Real-Time Low-Complexity Adaptive Approach for Enhanced QoS and Error Resilience in MPEG-2 Video Transport Over RTP Networks

Bulent Cavusoglu, Dan Schonfeld, *Senior Member, IEEE*, Rashid Ansari, *Fellow, IEEE*, and Deepak Kumar Bal

***Abstract*—In this paper, the problems of redundancy allocation for providing effective error-resilience and service class distribution for enhanced quality of service (QoS) in real-time MPEG-2 video transport are addressed. A real-time low-complexity content-based adaptive error-resilient approach is proposed for the transport of MPEG-2 video streams, encapsulated using real-time transport protocol (RTP) and delivered over heterogeneous networks. An algorithm is derived using spatial and temporal properties of MPEG-2 video for assigning weights to each packet based on the estimated perceptual error. These weights, which indicate the relative importance of RTP packets, together with the communication channel characteristics are used to determine the allocation of resources for providing improved error-resilience and for assigning data packets to various classes of service in order to enhance the quality of transmission. Parameters extracted from the RTP header are used to determine the weights, so that the proposed algorithm can be implemented in real-time. This algorithm is used for adaptively allocating redundant forward error correction packets as well as for marking and forwarding of RTP packets in differentiated services (DiffServ). Simulation results are presented to show the significant improvement in performance based on our proposed approach to video transport.**

***Index Terms*—Differentiated services, forward error correction (FEC), motion compensation, MPEG-2, real-time transport protocol (RTP), video communications, video compression.**

I. Introduction

EVOLVING technology in communication networks and multimedia applications is being rapidly adopted for wide public use. With increased sophistication of applications and growing demand for resources, it has become more challenging to provide adequate quality of service (QoS) for multimedia applications being delivered over communication networks. Packetized networks such as the Internet, in general, work on Best-effort basis for packet delivery. This means that transmission over the network is prone to packet losses depending on the network's condition at the time of transmission.

Manuscript received November 20, 2003; revised April 4, 2005. This paper was recommended by Associate Editor O. Al-Shaykh.

B. Cavusoglu is with the University of Illinois at Chicago (UIC), Chicago, IL 60607-7053 USA. He is also with the Department of Electrical and Electronics Engineering, College of Engineering, Ataturk University, Erzurum 25240, Turkey (e-mail: bulent.cavusoglu@gmail.com).

D. Schonfeld and R. Ansari are with the Electrical and Computer Engineering Department, University of Illinois at Chicago (UIC), Chicago, IL 60607-7053 USA (e-mail: ds@ece.uic.edu; ansari@ece.uic.edu).

D. K. Bal is with Takata Inc., Detroit, MI 48326 USA (e-mail: deepak_kumar_bal@yahoo.com).

Digital Object Identifier 10.1109/TCSVT.2005.856917

Real-time delivery of MPEG-2 video is very sensitive to both delays and losses. Guarantee of adequate quality of video communications in real-time applications is more complicated than in buffered applications since recovery of lost packets must be completed in a timely manner. It is desirable to increase the QoS by utilizing the available resources, such as bandwidth, in a judicious way. Reliable transport protocols, such as TCP, may work very well for applications that are not constrained by specific delay or jitter requirements. However, reliable protocols are unsuitable for real-time delivery of video content. They rely on packet retransmission and it is, therefore, difficult to meet the strict delay requirements of real-time video applications. When a packet is delivered to its destination later than the allowed delay bound, it is considered a lost packet. The main goal of many protocols is to minimize losses due to congestion. For compressed video, like MPEG-2 [1], minimizing packet loss does not necessarily imply improved video quality, owing to spatial and temporal dependencies in an MPEG-2 video stream. For example, in exploiting temporal redundancy in video, a reference frame is coded and several other frames are differentially coded. The reference frame data is more critical in decoding the sequence because an error in the reference frame may propagate further due to temporal coding. There have been several studies that have attempted to improve the performance of the received video under network loss by minimization of error propagation.

One possible approach to reducing the effects of temporal error propagation is to increase the amount of intra coded frames. Although this method will lower the coding efficiency, an optimal algorithm for adaptive selection of frames for intra coding can be devised by using peak signal-to-noise ratio (PSNR) at the decoder together with knowledge of the network conditions [2], [3]. Another approach is to drop packets that are less important and to send packets or frames that have a higher impact on the quality of the received video [4], [5]. While these approaches are sender-driven, it is necessary to know the error concealment capabilities at the receiver site for better performance. Previous studies have also focused on adaptively adjusting the number of redundant forward error correction (FEC) packets depending on QoS requirements [6]–[8]. In particular, previous efforts have demonstrated the value of adaptive FEC based on picture types [8]. In [9], adaptive FEC is applied by considering the effects of lost packets on spatial distortion. In multicast environments, where each host has different QoS requirements, the use of layered FEC to allow different levels of protection depending on the receivers' channel conditions is an especially effective application of adaptive FEC [10].

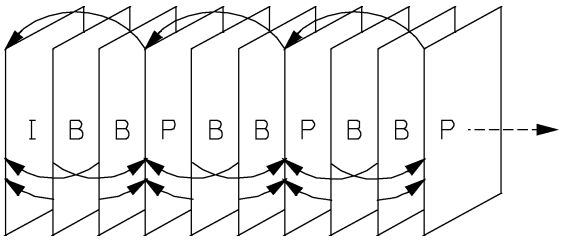


Fig. 1. Pictures in a typical GOP and temporal dependencies.

It is important to accurately capture the temporal and spatial dependencies of MPEG streams to estimate the extent of distortion due to loss [8], [11]. However, when trying to determine how reliably each packet should be sent, it is crucial that the decision be made in real-time. Making the determination by decoding the compressed (MPEG-2) video is not possible since it is a time-consuming process. This precludes the use of an exhaustive method based on decoding the compressed video to estimate the amount of distortion incurred by the potential loss of each packet. An attractive idea which avoids the time-consuming process of decoding is to consider the use of the header fields of the transport packets to infer the necessary information. Real-time transport protocol (RTP) is designed not only to carry key information, such as timing, sequence numbers etc., necessary for real-time applications, but it also provides specific packet content information through header extensions. Mechanisms for fast recovery from error often rely on the information provided in the header fields. RTP includes MPEG-1 and MPEG-2 video header and MPEG-2 specific video header extensions for real-time video communications. Thus, the use of RTP is attractive especially because the protocol provides quick access to crucial information of the payload via its headers at any point of transmission over the network. We use the RTP header and MPEG-2 specific header extension to gather vital information of MPEG-2 video streams and design our algorithm based on the parameters embedded in the RTP header. In this paper, we focus only on those header parameters that are used in our algorithm. The interested reader is referred to [12] for a detailed explanation of MPEG-2 payload for RTP.

MPEG-2 video exploits spatial and temporal redundancies in a video stream to achieve the desired compression. MPEG-2 video is constructed from video sequences, for which group of pictures (GOP) structures are defined consisting of a combination of I- (intra coded), P- (predictive-coded), and B- (bi-directionally predictive-coded) pictures (see Fig. 1). As seen in Fig. 1, there is either a direct or indirect dependency from any picture to the first picture within the GOP of an MPEG-2 video stream. Hence, the loss of a single packet may cause a disturbing visual degradation due to propagation of error. It is, therefore, important that the amount of distortion due to the possible loss of a packet is determined and the packets are serviced based on their potential distortion levels. Previous studies mainly focused on picture type and pointed out the difficulties of including temporal effects in determining the level of importance of a packet. In this paper, we propose an algorithm that uses both spatial and temporal information to estimate each packet's relative impact on the quality of received video. We estimate the relative

RTP Header

V	P	X	CC	M	PT	Sequence Number
Timestamp						
Synchronization Source Identifier						

RTP Video Specific Header for MPEG-1 & MPEG-2

MBZ	T	TR	AN	N	S	B	E	P		BFC		FFC
									FBV		FFV	

N: New picture header.
P: Picture type.

