Joint Source Coding and Packet Classification for Real-Time Video Transmission over Differentiated Services Networks

Fan Zhai, Carlos E. Luna, Yiftach Eisenberg,
Thrasyvoulos N. Pappas, Randall Berry, and Aggelos K. Katsaggelos*
Department of Electrical and Computer Engineering
Northwestern University
2145 Sheridan Road, Evanston, IL 60208, USA
Email: {fzhai, carlos, yeisenbe, pappas, rberry, aggk}@ece.northwestern.edu

*Correspondence: Prof. Aggelos K. Katsaggelos
Department of Electrical and Computer Engineering
Northwestern University
2145 Sheridan Road, Evanston, IL 60208, USA
Phone: (847) 491-7146
Fax: (847) 491-4455
Email: aggk@ece.northwestern.edu
EDICS: 5-HIDE; 5-QOSV; 5-STRM

Abstract

Differentiated Services (DiffServ) is one of the leading architectures for providing quality of service in the Internet. We propose a scheme for real-time video transmission over a DiffServ network that jointly considers video source coding, packet classification, and error concealment within a framework of cost-distortion optimization. The selection of encoding parameters and packet classification are both used to manage end-to-end delay variations and packet losses within the network. We present two dual formulations of the proposed scheme: the minimum distortion problem in which the objective is to minimize the end-to-end distortion subject to cost and delay constraints, and the minimum cost problem which minimizes the total cost subject to end-to-end distortion and delay constraints. A solution to these problems using Lagrangian relaxation and dynamic programming is given. Simulation results demonstrate the advantage of jointly adapting the source coding and packet classification in DiffServ networks.

Keywords

Multimedia streaming, error resilience, error concealment, QoS, unequal error protection (UEP), optimal resource allocation, joint source-channel coding.

I. Introduction

Continuous media (CM) Internet applications, such as streaming video and videoconferencing, are rapidly increasing in popularity. However, these applications can be difficult to deploy due to the lack of quality of service (QoS) guarantees in today's Internet. Several approaches have been developed to improve the QoS support in the Internet. One of the major standardization efforts in this area is the Differentiated Services (DiffServ or DS) architecture developed by the Internet Engineering Task Force (IETF) [1]. In this paper, we study the interaction of video source coding and network layer QoS in a DiffServ network. Our goal is to obtain an optimal balance between received video quality and overall cost.

Traditional "elastic" Internet applications such as web browsing and electronic mail are relatively insensitive to delay variations but require reliable transmission, which can be achieved through retransmissions. In contrast, CM applications, especially real-time applications, such as low-delay video conferencing and video telephony, are much more sensitive to delay but more tolerant to packet loss. The best-effort design of today's Internet does not always accommodate the QoS requirements of CM applications. The incompatibility between the nature of the current Internet and the QoS requirements for

CM applications has led to proposed modifications of the Internet itself [2–4]; the DiffServ architecture is considered one of the most viable approaches due, in part, to its scalability and flexibility.

DiffServ is a Layer 3 (IP layer) approach for supporting QoS by discriminately allocating resources to aggregated traffic flows according to service classes. Specifically, each packet is assigned a priority tag (a DS "code-point"), indicating a QoS class to which the packet belongs. Upon arriving at a router, a packet is queued and forwarded based on its assigned class. As a consequence, a router can provide different "per hop behaviors" to each aggregated traffic class. These per hop behaviors lead to an end-to-end statistical differentiation between the QoS of each class [1, 2, 4].

In this work, we consider the video traffic to be generated by a hybrid motion compensated video encoder, as in H.263 or MPEG-4. In these standards, a given macroblock (MB) can be intraframe coded, interframe coded using motion compensated prediction, or simply replicated from the previously decoded frame. These prediction modes are denoted as INTRA, INTER, and SKIP mode, respectively. Quantization and coding are performed differently for each MB according to its mode. Thus, the coding parameters for each MB are typically represented by its prediction mode and quantization parameter. To overcome the difficulties brought about by a lossy packet channel, source encoding should be designed taking into account channel characteristics to make it more error resilient [5–13]. For Internet video, error resilience usually includes selecting the optimal prediction mode and quantization parameter for each MB. Different encoding modes result in different levels of coding efficiency and robustness to packet loss [5–7, 13].

We consider real-time video transmission applications, where packet retransmissions are not applicable due to the application's strict delay requirements. In this setting, we present a framework for choosing both the source coding parameters as well as the appropriate DS class for each packet. Different DS classes result in different QoS levels, such as different probabilities of packet loss and network delay. The goal is to find a *cost-distortion* optimized selection of parameters. This framework can be viewed as a type of joint source-channel coding, or more specifically, unequal error protection (UEP), where the "channel coding" role is provided by the choice of DS class.

Several related approaches have been discussed in the literature. In [3], an adaptive packet forwarding mechanism was proposed for a DiffServ network where video packets are mapped onto different DiffServ service levels. However, the framework in [3] does not incorporate video source coding decisions. The authors in [14] proposed a rate-distortion optimized packet marking technique to deliver MPEG2 video sequences (only INTRA frames were used in this work) in a DiffServ IP network. Their goal was to minimize the bandwidth consumption in the premium class while achieving nearly constant perceptual quality. This work was extended by taking into account inter-frame motion compensation in [15]. Neither [14] nor [15] considered the selection of source coding parameters. In [16] and [17], cost-distortion optimized multimedia streaming over DiffServ networks was studied. Although the proposed framework in [16] and [17] is very general, it is based on pre-encoded media. Thus, the selection of encoding parameters is not considered. In addition, error concealment is not included. The work presented in this paper builds on the earlier work in [18], which considered cost-distortion optimized streaming in a simpler setting.

We present a novel framework for joint adaptation of source coding and packet priority assignment to maximize the system performance. Our formulation incorporates the random network delay for each packet into the calculation of the probability of packet loss; this delay is managed through selecting the source coding parameters (prediction mode and quantization stepsize) and packet priority. More specifically, finer quantizers lead to higher video reconstruction quality but longer delay (given the same QoS class), which results in higher loss probability. In addition, the packet QoS class also needs to be selected in a way to properly balance cost, delay, and video quality. For example, packets that are hard to conceal but can be easily encoded should use coarser quantizers and a higher QoS class. Packets that are easily concealable can be sent using a lower QoS class. For packets that are hard to encode, the best choice may be a finer quantizer and a higher QoS. The goal is to minimize the end-to-end distortion subject to cost and delay constraints, or alternatively, to minimize the overall cost given end-to-end distortion and delay constraints. Such a goal is achieved through joint selection of source coding parameters and QoS classes in order to optimally balance the received video quality and the overall cost. Our contributions include: 1) a comprehensive framework for real-time video transmission over DiffServ networks; 2) a solution approach based on Lagrangian relaxation and dynamic programming for solving optimization problems with two constraints; 3) the development of a tree-pruning technique for efficiently implementing dynamic programming (DP).

The major components of the proposed framework are described in Sect. II. In Sect. III, we present problem formulations for both the minimum distortion approach and the minimum cost approach. Section IV provides a detailed description of the algorithm used to solve the optimization problem. Simulation results and discussion are reported in Sect. V. We draw conclusions and speculate on potential future work in Sect. VI.

