Optimized H.264 Video Encoding and Packetization for Video Transmission Over Pipeline Forwarding Networks

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

***Abstract*—Previous works showed that the quality-of-service (QoS) requirements of multimedia applications can be optimally satisfied by pipeline forwarding (PF) by providing end-to-end delay guarantees as well as high network resource utilization. However, the unavoidable mismatch between reserved resources and the unpredictable traffic profile of a video stream has an impact on the resulting application layer quality. Therefore, a new low-complexity H.264 video encoding and packetization scheme based on a distortion-optimized macroblock grouping technique is designed here to maximize the performance of video transmission on PF networks. The scheme considers the perceptual importance of the different parts of the video data to group the most important information in few packets that are the natural candidates to receive the deterministic service provided by PF. Results show peak signal-to-noise ratio (PSNR) gains up to 2.5 dB over traditional video encoding and packetization schemes, as well as more graceful degradation in case of high network load.**

***Index Terms*—Flexible macroblock ordering, pipeline forwarding.**

I. Introduction

THE number of multimedia communication applications, among which video streaming, being deployed in today's packet networks is constantly increasing. These applications are often referred to as real-time to juxtapose them to traditional data applications as timely packet delivery is important for multimedia applications to work properly.

However, the large-scale development of multimedia services over packet networks, originally designed for generic data applications, faces numerous challenges stemming from the stringent quality-of-service (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—as they need at least a significant fraction of their data to reach the destination—i.e., they 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 and packet losses consequently small. This approach is not feasible if multimedia traffic grows faster than the rate at which technology enables proportionally more powerful network infrastructures. This might be the case not only because a larger fraction of broadband users might subscribe to current multimedia services, but especially because new, bandwidth-hungry services, such as high quality videoconferencing, virtual presence, high definition TV, 3-D video, distributed gaming, and remote surveillance, might become the dominant traffic sources in the future Internet.

A previous work [1] showed that pipeline forwarding (PF) of packets [2] can satisfy the QoS requirements of multimedia applications while ensuring high network utilization and enabling the implementation of highly scalable network devices [3]. These properties are key in today's networks to enable value-added services and to avoid that the traffic increase due to the above mentioned broadband applications either strains existing networks or forces the deployment of high cost, cutting-edge technology to upgrade them. PF properties stem from network nodes sharing a common time reference (CTR) and can be beneficial also when multicasting of packets is performed [4]. Thus broadcasting services and group communications, as well as point-to-point, possibly peer-to-peer, streaming and interactive multimedia applications can benefit from PF.

PF firstly enables overcoming the scalability limitations of the overprovisioning-based approach by providing efficient support for multimedia applications, i.e., high network utilization. Service providers can thus offer new multimedia services at competitive prices to a large 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

Manuscript received September 18, 2009; revised March 09, 2009. First published May 02, 2009; current version published July 17, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhu Liu.

The authors are with the Control and Computer Engineering Department, Politecnico di Torino, 10129 Turin, Italy (e-mail: masala@polito.it; andrea.vesco@polito.it; mario.baldi@polito.it; demartin@polito.it).

Digital Object Identifier 10.1109/TMM.2009.2021784

[1] See for example: "Will Internet TV Crash the Internet?" online 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, online at http://www.ofcnfoec.org/materials/2006KafkaPlenary.pdf.

1520-9210/$25.00 © 2009 IEEE

will be strained by huge amounts of traffic. PF is key in enabling the implementation of highly scalable network devices [5] that will be able to overcome the switching bottleneck resulting from the switching solutions and architectures currently deployed in network devices.

However, video transmission over PF networks does present the challenge of optimally matching the amount of resources reserved throughout the network to the specific video stream to be transported, which is made nontrivial by the highly variable amount of bits which in turn depends on the video content. Recently, [6] proposed a quality-oriented multimedia delivery framework that tackles this issue optimizing the trade-off between resource utilization and user perceived quality. The issue of multimedia packet scheduling for transmission over a PF network is explored by proposing two heuristic scheduling algorithms based on the perceptual information of the carried video samples. Special attention has been devoted to evaluate the trade-off between end-to-end delay and the number of video frames over which optimization is performed. Moreover, bandwidth allocation issues have been experimentally studied, evaluating the trade-offs between encoding quality and reserved bandwidth. Finally, the impact of the group of pictures (GOP) structure with different trade-offs between encoded video quality and bitrate fluctuations has been investigated showing their impact on performance.

However, [6] mainly focuses on optimization at the network layer. This paper, instead, complements the previous work by addressing the issue of improving the quality of video transmission by means of encoding and packetization schemes specifically designed for the characteristics of a PF network. The solutions presented here are independent of and can be used in conjunction with any further optimization at the network layer.

The main contribution of this work is to present a new video encoding and packetization scheme, based on a distortion-optimized macroblock grouping technique, to maximize the quality of video communication over a PF network. The proposed scheme considers both perceptual importance of the various parts of the video data and resources reserved in the PF network. The scheme takes advantage of the flexible macroblock ordering (FMO) option [7] of the H.264 standard [8] to perform an arbitrary grouping of macroblocks. In this context, this work shows how to use such an option, that was not originally designed with the purpose of enabling arbitrary macroblock encoding order, to properly reorder the macroblocks at the decoder. The underlying idea is to create both higher-importance and lower-importance packets by appropriately assigning macroblocks to packets. Clearly, higher-importance packets shall use reserved resources, i.e., shall be pipeline forwarded, thus receiving deterministic service, while other packets shall receive traditional, e.g., best effort or differentiated, service.

Another contribution of this work is to show how to tune various video coding and packetization schemes to optimize their performance over the PF network. For instance, a scheme based on region-of-interest (ROI) protection, that is often deployed to improve visual quality [9]–[11], is considered for transmission over PF. Both the ROI-based and the standard encoding and packetization schemes are compared to the proposed distortion-optimized macroblock grouping scheme showing the advantages of the latter.

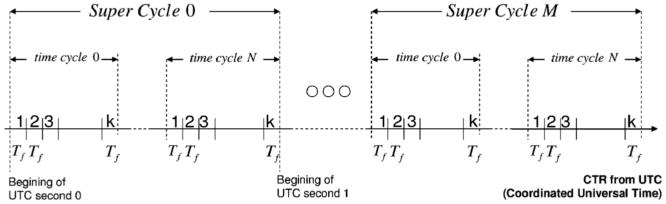

Fig. 1. Common time reference structure.

The paper is organized as follows. Section II discusses PF by presenting its operating principles, its suitability for video transmission, and the traffic conditioning deployed at the backbone network boundary. Section III presents in details the analysis-by-synthesis distortion estimation technique as well as the video encoding and packetization schemes. Extensive simulation results are presented in Section IV. Finally, conclusions and future work are discussed in Section V.

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 coordinated universal time (UTC) 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.

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: 1) 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 2) 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 1). 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 PF-capable nodes.

A deployment option of the basic pipeline forwarding operation is referred to as immediate forwarding. When it is deployed all packets received during TF t by node n are forwarded during TF $t+1$ to node $n+1$. The forwarding delay is equal to the propagation delay between two nodes plus one TF for all packets. Thus the end-to-end delay through the SVP τ_{PF} is given by

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

where N is the number of PF nodes on the path, Cd_i is the propagation delay between node $i-1$ and node i (the ingress node of the SVP being node 0 and the egress node being node $\mathrm{N}+1$), T_f is the duration of the TFs and J is the jitter, $0 \geq J \geq T_f$, see [1] for further information on jitter characterization. Therefore, for all purposes of video transmission the end-to-end delay through the SVP can be considered constant and deterministically upper-bounded, given the path the video flow takes through the network, as

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

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 one TF. Point-to-multipoint SVPs can be used to support multicast and broadcast packet delivery with guaranteed quality.

Non-pipelined packets, i.e., packets that are not sent over an SVP, 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-capable node performs statistical multiplexing of non-pipelined traffic. Therefore, SVPs are not at all as time division multiplexing (TDM) 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, i.e., 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 Optimization

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. 2. For example, if a video sequence is sampled at 30 frames per second, a super cycle lasts one second and contains 300 time cycles, a reservation can be made in a number of TFs $t, t+1, \ldots, t+r$ each ten time cycles, where $r+1$ is the number of allocated TFs; r is chosen such that the reservation is large enough to enable the transmission of a whole encoded video frame.

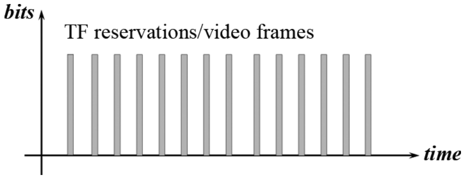

Fig. 2. Periodic allocation scheme for video transmission.

