Packet Classification Schemes for Streaming MPEG Video over Delay and Loss Differentiated Networks

Wai-tian Tan and Avideh Zakhor
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
e-mail: {*dtan avz*}*@eecs.berkeley.edu*

Abstract

Differentiated services (DiffServ) is currently under investigation by IETF to provide relatively simple and coarse traffic differentiation in the Internet. In this paper, we present a framework for transmitting MPEG video by dividing the bit-stream into sub-streams of different delay and loss requirements. The sub-streams are then transported using multiple DiffServ traffic classes of different bandwidth, transmission delay, and packet loss characteristics. The use of multiple traffic classes to carry video improves network utilization as packets are transmitted in traffic classes with QoS commensurate to their requirements, rather than all in the "best" class. This is related to the existing use of scalable coding for loss-differentiated networks. However, in this work, we consider non-scalable MPEG video due to the abundance of existing content, and also consider loss and delay differentiations simultaneously. We present a number of packet classification schemes for MPEG bit-stream based on delay and loss characteristics of the data, and compare them using commercial DVD content. We simulate transmission of DVD movies over a DiffServ enabled network, and show a distortion reduction of over 4 dB using a packet classification scheme optimized for loss. Another packet classification scheme optimized for delay is shown to reduce end-to-end playback time by 30 ms as compared to packet classifiers that treat the MPEG stream as homogeneous.

1 Introduction

A large and ever-increasing body of digital video content is being stored in the form of MPEG bit-stream. In particular, content in the popular DVD and VCD formats is compressed using MPEG-2 and MPEG-1 respectively, making MPEG the dominant format for entertainment video. Nevertheless, MPEG is primarily designed for storage purposes, and as such, is ill-suited for transmission over the best-effort packet networks.

One possible approach to facilitate communication of MPEG is to augment best-effort packet networks with service differentiation. In such a way, traffic that requires better quality of service (QoS) can be preferentially treated as compared to traffic that does not. The Integrated Services (IntServ) model of the Internet, for instance, is an attempt to provide end-to-end QoS guarantees in terms of bandwidth, packet loss rate, and delay, on a per-flow basis. Availability of such QoS guarantees would greatly facilitate the streaming of existing video content and is achieved through explicit resource allocation using the Resource Reservation Protocol (RSVP). However, the high complexity and cost of deployment of the RSVP-based service architecture, together with its lack of scalability, eventually led the Internet Engineering Task Force (IETF) to consider other service differentiation mechanisms in the Internet. The Differentiated Services (DiffServ) model, in particular, is specifically designed to have low complexity and to be easily deployable, at the expense of more relaxed QoS guarantee than IntServ. Under DiffServ, service differentiation is no longer provided on a per-flow basis, but rather on aggregates of flows identified by different *code-points* or tags. Thus, packets with the same tag are given the same treatment under DiffServ regardless of where they originate from.

There are a number of ways to exploit service differentiation for video communication in a DiffServ Internet. Given a fixed number of traffic classes, one strategy is to assign different data types to different classes. For example, one traffic class may be used for video, one for interactive data applications such as web browsing, and one for non-interactive data transfer such as email and file transfer, as illustrated in Fig. 1-(a). By providing lowest loss and delay to video followed by interactive data and non-interactive data respectively, the user expectations for the different applications can be better met. Despite data traffic is loss sensitive, it can be transported using higher loss classes. This is because data is insensitive

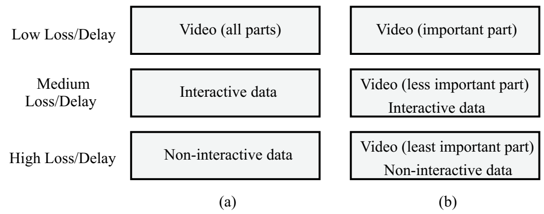

Figure 1: Transport of video under DiffServ using (a) single, and (b) multiple traffic classes.

to delay, and the use of end-to-end retransmission protocols such as TCP can effectively translate loss into additional delay and delay jitter.

Our approach in this paper is to use multiple traffic classes even within a single video application. This is achieved by decomposing a video bit-stream into multiple sub-streams of different delay and loss sensitivities, and assigning each sub-stream to a different traffic class with a different QoS characteristic, as illustrated in Fig. 1-(b). This "multi-class" approach generally achieves better utilization of network resources than the "single-class" approach due to the ability to transmit packets using traffic classes with commensurate QoS. Another advantage of the multi-class approach is the finely granular tradeoff between network resources and reception quality. This results in a spectrum of cost-performance tradeoff in transmitting video over DiffServ networks.

To facilitate transmission of the large body of existing MPEG content using multiple traffic classes in a DiffServ Internet, methods of classifying and tagging an MPEG bit-stream into different traffic classes are needed. In this paper, we present one such tagging method for transmission over a network that offers multiple traffic classes with different bandwidth, transmission delay, and packet loss characteristics. An overview of the proposed approach is presented is Section 2. Related work in the area of loss and delay differentiation for video communication is discussed in Section 3. Our proposed methods of providing delay and loss differentiation for MPEG video are discussed in Sections 4 and 5 respectively. Specific packet classification schemes for a loss and delay differentiated network are discussed in Section 6. Simulation results of transmitting a DVD movie over a loss and delay differentiated network are presented in Section 7. Finally, conclusions are given in Section 8.

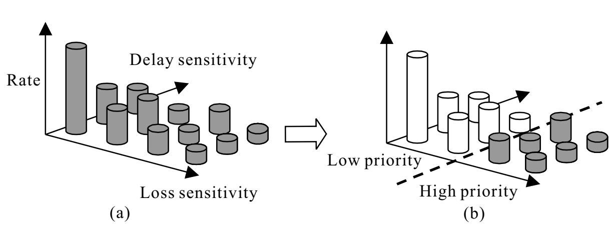

Figure 2: Using a map of delay-loss spread to classify traffic into low-loss and high-loss classes.