II. VIDEO TRANSMISSION SYSTEMS

Figure 1 depicts a block diagram of the proposed video transmission system. On the sender side, a raw bit-stream of live video is continuously fed into the video encoder, which generates a stream of encoded video packets. The controller assigns both coding parameters and a service class to each video packet based on the QoS information associated with the service classes, the error concealment strategy used at the decoder, and the fullness of the encoder buffer. The video packets are placed into a first-in-first-out (FIFO) encoder buffer until they are transmitted. After passing through the network protocol stack (e.g. RTP/UDP/IP), the packets are transmitted over a DiffServ network. Some packets may be dropped in the network (due to congestion) or at the receiver (due to excessive delay). Packets that reach the decoder in time are buffered. The video decoder reads video packets from the decoder buffer and displays the resulting video frames in real-time (i.e., the video is displayed continuously without interruption at the decoder). Lost packets are concealed at the decoder.

A. DiffServ Traffic Classes

Each video packet is classified into one of the available DiffServ traffic classes before being transmitted over the network. There are several parameters associated with each class. First, each class has a specified packet loss probability and a packet delay distribution. These parameters may be specified in a service level agreement (SLA) between the Internet service provider (ISP) and the users [2, 3] or estimated via feedback from the

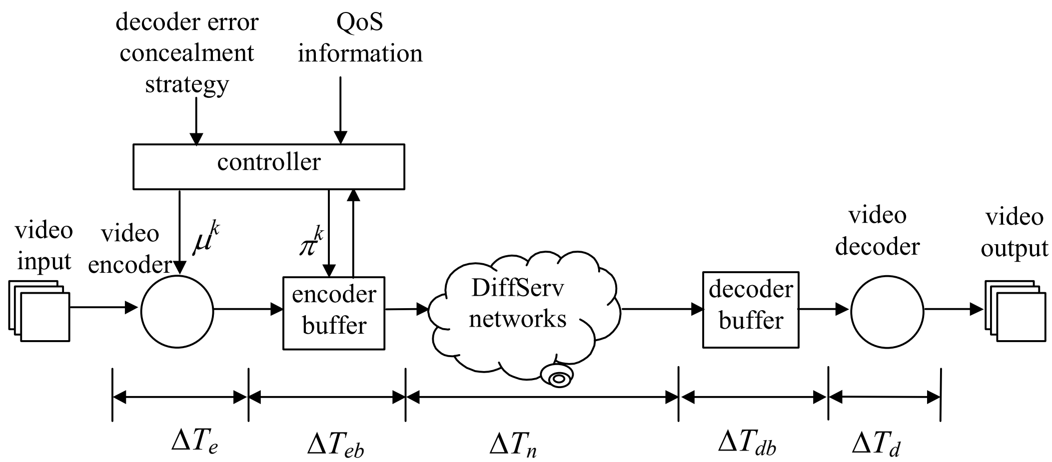

Fig. 1. Video transmission system block diagram.

network (e.g., using Real-time Transport Control Protocol (RTCP)). Also, each class is associated with a price per bit transmitted. These prices can also be prespecified in an SLA and are used by the service provider to achieve efficient network resource utilization. Typically, transmitting a packet in a higher priority service class results in a higher cost but a better QoS (lower delay and loss probability). The sender has to classify each packet according to its importance in order to better utilize the available network resources.

Each class also has a "rate" constraint limiting the amount of traffic that can be sent in the class. This can again be specified in the SLA and enforced at the ingress routers. Alternatively, the rate constraint could be estimated via feedback from the network as in TCP-Friendly protocols [12, 16]. In the experiments in Sect. V we use a model-based TCP-friendly congestion controller, where the throughput per class is estimated using a stochastic TCP model based on the steady-state packet loss probability and round-trip time [12, 16, 19]. Such rate constraints limit the throughput per class over a specified time-scale. A user can ensure that its traffic confirms to this constraint by using a traffic regulator, such as a Token Bucket algorithm.

In order to develop a tractable optimization model, we assume that all encoded video packets (for every class) are transmitted FIFO from a single buffer, as shown in Fig. 1. To model the rate constraint in this setting, we assume that a higher level controller specifies an *effective rate*, $R(\pi)$ for each class π. When a packet containing B bits is transmitted, the next packet cannot be sent until $B/R(\pi)$ seconds later (independently of the class of the next packet). Thus, in this context $R(\pi)$ should not be interpreted as the rate constraint for class π, but as a parameter that is adjusted by a congestion controller to ensure that any rate constraint is satisfied over a longer horizon. The actual transmission time of the packet will be B/R_l, where R_l is the link transmission rate which is typically much greater than $R(\pi)$. Thus $B/R(\pi) - B/R_l$ is the time during which the transmitter is idle (this models, for example, the time the transmitter must wait for a token to be available with a Token Bucket regulator). During a given interval of time, this ensures that the average rate of packets in class π is upper bounded by $R(\pi)$, where this bound is approached if every packet is in class π. When packets are sent from multiple classes over a time-interval, then the upper bound $R(\pi)$ may not be met, due to the transmission of packets from the other classes. If the rate constraint for a given class is not met, then the higher level controller can adjust the effective rate for that class for future frames (depending on the time-scale over which the rate constraint is defined).

An alternative model would be to view the system as having one transmission buffer for each class, where each buffer is served at the effective rate for that class. Though, in principle, the following framework can be extended to this setting, there are two difficulties with this approach: First, the packets in the actual system are all transmitted over the same link, which a multiple buffer model does not capture. Second and more importantly, the complexity of the resulting optimization problem increases exponentially with the number of queues and quickly becomes intractable. The single buffer model greatly simplifies the problem and also has the benefit of ensuring that all packets are more evenly spaced out over time. Also, we emphasize that we are discussing the optimization model. The actual system need not transmit packets in this manner, as long as the model provides a bound on the resulting performance.

B. Delay Components

In a video transmission system, the end-to-end delay (i.e., the time between when a frame is captured at the encoder and displayed at the decoder) should be constant, if the encoder and decoder are to operate at the same frame rate of F frames per second [20].

As shown in Fig. 1, the end-to-end delay T of each frame can be decomposed into

$$T = \Delta T_e + \Delta T_{eb} + \Delta T_n + \Delta T_{db} + \Delta T_d, \tag{1}$$

where ΔT_e, ΔT_{eb}, ΔT_n, ΔT_{db}, and ΔT_d are respectively the encoder delay, encoder buffer delay, network delay, decoder buffer delay, and decoder delay for each frame [18, 20]. Mechanisms, referred to as *rate control*, are needed to perform bit allocation among frames to ensure that the encoder and decoder buffer do not overflow or underflow. Examples of rate controllers include TM5 [21], and the work for variable bit-rate channels in [22, 23].

We assume that a rate controller specifies a deadline by which each frame must be transmitted. We translate the frame deadline into a delay constraints for each packet. We then proceed to study the problem of allocating bits among packets and specifying QoS classes given these delay constraints. Let M be the number of packets in a video frame and k the packet index. Without loss of generality, we assume that the processing times for both encoding and decoding a packet are constant and equal to $T_p = 1/(MF)$. Note that the k-th packet enters the encoder buffer after the previous $k-1$ packets and itself have been processed (i.e., at time kT_p). In addition, in order for the corresponding frame to be displayed on time, this packet must arrive at the decoder in time to allow itself and the following $M-k$ packets to be processed (i.e., $(M-k+1)T_p$ before display) [18, 20]. Thus, for each packet to experience constant end-to-end frame delay, it must be that

$$\Delta T_{eb}(k) + \Delta T_n(k) + \Delta T_{db}(k) = T - kT_p - (M-k+1)T_p = T - (M+1)T_p. \tag{2}$$

In order to avoid decoder buffer underflow, i.e., to satisfy $\Delta T_{db}(k) \geq 0$, the total encoder buffer delay and network delay must be

$$\Delta T(k) = \Delta T_{eb}(k) + \Delta T_n(k) \leq T_{max}, \tag{3}$$

where $T_{max} = T - (M+1)T_p$. For simplicity, we assume a sufficiently large decoder buffer, so that we do not need to consider decoder buffer overflows.