However, a video stream is inherently variable as the amount of bits required to encode each video frame changes significantly. Thus the maximum frame size should be used to determine the reservation, but this might yield to inefficient bandwidth allocation. Therefore, more efficient techniques could be implemented by reducing the reservation and sending video packets in excess as non-pipelined traffic.

In a possible design the PF network provides services with two quality levels, i.e., deterministic and best effort. Based on the pipeline forwarding operating principles, an SVP can be modeled as an *independent time-invariant channel with deterministic constant delay and loss/late probability* (P_{ll}^{PF}) equal to zero, as follows:

$$P_{ll}^{PF} = 0. \qquad (3)$$

On the contrary, non-pipelined channel can be modeled as an independent time-invariant packet drop channel with random delay. 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: 1) a loss probability P_{lost}^{NPF} independent of t, and 2) a variable end-to-end delay, yielding a total packet loss rate

$$p_{ll}^{NPF} = p_{lost}^{NPF} + \left(1 - p_{lost}^{NPF}\right) \cdot p_{late}^{NPF} \qquad (4)$$

where p_{late}^{NPF} is the probability that a non-pipelined packet reaches the destination too late for decoding.

When video sequences are transmitted, if not all the video packets can be accommodated into an SVP, errors and packet losses contribute to increase the distortion d_0 introduced by the video encoding process. The exact expected distortion value at the receiver for a given video sequence could be computed as the weighted average of the distortions corresponding to all the possible realizations of the network channel where the weights are the probability of a specific channel realization, as formulated in [12]. However, this procedure is impractical due to its computational complexity, hence a linear approximation is commonly used [12]–[16]:

$$E[d] = d_0 + \sum_{i=1}^{N} d_i \cdot p_{ll}^{i}$$

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

Let Ω be 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 $\alpha \cap \beta = 0$. 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

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

The work in [6] focuses on minimizing $E[d]$ by using PF for the transmission of packets with the highest d_i as well as on minimizing the loss/late probability p_{ll}^{NPF} experienced by non-pipelined packets. This work proposes another approach to minimize $E[d]$, i.e., the minimization of the distortion d_i of non-pipelined packets, which can be performed also in addition to the techniques proposed in [6]. The encoding and packetization schemes are designed to group the most important information in few packets, which are the natural candidates to receive the deterministic service provided by PF. Given an SVP which can transmit a certain amount of bits, even if not known at encoding time, the video encoding process is optimized to maximize d_i with $i \in \alpha$ which is equivalent to minimize d_i with $i \in \beta$.

C. Scheduling Operations at the SVP Interface

Fully benefiting from PF requires providing network nodes and end-systems with a CTR to maximize the quality of the received service [1]. Since this is not realistic in the near future, this work assumes asynchronous video sources and receivers connected to portions of the network performing traditional packet switching.

The generated packet stream is then time-shaped by the scheduling algorithm at the SVPI, i.e., packets are forwarded during the TFs in which resources have been allocated to their SVP. The scheduling algorithm is also responsible of selecting the set of packets with the highest distortion α to transmit on the SVP in order to minimize the expected distortion at the receiver.

To achieve the best results, the scheduling algorithm should run on the entire video sequence, which is obviously not possible in a real scenario. Normally, to avoid packets arriving at the destination beyond their playout deadline due to the variable delay introduced by the asynchronous access network, the fixed delay through the SVP and the scheduling algorithm waiting a large number of frame periods $(1/f_r)$, a trade-off is found by running the algorithm on a small part of a video sequence, which results is a locally optimal schedule [6]. The length of the video sequence on which the algorithm is run is determined based on the maximum end-to-end network delay tolerable by the application or a percentile thereof.

The SVPI estimates [17] the maximum delay experienced by packets through the asynchronous access network and calculates the fixed delay they experience on an SVP by (2). Then, 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. Given the arrival time t_i of the first packet of video frame i at the SVPI, the end-to-end network delay tolerable by the application τ_N, the maximum delay through the access network τ_A and through the SVP τ_{PF}, the value of the forwarding deadline $t_{fd,i}$ for each packet belonging to video frame i is calculated as follows:

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

$\tau_s = \tau_N - \tau_A - \tau_{PF}$ is the maximum time pipelined packets can spend at the SVPI while still satisfying (6).

Since the delay introduced by the PF backbone network is known in advance and it is smaller than the maximum delay introduced by a backbone deploying other packet queuing techniques [1], typically τ_s enables running the scheduling algorithm on longer sequences of video frames compared to when traditional network solutions are used. This results in a potentially more optimized solution and confirms the effectiveness of PF for the purpose of video communication.

In order to assess the gain in video quality stemming only from the proposed video encoding and packetization schemes, this work considers a low delay scenario in which the scheduler optimizes the scheduling over a single video frame. Each video frame is assumed to be encoded, packetized by means of the considered encoding and packetization schemes and immediately sent by the source. For simplicity's sake, in the rest of the paper, video packets are assumed to reach the SVP interface (SVPI) without losses and after a negligible delay, i.e., τ_A is equal to zero. 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.

Fig. 3 shows the time schedule for the transmission of packets at the SVPI; at time t_0 the SVPI computes $t_{fd,1}$ for each packet of the first video frame. Moreover, it also computes the number of allocated TFs before time $t_{fd,1}$ and the amount of reserved bits inside those TFs. Note that the sum of the reservation size inside the TFs allocated between two forwarding deadlines coincide with the reservation for an encoded video frame. Then the scheduler selects for PF the subset of packets α_{VF1} with the highest distortion, which fit into the TF reservation. At the end of the scheduling operations, the SVPI forwards, as non-pipelined traffic, the subset β_{VF1} of the lowest distortion packets immediately, i.e., without waiting for packets in α_{VF1} to be sent. The previous steps are repeated every time a new video frame arrives.

Different packet scheduling algorithms can be implemented to select which packets should be pipeline forwarded to maximize the utilization of the reserved bandwidth while minimizing the expected distortion at the receiver. In this work the first fit in (FFI) algorithm is deployed. This algorithm has been proven

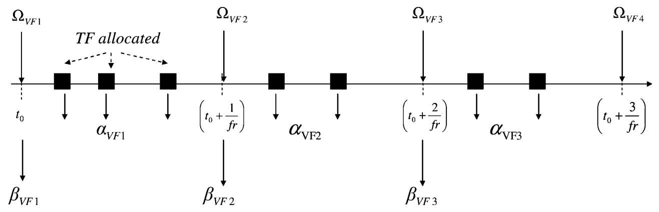

Fig. 3. Time schedule for the transmission of packets at the SVP interface.

to be a good candidate for actual deployment because of its good trade-off between low complexity and good video quality performance [6]. The FFI considers packets to be forwarded through an SVP in decreasing order of distortion d_i. Then it assigns packets to the first available TF with enough reserved capacity, until it cannot accommodate more packets or all packets have been assigned. Thus, the FFI complexity is linear in the number of packets waiting for scheduling and in the number of reserved TFs for each video frame.

III. The 264 Video Encoding for Pipeline Forwarding

A. Analysis-by-Synthesis Distortion Estimation

The quality of multimedia communications over packet networks is affected by packet losses. The amount of resulting 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 [19] approach, e.g., motion vectors are more important than residual coefficients. In order to provide a quantitative importance estimation method at a finer level of granularity, the importance of a video coding element, such as a macroblock or a slice, i.e., an integer number of consecutive macroblocks, could be defined as a value proportional to the distortion that would be introduced at the decoder by the loss of that specific element.

The analysis-by-synthesis technique [13] computes the distortion caused by the loss of each element, e.g., a macroblock, referred to as the distortion of the macroblock in the following, using the following steps:

1) decoding, including concealment, of the bitstream simulating the loss of the macroblock being analyzed (synthesis stage);
2) quality evaluation, that is, computation of the distortion caused by the loss of the macroblock; 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. Therefore, complexity is only due to the simulation of the concealment algorithm. In case of a simple temporal concealment technique the task is reduced to provide the data to the quality evaluation algorithm. Moreover, with this simple concealment technique, the distortion caused by the loss of a packet containing several macroblocks can be easily estimated by summing the distortion of each macroblock.

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 interdependencies usually existing between data units, the simulation of the loss of an isolated data unit might not be completely realistic. However, values estimated by the analysis-by-synthesis method, which is equivalent to the DC^0 method in [16], are shown to be very close to the actual distortion values, even if there is a slight tendency to overestimation. Note that all the considered distortion values accounts for the effect of the dependencies between macroblocks, i.e., the distortion due to error propagation. Nevertheless, experiments in [16] as well as other applications of the analysis-by-synthesis approach to MPEG coded video [15], [20], [21] confirm that such an estimation technique can be successfully used to develop quality optimized video communication algorithms.

