Spectrally Efficient Partitioning of MPEG Video Streams for Robust Transmission Over Multiple Channels

Wen Xu and Sheila S. Hemami
Cornell University, School of Electrical and Computer Engineering, Ithaca, NY 14853
email: {wxu, hemami}@ece.cornell.edu

Abstract

Reliable transmission of video over wireless networks must address both the limited bandwidth and the possibility of loss. When bandwidth sufficient to transmit the video is unavailable on a single channel, the video can be partitioned over multiple channels with possibly unequal bandwidths and error characteristics at the expense of more complex channel coding (error correction). This paper addresses the problem of efficiently partitioning forward error protected, pre-encoded video data for transmission over multiple channels. The assumption of pre-encoding precludes adjustment of source rates to the channels, since it is assumed that channel characteristics are not known until immediately prior to the start of transmission. The proposed partitioning exploits the structure of MPEG video, and frames in each group-of-pictures are reordered based on their decoding dependence and assigned to the channels to maximize the decodable frame rate. To be spectrally efficient, the frames are unequally error protected based on both frame types and channel packet loss rates. This technique is also applicable to scalable coding techniques. Simulation results demonstrate that the approach is effective under a variety of channel conditions and for a broad range of source material.

1 Introduction

Wireless systems are characterized by limited bandwidths and high loss rates, while streaming video transmission requires dedicated bandwidth and very reliable transmission. Bandwidth limitations can be overcome when multiple channels are aggregated into one higher bandwidth channel for transmission of a single video sequence [1]. Numerous error control mechanisms are available to compensate for the packet loss and delay common in wireless networks, trading information redundancy for reliability [2]. Automatic Repeat reQuest (ARQ) deals with channel errors by means of retransmitting lost packets. However, it requires a reverse channel to handle retransmission requests and results in longer delay, which can be unacceptable for video applications with latency constraints. Forward Error Correction (FEC) is an alternative to ARQ, which allows recovery from loss by introducing controlled redundant data without retransmission. Applying FEC across video data packets has been shown to be effective in an ATM environment [3]. In wireless networks, the error control mechanisms must also be spectrally efficient due to limited available bandwidth. Unequal Error Protection (UEP) combined with hierarchical encoding has been applied to video and achieves graceful degradation of picture quality and spectral efficiency in an error-prone environment [4]. However, each of these techniques is implicitly applied to transmitting video over a single channel. Bandwidth aggregation and error control can be combined to achieve higher rate reliable transmission. Optimization of this combination is addressed in this paper.

This paper proposes a framework that extends UEP to multiple channels while retaining UEP on the frame level. It is assumed that channel characteristics are monitored and estimated at intervals during the transmission process, giving bandwidth, latency, and packet loss rates (PLRs); this data is more detailed than what is known if Quality of Service (QoS) is guaranteed at the connection set-up. The variations in PLRs are considered in FEC selection for each channel. A frame reordering approach is proposed which uses frame decoding dependence to facilitate partitioning of video data across multiple channels and maximize the decodable frame rate. Dropped frames may be concealed by simply replaying previous frames at playback, or by any of the more sophisticated frame-loss concealment techniques [5]. An algorithm is proposed which optimally and efficiently partitions video data given any frame order under various channel conditions. A fast appoximation algorithm is also provided. Although the proposed approach is illustrated using MPEG video, it can be applied to MPEG-2 scalable coding schemes or extended to build more complex multiple channel streaming techniques.

This paper is organized as follows. Section 2 briefly reviews MPEG frame dependencies, then outlines the transmission model on which the partitioning problem and algorithm are based. In Section 3, the optimal-partitioning problem is formulated and solved. Experimental results are presented in Section 4. Section 5 concludes the paper.