RTP MPEG-2 Video-Specific Header Extension

X	E	f(0,0)	f(0,1)	f(1,0)	f(1,1)	DC	PS	T	P	C	Q	V	A	R	H	G	D

f(0,0): Forward horizontal f_code.
f(0,1): Forward vertical f_code.
f(1,0): Backward horizontal f_code.
f(1,1): Backward vertical f_code.

Fig. 2. RTP header fields.

weights of packets by extracting the picture type, and the spatial and motion-based parameters of MPEG-2 from RTP packet headers. These weights are then used to classify the packets and to allocate available resources. We finally evaluate the performance of the proposed algorithm and show that the gap between the proposed algorithm and an optimal algorithm that relies on offline exhaustive calculation is small.

II. Relative Weight Calculation for Video Frames

The proposed algorithm adopts the idea that every picture has a different level of importance in MPEG-2 video. The importance of a picture can be expressed in terms of the potential degradation caused in the video due to loss of information in that picture. The loss will cause distortion in the current picture as well as in the pictures that are directly or indirectly dependent on the current picture. Standard measures of distortion can be used if the video is fully decoded. Here we consider the extent of distortion by assessing the relative impact of the packet loss on distortion. We use the picture type "P" and "f_code" parameters embedded in RTP headers (see Fig. 2) and a temporal compression measure we provide to determine the relative importance of each picture. The "f_code" parameter is used in motion vector (MV) decoding and provides a measure of the size of the MV range in the current picture (see Table I). Bit rate of the video, spatial and temporal compression, and concealment techniques play a critical role in the quality of the video at the receiver. The MPEG encoder determines the first three factors listed and their effect varies depending on the implementation of the encoder. All of the MPEG standards are provided for the decoder. Senders have the freedom to use any encoder they wish provided it is compatible with the standard. This results in transmission of different video quality even for the same bit rate. It is not an easy task to determine the compression achieved and dependency between frames exactly by simply observing the transmitted packets without decoding the video stream. We assign a relative importance level to each frame by considering

TABLE I
f_codes (m AND n REPRESENTS 0 or 1)

f_code(m,n)	Vertical component of field vectors in pictures	All other cases
0	(forbidden)	
1	[-4:+3.5]	[-8:+7.5]
2	[-8:+7.5]	[-16:+15.5]
3	[-16:+15.5]	[-32:+31.5]
4	[-32:+31.5]	[-64:+63.5]
5	[-64:+63.5]	[-128:+127.5]
6	[-128:+127.5]	[-256:+255.5]
7	[-256:+255.5]	[-512:+511.5]
8	[-512:+511.5]	[-1024:+1023.5]
9	[-1024:+1023.5]	[-2048:+2074.5]
10-14	Reserved	
15	(used when a particular f_code will not be used)	

the level of dependency between frames and the degree of spatial detail in the current picture compared with other pictures in the video stream. Moreover, the amount of perceptual distortion at the receiver is estimated by determining the information that can be recovered using knowledge of the concealment technique used at the receiver. However, assessing performance for arbitrary concealment techniques at the receiver is not always possible due to the fact that specific protocols have to be defined for that purpose. We, therefore, assume the existence of a basic concealment method at the receiver and base our discussion and computations on that assumption. Details of the concealment technique assumed at the receiver are presented in the next section. Next, we discuss important issues relating to spatial compression, temporal compression, and concealment effects in our model. We then explain how these effects are incorporated into our model.

A. Spatial and Temporal Redundancy

When compression is lossy, each picture is subject to different amount of distortion in the spatial domain depending on the amount of detail in the current scene and the fidelity that can be provided at the given bit rate. If the picture can be represented with high fidelity using mostly the low-frequency discrete cosine transform (DCT) coefficients, then a lower distortion is achieved with MPEG-2 encoding. In typical MPEG-2 coding, it is common to choose a total of 15 pictures per GOP. Assuming 30 frames/s as the frame rate, the duration of each GOP is 0.5 s. It can be assumed that, with high probability, pictures in a GOP carry similar information content. In other words, the same scene contains mostly the same objects and background. Therefore, if all pictures in 0.5 s are compressed with intra coding, the distortion at a typical bit rate would be high. This limitation is overcome in MPEG-2 coding by exploiting temporal redundancy which accounts for its high compression gain.

B. Concealment Effects

Several concealment strategies have been proposed to overcome the effects of packet loss in an MPEG-2 video stream. They rely on the idea that neighboring macroblocks have high correlation and a combination of neighboring blocks may be used to conceal the error. Alternately, an estimate of lost blocks in the reference picture is obtained from MV information of neighboring macroblocks, assuming that the lost block is very likely to have had similar movement in temporal domain. The latter method is applicable only when MV information is available for the neighboring macroblocks. In [13], joint forward error correction and error concealment for compressed video is evaluated and it is shown that, when slices in MPEG-2 video are adjusted by taking into account discontinuities in the picture, the video quality improves. When such concealment methods are not employed at the receiver, the MPEG-2 decoder copies the macroblock which is at the same location in the most recently decoded picture to conceal the lost macroblock. We will assume that receivers adopt this simple concealment method, which is commonly used in most decoders.

C. Derivation of Relative Weights

Let us assume that there are a total of M pictures per GOP

$$M = (N_B + 1)(N_P + 1), \quad \text{for open GOP} \tag{1a}$$

$$M = (N_B + 1)(N_P + 1) - 2, \quad \text{for closed GOP} \tag{1b}$$

where N_P is the number of P-pictures per GOP and N_B is the number of consecutive B-pictures between the nearest I- and P-pictures.

The initial weights can be assigned by assuming that any error occurring in the reference picture will produce the same amount of distortion in the pictures that use the reference picture. For simplicity, it is also assumed that each lost macroblock will cause the same amount of distortion in the current picture regardless of its position.

The initial weights are, therefore, given by

$$w_I = M \qquad w_{P_i} = M - T_{P_i} \qquad w_B = 1 \tag{2}$$

where w_I, w_{P_i}, w_B are the initial weights for I-, P-, B-pictures, respectively. T_{P_i} is the temporal position of the ith P-picture in the current GOP. For instance, if a received picture is the first P-picture after an I-picture, then $T_{P_1} = N_B + 1$.

This assignment of initial weights is justifiable only if the macroblock in error is exactly copied from the reference picture. However, exact copying is rarely invoked when motion is present. Next, we will determine the amount of temporal dependency between the pictures and adjust the weights accordingly.

Let R_I, R_P and R_B be the sizes, in bits, of compressed I-, P-, and B-pictures, respectively. Let us define

$$I_{RP} = \frac{R_P}{R_I} \qquad I_{RB} = \frac{R_B}{R_I} \tag{3}$$

where I_{RP} and I_{RB} are called the *intra refreshing factors* for P- and B-pictures, respectively. As explained earlier, if every picture in a GOP were coded without the use of motion compensation, then the size of each compressed picture would be approximately the same, assuming there was no scene change within the GOP. It is possible for an MPEG-2 encoder to detect a scene change and force an I-picture wherever a scene change occurs. Intra refreshing factors provide an approximation of the percentage of "new" data in the picture. Some overhead, due to MV coding and picture header information is ignored in this approximation. A typical distribution of this data is shown in Fig. 3. As

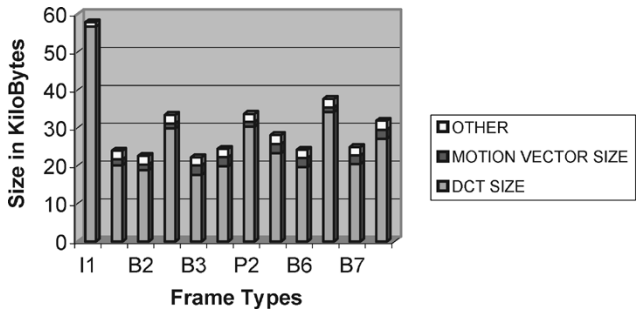


Fig. 3. Typical data distribution in frames per GOP.

seen from the figure, however, the overhead data is negligible. We have also verified throughout our simulations that ignoring the overhead does not significantly affect the results.