In this work a low-complexity model-based approach, first presented in [22], is used to estimate the distortion caused by packet losses in future frames due to error propagation. According to [22], the ratio of the distortion caused in future frames to the distortion caused in the current frame can be modeled as a function of only the number of frames affected by the error propagation. Such a result is shown to be consistent across a wide set of sequences. Therefore, to estimate the total distortion caused by the loss of a macroblock, the distortion induced in the current frame can be multiplied by a fixed coefficient which depends on the position of the macroblock within the GOP.

The complexity of the model-based estimation approach is due to two factors: 1) the simulation, for each macroblock, of the error concealment technique that would be performed at the decoder, for the current frame only; 2) a multiplication by a precomputed value depending on the position of the macroblock within the GOP. In case a simple frame copy error concealment technique is employed, an MSE computation is required between two frames already available at the encoder. This operation takes constant time for each macroblock. Thus the complexity of the model-based distortion estimation method for a frame is $O(\mathrm{M})$, where M is the number of macroblocks per frame.

However, note that distortion values can also be precomputed without using the model, i.e., by decoding the whole GOP, and stored in the case of prerecorded video, e.g., non-live streaming scenarios. In this case the complexity of computing the distortion values for each frame is $O(\mathrm{MN})$, where N is the number of frames per GOP. The accuracy of the model-based distortion estimation compared to precomputation by whole GOP decoding will be assessed in Section III-C.

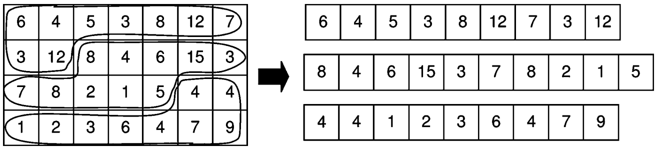

Fig. 4. Example of slices resulting from the traditional raster scan order macroblock grouping technique. Numbers indicate distortion values for each macroblock.

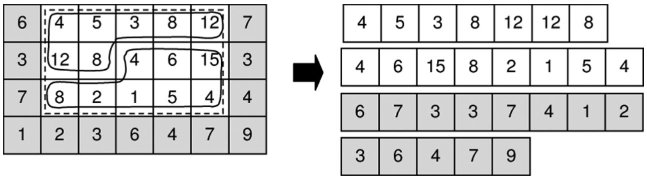

Fig. 5. Example of slices resulting from the ROI prioritization technique. Numbers indicate the distortion value for each macroblock, white and shaded colors represent the ROI and non-ROI areas, respectively.

B. Encoding and Packetization Schemes

Traditionally, video encoders perform coding operations regardless of how data is transmitted over the network. Then a module, called packetizer, is used to split the data stream produced by the encoder into different packets. However, the data stream can be decoded only by starting at predefined resynchronization points, e.g., at the beginning of a new picture. Video encoders usually have the possibility to group an arbitrary number of consecutively encoded macroblocks of a picture into the so-called slice, which is the smallest unit including a resynchronization code. Thus, each slice can be decoded independently of the others.

Slices can not be too small because this would reduce coding efficiency and can not be too large because this would require dealing with fragmentation. This work assumes that the maximum packet payload size is known by the video encoder in order to achieve the maximum efficiency. Therefore, data is grouped into slices whose size is the closest to the maximum packet payload size and each slice is inserted into one packet. With this scheme, in case of packet losses the decoding of correctly received packets is always possible, because each packet contains an independently decodable slice.

This work aims at improving the quality of the video communication by influencing the coding and packetization scheme in order to group together the most important macroblocks of a picture into few packets, which are the natural candidates to receive the deterministic service provided by PF. Three encoding and packetization schemes are investigated in the following.

1) Standard: The traditional encoding scheme groups macroblocks in the same slice in raster scan order, i.e., from left to right, top to bottom, regardless of macroblock distortion. The situation is illustrated in Fig. 4. Since in this scheme only the slice size can be easily controlled, the encoder is configured to produce packets whose size—including the header size—is as close as possible to the reservation size inside the allocated TFs, in order to maximize the reserved bandwidth utilization. This scheme is referred to as standard in the rest of the paper.

2) ROI Prioritization: Recent video coding standards include innovative coding options with respect to past standards. For instance, the H.264 standard [8] introduces the FMO option [7], which allows to control how macroblocks are grouped together into slices. This feature is achieved by adding a new layer between macroblocks and slices, i.e., macroblock groups.[2]

With the FMO option the encoder is not any more restricted to raster scan order as in Fig. 4. Using FMO, the encoder first assigns each macroblock in the picture to a certain group, then it encodes each group independently. For each group, only macroblocks belonging to the group are considered. They are coded in the raster scan order within the group, but they are not necessarily consecutive in the raster scan order within the picture. Then, for each group, macroblocks are put into slices and slices into packets.

The FMO option allows great flexibility in defining the groups. For instance, with "type 2" and two groups, macroblocks can be assigned either to the first group, a rectangular region of macroblocks called region of interest (ROI), or to the other group, i.e., the remaining macroblocks on the background. Note that, using the FMO option, a slight overhead—few bytes—is introduced for each frame to signal the position and size of the ROI. However, this overhead has a negligible effect on the compression performance of the encoder.

The deployment of a ROI is a well-known method [9]–[11] to improve the quality of video communications, for instance by assigning a better protection level to the ROI data. Ideally, the ROI should include the area on which the user's attention is focused, so that prioritizing the ROI minimizes the distortion as perceived by the user. However, automatically determining a ROI inside a video sequence based on the semantic of the video content is a tough task. To partially overcome this issue, in this work the ROI size and boundaries are determined using the macroblock distortion information computed by the analysis-by-synthesis technique.

Clearly, compressed video data included in the ROI area shall be pipeline forwarded, thus receiving deterministic service. In more details the ROI is determined as follows. First, if the whole frame fits into the allocated TFs the whole frame is considered as the ROI and consequently pipelined forwarded. If its size is too large, the ROI area—restricted to be a rectangular set of macroblocks—is progressively reduced, first decreasing width by one macroblock, then height, and again until it fits into the TFs allocated to the frame. For each new ROI size, different rectangle positions are possible, each one including a different set of macroblocks. Each possible position is evaluated by computing the total distortion of the macroblocks in the ROI, and the position with the highest total distortion value is selected, provided that it fits into the reservation size. At this point, both the size and position of the ROI have been determined, and the encoder proceeds to create packets whose size—including the header size—is as close as possible to the reservation size inside the TFs allocated to the video frame; see Fig. 5. This scheme is referred to as ROI-based in the rest of the paper.

3) Distortion-Optimized Macroblock Grouping: As discussed before, the objective of an encoding and packetization

[2] Note that in the H.264 standard they are called "slice groups" even if they represent a subset of macroblocks in the picture, as defined in Par. 3.138. Since the discussion focuses on macroblocks, we refer to them explicitly as groups of macroblocks to avoid confusion.

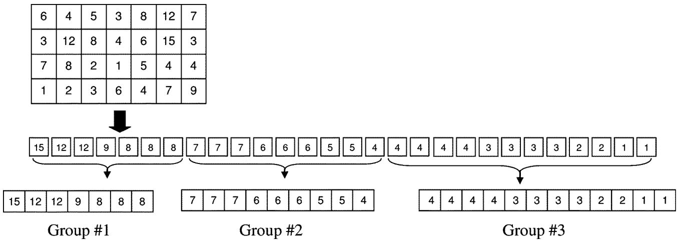

Fig. 6. Example of slices resulting from the distortion-optimized macroblock grouping technique. Numbers indicate the distortion value for each macroblock.

scheme tackling the issue of minimizing the distortion as perceived by the users on a PF network is to produce high-distortion packets which are candidates for pipeline forwarding as well as low-distortion packets to be forwarded as non-pipelined traffic. In the same network conditions, i.e., deploying the same scheduling algorithm at the SVPI and with the same packet loss rate P_{ll}^{NPF} on the non-pipelined channel, minimizing the expected distortion is equivalent to maximize d_i with $i \in \alpha$; see (5).

This can be achieved by rearranging the single macroblocks in a frame as follows. For each frame, macroblocks are sorted in decreasing order of distortion. Then, macroblocks are assigned, in that order, to the first packet, until the maximum packet payload size, i.e., the reservation size inside the allocated TFs, is reached. The previous step is repeated until all macroblocks have been assigned to a packet. With this procedure, the first packet will always have the highest possible distortion, the second one will have the second highest distortion, and so on, until the last packet, which will have the lowest distortion. The procedure is illustrated in Fig. 6.