2 MPEG and the Transmission Model

The partitioning problem and solution are illustrated using the MPEG compression standard [6], though they could easily be applied to other video coding algorithms in which encoded data has different importance in terms of decoding quality and consists of independent coded units. An MPEG stream consists of a sequence of intracoded (I), predictively coded (P), and bidirectionally predicted (B) frames, encoded as a group in a group-of-pictures (GOP). Each I-frame is encoded independently, a P-frame depends on the previous I- and P-frames in the GOP and a B-frame is coded with respect to both previous and the following I- and/or P-frames. Thus the frame sizes vary significantly. In practice, the size ratio of I-frames to B-frames is in the range of 1-10:1, depending on content. The MPEG encoding standard therefore provides an implicit transmission priority for the different types of frames. For example, the entire GOP cannot be decoded without the I-frame, while non-reception of a B-frame does not affect decodability of any other frames.

While the low bandwidth provided by wireless networks is often insufficient to support streaming video over a single channel, multiple channels can be used to provide a logical channel with a sufficient (or nearly sufficient) bandwidth. Assume that there are N ($N \geq 2$) channels having various characteristics (bandwidth, latency, PLR) available for transmitting one video sequence. It is assumed that channel characteristics are monitored and estimated at intervals during the transmission process, giving bandwidth, latency, and loss rates in real time. An algorithm that provides efficient bandwidth estimates, for example, is suggested in [7]. Each channel is thus parameterized by its bandwidth $BW_i(t)$, delay $D_i(t)$, and probability of packet loss $p_i(t)$ ($i = 1, 2, \ldots N$). A packet is considered lost either if it is physically lost due to buffer overflow or channel fading, or if it is delayed and arrives at the decoder after its decoding time.

FEC across packets is applied to each coded frame in order to recover lost packets. Because the multiple channels have different PLRs, data transmitted over each channel requires a different protection level. Because the MPEG frames have priorities in terms of GOP decoding, UEP is applied at the frame-level. After FEC protection the channel reliabilities are assumed to be approximately the same for a given frame type, though they vary for different frame types. As such, all frames of a given type are received with the same probability, and it is assumed that the error correction codes (ECCs) are selected to provide the desired reliability. Assume a UEP encoding scheme encodes packets by adding redundancy by factor $R_i^{(A)}(t)$ ($i = 1$, 2,$\ldots N$, $A =$ (I, P, B)). For instance, Reed-Solomon (RS) codes, which are both strong and have low delay, have been widely implemented for wireless transmission. If an I-frame in channel 1 is to be encoded by ($n_1^{(\mathrm{I})}$, $k_1^{(\mathrm{I})}$) RS-code, $R_1^{(\mathrm{I})} = (n_1^{(\mathrm{I})} - k_1^{(\mathrm{I})})/k_1^{(\mathrm{I})}$. $R_i^{(A)}(t)$ is determined by the current estimate of PLR $p_i(t)$ and the frame type A. Higher redundancy is given to more important frames. Furthermore, the same frame would assume different sizes after FEC encoding if allocated to channels with different PLRs. Implementation of a UEP encoding scheme requires a relation between PLRs, decoding visual quality, and various ECCs, which is beyond the scope of this work; rather this paper will address how to partition the video data assuming that the redundancy factors are known (note that this follows once specific ECCs are selected). Regardless of the specific ECCs selected, the following inequalities hold based on the above assumptions:

$$R_i^{(\mathrm{I})}(t) \geq R_i^{(\mathrm{P})}(t) \geq R_i^{(\mathrm{B})}(t) \tag{1}$$

$$R_i^{(A)}(t) \geq R_j^{(A)}(t) \quad \text{if and only if} \quad p_i(t) \geq p_j(t) \tag{2}$$

where $i,\ j = 1,\ 2,\ \ldots N$ and $A = (\mathrm{I, P, B})$.

The FEC-encoded video data is to be partitioned on a frame level; further segmentation of a single frame will produce increased overhead in order for receiver to decode multiple segments from different channels. An exception is made, however, to allow segmentation of a single frame when attempted transmission of the unsegmented frame would result in no frames being transmitted for a given GOP; this can occur for video sequences with extremely large I-frames when the available bandwidths are too small. Discussion of this case is provided in section 4.