Some macroblocks in P- and B-frames are coded as intra macroblocks because the MPEG encoder has determined that the change in the current macroblock is beyond a threshold set by the encoder to code the current macroblock differentially. If we assume that every block in the P- or B-frame can be recovered at the receiver except intra macroblocks, then the relative distortion is proportional to the intra refreshing factor I_{R^*} and can be estimated by

$$D_{1^*} = \alpha I_{R^*} \tag{4}$$

where "*" represents P or B, as appropriate.

The rest of the picture will be constructed by predictive-coded macroblocks. The amount of predictive-coded data carried over from the reference picture by using MVs can be approximated by $1 - I_{R^*}$. Determining the amount of distortion due to temporal information loss is more involved. If we assume that there is no concealment other than copying the macroblock from the previously decoded reference picture, then the relative perceptual distortion for temporally coded macroblocks can be estimated by

$$D_{2^*} = \beta(1 - I_{R^*}). \tag{5}$$

Now, the total distortion is given by

$$D_* = \alpha I_{R^*} + \beta(1 - I_{R^*}) \tag{6}$$

where α *and* β are *adjustment factors* used to denote the relative distortion amount. They cannot be determined independently and are correlated due to the fact that the MPEG encoder uses a decision criteria based on spatial correlations of neighboring macroblocks to determine if a macroblock shall be encoded as intra or predicted. The amount of distortion will be greater in case of a loss of an intra macroblock because it has less spatial correlation to the macroblocks that could have served as a best match than that of predicted macroblocks. This suggests that α should be greater than or equal to β. In other words, loss of a predicted macroblock will not cause a larger distortion than the loss of an intra coded macroblock in the current picture.

After normalizing by α we get the normalized relative distortion given by

$$\overline{D}_* = I_{R^*} + \frac{\beta}{\alpha}(1 - I_{R^*}) \tag{7}$$

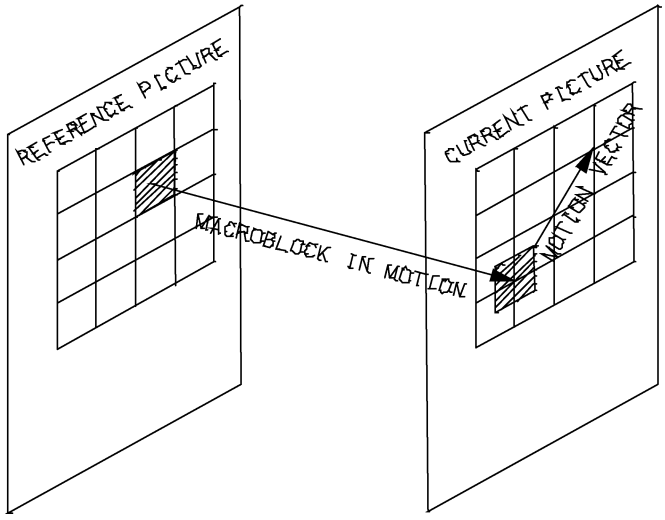


Fig. 4. Captured motion between pictures.

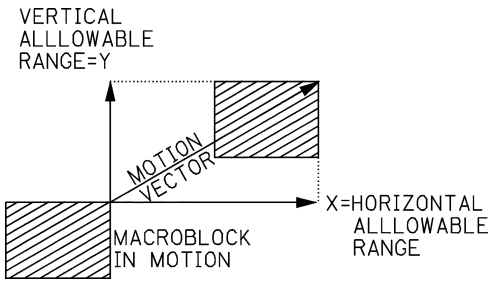


Fig. 5. Range of the MV.

where $\bar{D}$ becomes 1 if the intra refreshment factor I_R is 1, which means that the current picture is not temporally dependent on a reference picture and cannot be recovered by using a reference picture.

To determine the exact ratio of β/α, one needs to know the algorithm used by the MPEG-2 encoder at the sender precisely and the success of the concealment method at the receiver in the recovery of intra macroblock and predicted macroblock losses. Next, we estimate this ratio under assumptions which allow us to derive an algorithm for a generic sender-receiver case, where only generic features of the concealment and MPEG-2 encoder are used rather than the exact features of a particular sender-receiver pair.

To determine the β/α ratio, we define the maximum possible captured motion by using f_codes (see Fig. 4). In other words, we determine the extent of temporal compression applied by the MPEG encoder. Table I depicts the MV range for f_codes. Then, X and Y (see Fig. 5) are given by

$$X(m,0) = \frac{\left(2^{(\text{f\_code}(m,0)-1)}\right)}{2}$$
$$Y(m,1) = \frac{\left(2^{(\text{f\_code}(m,1)-1)}\right)}{2}, \qquad m = 0, 1. \tag{8}$$

where X and Y are distance vectors in the units of macroblocks. For example, if $\text{f\_code}(m,0)$ is 2, then $X(m,0) = 1$, which means that the current macroblock is shifted by at most 1 macroblock in the horizontal direction compared to the reference macroblock.

We also define the MV as

$$\mathrm{MV}_F = \sqrt{X(0)^2 + Y(0)^2} \qquad \mathrm{MV}_B = \sqrt{X(1)^2 + Y(1)^2}$$
$$\mathrm{MV} = \max(\mathrm{MV}_F, \mathrm{MV}_B) \tag{9}$$

where MV_F and MV_B are forward and backward maximum captured motion, respectively, and MV is the maximum captured motion. MV is a measure of the amount of possible motion of a macroblock that could be encoded using MVs. However, this should not be confused with the actual motion between pictures, since some of the motion can be coded by using intra macroblocks. Basically, MV indicates the amount of possible motion between the current frame and the reference frame. Assuming that the concealment technique used does not rely on MV estimation, the loss of a reference frame will be worse for higher MV values. This is due to the fact that spatial correlation between neighboring macroblocks is lower for high MV values.

Using this information and assuming that the perceptual error in the case where the largest MV is used and the perceptual error in the case where intra coding is used is the same in the event of a macroblock loss, we can estimate the ratio β/α by

$$\frac{\beta}{\alpha} = \frac{\mathrm{MV}}{\max(\mathrm{MV})}\lambda(R_I) \tag{10}$$

where $\lambda(R_I)$ is the *relative spatial correlation adjustment factor*. Basically, it provides an approximation of the relative spatial correlation among neighboring blocks in the current GOP or scene compared to other GOPs or scenes in the stream. Since we assume that there is no scene change within a GOP, $\lambda(R_I)$ can be approximated by

$$\lambda(R_I) = \frac{R_I^{\mathrm{current}}}{R_I^{\max}} \tag{11}$$

where R_I^{current} is the size of the I-picture, in bits, in the current GOP and $R_I^{\max}$ is the size of the I-picture with maximum number of bits among the previous I-pictures. $R_I^{\max}$ is recalculated for each GOP by comparing to the previous I-pictures and is reset after a predetermined time period.