2 Tagging MPEG for DiffServ Networks

A DiffServ network typically provides multiple traffic classes with different QoS in terms of loss and delay. One possible approach to exploit such diversity of QoS is to transcode the MPEG video into alternative formats that can be easily mapped to the different QoS parameters. For instance, transcoding MPEG video into a layered representation [1, 2] would allow natural mapping of layers to traffic classes with increasing loss rates. In particular, the ability of some scalable compression techniques to produce more than two layers [3] further increases the level of achievable traffic differentiation. However, this transcoding approach suffers from three drawbacks: (1) high complexity, (2) video quality degradation, and (3) additional latency introduced by transcoding.

In this paper, we seek a transcoding-free approach to sending video using multiple traffic classes in a DiffServ network. Our approach can be summarized as follows. An MPEG bit-stream is first split into sub-streams with different delay and loss sensitivities. Different methods of computing such sensitivities are discussed later in Sections 4 and 5. In such a way, a map of loss-delay spread is obtained showing the amount of video data with each value of delay and loss sensitivity, as illustrated in Fig. 2-(a). This map is a property of the video only and can be computed independently of the traffic classes prior to transmission. Since the map orders different parts of an MPEG video according to delay and loss sensitivities, it can be used for packet classification and tagging in delay and loss differentiated networks. For example, Fig. 2-(b) shows the classification of the MPEG video bit-stream into two traffic classes with different loss priorities.

More generally, suppose a video server streaming MPEG video over a DiffServ network has at its disposal N traffic classes C_1 to C_N. Each of the classes C_i will have different QoS parameters typically expressed in terms of delay and loss, and an associated maximum bandwidth R_i. Suppose further that the network offers differentiation in loss only with increasing loss rate ranging from C_1 to C_N. Thus, class C_i is rate limited to R_i. Then, a packet tagging scheme that attempts to minimize the effect of packet loss would proceed by first classifying the most loss-sensitive data to class C_1 until the available bandwidth R_1 is reached. It will then continue traffic classification by tagging and assigning the most loss-sensitive part of the remaining data to class C_2 until the available bandwidth R_2 is reached. The process will continue until the entire bit-stream is classified. In such a way, we are guaranteed that the more loss-sensitive part of a video bit-stream is transmitted using a traffic class with lower loss. For networks with delay differentiation only, tagging proceeds in a completely analogous manner.

For networks that offer differentiation in both delay and loss, the packet tagging algorithm is more challenging. Generally, if the available rates R_i for different traffic classes are given, then a tagging scheme that simultaneously optimizes loss and delay may not exist. It is possible to construct an aggregate cost as a function of loss and delay so that the different traffic classes can be ordered in terms of QoS, and the delay-loss spread map can be ordered in terms of importance. While such an approach would allow simple mapping of data to different traffic classes, the construction of such a cost function requires a trade-off between delay and loss. Since delay and loss give rise to different types of degradations in visual communication, we focus instead on schemes that either optimize for loss or delay. In Section 6, we will describe specific packet tagging schemes that optimize for either loss or delay for the case when the available bandwidth R_i for each traffic class is fixed.

Rather than assuming a pre-specified amount of bandwidth available to each traffic class, an alternative and less restrictive formulation is possible by assuming a pricing structure for the different traffic classes. The constraint then can be given by a fixed total budget to transmit the entire video. This alternative formulation allows maximization of reception quality for a given cost.

3 Related Work

The idea of providing differential treatment to different parts of compressed digital video with different error sensitivity is not new. In particular, it has been shown in [1, 4] that by separating the source material into a base layer and an enhancement layer, and by applying priority queuing [4] or different loss probabilities [1], results superior to the one layer approach can be obtained. In terms of MPEG, it has also been shown that two-layer scalable MPEG outperforms single-rate MPEG with loss prioritization [2, 5]. However, in this paper, we only consider the transport of non-scalable MPEG due to the abundance of existing content.

For existing content compressed using one-layer MPEG, different priorities can be assigned according to the different frame types. For example, [6] uses three different priorities for the I, P, and B frames of an MPEG video to provide unequal error protection. In [7], the performance of two priority assignment schemes are compared to each other for MPEG video multiplexed in a fixed network buffer. The two schemes are: (1) marking I, P as important and B as less important, and (2) marking I as important and P, B as less important.

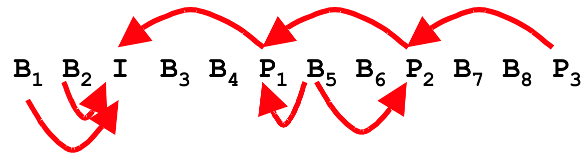

Figure 3: Dependence between different frame types in an MPEG video.

Another approach to traffic prioritization for MPEG video is to assign higher priority to the motion information than the texture information. For instance, [8] has applied unequal error protection to protect motion information for transmission of MPEG-4 video over wireless channels.

There is considerably less work in the area with separating a video into parts with different delay sensitivities. One notable exception is the delay-cognizant video compression of [9], where the generation of video sub-streams of different delay sensitivities is enabled by the use of asynchronous frame update, i.e., by rendering different parts of a given source video frame at different times according to perceived delay sensitivity. Unfortunately, the techniques of [9] cannot be easily applied to MPEG video, where each video frame needs to be rendered at fixed given times.

4 Delay Spread of MPEG Video

While MPEG does not make special provisions for transport over delay-differentiated networks, limited delay separation is achievable by exploiting the difference between the order in which video frames are stored in the bit-stream (*bit-stream order*), and the order in which they are displayed (*display order*).