The previous ROI-based scheme needs to know the amount of reserved bandwidth in advance so that the optimal ROI size can be determined. On the contrary, ordering macroblocks from the most to the least important one and putting them into packets always provide the best performance independently of the bandwidth reserved for the video frame because packets containing the most important macroblocks are always scheduled for pipeline forwarding. This indeed reduces encoding complexity with respect to the ROI-based scheme.

However, implementing the proposed encoding and packetization scheme with existing video coding standards faces some difficulties since they do not easily allow to rearrange macroblock encoding order which is needed to perform arbitrary grouping of macroblocks into packets. The FMO option, which was not originally designed to allow an arbitrary macroblock encoding order, can be used to arrange macroblocks into packets in decreasing order of distortion as follows. As stated before, the FMO option allows great flexibility in defining macroblock groups. In particular, completely arbitrary group definition ("type 6" in the standard) is also allowed. Each macroblock can be assigned to any group by means of a map and a maximum of eight groups are allowed. Moreover, note that if macroblocks are assigned to a certain group, and such a group is put into one slice, it is possible to arbitrarily decide which macroblocks of the frame are put into the slice. Unfortunately, if the macroblocks of a group need two slices to be coded, it is not possible to decide which macroblocks are in the first or in the second slice, since the standard impose the raster scan order inside the group. However, the group can be made sufficiently small so that all its macroblocks fit in only one slice. The remaining macroblocks are assigned to another group and the process is repeated until the eighth group. If, after eight iterations, some macroblocks are still not assigned to a group, they are forcedly assigned to the eighth slice group. Therefore, if more than one slice is needed to code the macroblocks in the eighth group, their assignment into slices cannot be arbitrarily decided since it has to be in raster scan order, however usually a large part if not all the higher-distortion macroblocks have already been inserted in the previous seven groups. Fig. 7 shows the macroblock assignment to groups corresponding to slices in Fig. 6. Note also that the proposed scheme produces a bitstream which is H.264/AVC compliant. This scheme is referred to as distortion-optimized macroblock grouping (DOMG) in the rest of the paper.

2	2	2	3	1	1	2
3	1	1	3	2	1	3
2	1	3	3	2	3	3
3	3	3	2	3	2	1

Fig. 7. Macroblock assignment to groups corresponding to slices in Fig. 6. Numbers represent the groups.

Clearly, a map signaling which macroblock is assigned to which group needs to be coded and sent to the decoder, otherwise macroblocks could not be correctly placed in the decoded frame. The map is inserted into the so-called picture parameter set (PPS), which is a structure first introduced in the H.264/AVC standard. Although the main purpose of the PPS is to increase compression performance and improve reliable delivery of the most important parameters of pictures, it can also be used for the purpose of including FMO maps.

C. Discussion of the Encoding and Packetization Schemes

This section presents a preliminary analysis of the characteristics of the three encoding and packetization schemes aimed at better understanding the simulation results.

Firstly, all the three encoding and packetization schemes have been adapted to the operating principles of the PF network. In particular, they have been configured to create packets whose size is as close as possible to the reservation size inside the allocated TFs to maximize the utilization of the reserved bandwidth.

Secondly, with the ROI-based and DOMG schemes, the time-variant characteristics of the application data, i.e., the different distortion of the various macroblocks, have also been exploited to control and maximize the distortion of the packets candidate for pipeline forwarding. Fig. 8 shows a sample of the statistical frequency of packet distortion values estimated with the model described in Section III-A, for the three schemes, for the *lts* test sequence. Moreover, Fig. 9 shows, for same the three schemes of the previous figure, the statistical frequency of the actual packet distortion values, obtained by simulating the decoding of the whole GOP for each packet. The two distributions show strong

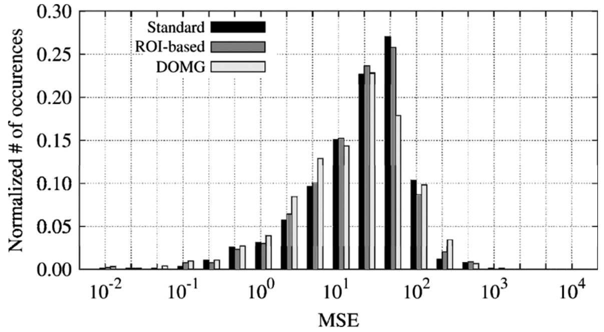

Fig. 8. Normalized number of occurrences of estimated packet distortion values for the *lts* sequence.

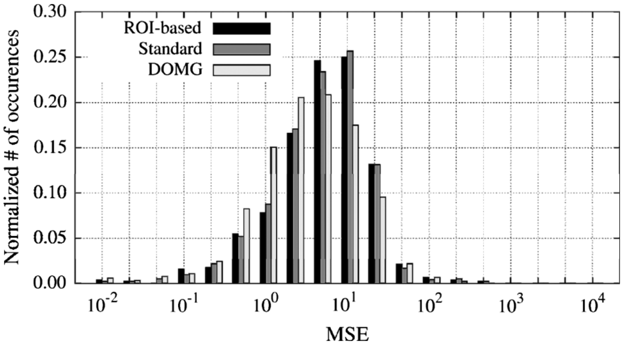

Fig. 9. Normalized number of occurrences of actual packet distortion values, computed by completely decoding the GOP for each considered packet, for the *lts* sequence.

similarities in the behavior of the various schemes, even though the actual distortion values are slightly lower than the estimated ones.

The statistical frequency of packet distortion values for the DOMG scheme is significantly different from the one of the standard scheme. For instance, the number of low-distortion packets is strongly increased, whereas the change in the case of the ROI-based scheme is not as significant as for the DOMG scheme. Moreover, since the ROI-based scheme is based on the "type 2" FMO, it is restricted to produce a rectangular ROI, therefore, for any sequence, usually the last packet containing ROI data is not completely full, thus wasting space that could accommodate other macroblocks which would increase the distortion associated with the packet.

Fig. 10 shows the statistical frequency of packet sizes for the three considered schemes, for the foreman sequence, when the reservation size inside the allocated TFs is 1250 bytes. The ROI-based scheme has the lowest number of packets whose size is close to the reserved size and, differently from the other schemes, it has numerous packets whose size is about 1100 bytes. This is a sequence-independent behavior of the ROI-based scheme, which relies on the "type 2" FMO that imposes rectangular ROI regions, thus it is often impossible to add another whole row of macroblocks to the ROI without exceeding the 1250 byte threshold. Since typically very few packets are smaller than 150 bytes, i.e., the remaining size in the reservation, the reserved bandwidth is slightly underutilized, which negatively impact on the performance of the communication, as shown in the results section. The DOMG scheme, instead, does not suffer from reserved bandwidth underutilization, since it can decide how to group data with macroblock granularity.

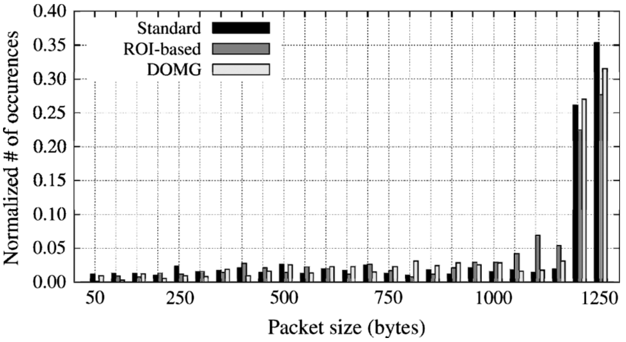

Fig. 10. Normalized number of occurrences of packet sizes for the three considered schemes, for the *foreman* sequence.

Finally, note that the deterministic service provided by the PF network is particularly suitable for transmitting the PPS information, either in-band when the PPS is put in the highest distortion packet, or out-of-band when a dedicated SVP is allocated for PPS transmission. The PPS is, in fact, extremely important since its loss prevents the decoding of the whole frame, thus the PPS is the perfect candidate to be pipelined forwarded.

IV. Experimental Results

In this section the performance of the encoding and packetization schemes are assessed and compared in the same network conditions, i.e., allocating the same bandwidth for each scheme and using the same scheduling algorithm. Under these conditions the performance gains are all due to the capability of the encoding and packetization schemes to exploit the peculiarities of the service offered by the PF network.

A. Model-Based Simulations

The first part of the performance evaluation focuses on determining the general behavior of the encoding schemes. A model implementing the FFI algorithm and a random packet drop channel are assumed to assess the performance independently of a particular network scenario.