The video data for each GOP is partitioned as a group, though the group could consist of integer numbers of GOPs or in general a segment of video data which is coded independently. The transmission process is thus described in discrete time with step size of the duration of a GOP, denoted as T. Bandwidth ($BW_i(t)$), PLR ($p_i(t)$) and redundancy factors ($R_i^{(A)}(t)$) are taken to be a constant within a step time T though they can vary throughout the duration of the entire video sequence; i.e., $BW_i(nT)$, $p_i(nT)$, and $R_i^{(A)}(nT)$ ($n = 0,\ 1,\ 2,\ \ldots$) (simplified as $BW_i(n)$, $p_i(n)$ and $R_i^{(A)}(n)$ in the following) represent bandwidth, PLR and redundancy factor in ($n+1$)-st GOP duration, respectively. Partitioning is performed when the sender buffer has one GOP (assume it has Q frames) to be transmitted and is completed within T seconds.

The physical decoder buffer size is assumed to be large enough so that the bottleneck during transmission is given solely by the bandwidths of the channels [8] (note that for pre-encoded sources the contraint from encoder buffer is eliminated). Under this assumption, connection bandwidths are the primary constraints in the optimization problem presented in the following section.

3 The Optimal Partitioning Problem

In the frame-level video partitioning problem a GOP consisting of Q frames is to be transmitted over N channels with differing bandwidths, PLRs, and latencies. Each GOP is to be transmitted subject to a delay constraint to allow continuous playback at the receiver (i.e., no delays between successive decoding of GOPs occur). It is assumed that the total available bandwidth is insufficient to transmit the entire coded GOP, so frames have to be dropped by the sender; the partitioning is performed to maximize the decodable frame rate. In this section the optimization problem is formulated.

3.1 Problem Formulation

Based on the transmission model, there are N^Q ways to divide an entire GOP into N non-overlapping sets corresponding to N channels at each time step. When the bandwidth is limited as assumed, a possible partition involves q ($0 \leq q \leq Q$) frames, the determination of which requires a complete search over all possible values. Let $S_i(n)$ ($i = 1, 2, \ldots N$) denote the set of frames that are allocated to channel i and S be the set of all frames to be transmitted in a GOP. These sets have the following properties:

$$S_i(n) \cap S_j(n) = \phi \qquad \text{if } i \neq j \tag{3}$$

$$S_1(n) \cup S_2(n) \cup \ldots \cup S_N(n) = S \tag{4}$$

where $i, j = 1, 2, \ldots N$.

$F^{(A)}$ refers to the non-FEC-coded data size of frame A. When a frame A is assigned to channel i, data is added in the fractional amount of redundancy factor $R_i^{(A)}(n)$, which increases its frame size to $(1 + R_i^{(A)}(n)) \cdot F^{(A)}$. Thus the total amount of data to be allocated to channel i ($1 \leq i \leq N$), denoted as C_i, is

$$C_i = \sum_{j \in S_i} (1 + R_i^{(A_j)}(n)) \cdot F^{(A_j)} \tag{5}$$

The partitioning optimization is performed for each GOP in a video sequence. To simplify notation below, the time index n is omitted.

It is desirable to transmit the entire GOP, but in many situations the channel bandwidths are so limited that some of the frames must be dropped. In this case the objective is to transmit as many decodable frames as possible in one GOP using the limited bandwidths, or more precisely, the limited bandwidth-time product, $BW \cdot T$. In order to effectively use the bandwidths, there is no need to deliver a frame if any one of the frames required to decode it is not delivered; the transmitted frames must be decodable without the dropped frames. At playback the missing frames can be replaced by replaying the latest decoded one, or by using a more sophisticated frame reconstruction technique [5].