Fig. 3 shows a typical frame pattern commonly used in the production of DVD movies. The frames are shown in their display order, with the arrows indicating the dependence between them. In general, Intra or I frames are coded by themselves while Predicted or P frames depend only on the previous P or I frames. Bi-directionally predicted or B frames may depend on the previous non-B frame, the next non-B frame, or both. As shown in Fig. 3, since both B_1 and B_2 depend on I, the bit-stream is organized so that the bits for I are available before those of B_1 and B_2 despite the fact that I is displayed after B_1 and B_2. Similarly, bits for P_1 appear before those

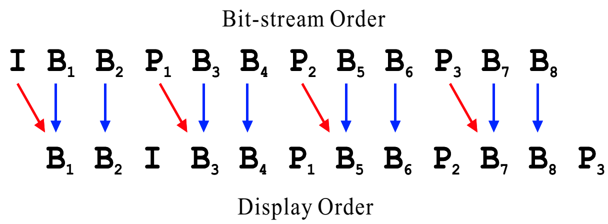

Figure 4: Exploiting the difference between display and bit-stream order for an MPEG video.

of B_3 and B_4 in the bit-stream for the same reason. This difference between the bit-stream and display order of MPEG video means that certain frames can afford to have larger transmission delays than others. Fig. 4 shows the case where compressed video frames are transmitted in bit-stream order at periodic intervals, with arrows mapping frames in bit-stream order to the instants they are first needed in display order. We see that generally, I and P frames can tolerate extra delay corresponding to one video frame as compared to B frames. For a full frame rate movie at 30 fps, this corresponds to a 33 ms difference between the delay requirements of B frames and non-B frames. While such delay spread is small compared to the duration of a movie, it is sizable in the timescale of network delays.

Fig. 5 shows the delay spread for three DVD movies with increasing motion content: *Sting*, *Batman*, and *First Strike*. The delay sensitivity of the B frames is normalized to be zero while the delay sensitivities of the I and P frames are -33 ms, indicating that both frame types can tolerate an additional 33 ms of delay. We see that B frames account for between 40 to 50% of the total bits, yielding roughly equal amount of data in the more delay-sensitive and the less delay-sensitive categories.

The delay spread of Fig. 5 is obtained using only the frame-level information of bit-stream and display order. As such, implementation complexity is extremely low since both the bit-stream and display order are available in the MPEG picture headers, and can be read off directly. The delay spread can be improved by exploiting the fact that each MPEG frame is typically organized into multiple *slices*, and each

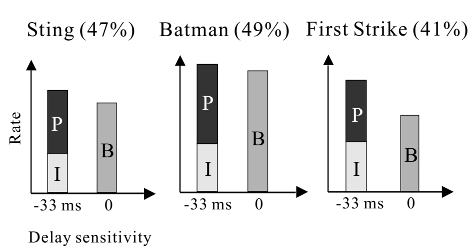

Figure 5: Delay spread for three DVD movies. Percentage of bytes in B frames are shown in brackets.

slice could potentially have different delay tolerances. For example, the frame size for DVD movies fitted for the TV screen is 720×480. Each video frame typically comprises of 30 slices, each corresponding to a horizontal strip of size 720×16. By partially decoding the MPEG bit-stream to obtain the motion vectors, it is possible to determine exactly when each slice in an I or P frame is needed. For example, if B_3 does not depend on one particular slice in P_1 in Fig. 4, that slice may not be needed until B_4 is decoded, resulting in a delay tolerance of two frames over the B frames. Furthermore, if B_4 does not depend on the same slice, it can gain yet another one frame worth of delay tolerance.

The resulting delay spreads obtained using motion vector information for the 3 DVD titles are shown in Fig. 6. We see that only minor additional spread is obtained as compared to the delay spread of Fig. 5 for the three DVD movies tested. This is probably due to the fact that at high frame rates such as 30 fps for DVD movies, successive video frames are well correlated so that almost all slices in I and P frames are used as reference for motion prediction. The largest improvement comes from the action movie *First Strike* which by virtue of its high motion content, tends to have larger portions of non-referred slices in the I and P frames.

The delay spread achieved through the above technique can potentially reduce network resources to transmit video. Specifically, consider two traffic classes with a delay guarantee of Δ_L seconds and Δ_H seconds respectively, where $\Delta_L < \Delta_H$. Assume $\Delta_H - \Delta_L \leq$ 33 ms. Then, no additional delay is incurred by sending the B frames in the low-delay class and the other frames in the high-delay class, as compared to transmitting all frames in the low-delay class. This is because I and P frames can tolerate an additional delay of at least 33 ms. Since B frames account for 40 to 50% of the bits in the 3 DVD movies used in Fig. 6, more than half of the traffic can be transmitted

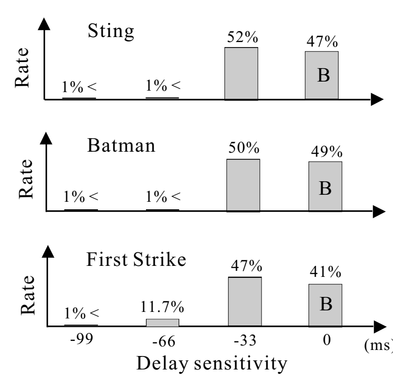

Figure 6: Delay spread using motion vector information.

in the high-delay class without incurring any penalty. Thus, the delay decomposition technique described in this section reduces the network resources required to transmit video over networks that can process packets in a differentiated manner.