As discussed in Sect. II-A, when leaving the encoder buffer, packets are transmitted at the link transmission rate, R_l, but the transmitter is required to idle according to the effective transmission rate of the given service class. Thus, from the point of view of the controller, we model packets from each QoS class, π^k, as being actually transmitted at

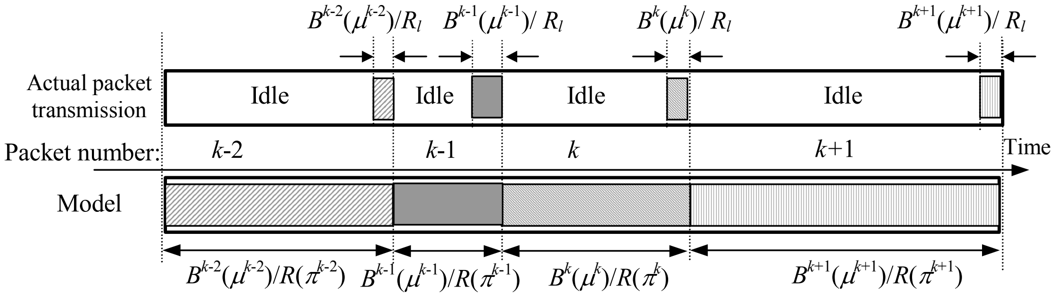

Fig. 2. Model of packet transmission behavior in the encoder buffer. The length of each block corresponds to the transmission time of the packet.

the effective transmission rates, $R(\pi^k)$. Thus, a packet containing B^k bits has an effective transmission time of $B^k/R(\pi^k)$, as shown in Fig. 2. [1]

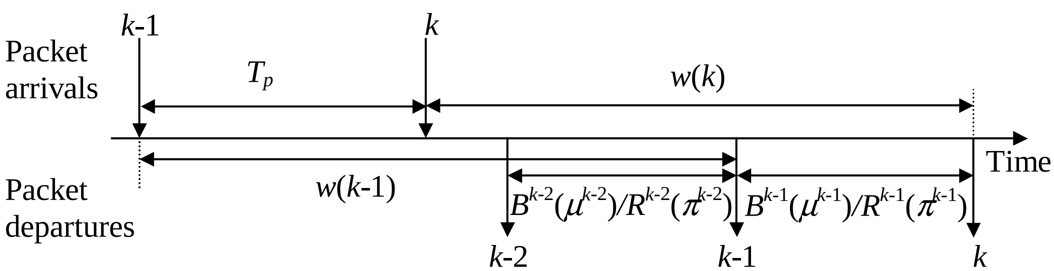

Fig. 3. Illustration of the buffer delay calculation for each packet. The top arrows indicate the time at which a packet arrives at the encoder buffer, the lower arrows indicate the time at which the packet departs (including transmission time).

Next, based on this model, we specify the maximum allowable network delay for each packet, $\tau(k)$, i.e., if $\Delta T_n(k) > \tau(k)$, the packet will be lost due to excessive delay. Let $w(k)$ be the waiting time in the encoder buffer for the k-th packet before it is transmitted.The encoder buffer delay (waiting time plus transmission delay) for the k-th packet $\Delta T_{eb}(k)$ can be written as

$$\Delta T_{eb}(k) = w(k) + \frac{B^k(\mu^k)}{R^k(\pi^k)}, \tag{4}$$

where $B^k(\mu^k)$ and $R^k(\pi^k)$ are the packet length in bits and the transmission rate in bits/sec for packet k respectively. The k-th packet has a particular class $\pi^k \in \Pi$ and coding parameter $\mu^k \in \mathcal{Q}$, where Π and $\mathcal{Q}$ are the set of available service classes and source

[1]Note, in calculating the delay, we assume that the idle time occurs before the actual packet transmission. The model could easily be modified for the case where the idle time follows the packet transmission.

coding parameters (quantization step size and prediction mode), respectively. From (3), the maximum allowable network delay for packet k is then

$$\tau(k) = T_{max} - w(k) - \frac{B^k(\mu^k)}{R^k(\pi^k)}. \tag{5}$$

As shown in Fig. 3, the waiting time for the k-th packet can be recursively calculated as

$$w(k) = w(k-1) + \frac{B^{k-1}(\mu^{k-1})}{R^{k-1}(\pi^{k-1})} - T_p. \tag{6}$$

C. Channel Model

We model the network as an independent time-invariant packet erasure channel with random delays, as in [16, 17]. A packet is considered lost if it does not arrive at the decoder on time. Thus, the packet loss probability is made up of two components: the packet loss probability in the network and the probability that the packet experiences excessive delay. Combining these two factors, for each service class $\pi \in \Pi$, the overall probability of packet loss is

$$\rho_\pi^k = \epsilon_\pi + (1 - \epsilon_\pi) Pr\{\Delta T_n(k) > \tau(k)\}, \tag{7}$$

where ϵ_π is the probability of packet loss in the network for service class π and $\tau(k)$ is the maximum allowable network delay for packet k.

Packet losses in the network can be modeled in various ways, e.g., a Bernoulli process, a 2-state or k-th order Markov chain [24]. The network delay may also be randomly varying and follow a self-similar law where the underlying distributions are heavily-tailed rather than following a Poisson distribution [25, 26]. The proposed framework is general and not limited to any specific packet loss or network delay model. All that is needed is a stochastic model of the packet losses and delays. In addition, we assume that the service classes are independent, and that the delay of each service class is i.i.d. for each transmitted packet.

D. Packet Loss and Delay Model

In our simulations, packet loss in the network is modeled by a Bernoulli process, i.e., each packet is independently lost with probability ϵ_π. When DiffServ is employed, ϵ_π differs from class to class, with a lower probability of packet loss for higher cost classes. This can be achieved by priority packet dropping schemes implemented in the routers [2, 27].

For simplicity, the network delay in our simulations is modeled as a *shifted Gamma distribution* with rightward shift γ_π and parameters n_π and α_π, defined as

$$f_\pi(\tau|\text{received}) = \frac{\alpha_\pi}{\Gamma(n_\pi)}(\alpha_\pi(\tau-\gamma_\pi))^{(n_\pi-1)}e^{-\alpha_\pi(\tau-\gamma_\pi)} \quad \text{for } \tau \geq \gamma_\pi. \tag{8}$$

This model arises when a flow is sent through n_π routers, each modeled as an M/M/1 queue with service rate α_π and a total end-to-end processing time of γ_π [16]. Here, π is the DS class index. Short-term estimates of n_π, γ_π, and α_π can be obtained by periodically estimating the mean and variance of the forward trip time. For more details, see [12, 16].

III. Problem formulation

In this section, we present the problem formulation for transmitting video over a DiffServ network. We assume that the sender is charged for each transmitted bit based on its service class. For example, the cost can be measured in dollars per byte as in [17, 18], or dollar per bit as in our experimental results. We want to minimize the overall end-to-end distortion while meeting total cost and delay constraints, or alternatively, to minimize the total cost given distortion and delay constraints.

A. Packetization and Error Concealment

We consider a packetization scheme where each row of MBs is coded as one packet, and every packet is independently decodable. The framework presented here can easily be extended to include other packetization schemes. The only requirement is that the packet boundaries are known *a priori*, i.e., which MBs are grouped into the same packet.