Recall that the initial weights were defined by assuming that every macroblock is copied to the current picture from the reference picture and any loss that occurred in the reference frame cannot be recovered. However, $\bar{D}$ reflects the actual distortion in the current frame since it determines the distortion by considering the portion of the picture that could be recovered from its reference frame. The weights for the current picture can then be calculated by

$$W_I' = w_I \qquad W_{P_i}' = w_{P_i}\bar{D}_{P_i}, q \quad W_{B_i}' = w_B\bar{D}_{B_i}. \tag{12}$$

Notice that a lower perceptual distortion factor results in a lower weight for the current picture. When all macroblocks are copied from the reference picture (in other words, when $I_{R^*} = 0$ and $\mathrm{MV} = 0$), the perceptual distortion becomes 0, which indicates that receiving only the reference frame is sufficient to recover the current frame; thus, the current frame has no weight. However, this formulation is incomplete since the initial values were determined by assuming that all macroblocks are copied from the first reference picture. In other words, it is assumed that every picture has a distortion factor of 0, except the I-picture at the start of the GOP. The effective distortion values have to account for error propagation and the resulting distortion must be fed back to the previous pictures in the GOP and the initial values need to be updated. It is necessary to wait until the end of the GOP in order to determine the weights for all pictures in the GOP. However, this requirement is impracticable in real-time systems. In this case, we will update the initial values by only waiting until the next P-picture. No delay is introduced by this procedure since in the encoding order P-pictures are generated immediately after their reference pictures. The feedback values are given by

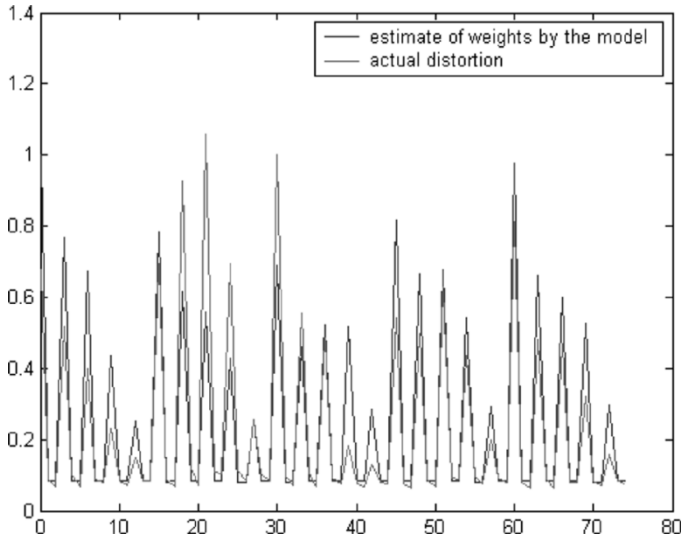


Fig. 6. Trace of the actual distortion and the estimate of the model. Both traces are normalized to 1.

$$f_I = 1 - \bar{D}_{P_1} \quad f_{P_i} = 1 - \bar{D}_{P_{i+1}}. \tag{13}$$

Finally, the relative importance weights are given by

$$W_I = 1 + (W_I' - 1)\, f_I \qquad W_{P_i} = W_{P_i}' + \left(W_{P_i}' - 1\right) f_{P_i}$$
$$W_B = W_{B_i}'. \tag{14}$$

When we observe the end result for the weights given in (14), we note that the first part of the summation represents the effect of a loss in the picture itself and the second part of the summation corresponds to the effect of a loss due to temporal error propagation. Fig. 6 illustrates a sample normalized trace of the model's estimated weights and the actual distortion of the frames. Both traces are normalized to 1.

We shall now use these weights for adaptive allocation of redundant FEC packets as well as for marking and forwarding of RTP packets in differentiated services (DiffServ).

III. Adaptive Forward Error Correction (AFEC)

Packetized networks are modeled in terms of packet losses, not bit errors; in other words, if a packet arrives at its destination in time, then it is useful. Recovery from packet losses can be achieved via the use of erasure codes [14]. In an (n, k) erasure code, where k is the number of message packets and $n - k$ is the number of redundant packets, message packets are fully

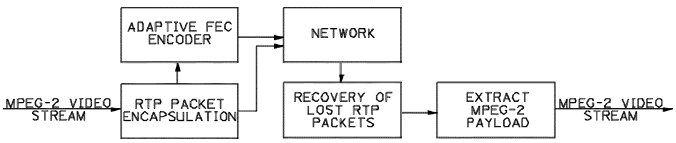


Fig. 7. Network model for FEC.

recoverable if and only if at least k out of n packets arrive at the destination [14], [15].

Many effective FEC codes, such as Reed–Solomon, Hamming, and CRC [16] have been proposed to recover packet losses. These codes have erasure correcting capabilities as well. FEC codes introduce redundant packets in order to recover lost packets. Increasing the amount of redundancy will achieve higher recovery rates; however, it is evident that increasing redundancy will cause congestion and eventually degradation of video quality. The amount of redundancy introduced can be determined by using various performance parameters and constraints, such as required QoS, delay, packet loss ratio, and network conditions. Moreover, once the extent of redundancy allowed is known, it should be used in the best way.

We apply the weights derived by our algorithm, which essentially capture the relative importance of each picture in the video stream, to determine the best way to distribute the redundant FEC packets in the video stream.

We have constructed our network model based on RTP encapsulated MPEG-2 and FEC packets (see Fig. 7) for adaptive FEC. The interested reader is referred to [17] for the details of the generic FEC method used with RTP. It is assumed that packet losses are uniformly distributed and that errors do not occur in bursts. Losing a single macroblock in a slice implies the loss of the entire slice since macroblocks in each slice are differentially coded. When losses occur, the MPEG decoder may proceed to the next synchronization point which is the next slice and continue decoding from there. When we use relative importance weights for differentiated services in the next section we allow bursty traffic; thus, burst of packet drops occur and we evaluate the results with bursty errors. Here, we show that our proposed algorithm takes the error correction capabilities and channel characteristics into account in its decision criteria.

We first define a degradation density function based on the weights and the packet loss ratio (PLR). The probability of not recovering a lost packet in an (n, k) erasure code is simply given by

$$P_L = \sum_{i=n-k+1}^{n} \binom{n}{i} \mathrm{PLR}^i (1-\mathrm{PLR})^{n-i}. \tag{15}$$

Then, the resulting degradation density function is defined by

$$\mathrm{DDF} = P_L W_*. \tag{16}$$

Once DDF is obtained, it has to be set to a constant, preferably to an optimized value so that the degradation at the receiver is approximately the same regardless of whichever packet is lost. Solving the optimization problem is beyond the scope of this paper. Instead, we will determine an adaptive working point DDF_0 based on the amount of FEC we are allowed to use.

We set our FEC to $(n, n-1)$. In other words, we produce only one FEC packet per $n-1$ RTP packet. After generating the RTP packets, we apply our algorithm by assuming that the amount of redundant packets to be allocated is given. Given the allocated amount of FEC, we set an initial threshold value, DDF_0, for DDF. Whenever a packet arrives, DDF is calculated and added together. If the total is less than DDF_0, we store a copy of the received packet and transmit the original packet. Once the total is greater than DDF_0, one FEC packet is produced with the copied packets in the buffer by simply applying XOR operation among them. At the end of each GOP the percentage of FEC packets generated are compared with the allocated percentage and threshold value is adjusted to be as close as to the percentage allocated. If K packets are generated per L input packets and the allocated percentage is $A\%$, then the new threshold value is set to

$$\mathrm{DDF}_0 = \frac{\mathrm{DDF}_0 A(\frac{L}{K})}{100}. \tag{17}$$

We applied our algorithm to MPEG-2 streams, which we generated, with scene detections. We used scenes from the movies Matrix and Mummy to generate multiple MPEG-2 streams. The streams have 30 frames/s and they were generated at different bit rates varying from 1.5 to 4 Mb/s. The results represent the average behavior of the simulations we conducted. In our simulations, the amount of FEC redundancy used was 25%. We compared the performances in the following cases with the proposed adaptive algorithm, AFEC.

Static FEC: Redundant packets are distributed equally regardless of the weight of the packets.

Static IPBFEC: The only criterion used for distribution of FEC packets is the picture type. Redundant packets are first assigned to I-pictures, then to P-pictures and very few of them to B-pictures.

Optical FEC: Actual distortion amount caused by each packet lost is calculated. Calculation is performed for all packets by dropping one packet and calculating the mean-squared error distortion at the receiver caused by the dropped packet. FEC packets are allocated in proportion to the amount of distortion caused by the packet loss.