5 Loss Spread of MPEG Video

In this section, we will propose and compare several methods for decomposing the bit-streams for DVD movies based on their loss sensitivity characteristics. There are a number of possible mechanisms for such decompositions in the MPEG context. One possible approach is to assign higher priority to motion information in favor of texture information [8]. Another approach is to assign highest priority to I frames, followed by P frames and B frames respectively [6, 7]. The latter approach can be generalized using the notion of *reference count*, in which each frame or slice is assigned importance according to how many frames or slices are dependent on it. In this section, we will compare five schemes for loss differentiation summarized as follows:

1. *Uniform*: Treats an MPEG bit-stream as homogeneous and assigns equal importance to all bits. This scheme serves as a base-line for our comparisons.
2. *Frame Level Reference*: The importance of all slices in a specific frame equals the number of frames that refer to it. For example, every B frame has a reference count of one since no other frames besides itself refers to it. On the other hand, the I frame in Fig. 5 has a reference count of 12 since it is used to predict every frame in the given GOP. Similarly, P_1 and P_2 have reference counts of 9 and 6 respectively.
3. *Slice Level Reference*: In this scheme, if a macroblock in frame i is predicted from K macroblocks in a previously coded frame j, each of the K macroblocks in frame j will receive a contribution of $1/K$ to their reference count. The importance of each slice is then computed as the average reference count of its macroblocks.
4. *Motion*: Assigns more importance to motion information than texture.
5. *Motion + Slice Level Reference*: Motion vector is assigned highest importance. The importance of texture information is ordered by the method of Slice Level Reference.

To compare the performance of the above schemes, we simulate the transmission of the DVD movie *First Strike* using a simple loss-differentiated network with two traffic classes of 0.3% and 3% packet loss. A two-state Markov process with parameters given in [10] is used to generate packet losses. In two independent simulations, we limit the ratio of low-loss traffic to be 50% and 30% of the total traffic respectively. Fig. 7 shows the spread of loss sensitivity for *First Strike* using the five schemes. The dashed and dotted lines show the boundary separating low and high loss traffic when the available bandwidth in the low-loss traffic class is 50% and 30% of the total bit-rate respectively. No unit or values are given for the loss sensitivities in Figs. 7-(a) to (e) due to the fact that (1) the interpretation of loss sensitivity is different for different schemes and hence cannot be directly compared against each other, and (2) we are only interested in the amount of separation or spread that is achievable within each scheme, and not the actual value of the loss sensitivity. As expected, *motion + slice level reference* results in the larger spread than any of the other schemes.

Figs. 8 and 9 show the results of applying the five loss differentiation schemes for 50% and 30% low-loss traffic respectively. The distortion measured in mean square error (MSE) as a function of time is shown for the movie *First Strike* where the loss rate for the low-loss and high-loss traffic classes are 0.3% and 3% respectively. The MSE induced by packet loss is computed for every video frame, and an average number is reported every minute. When a slice is lost, it is approximated either by the corresponding slice in the previous reconstructed frame, or the previous slice in the current frame, depending on which is estimated to be more similar. The main conclusion from Figs. 8 and 9 is that the *uniform* scheme suffers from the

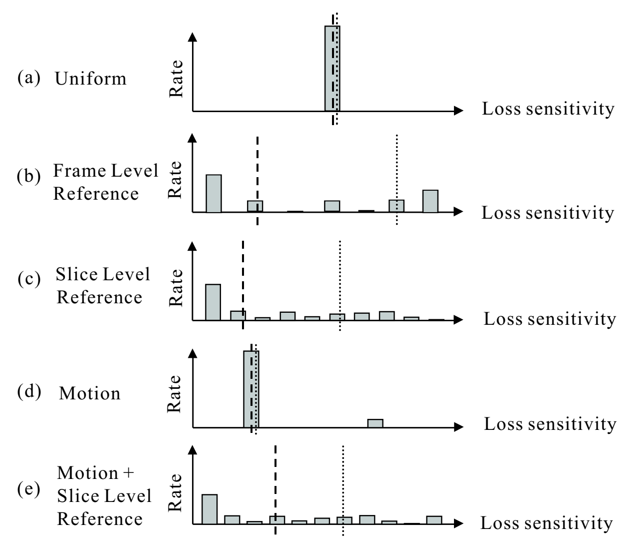

Figure 7: Loss spread for DVD movie *First Strike* using different methods of loss differentiation.

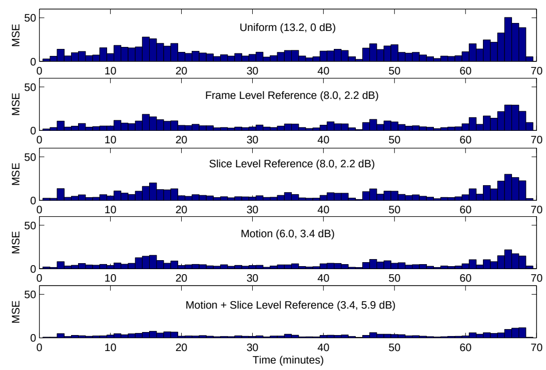

Figure 8: Comparison of loss differentiation schemes using 50% low-loss traffic. Average MSE, and distortion reduction in dB are shown in brackets.

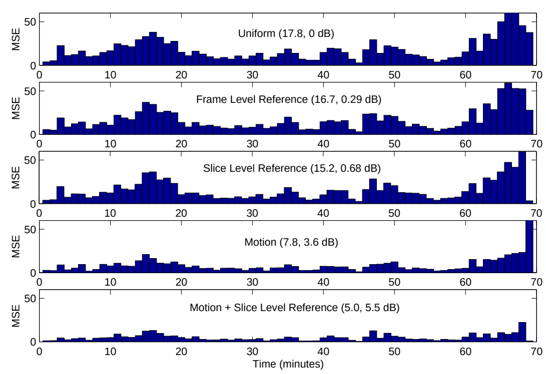

Figure 9: Comparison of loss differentiation schemes using 30% low-loss traffic.

highest distortion whereas *motion+slice level ref* provides the lowest distortion in both cases. This is because the former offers no loss differentiation while the latter offers the most loss differentiation. There are two other observations we can make from Figs. 8 and 9. First, the *slice level ref* scheme consistently outperforms the *frame level ref* scheme due to its larger degree of loss differentiation at the expense of larger complexity. Second, it is found that loss differentiation based on motion and texture information is more effective than those based purely on reference counts. Nevertheless, the lowest distortion is achieved by combining both loss-differentiation methods, as in the *motion + slice level ref* scheme.