The frames in a GOP are therefore reordered based on their encoding dependencies. The reordering occurs prior to the partitioning process. In the reordered sequence the j-th ($1 < j \leq Q$) frame is decodable on the condition that all frames from 1 to $j - 1$ are transmitted. Therefore, frames from the end of the reordered sequence can be removed without affecting the decodability of remaining frames. I- and P-frames are placed in early positions obviously, but the order of B-frames is arbitrary as B-frames do not depend on one another for decoding. For example, a GOP ($Q = 12$) is encoded as: $\mathrm{IB_1B_2P_1B_3B_4P_2B_5B_6P_3B_7B_8}$. The reordered sequence could be : (a) $\mathrm{IP_1P_2P_3B_1B_3B_5B_7B_2B_4B_6B_8}$, (b) $\mathrm{IP_1P_2P_3B}_{min} \ldots \mathrm{B}_{max}$, where B_{min} has the smallest frame size among all B-frames, B_{max} has the largest and the B-frames are sorted in ascending order of frame data amount. In the first case, the reordering minimizes the interval of unplayable frames. The corresponding decoded sequence is $\mathrm{IB_1\text{-}P_1B_3\text{-}P_2B_5\text{-}P_3B_7}$, for example, when the first 8 frames are transmitted. The interval of unplayable frames is one. The second reordering gives the maximum frame rate among all possible B orders, but the decoded video may have two-frame gap depending on encoded frame sizes.

Following reordering, the optimization problem is to maximize the number q ($1 \leq q \leq Q$) of transmitted frames, i.e. the cardinality of set S, such that the bandwidth constraints remain satisfied. The formulation is given as follows:

$$\max_{1 \leq q \leq Q} q \tag{6}$$

$$\text{subject to } C_i \leq BW_i \cdot T$$

where C_i are given by (5), $i = 1, 2, \ldots N$. The optimal solution may not be unique.

The above formulation is general and can be extended to work with other coding techniques. For instance, the MPEG-2 video-compression standard [6] has established four types of scalability: spatial, temporal, SNR, and data partitioning, in which some or all frames are partitioned into base and enhancement layers in different ways. By combining certain frames' two layers and rearranging the order of frames (or sublayers), the above described formulation can achieve a balance between frame rate and picture quality. The following discussions assume single layer MPEG encoding; the extension to scalable encoding is straightforward.

3.2 Delay Constraint

Along with the bandwidth constraints, delay poses another constraint for real-time video transmission. In typical video transmission, the end-to-end delay each frame experiences consists of several components, such as en/decoder delay, buffering delay, queuing delay and propagation delay. Here, only those components that vary among multiple connections are considered; common components are ignored.

The delay is defined as the time interval between the start of a partitioned GOP transmission and the moment when all delivered frames of the GOP arrive. It consists of two components: transmission delay, which is the time needed for an amount of data to "clock in" a channel and is given by (total bits)/bandwidth, and channel propagation delay, which comprises any intermediate delay as data travels through a certain connection.

Assume $D_i^p(t)$ is the propagation delay of the i-th channel. In general, these values are variable due to factors such as congestion at intermediate buffers. As noted previously, the delay is assumed constant during one GOP duration and the time index is heretofore omitted. Therefore, the delay experienced by video frames assigned to channel i ($i = 1, 2, \ldots N$) is given by

$$D_i = \frac{C_i}{BW_i} + D_i^p \tag{7}$$

Let T_D be an upper bound to the delay of all channels, which contributes to the initial delay before the decoder starts decoding a GOP and applies to all successive GOPs. Then

$$D_i \leq T_D \tag{8}$$

By (7) and (8), the constraints in (6) are modified as

$$C_i \leq BW_i \cdot (T_D - D_i^p) \tag{9}$$

for $i = 1, 2, \ldots N$. The tradeoff between delay and frame rate is obvious in this equation. A large T_D, which implies a longer initial latency, yields looser constraints and thus achieves a higher frame rate. An increased propagation delay for a fixed T_D reduces a channel's capability to carry more video data.

The selection of T_D is related to many factors such as channel propagation delays, initial latency and continuous playback at the receiver. It is therefore desirable to avoid the determination of T_D in order to simplify the formulation and algorithm. Equation (9) can be shown to coincide with the previous constraints in (6) where $(T_D - D_i^p)$ is replaced by the GOP duration T under certain assumptions.

Consider two consecutive GOP transmissions. For simplicity of notation, let the first GOP start at $t = 0$. The following time instants are of interest, as shown in Figure 1:

t_1 - the maximum amount of time when all channels finish "clocking-in" data in the first encoded GOP as allocated.

t_2 - start of transmission of the second GOP.

t_3 - time by which all delivered frames in the first GOP have arrived at receiver.

t_4 - time by which all delivered frames in the second GOP have arrived.