We measured end-to-end PSNR values for performance comparison. Fig. 8 shows the end-to-end PSNR values for suggested algorithm AFEC (Adaptive FEC), static FEC, static IPBFEC and Optimal FEC. It can easily be seen that AFEC outperforms the other two methods used for the distribution of FEC and performs approximately 1.5 dB below the optimal FEC.

IV. Adaptive Packet Marking in Differentiated Services Networks

Differentiated services architecture (DiffServ) [18], has been developed by the internet engineering task force (IETF). DiffServ is intended to provide scalable and flexible service differentiation. It allows handling different classes of traffic in different ways within the Internet. DiffServ architecture does

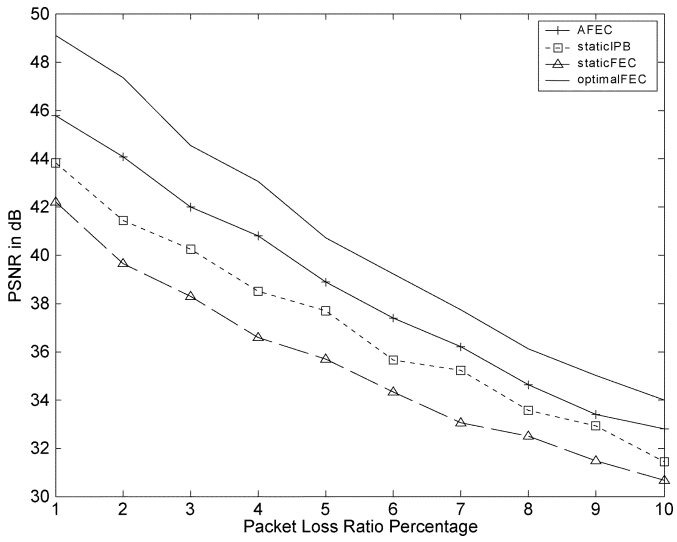


Fig. 8. FEC simulation -PSNR.

not define specific services or service classes. It provides the functional components needed to build such services. DiffServ architecture defines a set of behaviors called per hop behavior (PHB). Each PHB corresponds to a particular forwarding treatment of the packets. DiffServ treats each packet according to the packet's differentiated services code point (DSCP) located in the IP header. Basically, in DiffServ, traffic is divided into different classes depending on the DSCP, determined at the source or boundary router. Furthermore, each packet can have different drop precedence in the same class allowing less important packets to be dropped first in case of congestion. DSCP includes information dictating how a particular packet should be forwarded in the network. There are two different forwarding mechanisms proposed by IETF: expedite forwarding (EF) and assured forwarding (AF). EF provides low loss, latency, and jitter by minimizing the queue sizes that aggregate traffic encounters. On the other hand, currently there are four distinct classes defined for AF and three different relative dropping priorities associated with each of these classes. AF provides guaranteed aggregate service criteria depending on the service level agreement (SLA) between the source and internet service provider (ISP). It is evident that using high quality links will cost more; in other words, EF is the most expensive forwarding method and AF is more expensive than Best-effort services. All classes operate independently from each other on the router.

In [19], MPEG-2 video performance in DiffServ is evaluated and it is concluded that with proper queuing management DiffServ can satisfy the QoS required by video applications. The improvement attained in QoS by assigning different drop precedence for each picture type in DiffServ is shown in [20]. It is also shown in [21] that the use of FEC and DiffServ with the structural hierarchy of MPEG-2 video has a significant improvement on QoS. The improvement in QoS when packet forwarding is performed adaptively depending on the video stream is evaluated in [22], [23].

We use the relative importance weights for each packet when determining the particular forwarding class and drop precedence assignments. We subsequently forward the traffic using

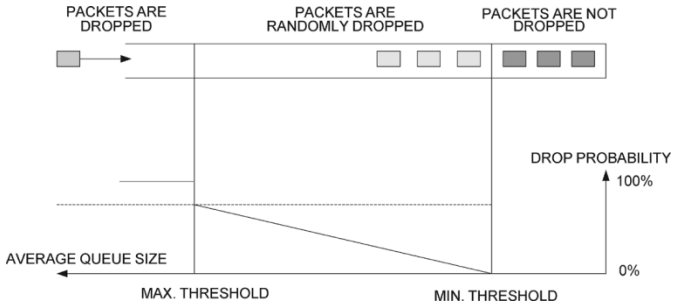


Fig. 9. RED packet dropping algorithm.

AF classes and Best-effort forwarding. Although the proposed algorithm can allow the use of all 4 AF classes defined, we will utilize only 3 AF classes in order to improve the comparison between the proposed algorithm and forwarding by using only the picture type (I,P,B).

We first propose an estimated cost function by

$$EC = \sum_{i=1}^{3} \left(C_i^{\mathrm{AF}} x_i + C^{\mathrm{BE}} \left(1 - \sum_{j=1}^{3} x_j \right) \right) \qquad (18)$$

where C_i^{AF}, C^{BE}, and x_i represents the cost associated with AF classes, cost for Best-effort forwarding and percentage of the source's traffic marked to be forwarded with that particular class, respectively. A quality metric can accompany this cost function and depending on QoS requirements a cost benefit analysis can be performed to find the optimum resource usage. Moreover, for a fixed cost, in other words for a fixed SLA, we can distribute the packets in a way that uses the available bandwidth for providing the best quality possible for the video. We have performed the simulations for a "fixed cost" setup in the following way.

Let us assume that price for each unit of bandwidth is constant and the amount of bandwidth is the only criteria for differentiating the classes. This is a reasonable assumption considering that if we want to restrict the delay and drop probability as well, we would need to provide sufficient bandwidth. Then, C_i^{AF} and C^{BE} in (18) is proportional to the bandwidth amount that is assigned to the particular class. We also assume that SLA is based on fixed cost. Assuming the link costs remain the same once they are assigned, we determine the fixed cost for our simulations by determining the term x_i in (18) as

$$x_i = \frac{1}{C_i^{\mathrm{AF}} \left(\sum_{i=1}^{3} \frac{1}{C_i^{\mathrm{AF}}} + \frac{1}{C^{\mathrm{BE}}} \right)}. \qquad (19)$$

We determine the fixed cost operating point in this manner for our simulations in order to compare the algorithms fairly. Within this fixed cost operating point, the product of C_i^{AF} and x_i will give a constant result no matter which class is used. This assignment for class usage is forced with a token bucket algorithm at the edge router.

In DiffServ domain, the edge router does the job of marking, shaping/dropping the traffic as per the SLA between customer and service provider. The token bucket algorithm used consists of a bucket (normal burst size) that can hold up to b tokens that are created at a rate of r tokens per second (average rate). The