In the model-based simulations, packets of each video frame are first sorted in decreasing order of distortion. The most important packets, as well as the PPS, are scheduled for pipeline forwarding and consequently considered as correctly received at the decoder. The remaining packets are subject to random losses. Hence the model also accounts for the overhead due to the PPS information, as well as it allows to evaluate the performance as a function of an arbitrary loss rate. The reserved bandwidth is the same for all experiments related to the same video sequence and, for each sequence, its value is chosen on the basis of the average frame size produced by the standard encoding and packetization scheme. The transmission of various video sequences, encoded with different parameters, is evaluated by means of that model.

Experiments are performed with four video sequences known as *foreman*, *mad*, *lts*, and *city*, encoded at CIF resolution (352×288), 30 fps, with the standard H.264 codec JM v. 11.0 [23].

TABLE I
MAIN PARAMETERS OF THE USED VIDEO SEQUENCES

Sequence	Scheme	Bitrate (Application) (Kb/s)	Bitrate (Network) (Kb/s)	Encoding PSNR (dB)
Foreman	Std., 1 slice	594.64	616.88	35.98
	Standard	588.04	611.30	35.87
	ROI-based	603.70	613.19	35.87
	DOMG	588.41	612.09	35.49
Mad	Std., 1 slice	283.10	294.34	37.33
	Standard	281.55	293.81	37.26
	ROI-based	298.13	296.54	37.26
	DOMG	273.64	295.24	36.95
Lts	Std, 1 slice	667.31	698.71	36.02
	Standard	660.41	691.74	35.86
	ROI-based	676.61	694.46	35.86
	DOMG	673.50	705.71	35.56
City	Std., 1 slice	563.49	585.76	34.46
	Standard	554.21	575.98	34.32
	ROI-based	570.50	578.03	34.31
	DOMG	566.56	588.14	34.06

The H.264 video codec is configured to use a fixed quantization parameter (QP), hence the video quality is approximately constant. First, sequences have been encoded with the DOMG scheme using a fixed QP equal to 29 for all macroblocks of all frames, leading to the bitrates shown in Table I. Then, to achieve the same bitrate for the other schemes, the base QP, equal to 29, was decreased by one for all macroblocks belonging to a number of frames, uniformly distributed within each GOP, so that globally about the same bitrate of the DOMG scheme is achieved for all schemes for a given sequence.

Table I reports, for each combination of sequence and encoding scheme, the bitrates as seen at the application level, the bitrate including the IP/UDP/RTP packet overhead and the corresponding PSNR value. Note that the bitrate at the application as well as at the network level also include the bits dedicated to the PPS information for the case of the ROI-based and DOMG schemes. Table I considers the three encoding and packetization schemes described in Section III-B as well as a fourth encoding scheme producing a single slice for each frame, referred to as *Standard 1 slice* in the rest of the paper. This scheme is considered here in order to assess the overhead caused by using more than one slice for each frame, as it is done with the other schemes. The table shows that the quality loss due to the use of multiple slices per frame (standard scheme) with respect to the 1-slice scheme is about 0.1 dB PSNR. The network bitrate for the case of the *Standard 1 slice* scheme has been obtained by adding the network header to slice fragments whose size is equal to the network maximum transmission unit (MTU). Note also that the PPS overhead causes a reduction of the encoding quality of about 0.3–0.4 dB PSNR for the DOMG scheme with respect to the standard scheme, while it is negligible for the ROI-based scheme.

According to [6], the most suitable encoding scheme for the allocation provided by the PF transmission, depicted in Fig. 2, is to set the video codec to produce 99 P-frames after each I-type frame. A rate control scheme could also be used, as done in [6], achieving a smoother bitrate profile. However, the smoothness is achieved by trading off encoding quality and generally yields lower quality at the receiver than the employed scheme [6].

To reduce the impact of errors in the video section containing P-type frames, an intra refresh method is employed, which refreshes 33 macroblocks in each picture, taken in raster scan order, thus achieving a full frame refresh every 12 frames for a CIF resolution sequence. The intra-refresh method is the same for all the encoding schemes, therefore the particular slice configuration of each scheme is not considered. Moreover, the QP for the intra-refreshed macroblocks is equal to the one of the other macroblocks in the same frame.

Packet losses are concealed, unless otherwise stated, using a temporal concealment technique, i.e., missing pixels are replaced with the ones in the same position in the previous frame.

For the *Standard 1 slice* encoding scheme, slices are generally larger than the network MTU, thus a fragmentation strategy is needed before transmission. Two strategies have been employed. In the first one, referred to as "A" in the rest of the paper, the IP layer fragments the transmission unit, namely, the slice, in multiple IP packets using the IP fragmentation feature. This implies that even if only a single fragment of the transmission unit is missing at the receiver, the receiver IP layer discards the whole unit, i.e., the slice is entirely lost and the whole frame has to be concealed. The second strategy, referred to as "B" in the rest of the paper, fragments data units at the RTP level, that is, a slice is encapsulated in multiple RTP packets (whose size is smaller than the MTU). With this strategy, no received packets are discarded in the receiver IP layer. This allows to decode each slice up to the point of the first missing packet of the slice itself. In fact, the rest of the slice after the first packet loss, even if data are received, is undecodable due to the slice internal dependencies.

1) Impact of Packet Loss Models: Two different packet loss models are adopted. In the first simulation set, data sent as non-pipelined are subject to uniformly distributed random packet losses. For each video sequence and desired packet loss rate (PLR), 30 loss traces are generated.

The results in Fig. 11 indicate that the proposed DOMG scheme provides consistent PSNR gains with respect to the standard scheme, up to 2 dB. Although the absolute value of PSNR decreases if the packet loss rate is increased, the gain of the proposed scheme is more pronounced at high packet loss rates, which shows that the proposed encoding scheme provides more graceful performance degradation. The performance of the ROI-based scheme, instead, is strongly influenced by the underutilization of the reserved bandwidth. The difference is limited in the case of low packet loss rates, while it is significant at high packet loss rates.

These results also show that both the data importance, i.e., distortion, and packet sizes are to be considered at application level to address the issue of optimizing the video communication quality on PF networks. In general, it is not easy to adapt existing schemes, such as the ROI-based one, for PF networks.

Concerning the proposed DOMG scheme, results also show that it can achieve the same performance of the standard scheme at a much higher packet loss rate, e.g., 10% instead of 5% for the *foreman* and *lts* sequences. Considering the *city* sequence, the gain is even higher, 20% instead of 5%. For the *mad* sequence, due to the mainly static video scene, the gain is more limited.

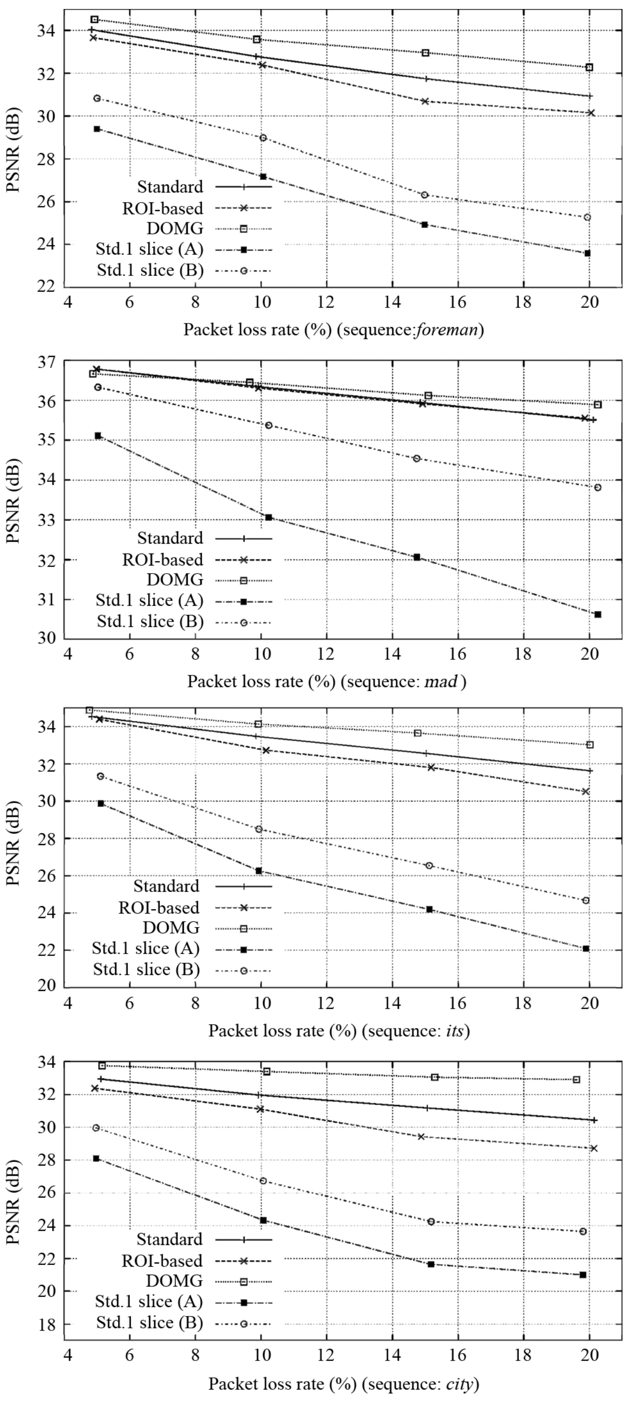

Fig. 11. PSNR performance as a function of the actual average PLR (uniform packet loss traces) for model-based simulations.