Without loss of generality, we consider a simple but efficient error concealment scheme similar to [5, 18]. When a packet is lost, the decoder first checks to see if the previous packet (the above row of MBs in our packetization scheme) was received. If it is, each MB of the corrupted packet is replaced with the MB from the previous frame indicated by the concealment motion vector. The concealment motion vector of a MB is the median of the motion vectors of the three closest MBs in the previous packet. If the previous packet is also lost, then the concealment motion vector is zero, i.e., the MB in the same spatial location in the previously reconstructed frame is used to conceal the current loss. Notice that for this packetization and error concealment scheme, the choice of prediction mode and the probability of loss for packet $k-1$ affects the distortion for the k-th packet if it is lost. We emphasize that the framework presented here is not limited to this specific concealment scheme.

B. Expected Distortion

To measure video quality, we consider the expected end-to-end distortion, where the expectation is taken with respect to the probability of packet loss (including losses due to delay). Specifically, we consider

$$E[D^k] = (1-\rho^k)E[D_r^k] + \rho^k E[D_l^k], \tag{9}$$

where $E[D^k]$ is the expected end-to-end distortion for the k-th packet, $E[D_r^k]$ is the expected distortion when the packet is received correctly, $E[D_l^k]$ is the expected distortion when the packet is lost, and ρ^k is the probability of loss of the k-th packet. Due to channel losses, the reference frames at the decoder and the encoder may not be the same. Therefore, both D_l^k and D_r^k are random variables.

In our simulations, the distortion measurement is based on per-pixel accurate distortion calculations, which ensure accurate estimation of the overall end-to-end distortion [5, 7, 13, 18]. For the packetization and concealment scheme discussed above, $E[D^k]$, can be described as

$$E[D^k] = (1-\rho^k)E[D_r^k] + \rho^k(1-\rho^{k-1})E[D_c^k] + \rho^k\rho^{k-1}E[D_z^k], \tag{10}$$

where $E[D_c^k]$ and $E[D_z^k]$ are the expected distortions after concealment when the previous packet is either received correctly or lost, respectively. If the distortion is defined using the mean squared error criterion, then the first and second order expected values of each pixel can be recursively calculated using Equations (2)-(7) described in [13], and the expected distortion for the k-th packet can be calculated by summing up the expected distortions of all the pixels in this packet, as shown by Eq. (1) in [13]. In our simulations, we assume that all the packets in the first frame are correctly received. Thus, based on the statistics of frame n, the expected pixel value of frame $n+1$ can be recursively calculated.

C. Minimum Distortion and Minimum Cost Problems

The first problem we consider is to provide the best quality (minimum end-to-end distortion) for given cost and delay constraints. We refer to this as the "minimum distortion problem". This problem can be formulated as

$$\min_{\{\boldsymbol{\mu},\boldsymbol{\pi}\in\mathcal{Q}\times\Pi\}} \sum_{k=1}^{M} E[D^k(\boldsymbol{\mu},\boldsymbol{\pi})] \tag{11a}$$

$$\text{s.t.} \quad \sum_{k=1}^{M} c^k(\pi^k)B^k(\mu^k) \le C_0(i) \tag{11b}$$

$$\sum_{k=1}^{M} B^k(\mu^k)/R^k(\pi^k) \le T_0(i), \tag{11c}$$

where $c^k(\pi^k)$ is the cost per bit of the k-th packet, M is the number of packets in a frame, $C_0(i)$ and $T_0(i)$ are the total cost constraint and transmission delay constraint for the i-th frame, respectively. We use $\boldsymbol{\mu} = \{\mu^1, \mu^2, ..., \mu^M\}$ and $\boldsymbol{\pi} = \{\pi^1, \pi^2, ..., \pi^M\}$ to denote the coding parameters and QoS classes for all the packets in a frame, respectively. The transmission delay constraint can be obtained from a higher-level rate controller, and may vary from frame to frame. Thus, bits can be traded off between frames based on how difficult they are to code. Note that $T_0(i)$ differs from T in (1), which is the required end-to-end delay. A higher-level rate controller assigns deadlines by which every frame should leave the encoder buffer. Assuming $T_{rc}(i)$ is the corresponding transmission delay budget for the i-th frame, $T_0(i)$ can be recursively obtained as

$$T_0(i) = \begin{cases} T_{rc}(0) & \text{if } i = 0 \\ T_{rc}(i) + T_0(i-1) - \sum_{k=1}^{M} \frac{B_{i-1}^k(\mu^k)}{R_{i-1}^k(\pi^k)} & \text{if } i \ge 1 \end{cases}, \tag{12}$$

where B_{i-1}^k and R_{i-1}^k denote the bits and transmission rate for the k-th packet in frame $i-1$ (Notice that for simplicity, we have not been using the frame index for B^k and R^k in the remaining of the paper.) Therefore, the smaller the transmission delay for a frame, the greater the time left for the next one to reach the receiver before the deadline set by the rate controller.

We will also consider the "minimum cost problem," i.e., minimizing the total cost (left hand side of (11b)) subject to an end-to-end distortion constraint ((11a) with a constraint) and a transmission delay constraint (11c). An analogous problem can be formulated in this case. This approach is useful when a desired level of video quality must be maintained.

Note that the proposed formulation is not limited to performing optimization on one frame only. The reason we chose to do so is mainly because of computational complexity considerations. In addition, for real-time CM applications, future frames are not always accessible when the current frame is being processed.

IV. Proposed Algorithm

In this section, we present a solution approach for the minimum distortion problem in (11) based on Lagrangian relaxation and deterministic dynamic programming (DP). The minimum cost problem can be solved in a similar fashion.

A. Lagrangian Relaxation

First, by introducing Lagrange multipliers $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ for the cost and delay constraints, respectively, the constrained problem (11) can be converted into an unconstrained Lagrangian problem given as,

$$\min_{\{\boldsymbol{\mu},\boldsymbol{\pi} \in \mathcal{Q}\times\Pi\}} L(\boldsymbol{\mu},\boldsymbol{\pi},\lambda_1,\lambda_2) = \sum_{k=1}^{M} \{E[D^k] + \lambda_1 c^k(\pi^k)B^k(\mu^k) + \lambda_2 B^k(\mu^k)/R^k(\pi^k)\}. \tag{13}$$

The solution of (11) can be obtained, within a convex hull approximation, by solving (13) with the appropriate choice of Lagrange multipliers, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$. This can be accomplished by using a variety of methods such as cutting-plane methods and subgradient methods [28]. The solution usually has high computational complexity.

Alternatively, by using only one Lagrange multiplier $\lambda \geq 0$ for the cost constraint, (11) is converted to

$$\begin{aligned} &\min_{\{\boldsymbol{\mu},\boldsymbol{\pi} \in \mathcal{Q}\times\Pi\}} \sum_{k=1}^{M} E[D^k(\boldsymbol{\mu},\boldsymbol{\pi})] + \lambda c^k(\pi^k)B^k(\mu^k) \\ &\text{s.t.} \quad \sum_{k=1}^{M} B^k(\mu^k)/R^k(\pi^k) \leq T_0(i). \end{aligned} \tag{14}$$

With an appropriate $\lambda \geq 0$, (11) can be solved within a convex hull approximation by solving (14). Since the objective function in (14) is a typical Lagrangian with one Lagrange multiplier, the appropriate $\lambda \geq 0$ can be found by using the bi-section iterative search or other fast search algorithms [29, 30]. It is worth noting that although (14) involves only one Lagrange multiplier, it does not mean that (14) is always easier to solve than (13) (Actually, in most cases, (14) is harder to solve than (13).) This is because extra dependency may be introduced by the constraint in (14) compared with (13), which complicates the optimization. However, if the extra dependency due to the constraint can be reduced, the advantage in solving (14) instead of (13) may be significant.