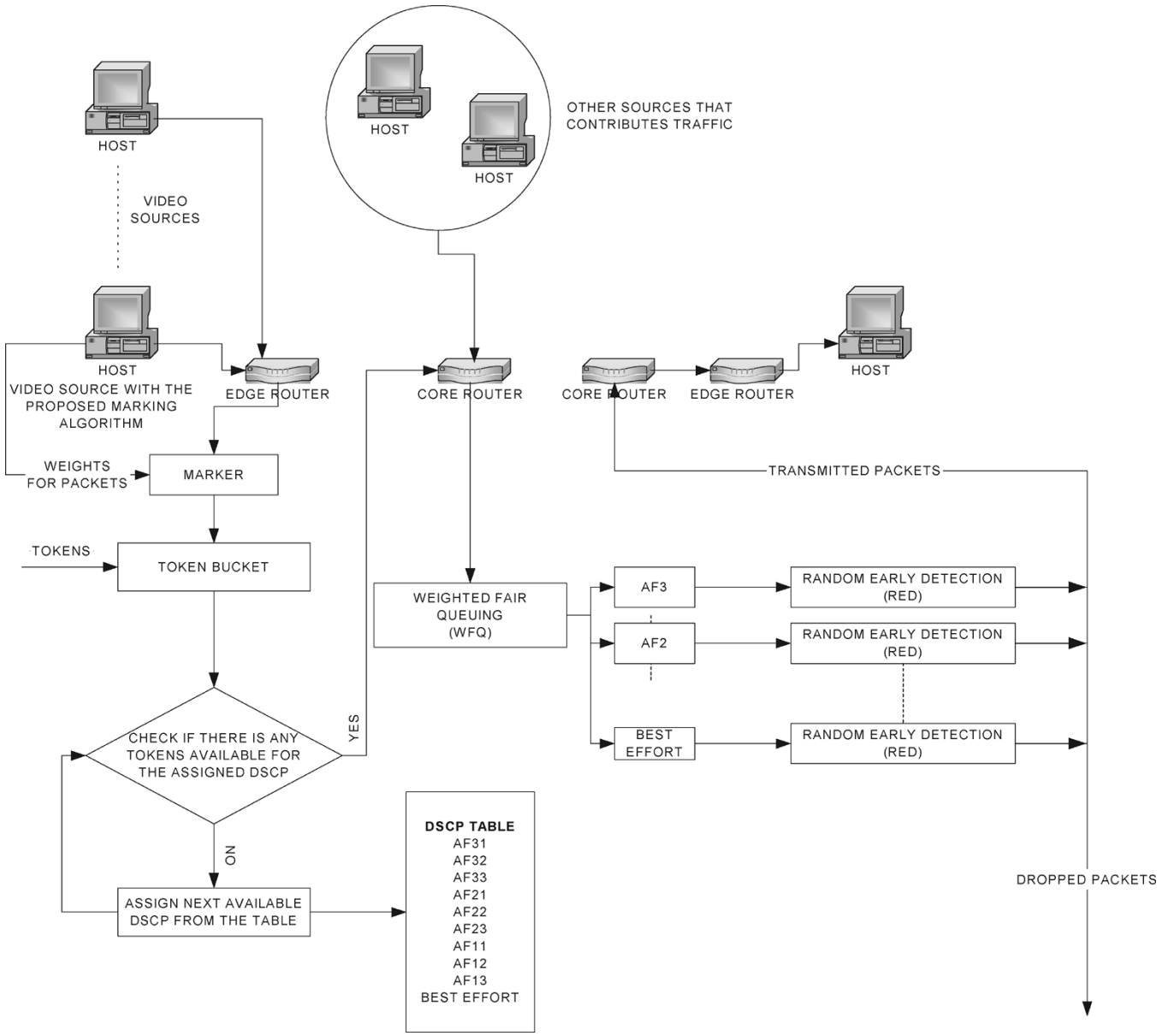


Fig. 10. DiffServ simulation setup.

core router does the job of packet scheduling and routing. In our simulation setup, committed access rate (CAR) (defined for Cisco routers) does the job of edge router by managing bandwidth through rate limiting.

Within the core of DiffServ domain packet scheduling is done using weighted fair queuing (WFQ). Packets are sorted into classes first and a scheduler alternates among the classes. Each class i, is assigned a weight, q_{w_i}. Under WFQ, during any interval of time during which there are class i packets to send, class i will be guaranteed to receive a fraction of service equal to $q_{w_i} / \sum_j q_{w_j}$, where the sum in the denominator is taken over all classes that also have packets queued for transmission.

Congestion in queues is avoided using random early detection (RED) [24], which drops packets as congestion begins to increase. It calculates average queue size using a low-pass filter with an exponential weighted moving average. When the average queue size exceeds a preset threshold, it drops or marks each arriving packet with a certain probability, where exact probability is a function of average queue size. As we see from Fig. 9, the packet dropping probability becomes 100% after the average queue size exceeds Max_{th}. The average queue size is calculated such that short-term increases in the queue size that result from bursty traffic or from transient congestion do not result in significant increase in average queue size.

The experiments were conducted using OPNET modeler 8.0.C running on Sun Solaris machines. The video conferencing application defined in OPNET was used where two workstations send video packets to each other using user-defined parameters.

The packet marked with an AF class is checked against the parameters defined for that class. If it conforms, then it is transmitted with that AF class. If it exceeds, then it is checked against the next set of parameters of the lower class (see Fig. 10). This process continues until the packet finds a match with a particular set of parameters. If there is no match, then it is transmitted as a Best-effort packet. By using the CAR profile, we shape the

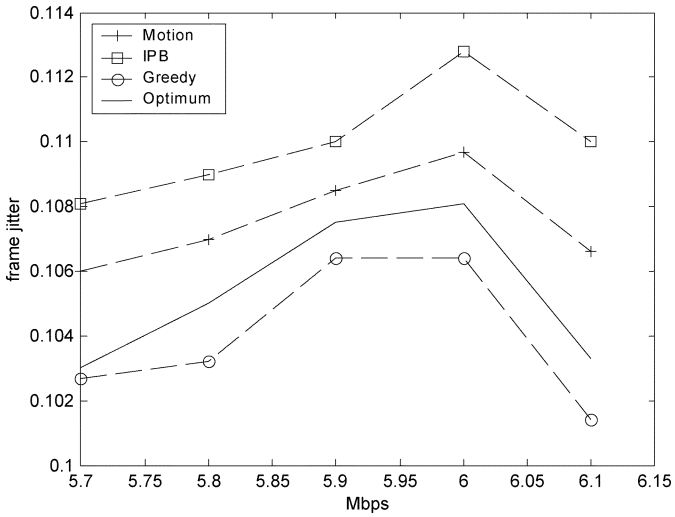


Fig. 11. DiffServ simulation I-frame jitters.

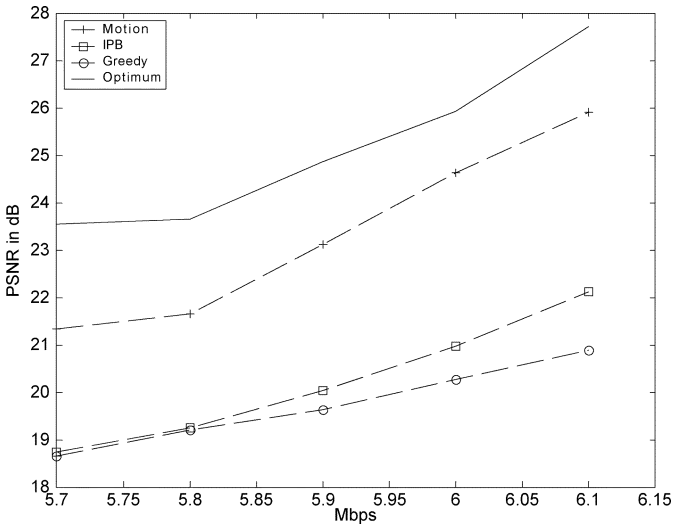


Fig. 13. DiffServ simulation I-PSNR.

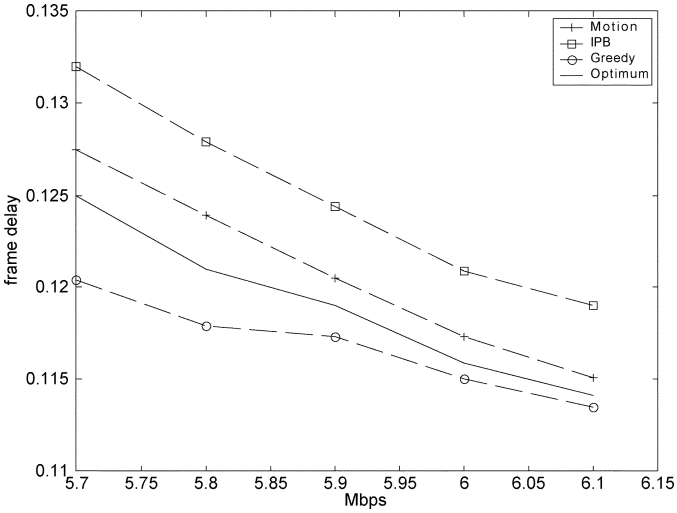


Fig. 12. DiffServ simulation I-frame delays.