Fig. 10 shows the distortion of sending the entire *First Strike* movie using the 0.3% loss traffic class only. The average distortion in this case is 2.5, compared to 3.4 in the *motion + slice level ref* scheme of Fig. 8. Thus, by sending all traffic using the premium class, a distortion reduction of 1.3 dB is achieved at the cost of consuming twice as much bandwidth of the "premium" low-loss traffic. The distortion reduction is expected to become smaller as the difference in loss rate between the low-loss and high-loss classes becomes smaller.

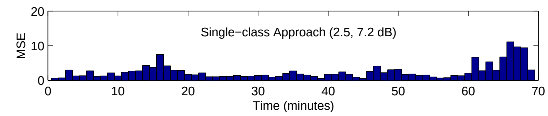

Figure 10: Distortion when 0.3% loss is applied to DVD movie *First Strike*. Average MSE, and distortion reduction in dB compared to the *uniform* scheme of Fig. 8 are shown in brackets.

6 Packet Classifiers Optimized for Loss and Delay

In this Section, we will describe packet classifiers for the DiffServ network with four traffic classes with loss rates of p_L and p_H, and delays of d_L and d_H, where $p_L < p_H$ and $d_L < d_H$, as depicted in Fig. 11. As an example, we will denote the available bandwidth of the low-loss, high-delay class by R_{LH}.

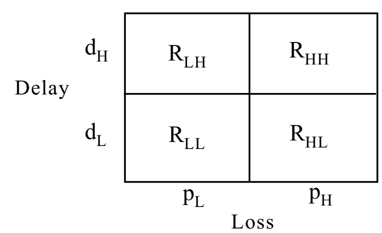

Figure 11: Four traffic classes that differ in loss rate ($p_L < p_H$) and delay ($d_L < d_H$).

When the available bandwidth for each traffic class is given, it may not be possible to simultaneously optimize for both delay and loss. In that case, a *loss-optimized* packet classifier assigns the more loss-sensitive data to the low-loss traffic classes first. As shown in Fig. 12-(a), this is achieved by first assigning the most loss-sensitive data to the low-loss traffic classes until a rate of $R_{LL} + R_{LH}$ is reached. Then, each of the low-loss and high-loss parts is further split according to delay sensitivity, as illustrated in Fig. 12-(b). In such a way, it is guaranteed that the more loss-sensitive data is transmitted using a lower loss traffic class.

A *delay-optimized* packet classifier can be constructed in an analogous fashion, and is illustrated in Fig. 13. Comparing Figs. 12-(b) and 13-(b), we observe that the loss-optimized and delay-optimized packet classifiers are generally different. In fact, the *loss-optimized* and *delay-optimized* packet classification schemes achieve minimum distortion and minimum delay respectively. More general packetization schemes achieving intermediate distortion and delay are also possible, as shown in Fig. 14. For example, consider packet classifiers that minimize a linear function of delay and

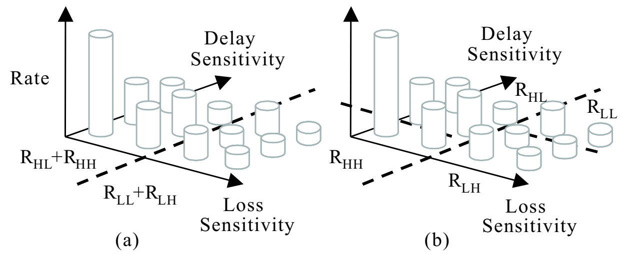

Figure 12: Loss-optimized classification of into four traffic classes with delay and loss differentiation.

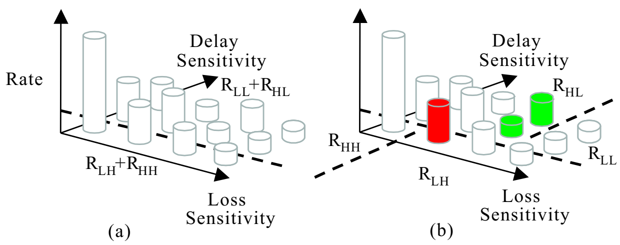

Figure 13: Delay-optimized classification of into four traffic classes with delay and loss differentiation.

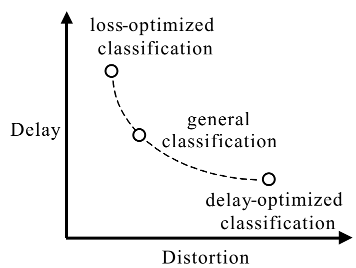

Figure 14: Performance of different packet classification scheme in terms of distortion and delay.

loss:

$$cost = (1-\rho)\times loss\ sensitivity + \rho\times delay\ sensitivity.$$

By varying the relative cost ρ from 0 to 1, a family of packet classifiers that begin with the *loss-optimized* classifier and end with the *delay-optimized* classifier is obtained, as shown by the dashed curve in Fig. 14[1]. Using other cost functions will result in different curves. The problem of characterizing the achievable region in the distortion-delay plane in Fig. 14 is open and as such is a potential subject for further investigation.

7 DiffServ Simulations

Even though DiffServ is still under development in IETF, two distinct approaches of providing traffic differentiation have evolved. The first approach is the *expedited service*, which strives to provide absolute performance guarantees via resource reservations but without the per-flow states as required by RSVP. The second approach is the *assured service*, which aims at providing relative service differentiation. Since each DiffServ domain can independently choose its own service disciplines, the DiffServ Internet is likely to consist of a mix of networks providing expedited, assured, and best-effort services. As such, the most general assumption about QoS on an end-to-end basis is relative differentiation.

In this section, we simulate a DiffServ network that provides relative differentiation. There are many possible implementations of relative differentiation, e.g., those that use the random early drop (RED) routers with In/Out bit or *RIO* [11]. In our experiments, we have chosen a special form of assured service known as the Proportional Differentiation Model [12] due to its ability to control the relative QoS in different traffic classes.