For simplicity, let $D(k)$, $C(k)$, and $T(k)$ denote the expected distortion, cost and transmission delay, for packet k, respectively. The Lagrangian in (14) can be expressed as $L(\boldsymbol{\mu}, \boldsymbol{\pi}, \lambda) = \sum_{k=1}^{M} J(k)$, where $J(k) = D(k) + \lambda C(k)$.

Next, we consider evaluating $J(k) = D(k) + \lambda C(k)$. Note that this Lagrangian is not separable because the distortion, $D(k)$, depends on the encoding modes and QoS classes chosen for the previous packets. From (7), (9), and (14), the cost of each packet $J(k)$ is a function of π^k, μ^k, $\tau(k)$ and $E[D_l^k]$. As shown in (5), $\tau(k)$ is a function of $w(k)$. In addition, $w(k)$ is recursively calculated from (6). Therefore we have

$$J(k) = J(\mu^1, \pi^1, ..., \mu^{k-1}, \pi^{k-1}, \mu^k, \pi^k), \tag{15}$$

i.e., the cost of each packet depends not only on its own coding and priority decision but also on the decisions for all previous packets. Ignoring the delay constraint, optimizing $\sum_{k=1}^{M} J(k)$ can be done via dynamic programming. However, because of the dependencies involved, this essentially results in an exhaustive search through all quantizers and priority choices. The time complexity of such a search is $O(|\Pi \times \mathcal{Q}|^M)$, where $|\Pi \times \mathcal{Q}|$ denotes the cardinality of the set, $\Pi \times \mathcal{Q}$ [29]. Although tree pruning techniques may be used in some cases, e.g., if the unconstrained problem satisfies monotonicity, the complexity reduction is still very limited [30]. In addition, the dependencies caused by the transmission delay constraint makes the DP problem more complicated, i.e., fewer branches can be pruned.

In the next section, we present a different DP algorithm, in which the packet waiting time $w(k)$ is used as part of the state description. This allows us to reduce the dependencies in both the objective function (the Lagrangian) and the constraint function in (14), making the proposed algorithm much more efficient.

B. Approximate Solution by Quantizing $w(k)$

As shown in (9), the error concealment scheme introduces dependencies between packets. Specifically, the calculation of $E[D_l^k]$ depends on the coding parameter and QoS class of the neighboring packet(s), if temporal concealment based on the motion vectors of neighboring packet(s) is used. For the particular error concealment scheme discussed above, $E[D_l^k]$ depends on the prediction mode (INTRA/SKIP or INTER) and QoS class π^{k-1}, for the previous packet, through the calculation of ρ^{k-1} and $E[D_c^k]$, as shown in (10). With the introduction of the packet waiting time $w(k)$, the cost of the k-th packet can be described as

$$J(k) = J(\mu^{k-1}, \pi^{k-1}, \mu^k, \pi^k, w(k)). \tag{16}$$

If we evaluate the objective function with respect to $w(k)$, evaluating the Lagrangian $J(k)$ in (16) will be much more efficient than in (15), due to the reduced dependencies. Note that $E[D_l^k]$ does not depend on which quantization parameter is used for the previous packet. This characteristic can be employed in an efficient tree-pruning algorithm.

Next, we discuss the dependency introduced by the transmission delay constraint in (14). Recall that now we are evaluating the DP problem with respect to the system state, $w(k)$. Let $s(k)$ be the accumulated transmission delay

$$s(k) = \sum_{j=1}^{k} T(j), \quad \text{for} \quad k = 1, ..., M. \tag{17}$$

The constraint (14) can be rewritten in terms of $s(k)$, using the fact that $s(k) = s(k-1) + T(k)$. Thus, the accumulated transmission delay up to packet k depends on which path is selected from state 1 to state $k-1$ only through $s(k-1)$ and the transmission delay for packet k, $T(k)$. The constraint in (14) is then rewritten as

$$s(k) \leq T_0, \quad \text{for} \quad k = 1, ..., M, \tag{18}$$

where for simplicity, we have ignored the frame index, i.e., $T_0(i) = T_0$. Recall that in our solution, the system state includes the packet waiting time. Equation (6) can now be written as

$$w(k) = w(k-1) + T(k-1) - T_p = w(1) + \sum_{j=1}^{k-1} T(j) - (k-1)T_p$$
$$= w(1) + s(k-1) - (k-1)T_p, \quad \text{for} \quad k = 2, ..., M+1. \tag{19}$$

Therefore, $w(k)$ and $s(k-1)$ have a one-to-one correspondence, and so the dependency in the constraint in (14) can be removed by using $w(k)$ as part of the system state. Based on (19), (18) can be expressed in terms of $w(k)$ as

$$w(k+1) - w(1) + kT_p \leq T_0, \quad \text{for} \quad k = 1, ..., M. \tag{20}$$

From the above observations, we propose a DP solution of (14) based on the use of $w(k)$ as part of the system state. Our approach is based on a similar technique used in [31].

Since $w(k) \in [0, T_{max}]$ is real-valued, the resulting state space is infinite. For computational reasons, we quantize $w(k)$ into a set of N_W values, $\mathcal{W} = \{w_0, w_1, ..., w_{N_W-1}\}$, with $w_i = (iT_{max})/(N_W - 1)$. Finer quantization of $w(k)$ leads to a better approximation of the optimal solution, at the cost of more computations. The effect of this approximation is to restrict the set of feasible choices for each system state. Therefore, the resulting solution will be a conservative approximation to the optimal solution. The optimization problem can now be re-formulated as

$$\min_{\{u(k) \in \mathcal{U}(w(k))\}} \sum_{k=1}^{M} J_k = \sum_{k=1}^{M} J(\mu^{k-1}, \pi^{k-1}, \mu^k, \pi^k, w(k))$$
$$\text{s.t.} \quad \mathcal{U}(w(k)) = \left\{ u(k) \in \Pi \times \mathcal{Q} : 0 \leq \frac{B^k(\mu^k)}{R^k(\pi^k)} + w(k) - T_p \leq \min\left(T_{max}, T_0 + w(1) - kT_p\right) \right\}, \tag{21}$$

where $\mathcal{U}(w(k))$ is the set of feasible choices, for the k-th packet at state $w(k)$.

C. Proposed Tree Pruning Technique

Figure 4 depicts the directed acyclic graph (DAG) of the state diagram. In this diagram, three stages corresponding to packets $k-1$ to $k+1$ are shown. At each stage, the possible system states[2] are represented by an encoder buffer waiting time ($N_W = 4$ in this example).

[2]To be precise, the system state at the k-th stage includes μ^{k-1} and π^{k-1}, but we suppress this to simplify our discussion.

Each branch in the graph corresponds to a choice from $\mathcal{U}(w(k))$ for the packet from which the branch starts. For each choice of $u(k) \in \mathcal{U}(w(k))$, the cost incurred for packet k is given by (21). Note that for a given branch, e.g., the branch starting from w_3 at packet k and ending at w_2 at packet $k+1$, the cost associated with this branch is unknown until the encoding mode μ^{k-1} and QoS class π^{k-1} of packet $k-1$ is known. The optimization is achieved by choosing the path through the trellis with the minimum cost among all feasible paths.