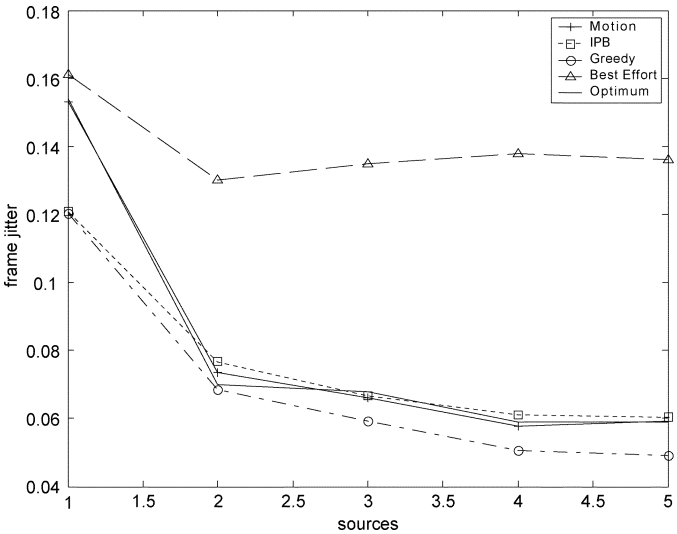


Fig. 14. DiffServ simulation II-frame jitters.

traffic into AF classes with AF3X getting the highest priority followed by AF2X followed by AF1X followed by Best-effort.

We have performed two sets of simulations. We set the parameters to have average delay of 0.1 seconds at core routers for both simulations I and II. The videos used for our simulations are constant bit rate (CBR) videos with 1.5 Mb/s.

In simulation I, we wanted to see how the proposed Motion Algorithm performs when compared with following simulation setups.

IPB Algorithm: Packets marked based on picture type.

Greedy Algorithm: Packets are marked to be forwarded with the highest AF class. If there is not enough budget left in the assigned class, then packets are passed to lower classes by CAR profile at edge router.

Optimal Algorithm: Actual distortion amount caused by each packet lost is calculated. Calculation is performed for all packets by dropping one packet and calculating the mean-squared error distortion at the receiver caused by the dropped packet. Packets are first ordered by the distortion amount they would cause if they were dropped, and then they are marked within the budget such that the packet with higher distortion amount is forwarded with higher class marking.

When we compared the end-to-end PSNR values, simulation I showed that when these algorithms are applied to different sources and sent through the same network simultaneously, the motion algorithm yields better PSNR performance than the IPB and Greedy algorithms. As far as frame delay and jitter are concerned, the Greedy algorithm performs better at the expense of the PSNR. This is not a surprising result because the greedy algorithm attempts to send more packets with better timing by using higher classes first. However, PSNR of reconstructed frames is expected to be worse since the packets that have more impact on the quality of the video are sent with low class markings more often than is the case with other methods. SNR performance for motion algorithm is within 1.5–2 dB range to the optimal algorithm (see Figs. 11–13).

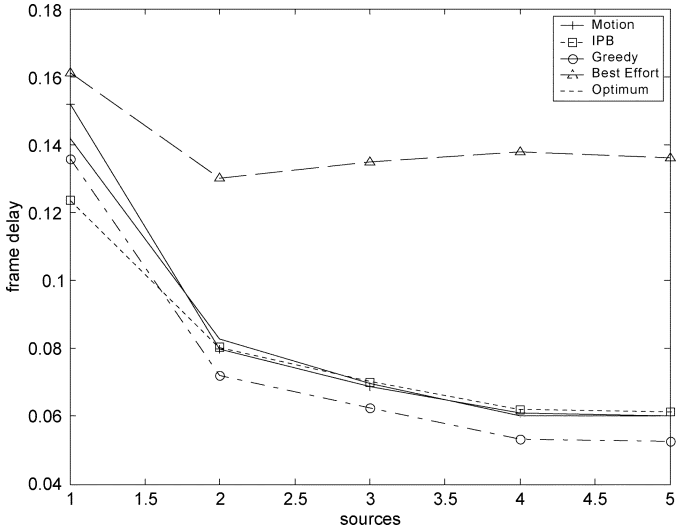


Fig. 15. DiffServ simulation II-frame delays.

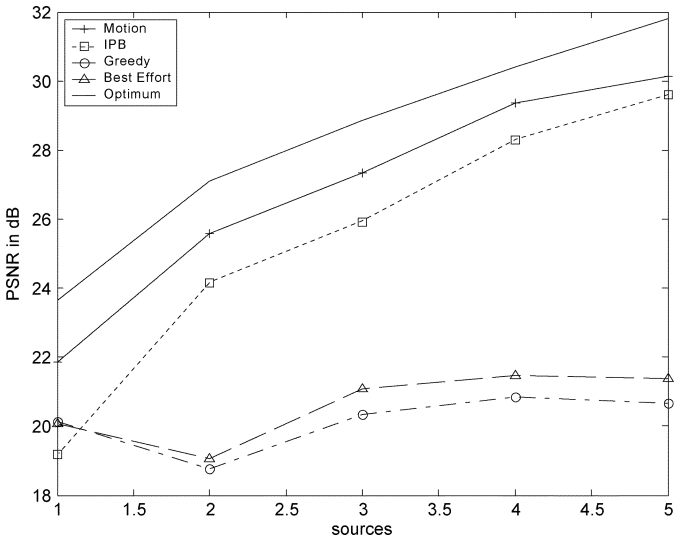


Fig. 16. DiffServ simulation II-PSNR.

In our second simulation, one source generates packets according to one of the algorithms and we vary the number of other sources generating background traffic to see how the changes occur in the algorithms when they are competing with only background traffic. We also provided Best-effort forwarding simulations as a benchmark, where all packets are sent over one communication channel with no packet marking. RED algorithm is still used to prevent tail drops. Simulation II showed that average frame delays and frame jitters converge, especially with increasing number of sources, regardless of how they are marked initially. This shows that when a large number of sources exists in the network, multiplexing gains of bursty sources increase and bursty packet losses due to bursty traffic decrease. Thus, the gap among the algorithms closes as far as the delay and jitter are concerned, with increasing number of sources. Moreover, the SNR performance of the motion algorithm stays superior to IPB, Greedy, and Best-effort algorithms. We conclude that the delay and jitter differences may become negligible among the algorithms using DiffServ, when a large number of sources exists in the network under a fixed budget constraint. Hence, the algorithm can be applied without undue concern about the delay and jitter in order to get better SNR performance (see Figs. 14–16).

V. Conclusion

Picture type and motion-based parameters that are embedded in RTP headers provide critical information about the encoded bit stream in the packet's payload. We have shown that by extracting these parameters from RTP headers, relative importance weights can be derived in real-time for RTP packets with MPEG-2 payload by means of our proposed algorithm. Once the relative weights have been determined, they can be used to enhance QoS and error resilience for video transport. We have shown that the performance of the proposed algorithm is limited to information that is available in real-time and is not optimal. Optimal performance, however, can be achieved only with an offline process where decoding of the compressed video is required. Algorithms that require new transport protocols or changes in the existing protocols in the TCP-UDP/IP environment are not desirable due to issues in implementation and adaptation. An important aspect of our proposed algorithm is that it does not require an additional transport protocol and is designed to work with existing RTP/UDP/IP protocols. Future research may include performance improvement by adjusting the parameters in the algorithm by using feedback information from the receiver. Additionally, emerging video coding standards such as H.264 are becoming increasingly important and may be investigated to implement similar algorithms. Moreover, the basic algorithm presented in this paper can be extended to layered coding techniques. In this case, it should be compared with existing UEP methods for progressive coding.

References

[1] *Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2, Recommendation ITU-T H.262, 1995.

[2] K. Stuhlmueller, N. Faerber, and B. Girod, "Adaptive optimal intra update for lossy video transmission," *Proc. SPIE*, vol. 4067, no. I, pp. 286–295, 2000.