Note also that the overhead due to PPS information, which approximately causes 0.3–0.4 dB PSNR loss according to Table I at the considered bitrate with respect to the standard scheme, is more than adequately counterbalanced by the performance gain offered by the DOMG scheme for all the tested video sequences, except for *mad* at low packet loss rates where the overhead effect slightly prevails.

For the case of the *Standard 1 slice* encoding scheme, despite the 0.5 dB PSNR improvement over the DOMG scheme in error-free conditions, the communication performance is strongly influenced by the fact that single packet loss events, which affect a limited part of the slice, cause to the loss of either the whole slice (case “A”) or a potentially large portion of the slice (case “B”). For case “A”, the performance decrease with respect to the DOMG scheme ranges, for 5% PLR, between about 2 and 6 dB and the gap increases at higher PLR. The same behavior characterizes case “B”, showing significant performance losses, for 5% PLR, between about 1 and 4 dB. Therefore, it is highly recommended that the encoding and packetization strategies cooperate to maximize slice size while not exceeding the maximum packet size.

A second simulation set investigates the case in which data sent as non-pipelined is subject to bursty packet losses. The desired PLR is achieved with the same procedure of the previous case, however a Gilbert–Elliott model [24] instead of an uniform random error model is used to generate packet loss traces. PLR is set as in the previous case, while the target average burst length of packet losses is set equal to 2.02. That value is the mean average burst length observed in PF network simulations. The actual average burst lengths measured after generating the packet loss traces are 1.93, 1.94, 1.92, 1.92 for the simulations of the *foreman*, *mad*, *lts*, and *city* sequences, respectively.

Fig. 12 shows the performance of the schemes under analysis. The DOMG scheme provides the best performance for all four sequences, with gains very similar to the case of the uniform packet losses.

2) Impact of the Concealment Technique: The distortion estimation method presented in Section III-A assumes that a given concealment technique at the decoder is used. This is indeed a common assumption in literature when a rate-distortion optimization framework is employed (see, for instance, [14], [16], and [18]). However, this might not be the case in practical situations, therefore this section assesses the impact of using another error concealment method at the decoder.

In this section, a motion compensated temporal concealment technique is used. For each missing macroblock, first motion parameters are estimated from the macroblock information in the same position in the previous frame. Then, those parameters are used to build a motion compensated estimation of the missing macroblock on the basis of the pixel values in the previous frame.

Fig. 13 shows a comparison of the three schemes presented in Section III-B for the four tested video sequences. The performance gap between the three schemes tends to reduce since the motion-compensated concealment technique provides, in general, better performance than the simple temporal error concealment technique. However, the results are still consistent with the case of the simple temporal concealment technique. The DOMG scheme outperforms the other schemes, except in the case of the *mad* sequence. For this sequence, however, the performance of all the schemes is very similar (within 0.3 dB) as well as it is very close to the encoding PSNR of each scheme at low PLR values.

B. Network Simulations

The transmission of the video sequences encoded by means of the standard, ROI-based and DOMG schemes is also assessed

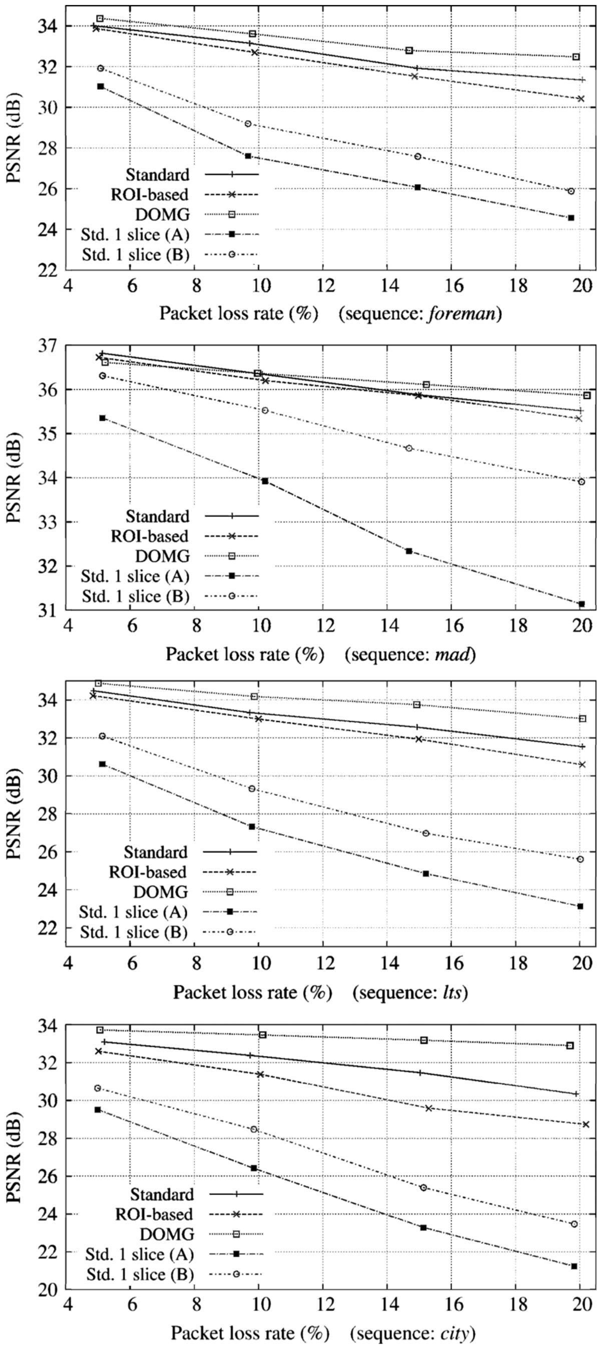

Fig. 12. PSNR performance as a function of the actual average PLR (bursty packet loss traces) for model-based simulations.

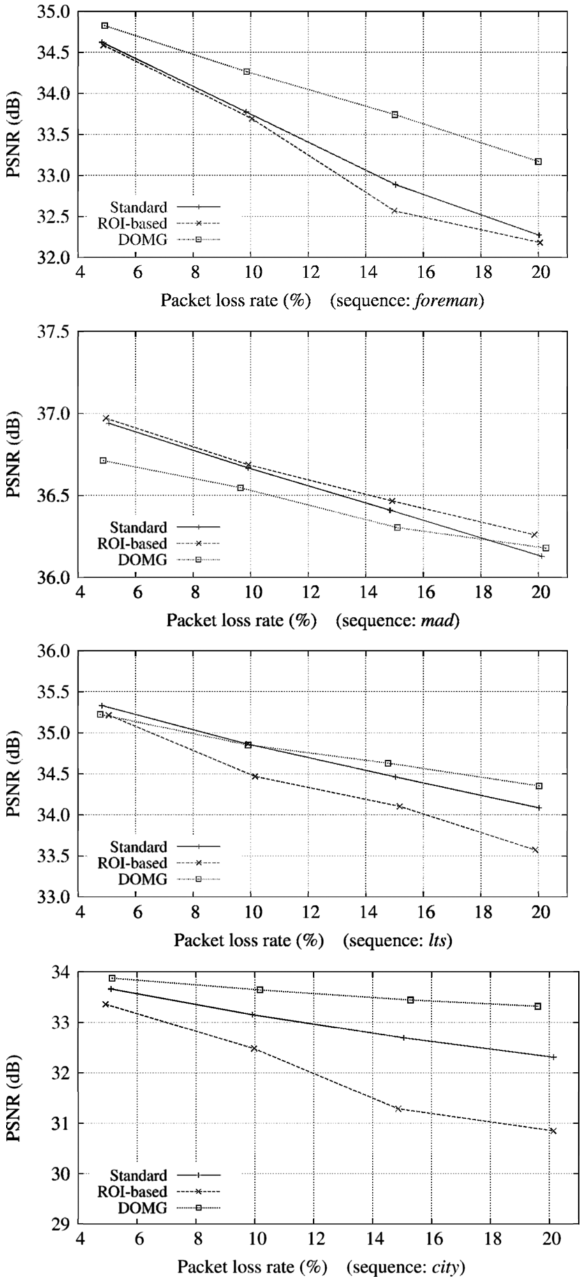

Fig. 13. PSNR performance as a function of the actual average PLR (uniform packet loss traces), model-based simulations, motion-compensated temporal concealment technique.