Due to the special structure of the objective function of (21), the relaxed problem can be solved using techniques from forward DP (i.e., the Viterbi algorithm). From the DP recursion, the M-dimensional optimization problem, which has time complexity $O(|\Pi \times \mathcal{Q}|^M)$, is reduced into M 3-dimensional optimization problems, where M is the number of packets in a frame [29]. Notice that in (16), for each $w(k)$, there are at most $|\Pi \times \mathcal{Q}|$ possible choices for (μ^{k-1}, π^{k-1}) and also $|\Pi \times \mathcal{Q}|$ choices for (μ^k, π^k). Thus, the complexity in evaluating (21) is $O(M \cdot |\mathcal{W}| \cdot |\Pi \times \mathcal{Q}|^2)$. In addition, because the cost associated with a branch depends only on the prediction mode and QoS class of its previous packet, the solution can be further simplified by using the following tree pruning algorithm.

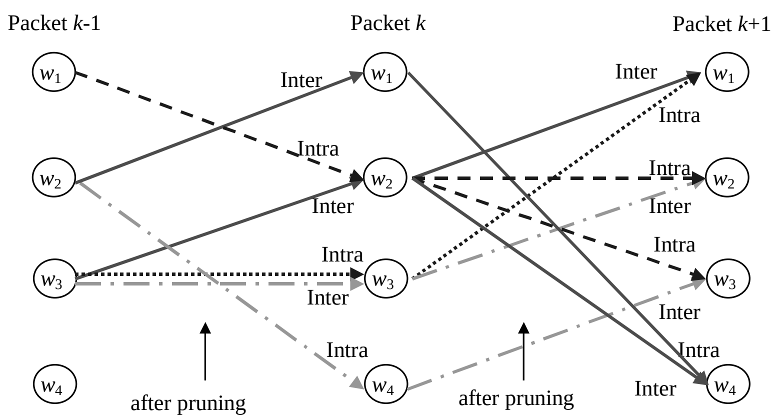

Fig. 4. Illustration of proposed tree pruning algorithm.

As shown in Fig. 4, we focus on how to prune between packet k and $k+1$, assuming the pruning is already done between packet $k-1$ and k. The cost of each branch emanating from $w(k+1)$ depends only on the prediction mode for packet k and not on its quantization parameter. Therefore, for all feasible branches $u(k) \in \mathcal{U}(w(k))$ ending at the same state, we can prune out all choices except the one with minimum cost associated with INTRA/SKIP mode and the one with minimum cost associated with INTER mode for each QoS class. For simplicity, only one branch with INTRA/SKIP and one with INTER mode are shown in Fig. 4. Figure 4 shows the remaining paths after pruning has been completed for packets k and $k+1$. The branches with the same line style belong to one path. For each state of packet $k+1$, there are at most $1+|\Pi|$ incoming paths, one associated with INTRA/SKIP mode for packet k, and $|\Pi|$ with INTER mode for each QoS class for packet k.

In going through the above steps, starting from the first packet and moving toward the final packet in a frame, we always keep at most $(1+|\Pi|)\cdot|\mathcal{W}|$ paths at each stage of the DP algorithm. The optimal solution for the frame is obtained by forward DP. Thus the time complexity[3] is further reduced from $O(M\cdot|\mathcal{W}|\cdot|\Pi\times\mathcal{Q}|^2)$ to $O(M\cdot(1+|\Pi|)\cdot|\mathcal{W}|\cdot|\Pi\times\mathcal{Q}|)$.

V. Experimental results

In this section, we report experimental results that demonstrate the performance of the proposed formulation. The simulations are based on the ITU H.263+ codec [32]. We consider the Foreman sequence in QCIF (176×144) format with frame rate $F = 30$ fps. Each row of MBs is encoded as a separate packet. The packets can be transmitted at different priorities specified as $\Pi = \{1, 2, 3, 4\}$, whose parameters are defined in Table I. Each class has a different transmission rate and loss probability. We choose these parameters using a model-based TCP-friendly congestion controller as in [12, 16]. The network delay is modeled by the shifted Gamma distribution shown in (8). The costs for each class are set proportional to the average throughput of the class, which takes into account the transmission rate, probability of packet loss, and network delay distribution. As for the quantizer set $\mathcal{Q}$, we consider quantization steps $\{8, 12, 18, 24\}$ for INTRA mode, $\{4, 6, 8, 10\}$ for INTER mode, and SKIP mode. We set $T_{max} = 333$ ms and $N_W = 300$. In the simulations, distortion is defined as the mean squared error (MSE) between the original and reconstructed sequence, and we evaluate the video quality in terms of PSNR, which is defined by PSNR$= 10\log\frac{255^2}{\text{MSE}}$ dB.

[3]Note the pruning itself has a complexity of $O(M\cdot|\mathcal{W}\times\Pi\times\mathcal{Q}|)$.

class	1	2	3	4
probability of packet loss	0.2	0.1	0.05	0.001
transmission rate (Kbps)	210	280	350	420
cost(microcents per kilobits)	25	50	75	100
γ (milliseconds)	40	30	20	10
n	2	2	2	2
α (1/milliseconds)	1/40	1/30	1/20	1/10
mean delay(milliseconds)	120	90	60	30

TABLE I

Parameters of four service classes.

A. Reference Systems

To illustrate the advantage of jointly selecting the source coding parameters and QoS class, we use a reference system where only one QoS class is available. In this reference system, source coding decisions are made to minimize the expected end-to-end distortion subject to the transmission delay constraint (11c). The transmission delay constraint for the i-th frame, $T_0(i)$, is recursively obtained as in (12). Since our focus here is not on rate control, we set $T_{rc}(i) = 1/F$ seconds for all frames. Note that the reference system is also a contribution of this work, since it is an optimized system based on the proposed formulation, which is of great use by itself in applications with only one QoS class. It will outperform any system based on heuristics.

B. Experiments

In this subsection, we compare the proposed DiffServ approach, with multiple QoS classes (11), to the reference system, which uses only one QoS class. We consider four reference systems, each of which uses only one of the four service classes. Each reference system generates a different optimized distortion $D_0(i)$, as well as the corresponding cost $C_0(i)$ and delay $T_0(i)$. The results from reference systems are used as constraints ($C_0(i)$ and $T_0(i)$ for the minimum distortion approach; $D_0(i)$ and $T_0(i)$ for the minimum cost approach) for the corresponding DiffServ system. In other words, we are comparing the four reference systems with four different optimized DiffServ systems, with matching constraints in each case.

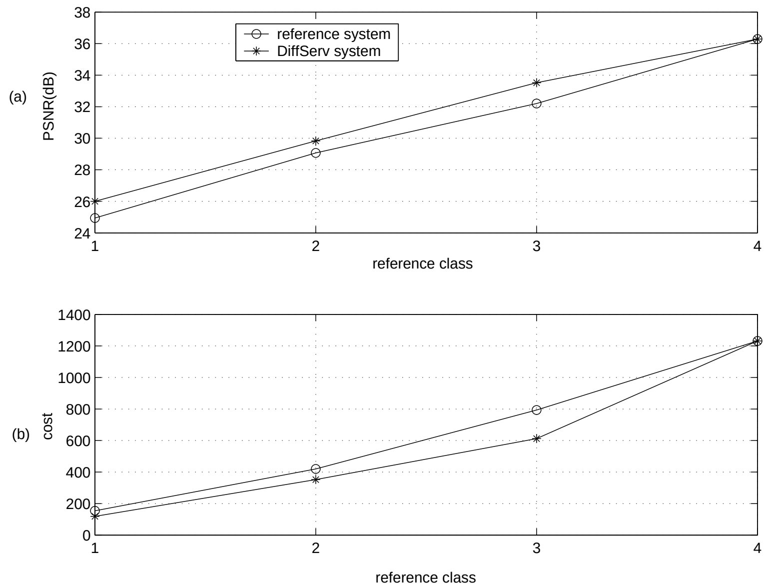

Fig. 5. Comparison of DiffServ approach with reference system: (a) Minimum distortion approach (b) Minimum cost approach.