7.1 Overview of Proportional Differentiation Model

Suppose $q_i(t, t+\tau)$ is the observed QoS, such as packet loss rate or queuing delay, for traffic class i during the time interval $(t, t+\tau)$, where $\tau > 0$ is the duration of QoS observation period. The proportional differentiation model maintains that for any two traffic classes i and j:

$$\frac{q_i(t, t+\tau)}{q_j(t, t+\tau)} = \frac{c_i}{c_j}$$

where c_i and c_j are specified quality parameters.

Mechanisms for realizing the proportional differentiation model for both queuing delay and loss rate are given in [12] and can be summarized as follows. To achieve loss differentiation, when a queue is full and a new packet arrives, one packet is dropped from the class k with the lowest normalized loss rate where:

$$k = \arg\min_i \frac{l_i(t, t+\tau)}{\lambda_i} \quad (1)$$

and l_i and λ_i are the observed loss rate and loss quality parameter for class i. To achieve delay differentiation, the packet p^* with the largest normalized waiting time in the set of all queued packets Q is chosen for transmission:

$$p^* = \arg\max_{p\in Q} \frac{w(p)}{\delta(p)} \quad (2)$$

where $w(p)$ is the time packet p has spent in the queue, and $\delta(p)$ is the delay quality parameter for the traffic class p belongs to.

7.2 Simulation Setup

In this section, we describe our simulations on performance improvement achieved by packet classifier

[1] Note that the dashed curve in Fig. 14 is not generated using any experimental or simulation results. Rather, it is hand-drawn purely for illustrative purposes.

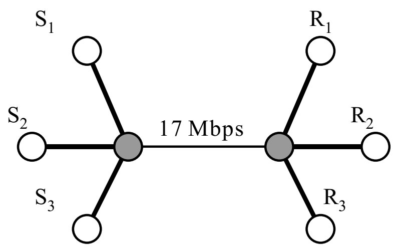

Figure 15: Simulated transmission of three movies. DiffServ enabled routers are shaded.

described in Section 6

The simulations are performed using the setup shown in Fig. 15, with a bottleneck link of 17 $Mbps$. Routers implementing the proportional differentiation model described in Section 7.1 are shown in shaded circles. Four traffic classes with both loss and delay differentiations are provided. The quality parameters of the four classes are given by $(\lambda,\ \delta)$ = (1,1), (1,2), (10,1) and (10,2) respectively. According to Eqn. 1, the traffic classes with $\lambda = 1$ will statistically have one-tenth the loss rate of traffic classes with $\lambda = 10$ during periods of network overload. Similarly, Eqn. 2 maintains that traffic classes with $\delta = 1$ will statistically have half the queuing delay as traffic classes with $\delta = 2$ when backlog is present. Thus the set of traffic classes assume the form given in Fig. 11.

Three sources are used, with S_1 and S_2 streaming the movie *Sting*, and S_3 streaming the movie *First Strike* to receivers R_1, R_2 and R_3 respectively. Senders S_1 and S_2 start streaming 5 and 1 minutes before that of S_3 respectively. In three independent runs of the experiment, the *uniform* scheme of Section 5, the *delay-optimized* and the *loss-optimized* schemes of Section 6 are employed in S_3 respectively. In each of the three runs, cross-traffic is provided by S_1 and S_2, both employing the uniform scheme. In all cases, loss-differentiation according to *motion + slice level ref* and the simpler delay-differentiation according to B and non-B frames are used.

In every run of our simulation, each source is given an available bandwidth equal to the average bit rate of the movie, i.e., 6.2 $Mbps$ for *First Strike* and 4.6 $Mbps$ for *Sting*. The available bandwidth for each source is distributed equally in the four traffic classes. Thus the average total traffic in the bottleneck link is 15.4 $Mbps$ for the three streams combined, which is lower than the bottleneck capacity of 17 $Mbps$. However, due to the varying bit-rate nature of the video sources, congestion are still experienced at various times.

One compressed video frame at bit-stream order is transmitted every 1/30 second and the video packets are tagged at the sources into one of the four classes according to the respective packet classification algorithm employed. Packets from different sources and with the same tag are given the same treatment at the bottleneck link. For each run of the experiment, even though S_3 uses the same about of bandwidth in each of the traffic classes, due to the employment of different packet classification schemes, the resulting distortion and delay may be different. One hour of video transmission is simulated and the results in terms of loss and delay are reported in Section 7.3

7.3 Simulation Results

Fig. 16 shows the MSE trace for the *First Strike* movie streamed by S_3 using the three packet classification schemes. In all three case, an identical bandwidth of one-fourth the average rate of *First Strike* or 1.55 $Mbps$ is used in each of the four traffic classes. The average MSE are 2.15, 1.6 and 0.76 for the uniform, delay-optimized and loss-optimized schemes respectively. Thus, the reduction of distortion for the *delay-optimized* and *loss-optimized* schemes over the *uniform* scheme are 1.3 dB and 4.5 dB respectively. As expected, the *loss-optimized* scheme achieves lowest distortion due to strategic classification of more important data to low-loss traffic classes.