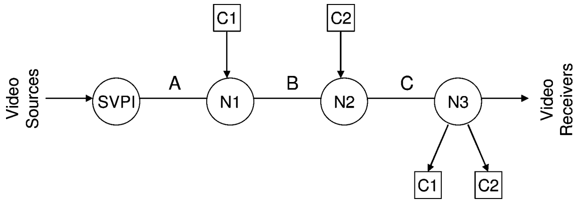

Fig. 14. Network simulation scenario.

by emulating the actual behavior of network devices with the network simulator ns [25].

The simulated network topology, depicted in Fig. 14, is composed of an SVP interface and three PF capable nodes. The capacity of the links is set to 10 Mb/s and their length is such that the end-to-end propagation delay is 60 ms. The parameters of the video sequences are the same as in the model-based simulations.

The four video sequences are encoded using the three encoding and packetization schemes, and the resulting 12 sequences are sent on the network at the same time. The PPS is sent in-band. Video transmissions assume the use of the IP/UDP/RTP protocol stack, which is commonly used for real-time multimedia communications.

The mean aggregate rate of the video flows is about 6.5 Mb/s, while the overall allocated bandwidth is 6.6 Mb/s. Interfering traffic (C1 and C2 in Fig. 14), handled as non-pipelined, is also injected in the network, causing congestion on the bottleneck link (Link C). The interfering traffic rate ranges from 1 Mb/s to 4 Mb/s in the simulations. Non-pipelined traffic is served in a FIFO basis at each node during any unused portion of a TF. Therefore, the instantaneous number of packets waiting for service in the non-pipelined queue might become large. Hence, losses might happen in bursts due to the instantaneous overload of the non-pipelined queue. Moreover, packets arrived at the receiver beyond a delivery deadline set to 100 ms were considered lost.

Note that the proposed scenario allows to evaluate the performance when non-pipelined packets belonging to a video flow compete for the same available resources with non-pipelined traffic of other video flows, as well as with the interfering traffic at subsequent nodes, as it potentially happens in real networks. Moreover, the performance of all the encoding and packetization schemes is simultaneously assessed in the same network conditions.

Fig. 15 shows the performance for the four tested video sequences in terms of PSNR values as a function of the interfering traffic rate. Each point represents the mean PSNR value computed over 20 repetitions of the sequence. The DOMG scheme provides consistently better performance compared to the standard and the ROI-based schemes in all conditions. Depending on the video sequence, the gain provided by the DOMG scheme ranges from about 1 dB for the *foreman*, *mad*, and *lts* sequences up to about 2.5 dB PSNR for the *city* sequence. Another advantage is that the performance gain over the other schemes increases as the amount of interfering traffic increases, thus the DOMG scheme provides more graceful performance degradation. The performance gain provided by the DOMG scheme is mainly due to the macroblock reordering technique, which influences the statistical frequency of the packet distortion values, as shown in Section III-C. As shown in Fig. 16 for the *city* sequence, the distortion values of packets sent as non-pipelined are lower than in the case of the other schemes, hence their impact on video quality is reduced in case of loss.

Moreover, Table II shows the efficiency with which the schemes use the reserved bandwidth. The proposed DOMG and the standard schemes achieve an efficient utilization of the reserved bandwidth, therefore the amount of data travelling as non-pipelined traffic is minimized and, consequently, losses affect a smaller share of the video data, with obvious benefit on the video quality. The ROI-based scheme, instead, has a tendency to underutilize the reserved bandwidth, as shown in Section III-C, which causes a performance reduction, as shown in Fig. 15.

Since the proposed DOMG scheme provides the best performance among the three schemes only if the network congestion level is higher than a given threshold, an adaptive strategy could be designed to dynamically choose the best encoding and packetization scheme not to incur in unnecessary encoding overhead if network conditions are good. Such an adaptation strategy could rely, for instance, on statistical information collected by the receiver and sent back as a feedback using, e.g., the standard RTP transmission control protocol (RTCP).

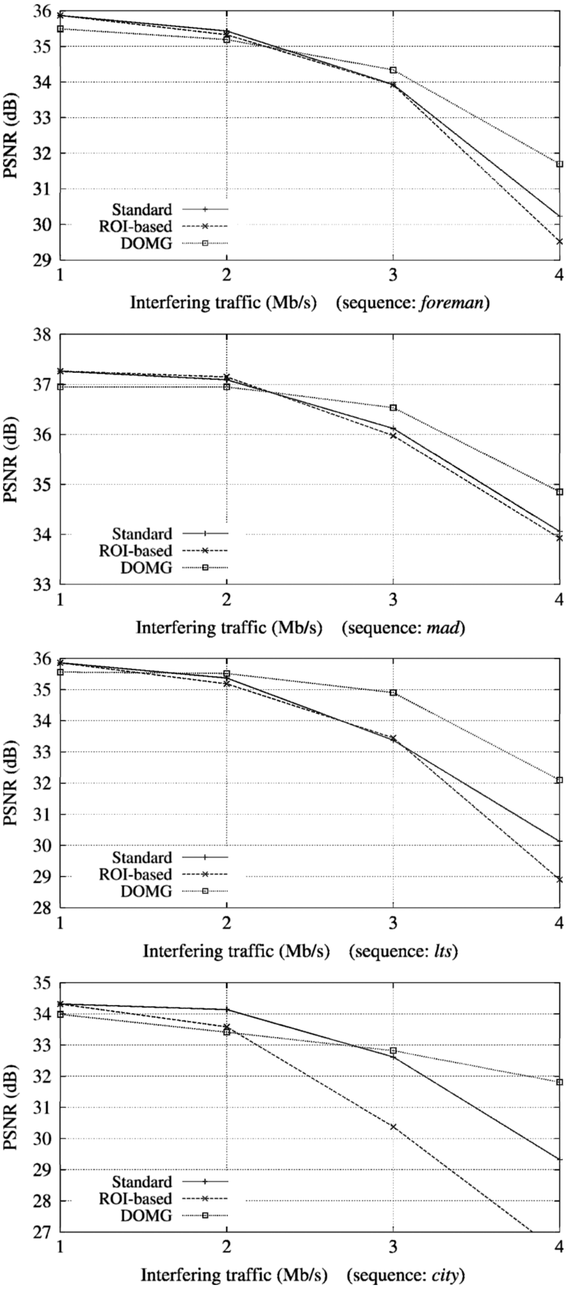

Fig. 15. PSNR performance in the network simulations for all four sequences.

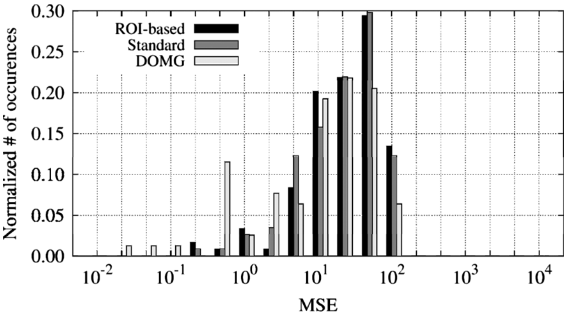

Fig. 16. Normalized number of occurrences of distortion values for non-pipelined packets, for the *city* sequence.

TABLE II
EFFICIENCY WITH WHICH THE SCHEMES USE THE RESERVED BANDWIDTH

Sequence	Scheme		
	Standard	ROI-based	DOMG
Foreman	99.1 %	97.8 %	99.4 %
Mad	99.2 %	98.6 %	99.7 %
Lts	97.9 %	98.5 %	99.2 %
City	99.4 %	98.6 %	99.8 %

V. CONCLUSIONS AND FUTURE WORK

This paper presented a low-complexity H.264 video encoding and packetization scheme optimized for deployment over networks implementing pipeline forwarding of packets. The perceptual importance of the video data, coupled with a distortion-optimized macroblock grouping technique relying on the FMO tool of the H.264 standard, is used in the packet creation process to group the most important information in few packets. Such packets are the natural candidates to receive the deterministic service ensured by pipeline forwarding. The solution optimizes video communications in terms of perceived video quality whereas it provides efficient utilization of the reserved resources. The performance of the solution has been assessed with extensive simulations, some based on a pipeline forwarding network model, others on emulating the actual behavior of network devices. Results showed significant PSNR gains—up to 2.5 dB—when compared to a traditional encoding and packetization scheme, as well as more graceful performance degradation when network load increases. Comparisons with a ROI-based scheme also showed the effectiveness of the proposed approach. To conclude, it is worth highlighting that the proposed solution relies solely on low-complexity algorithms, which makes it particularly suitable for deployment in real networks where scalability is an important issue.