First, we compare the proposed minimum distortion approach, with multiple service classes, to the reference systems. We illustrate the performance of the two systems in Fig. 5(a), which shows the average decoded PSNR for the Foreman sequence (300 frames) averaged over 50 random channel error realizations (We obtained almost identical results by considering the expected PSNR calculated at the encoder.) The DiffServ approach outperforms the corresponding reference systems by 1.06 dB, 0.76 dB, 1.31 dB, and 0 dB of average PSNR, respectively. We also compare the proposed minimum cost approach, with multiple service classes, to the reference systems. This is shown in Fig. 5(b), where the average cost per frame for the Foreman sequence is plotted. In this case, for the same distortion and delay, the minimum cost approach has a cost saving of 23%, 18%, 20%, and 0%, over the corresponding reference system, respectively. In our preliminary work [18], we have observed similar performance gains using different parameters. Of course, different choices of the parameter sets in Table I may result in different gains.

Next, we show the temporal behavior of these approaches for one channel error realization. Figure 6 shows the PSNR per frame for one channel error realization of the minimum

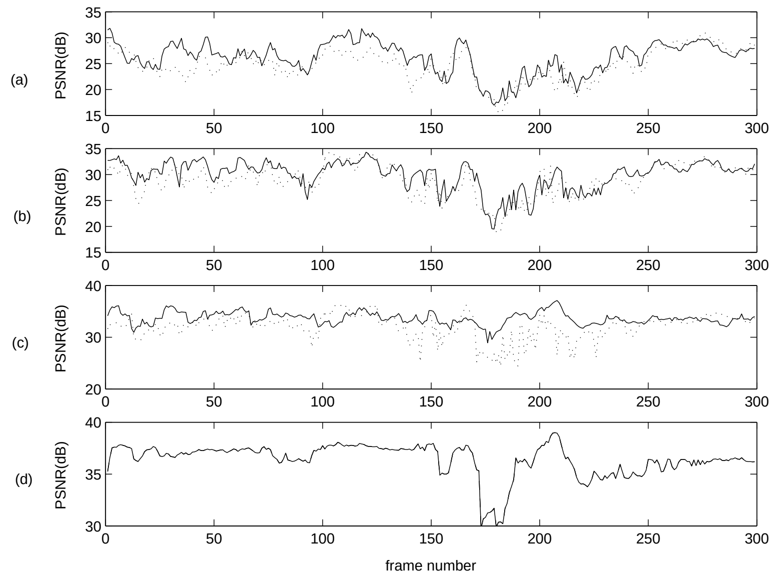

Fig. 6. One channel realization of minimum distortion approach (solid lines) and reference system (dotted lines), with reference (a) class 1, (b) class 2, (c) class 3, (d) class 4.

distortion approach and its corresponding reference systems. Figure 7 shows the cost per frame for the minimum cost approach and its corresponding reference systems. In Fig. 8, the number of packets that the DiffServ approach allocates to each service class is shown for each reference system.

As shown in these figures, the proposed DiffServ system, which jointly adapts the source coding and packet classification, can significantly outperform an approach which uses a fixed service class, except when the fixed class is class 4. When the reference system is class 2 or 3, the performance gain is mainly due to the flexibility in choosing the service class per packet. As shown in Fig. 8, the DiffServ approach allocates a significant number of packets to each of the available service classes for systems 2 and 3. If the reference system is class 1, the DiffServ approach still provides significant gains. As shown in Fig. 8, nearly all packets are still assigned to class 1, although a few are assigned to the higher service classes. Even this slight flexibility in priority assignment enables the DiffServ system to improve performance. This is because the packets that are assigned to higher QoS classes have smaller transmission delays. As a result, the encoder buffer occupancy is reduced, and the probability of packet loss due to excessive delay has less effect than if all packets are transmitted at the slowest rate (class 1). Under this situation, the transmission delay

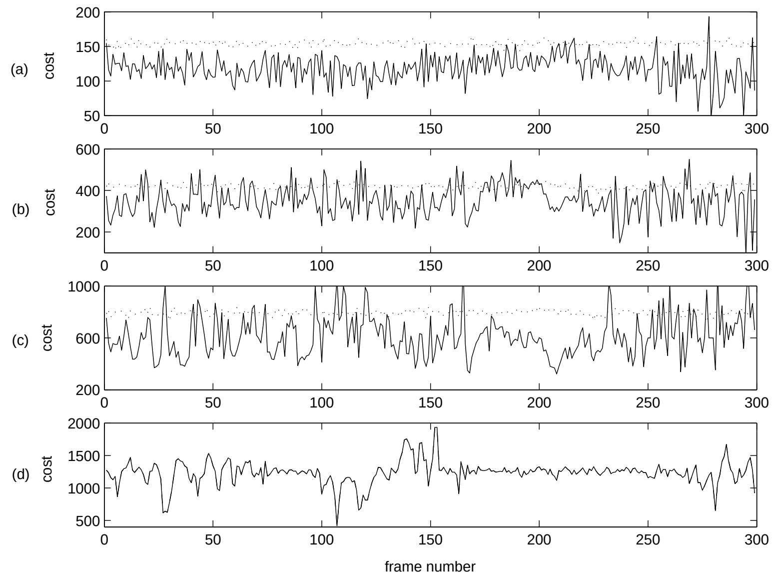

Fig. 7. One channel realization of minimum cost approach (solid lines) and reference system (dotted lines), with reference (a) class 1, (b) class 2, (c) class 3, (d) class 4.

constraint is not always active in the optimization problem (11).

Unlike reference system 1, 2, and 3, the DiffServ approach does not provide any gain over reference system 4. In this case, the reference system transmits all packets at the highest QoS. Hence, the corresponding constraints are such that there is no advantage in sending a packet at a lower service class; this will only result in longer transmission delay and higher probability of loss. Therefore, as shown in Figs. 6-8, the DiffServ approach ends up with the same solution as reference system 4. Similar gains are obtained when the Akiyo, Container, Coastguard, and Mother and daughter sequences are tested, as shown in Table II.

To conclude, the gains illustrated in the above experiments are mainly due to the use of multiple QoS channels in the proposed system. Specifically, for packets that are hard to conceal but easily encoded, the encoder tends to use coarser quantizers and higher QoS. These packets usually correspond to high motion areas that can be predicted from the previous frame. If such a packet is lost, it may be hard to conceal since the associated motion vector is lost. For packets that are hard to encode, the encoder may want to use a finer quantizer to reduce the quantization error and send them with higher QoS as well. The associated cost in this case is relatively high. These packets are usually hard