We next look at the delay performance of the three schemes in terms of a metric we call *variable latency*. Denote by T the time between successive video frames at the source. Due to the difference in the display order and bit-stream order of MPEG video, the *allowable delay*, or the time difference between the transmission of a packet p and the time at which p is needed at the decoder, may vary from packet to packet, as illustrated in Fig. 17. Consider a packet p belonging to the I frame in Fig. 17. Denote by τ_A the time interval between when packet p is transmitted and the time it is needed at the decoder. If p is not needed by B_1, but is needed by B_2, then the allowable delay is $\tau_A + T$. Finally, if p is needed by neither B_1 nor B_2, then the allowable delay is $\tau_A + 2T$. Generally, if K

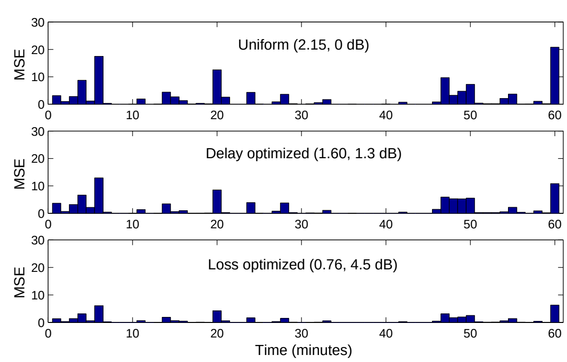

Figure 16: Distortion traces for three packetization schemes streaming a DVD movie over a DiffServ network.

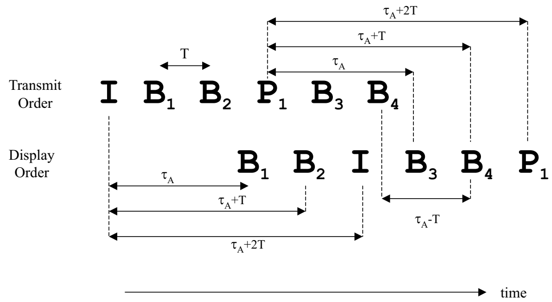

Figure 17: Different allowable delays for packets belonging to different frame types.

B-frames are used, the allowable delay is of the form $\tau_A + \kappa^{(p)}T$, where $\kappa^{(p)}$ is an integer between 0 and K inclusively. A completely analogous result holds for packets belong to P frames and the allowable delay is also of the form $\tau_A + \kappa^{(p)}T$. For packets belonging to B frames however, the difference in the display and bit-stream order dictates that they have a smaller allowable delay of $\tau_A - T$. Since actual delay of a packet p can be considered to be the sum of a constant propagation delay τ_p and a variable queuing delay $\tau_q^{(p)}$, p is considered to be on time if the allowable delay is no less than the actual delay, or:

$$\begin{cases} \tau_A + \kappa^{(p)}T \geq \tau_p + \tau_q^{(p)} & \text{if } p \text{ not part of } B \text{ frame} \\ \tau_A - T \geq \tau_p + \tau_q^{(p)} & \text{if } p \text{ part of } B \text{ frame.} \end{cases} \tag{3}$$

When the simple delay differentiation scheme of differentiating between B and non-B frames is used, as in the case of our simulations, the allowable delay for I and P frames can be simplified to τ_A. In the example of Fig. 17, the simple delay differentiation scheme conservatively assumes that every packet in the I frame might be needed to decode B_1 and B_2, and all the packets in both the I frame and P_1 might be needed to decode B_3 and B_4. In such cases, Equation 3 can be reduced to:

$$\begin{cases} \tau_A \geq \tau_p + \tau_q^{(p)} & \text{if } p \text{ belongs to an } I \text{ or } P \text{ frame,} \\ \tau_A - T \geq \tau_p + \tau_q^{(p)} & \text{if } p \text{ belongs to a } B \text{ frame.} \end{cases} \tag{4}$$

To further simplify Equation 4, we define for a packet p the *variable latency*, $l^{(p)}$, to be:

$$l^{(p)} = \begin{cases} \tau_q^{(p)} & \text{if } p \text{ belongs to an } I \text{ or } P \text{ frame,} \\ \tau_q^{(p)} + T & \text{if } p \text{ belongs to a } B \text{ frame.} \end{cases}$$

Thus, the variable latency of an I or P packet is simply the queuing delay of that packet, and the variable latency of a B packet is the queuing delay plus the a time offset of T to adjust the difference between the bit-stream and display order of MPEG video. Then Equation 4 can be simplified to:

$$\tau_A \geq \tau_p + l^{(p)} \tag{5}$$

For a video frame f, the variable latency $L^{(f)}$ is defined to be the maximum variable latency of all the packets that the frame contains. Thus, a frame f is on time for display if the following holds:

$$\tau_A \geq \tau_p + L^{(f)} \tag{6}$$

If the variable frame latencies for all frames are consistently small, then a small τ_A is sufficient to guarantee smooth playback, yielding low end-to-end delay and small buffer storage at the receiver. On the other hand, if the variable frame latencies have large maximum values, then large end-to-end delay and buffer sizes are required for smooth playback.

Figs. 18 and 19 show the variable frame latencies achieved by the three packet classification schemes at times near 6 and 38 minutes from the start of the *First Strike* movie respectively. For each of the schemes, an identical bandwidth of 1.55 $Mbps$ is used for each of the traffic classes. The load on the network is relatively high at t = 6 minutes and relatively low at t = 38 minutes, as can be inferred from Fig. 16 by the high and low distortion during the respective times. As expected, the variable latency value during the heavily loaded periods is generally larger than that of the lightly loaded periods for all packet tagging schemes. However, for the delay-optimized scheme, the maximum variable latency during the congested period is about 58 ms, while the maximum latency for the uniform and loss-optimized schemes are about 75 and 85 ms respectively during the same period. This means the end-to-end latency for the delay-optimized stream can be about 30 ms smaller than the other streams. During the lightly congested period, the latency difference between high-delay and low-delay classes is small, leading to little difference in the actual end-to-end latency.

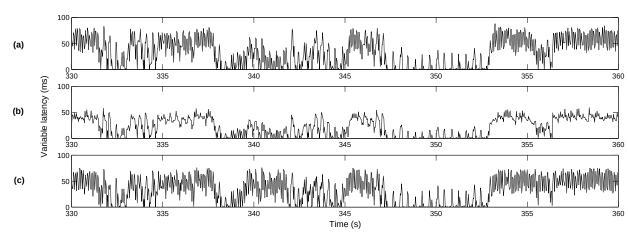

Figure 18: Variable frame latencies during periods of heavy congestion at time 6 minutes for (a) *uniform*, (b) *delay-optimized*, and (c) *loss-optimized* schemes.