[3] Y. L. Liang, M. Flierl, and B. Girod, "Low-latency video transmission over lossy packet networks using rate-distortion optimized reference picture selection," in *Proc. 2002 Int. Conf. Image Processing*, vol. 2, Sep. 2002, pp. 181–184.

[4] A. Mehaoua, S. Zhang, and R. Boutaba, "FEC-PSD: a FEC-aware video packet drop scheme," in *Proc. Global Telecommunications Conf.*, vol. 4, 1999, pp. 2091–2096.

[5] B. Girod, M. Kalman, N. J. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," in *Proc. 2002 Int. Conf. Image Processing*, vol. 1, Sep. 2002, pp. 9–12.

[6] A. Albanese, J. Blomer, J. Elmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1737–1744, Jun. 1996.

[7] K. Park and W. Wang, "QoS-sensitive transport of real-time MPEG video using adaptive forward error correction," in *Proc.IEEE Multimedia Systems '99*, 1999, pp. 426–432.

[8] M. Andronico, A. Lombardo, S. Palazzo, and G. Schembra, "Performance analysis of priority encoding transmission of MPEG video streams," in *Proc. Global Telecommunications Conf.*, 1996, pp. 267–271.

[9] P. Frossard and O. Verscheure, "AMISP: a complete content-based error-resilient scheme," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 989–998, Sep. 2001.

[10] W.-T. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 373–386, Mar. 2001.

[11] K. Mayer-Patel, L. Le, and G. Carle, "An MPEG performance model and its application to adaptive forward error correction," in *Proc. 2002 ACM Int. Conf. Multimedia*, 2002, pp. 1–10.
[12] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, "RTP payload format for MPEG1/MPEG2 video,", Rep. no. RFC 2250, Jan. 1998.
[13] Y. Mei, W. E. Lynch, and T. Le-Ngoc, "Joint forward error correction and error concealment for compressed video," in *Int. Conf. Information Technology: Coding and Computing*, 2002, pp. 410–415.
[14] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Comp. Commun. Rev.*, vol. 27, pp. 24–36.
[15] B. Johannes, K. Malik, K. Richard, K. Marek, L. Michael, and Z. David, "An XOR-based erasure-resilient coding scheme," Int. Comp. Sci. Inst., Berkeley, CA, Tech. Rep. TR-95–048, 1995.
[16] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1984.
[17] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction,", Tech. Rep. RFC 2733, Dec. 1999.
[18] D. Grossman, New terminology and classifications for DiffServ, Internet Engineering Task Force, Apr. 2002.
[19] H. Yu, D. Makrakis, and L. O. Barbosa, "Experimental evaluation of MPEG-2 video over differentiated services IP networks," in *Proc. IEEE Pacific Rim Conf. Communications, Computers and Signal Processing*, vol. 2, 2001, pp. 469–472.
[20] A. Ziviani, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "Improving the delivery quality of MPEG video streams by using differentiated services," in *Proc. 2nd Eur. Conf. Multiservice Networks ECUMN*, 2002, pp. 107–115.
[21] A. Ziviani, B. E. Wolfinger, J. F. de Rezende, O. C. M. B. Duarte, and S. Fdida, "On the combined adoption of QoS schemes to improve the delivery quality of MPEG video streams," in *Proc. Int. Symp. Performance Evaluation of Computer and Telecommunications Systems—SPECTS*, 2002, pp. 25–32.
[22] J. Shin, J. Kim, D. C. Lee, and C. C.-J. Koo, "Dynamic quality of service mapping framework for relative service differentiation-aware media streaming," in *Proc. Int. Conf. Information Technology: Coding and Computing*, Apr. 2001, pp. 30–34.
[23] J. Shin, J. Kim, D. C. Lee, and C.-C. J. Kuo, "Adaptive packet forwarding for relative differentiated services and categorized packet video," in *Proc. IEEE Int. Conf. Communications*, vol. 3, 2001, pp. 763–767.
[24] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

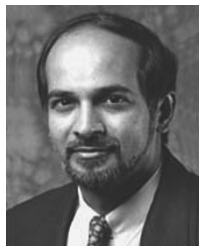
Bulent Cavusoglu received the B.S. degree in electrical and communication engineering from Yildiz Technical University, Istanbul, Turkey, in 1994, and the M.S. degree from Illinois Institute of Technology, Chicago, in 1997 and the Ph.D. degree from University of Illinois at Chicago in 2005, both in electrical and computer engineering.

His current research interests are in the areas of multimedia communication networks, multimedia compression, networking and image/video processing.

Dan Schonfeld (M'90–SM'05) as born in Westchester, PA, on June 11, 1964. He received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Johns Hopkins University, Baltimore, MD, in 1986, 1988, and 1990, respectively.

In August 1990, he joined the Department of Electrical Engineering and Computer Science at the University of Illinois, Chicago, where he is currently an Associate Professor in the Departments of Electrical and Computer Engineering, Computer Science, and Bioengineering, and Co-Director of the Multimedia Communications Laboratory (MCL) member of the Signal and Image Research Laboratory (SIRL). He has authored over sixty technical papers in various journals and conferences. He has served as consultant and technical standards committee member in the areas of multimedia compression, storage, retrieval, communications, and networks. He has previously served as President of Multimedia Systems Corporation and provided consulting and technical services to various corporations including AOL Time Warner, Chicago Merchantile Exchange, Dell Computer Corporation, Getco Corporation, EarthLink, Fish & Richardson, IBM, Jones Day, Latham & Watkins, Mirror Image Internet, Motorola, Multimedia Systems Corporation, nCUBE, NeoMagic, Nixon & Vanderhye, PrairieComm, Teledyne Systems, Touchtunes Music, Xcelera, and 24/7 Media. His current research interests are in multimedia communication networks; multimedia compression, storage, and retrieval; signal, image, and video processing; image analysis and computer vision; pattern recognition and medical imaging.

Dr. Schonfeld has also served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING on Nonlinear Filtering as well as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING on Multidimensional Signal Processing and Multimedia Signal Processing. He was a member of the organizing committees of the *IEEE International Conference on Image Processing* and *IEEE Workshop on Nonlinear Signal and Image Processing*. He was the plenary speaker at the *INPT/ASME International Conference on Communications, Signals, and Systems*.



Rashid Ansari (S'78–M'81–SM'93–F'99) received the B.Tech and M.Tech degrees in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1975 and 1977, respectively, and the Ph.D. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1981.

He has been at the University of Illinois at Chicago since 1995. He is currently Professor in the Department of Electrical and Computer Engineering, and in the past he has served as Director of Graduate Studies and as Interim Head. He was a Research Scientist at Bell Communications Research (Telcordia) from 1987–1995. Prior to that, he served on the faculty of Electrical Engineering at University of Pennsylvania. His research interest is in the general areas of signal processing and communications, and topics of research include image and video processing and analysis, video compression, multimedia signal processing and communication, data hiding, multirate filter banks and wavelets, OFDM transmission, and speech and audio analysis.

Dr. Ansari has been Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE SIGNAL PROCESSING LETTERS, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He was a member of the Editorial Board of the *Journal of Visual Communication and Image Representation* (1989–1993). He has served as member of the Digital Signal Processing Technical Committee of the IEEE Circuits and Systems Society. He was member of program committees of several IEEE conferences, and he served on the organizing and executive committees of the SPIE Visual Communication and Image Processing (VCIP) conferences. He was General Chair (jointly with M. J. T. Smith) of the 1996 SPIE/IEEE VCIP Conference.

Deepak Kumar Bal received the B.E. degree in electrical engineering from Regional Engineering College, Rourkela, India, in 1998 and the M.S. degree in electrical and computer engineering from University of Illinois at Chicago in 2003.

From 1998 to 2000, he worked for Wipro Technologies, Bangalore, India, as a Systems Engineer. Currently, he is working as a Software Engineer at Takata Inc., Detroit, MI. His areas of interest are computer networking and multimedia communication.