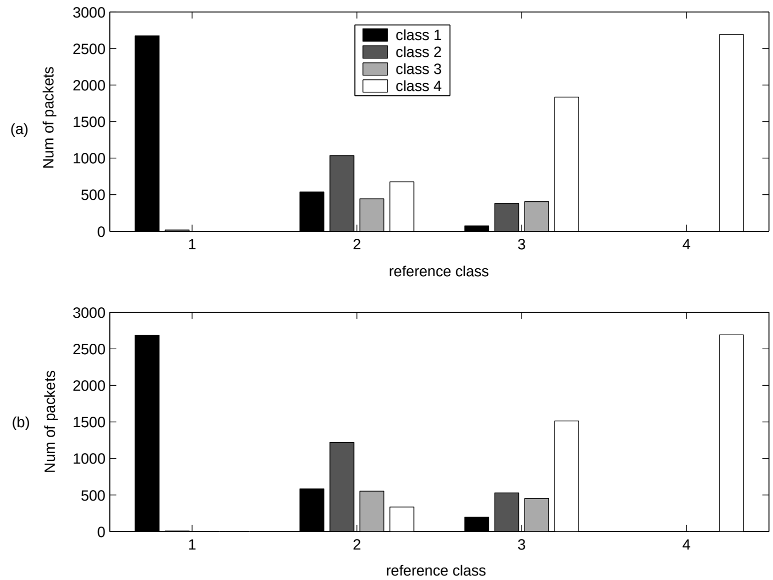

Fig. 8. Distribution of packet classification in the DiffServ system: (a) Minimum distortion approach (b) Minimum cost approach.

to predict from the previous frame, and are therefore difficult to conceal as well. Packets that are easier to conceal can be sent using a lower QoS class. For these packets the cost is relatively low. In this way, the encoder can select different quantizers and QoS classes in order to optimally balance the received video quality and the overall cost.

VI. Conclusions and future work

We have proposed a cost-distortion optimization framework for real-time video transmission over DiffServ networks. Our approach jointly considers error resilient source coding, packet classification and error concealment. Due to the random network delay in the Internet, end-to-end packet delay is not completely controllable by the transmitter. It is managed in the proposed framework through provisioning based on the fullness of the encoder buffer. By jointly adapting the source coding and packet classification, the optimization results in UEP for different packets, giving more protection to the most important parts of the bit-stream, thus achieving the maximum video quality at the receiver end.

Several issues remain to be elaborated on and are currently under investigation. First, the packet loss is modeled as a Bernoulli random process, which does not reflect the bursti-

Sequence	PSNR gains (dB) minimum distortion approach				Cost savings minimum cost approach			
	class 1	class 2	class 3	class 4	class 1	class 2	class 3	class 4
Foreman	1.06	0.76	1.31	0	23%	18%	20%	0%
Akiyo	0.92	1.91	0.82	0	31%	14%	12%	0%
Container	0.76	0.76	1.22	0	36%	16%	10%	0%
Coastguard	0.62	0.75	0.80	0	21%	19%	24%	0%
Mother and daughter	0.68	0.63	0.75	0	32%	11%	10%	0%

TABLE II

Average PSNR gains and cost savings of the proposed DiffServ systems compared with the corresponding reference systems (All sequences are in QCIF format, at 30 fps. Akiyo sequence has 100 frames, and each of the others has 300 frames.)

ness of errors due to network congestion. Two-state or higher order Markov chains are more realistic alternatives. In the future we also plan to investigate how different parameter settings affect the performance advantages of the proposed system. Additionally, a frame-level rate controller has not been incorporated into our proposed framework yet. Although rate control is usually designed separately from source coding and error control, it may improve the overall performance of a system if it is designed jointly with the other parts of the system. To the best of our knowledge, rate control in this context is still an open issue.

References

[1] S. Blake and et al., "An architecture for differentiated services," Tech. Rep., RFC 2475, IETF, Dec. 1998, http://www.rfc-editor.org/rfc/rfc2475.txt.

[2] B. E. Carpenter and K. Nichols, "Differentiated Services in the Internet," *Proc. IEEE*, vol. 90, no. 9, pp. 1479–1494, Sept. 2002.

[3] J. Shin, J. W. Kim, and C.-C. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Trans. on Multimedia*, vol. 3, no. 2, pp. 219–231, June 2001.

[4] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (DS field) in the Ipv4 and Ipv6 headers," Tech. Rep., RFC 2474, IETF, Dec. 1998, http://www.rfc-editor.org/rfc/rfc2474.txt.

[5] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and transmission power management for energy efficient wireless video communications," *IEEE Trans. on Circuits and System for Video Technology*, vol. 12, no. 6, pp. 411–424, June 2002.

[6] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video

communications over error-prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 952–965, June 2000.

[7] R. O. Hinds, T. N. Pappas, and J. S. Lim, "Joint block-based video source-channel coding for packet-switched networks," *Proc. SPIE*, vol. 3309, pp. 124–133, Jan. 1998.

[8] R. O. Hinds, *Robust model selection for block-motion compensated video encoding*, Ph.D Thesis, MIT, Cambridge, MA, June 1999.

[9] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 977–995, June 2000.

[10] Y. Wang, G. Wen, S. Wenger, and A. K. Katsaggelos, "Review of error resilience techniques for video communications," *IEEE Signal Processing Magazine*, vol. 17, pp. 61–82, July 2000.

[11] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc. IEEE*, vol. 88, pp. 1855–1877, Dec. 2000.

[12] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Trans on Multimedia*, vol. 3, no. 3, pp. 339–355, Sept. 2001.

[13] 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, pp. 966–976, June 2000.

[14] D. Quaglia and J. C. De Martin, "Delivery of MPEG video streams with constant perceptual quality of service," in *Proceedings of IEEE International Conference on Multimedia and Expro*, Lausanne, Switzerland, Aug. 2002, vol. 2, pp. 85–88.

[15] D. Quaglia and J. C. De Martin, "Adaptive packet classification for constant perceptual quality of service delivery of video streams over time-varying networks," in *Proceedings of IEEE International Conference on Multimedia and Expro*, Baltimore, MD, July 2003, vol. 3, pp. 369–372.

[16] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. on Multimedia*, 2001, Submitted.

[17] A. Sehgal and P. A. Chou, "Cost-distortion optimized streaming media over Diffserv networks," in *Proceedings of IEEE International Conference on Multimedia and Expro*, Lausanne, Switzerland, Aug. 2002, vol. 1, pp. 857–860.

[18] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint source coding and packet marking for video transmission over Diffserv networks," in *Proceedings of Tyrrhenian International Workshop on Digital Communications*, Capri, Italy, Sept. 2002, pp. 59–65.

[19] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," Tech report, International Computer Science Institutes, Berkley, CA, March 2000.

[20] C. Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, no. 5, pp. 756–773, May 1999.

[21] ISO/IEC, *MPEG Video Test Model 5*, ISO/IEC JTC1/SC29/WG11, MPEG93/457, Apr. 1993, Draft.

[22] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuit Syst. Video Technol*, vol. 2, pp. 361–372, Dec. 1992.

[23] B. G. Haskell and A. R. Reibman, "Multiplexing of variable rate encoded streams," *IEEE Trans. on circ. and Syst. for Video Techn.*, vol. 4, pp. 417–424, Aug. 1994.

[24] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE Infocom'99*, New York, NY, March 1999, vol. 1, pp. 345–352.

[25] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE Trans. Networking*, vol. 3, pp. 226–244, June 1995.

[26] G. Hooghiemstra and P. Van Mieghem, "Delay distributions on fixed Internet paths," Tech. Rep. report 20011020, Delft University of Technology, 2001.

[27] D. Ghosh, V. Sarangan, and R. Acharya, "Quality-of-service routing in IP networks," *IEEE Trans. on Multimedia*, vol. 3, pp. 200–208, June 2001.

[28] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[29] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, 1997.

[30] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[31] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint source coding and data rate adaptation for energy efficient wireless video streaming," *IEEE J. Select. Areas Commun.*, 2003, To appear.

[32] ITU, *Video Coding for Low Bitrate Communication*, ITU Telecom. Standardization Sector of ITU, Sept. 1997, Draft ITU-T Recommendation H.263 Version 2.