Although the DOMG scheme provides efficient utilization of the resources an incorrect estimation of the bitrate fluctuations at encoding time might affect the utilization of the reserved bandwidth. Future work will be devoted to evaluating the effectiveness of dynamic bandwidth reconfiguration strategies in mitigating this issue. Moreover the suitability of scalable codecs, such as SVC which intrinsically provide a mechanism for bandwidth adaptation, will be evaluated for provisioning of video on-demand services over PF networks.

REFERENCES

[1] M. Baldi and Y. Ofek, "End-to-end delay of video-conferencing over packet switched networks," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 479–492, Aug. 2000.

[2] C.-S. Li, Y. Ofek, A. Segall, and K. Sohraby, "Pseudo-isochronous cell forwarding," *Comput. Netw. ISDN Syst.*, vol. 30, no. 24, pp. 2359–2372, Dec. 1998.

[3] M. Baldi and Y. Ofek, "Blocking probability with time-driven priority scheduling," in *Proc.f SCS Symp. Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, Vancouver, BC, Canada, Jul. 2000.

[4] M. Baldi and Y. Ofek, "Adaptive group multicast with time-driven priority," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 31–43, Feb. 2000.

[5] M. Baldi, G. Marchetto, and Y. Ofek, "A scalable solution for engineering streaming traffic in the future internet," *Comput. Netw.*, vol. 51, no. 14, pp. 4092–4111, Oct. 2007.

[6] M. Baldi, J. C. De Martin, E. Masala, and A. Vesco, "Quality-oriented video transmission with pipeline forwarding, special issue on Quality Issues in Multimedia Broadcasting," *IEEE Trans. Broadcasting*, vol. 54, no. 3, pp. 542–556, Sep. 2008.

[7] P. Lambert, W. De Neve, Y. Dhondt, and R. Van de Walle, "Flexible macroblock ordering in H.264/AVC," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 358–375, Apr. 2006.

[8] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T & ISO/IEC Std. H.264 & 14496-10, May 2003.

[9] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 134–139, Jan. 2008.

[10] A. Jerbi, J. Wang, and S. Shirani, "Error-resilient region-of-interest video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1175–1181, Sep. 2005.

[11] M. M. Hannuksela, Y.-K. Wang, and M. Gabbouj, "Sub-picture: ROI coding and unequal error protection," in *Proc. Int. Conf. Image Processing (ICIP)*, Jun. 2002, vol. 3, pp. 537–540.

[12] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 390–404, Apr. 2006.

[13] E. Masala and J. C. De Martin, "Analysis-by-synthesis distortion computation for rate-distortion optimized multimedia streaming," in *Proc. IEEE Int. Conf. Multimedia & Expo (ICME)*, Baltimore, MD, Jul. 2003, vol. 3, pp. 345–348.

[14] R. Zhang, S. L. Regunathan, and K. Rose, "End-to-end distortion estimation for RD-based robust delivery of pre-compressed video," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Nov. 2001, vol. 1, pp. 210–214.

[15] E. Masala, D. Quaglia, and J. C. De Martin, "Adaptive picture slicing for distortion-based classification of video packets," in *Proc. IEEE Workshop Multimedia Signal Processing (MMSP)*, Cannes, France, Oct. 2001, pp. 111–116.

[16] J. Chakareski, J. G. Apostolopoulos, S. Wee, W.-T. Tan, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1257–1269, Oct. 2005.

[17] A. Charny and J. Y. Le Boudec, "Delay bounds in a network with aggregate scheduling," in *Proc. Quality of Future Internet Services (QofIS)*, Berlin, Germany, Sep. 2000.

[18] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[19] R. Aravind, M. R. Civanlar, and A. R. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 5, pp. 426–435, Oct. 1996.

[20] P. Bucciol, E. Masala, E. Filippi, and J. C. De Martin, "Cross-layer perceptual ARQ for video communications over 802.11e wireless networks," *J. Adv. Multimedia*, vol. 2007, Article ID 13969.

[21] F. De Vito, L. Farinetti, and J. C. De Martin, "Perceptual classification of MPEG video for differentiated-services communications," in *Proc. IEEE Int. Conf. Multimedia & Expo (ICME)*, Lausanne, Switzerland, Aug. 2002, vol. 1, pp. 141–144.

[22] F. De Vito, D. Quaglia, and J. C. De Martin, "Model-based distortion estimation for perceptual classification of video packets," in *Proc. IEEE Int. Workshop Multimedia Signal Processing (MMSP)*, Siena, Italy, Sep. 2004, pp. 79–82.

[23] JVT Reference Software v. 11.0, 2007. [Online]. Available: http://iphome.hhi.de/suehring/tml/download.

[24] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell. Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sep. 1960.

[25] UCB/LBNL/VINT Network Simulator-ns (Version 2), 1997. [Online]. Available: http://www.isi.edu/nsnam/ns.

Enrico Masala (S'01–M'04) received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 2004.

In 2003, he was a Visiting Researcher at the Signal Compression Laboratory, University of California, Santa Barbara, where he worked on joint source channel coding algorithms for video transmission. He is currently a post-doc researcher at the Politecnico di Torino. His main research interests include multimedia processing (especially video) and coding and transmission over wireline and wireless packet networks.

Dr. Masala is a member of the Multimedia Communications Technical Committee within the IEEE Communications Society.

Andrea Vesco (S'06–M'09) was born in Torino, Italy, on July 26, 1978. He received the M.S. degree in telecommunications engineering from Politecnico di Torino, Turin, Italy, in 2003. He is currently pursuing the Ph.D. degree at the Department of Control and Computer Engineering of Politecnico di Torino.

His research interests are quality-of-service in packet switched network and multimedia communications.

Mario Baldi (M'99) received the M.S. degree (summa cum laude) in electrical engineering and the Ph.D. degree in computer and system engineering from Politecnico di Torino, (Technical University of Turin), Turin, Italy, in 1993 and 1998, respectively.

He is an Associate Professor of information processing systems at the Department of Control and Computer Engineering of Politecnico di Torino, Vice Dean of the PoliTong Sino-Italian Campus at Tongji University, Shanghai, China, and coordinator of the B.Sc. programme in Information Technology Engineering. He is a member of the Computer Networks Group (NetGroup) at the Department of Computer Engineering of Politecnico di Torino. He was Vice President for Protocol Architecture at Synchrodyne Networks, Inc., New York, from November 1999 until October 2001. He has been Guest Professor at Tongji University, Shanghai, China; Honorary Visiting Professor at La Trobe University, Melbourne, Victoria, Australia; Adjunct Professor at University of Illinois at Chicago; Visiting Professor at Institut de Technologie du Cambodge, Phnom Penh, Cambodia; and Visiting Researcher at the IBM T. J. Watson Research Center, Yorktown Heights, NY, at Columbia University, New York, NY, and at the International Computer Science Institute (ICSI), Berkeley, CA. He co-authored over 100 papers on various networking related topics and two books, one (currently at the second edition) on internetworking and one on switched local area networks. He is co-inventor of 16 patents issued by the United States Patent Office, one patent issued by the European Patent Office, and three pending applications to the United States Patent Office in the fields of high performance networking and software security. His research interests include internetworking, high performance switching, optical networking, quality-of-service, multimedia over packet networks, voice over IP, trust in distributed software execution, and computer networks in general.

Juan Carlos De Martin (M'92) received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 1996.

He spent two years (1993–1995) as visiting scholar at the Signal Compression Laboratory of the University of California Santa Barbara, and two years (1996–1998) in Dallas, TX, with Texas Instruments as Member of Technical Staff as well as an Adjunct Professor at the University of Texas (1999). Between 1998 and 2005, he was a principal researcher at the National Research Council (CNR) of Italy in Torino, where he led the multimedia communications research group. He is currently an Associate Professor at the Information Engineering School of the Politecnico di Torino, where he coordinates the Internet Media Research Group. His research activities are focused on multimedia processing and transmission, with a special emphasis on high-performance streaming techniques. He is also active in exploring the interaction between digital technologies and society. In this regard, in November 2006, he founded and currently directs the NEXA Center for Internet and Society of the Politecnico di Torino; he is also the coordinator of COMMUNIA, the European thematic network on the digital public domain funded by the European Commission (2007–2010). He is the author or co-author of over 80 international scientific publications and of several patents; he is also an expert evaluator of research programs for the Italian Ministry of University and Research, for the Ministry of Industrial Activities, and for the Swiss Science Foundation.

Dr. De Martin serves as a member of the IEEE Multimedia Communications and of the IEEE Signal Processing Education Technical Committees.