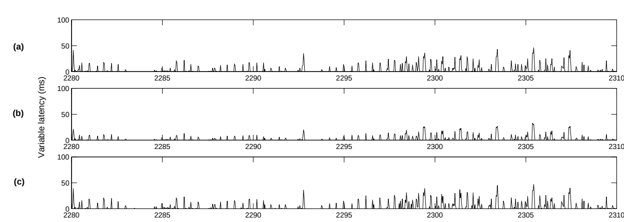

Figure 19: Variable frame latencies during periods of light congestion at time 38 minutes for (a) *uniform*, (b) *delay-optimized*, and (c) *loss-optimized* schemes.

According to Equation 6, a video frame f is late if the following holds:

$$\tau_A - \tau_p < L^{(f)}$$

We call the quantity $\tau_A - \tau_p$ the *delay parameter*, since the larger τ_A is, the larger end-to-end delay becomes. If the delay parameter is set to be small, then end-to-end delay is small, but the likelihood of a late frame is large. Conversely, one can set the delay parameter to be large, and hence decrease the likelihood of a late frame at the expense of large end-to-end delay. To characterize the percentage of late video frames as a function of the delay parameter, we compute the cumulative density function of the variable frame latency $L^{(f)}$ for each of the traces shown in Figs. 18 and 19. The results are shown in Fig. 20 which can be used to characterize the trade-off between end-to-end latency and the percentage of frame that will be late for the different packet classifiers. For instance, Figs. 20-(a) show that if we are willing to receive up to 10% and 0% late frames during periods of congestion, we need a delay parameter of about 45 and 60 ms respectively under the delay-optimized scheme. Thus, a reduction in delay by 15 ms can be achieved at the cost of 10% late frames under the delay-optimized scheme. From Fig. 20-(a), we see that in order to achieve between 0 to 20% late frames during the congested period, the delay parameter required by the delay-optimize scheme is about 20 ms less than that of the loss-optimized scheme. During the same period and in the same range of late frames, the delay parameter required by the loss-optimized scheme is only about 10 m less than that of the uniform scheme. This indicates that the delay-optimized scheme is far more effective than the loss-optimized scheme in reducing end-to-end delay.

8 Conclusions

In this paper, we have presented a framework for decomposing MPEG video into a map of delay-loss spread. This delay-loss spread map can be used to tag packets for transmission using multiple traffic classes with different delay, loss and bandwidth. We have presented a mechanism of splitting an MPEG stream into parts with different delay requirement, and also

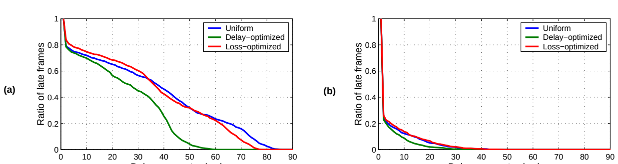

Figure 20: Ratio of late frames as a function of delay parameter for (a) congested and (b) uncongested periods.

compared experimentally several mechanisms for achieving loss and delay differentiation using commercial DVD movies. Finally, we presented simulation results, using an implementation of a DiffServ network and a DVD movie. We showed that distortion reduction of 4 dB is achievable by a packet classifier that optimizes for loss as compared to a classifier that treats the MPEG stream as homogeneous in importance. Another packet classifier that optimizes for delay achieves a reduction in end-to-end delay of 30 ms. The trade-off between end-to-end delay and the percentage of late frames are characterized using simulation for packet classifiers that optimize for loss and delay respectively.

References

[1] M. Ghanbari, "Two-layer coding of video signals for VBR networks," *IEEE J. Selected Area Comm.*, vol. 7, no. 5, pp. 771–81, June 1989.

[2] R. Aravind, M. Civanlar, and A. Reibman, "Packet loss resilience of MPEG-2 scalable video coding algorithm," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 6, no. 5, pp. 426–35, Oct. 1996.

[3] H. Radha and Y. Chen, "Fine-granular-scalable video for packet networks," in *Packet Video Workshop '99*, New York, NY., Apr. 1999.

[4] A. Dagiuklas and M. Ghanbari, "Priority mechanisms in ATM switches carrying two layer video data," *Electronic Letters*, vol. 29, no. 3, pp. 273–4, Feb. 1993.

[5] D. Wilson and M. Ghanbari, "An efficient loss priority scheme for MPEG-2 variable bit rate video for atm networks," in *Proc. IEEE Globecom '96*, London, U.K., Nov. 1996, pp. 1954–8.

[6] C. Leicher, "Hierarchical encoding of MPEG sequences using priority encoding," Tech. Rep., TR-94-058, ICSI, Nov. 1994.

[7] M. Krunz and H. Hughes, "A performance study of loss priorities for MPEG video traffic," in *Proc IEEE International Conference on Communications*, Seattle, WA, June 1995, pp. 1756–60.

[8] W. Heinzelman, M. Budagavi, and R. Talluri, "Unequal error protection of MPEG-4 compressed video," in *Intl. Conf. Image Processing '99*, Kobe, Japan, Oct. 1999, vol. 2, pp. 503–4.

[9] Y. Chang, *Delay Cognizant Video Coding*, Ph.D. thesis, University of California, Berkeley, 1998.

[10] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proc. IEEE Infocom '99*, New York, NY, Mar. 199, pp. 345–52.

[11] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *ACM/IEEE Trans. Networking*, vol. 6, no. 4, pp. 362–73, Aug. 1998.

[12] C. Dovrolis and P. Ramanathan, "A case for relative differentiated services and the proportional differentiation model," *IEEE Network Magazine*, pp. 26–34, Oct. 1999.