Their relative relations with T_D and T are:

$$t_3 \leq T_D \tag{10}$$

$$t_4 - t_2 \leq T_D \tag{11}$$

$$t_4 - t_3 \leq T \tag{12}$$

where the first two inequalities come from (8). The third provides continuous playback at the decoder, where consecutive GOPs arrive within the playback duration of one GOP. Assume equalities are taken in (10) and (11), which implies T_D is taken to be as small as possible, and

$$t_3 = t_4 - t_2 \tag{13}$$

Combined with (12), this yields

$$t_2 = t_4 - t_3 \leq T \tag{14}$$

The difference $(t_2 - t_1)$ is the processing time associated with partitioning a GOP and performing the FEC encoding. Thus, t_1 as defined satisfies

$$C_i \leq BW_i \cdot t_1 \tag{15}$$

Therefore from (14) this yields

$$C_i \leq BW_i \cdot t_1 \leq BW_i \cdot T \tag{16}$$

for $i = 1, 2, \ldots N$, which is exactly the constrait used in (6). The second equality is obtained when the processing time is negligible and thus $t_1 \cong t_2 \cong T$.

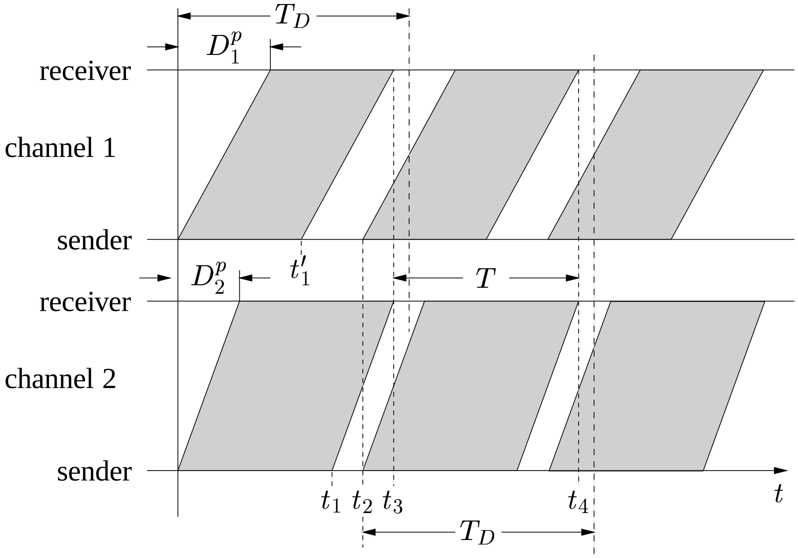

Figure 1: Consecutive GOP time instants ($N = 2$ channels)

The error caused by replacing $(T_D - D_i^p)$ with T can be ignored when the above equality assumptions hold. In practice, however, there is a small gap between t_3 and T_D, as exaggerated in Figure 1, due to the partition unit (a frame) in the formulation. The loss of bandwidth-delay product is usually less than a B-frame size. For the last equality in (16) to hold, a fast partitioning algorithm is necessary, which is discussed in the next section. The processing time to perform FEC encoding is ignored as current RS encoders can easily process tens or even hundreds of Mb/s. Note, however, that t_1 is the time instant when all channels finish transmission of current GOP data. Channels with longer propagation delays (channel 1 in Figure 1, for example) finish earlier, assume at t_1', and the difference $(t_2 - t_1')$ is thus enlarged. The replacement of $(T_D - D_i^p)$ with T results in an overestimated solution for these channels. Therefore, it is preferred that the available channels have propagation delays D_i^p with small variation.

3.3 Partition Algorithms

As noted above, the search space is $O(N^Q)$ for the solution to the optimization problem, which grows exponentially when the number of channels or frames in a GOP increases. It is therefore crucial for a partition algorithm to be fast from both the delay and computational efficiency aspects. Two algorithms are considered and each provides desirable performance under certain circumstances.

A fast solution is to use greedy algorithm, which allocates frames beginning with the best channel and ending with the worst. Because the more reliable channels are more spectrally efficient, this gives an approximate partition solution with polynomial running time. However, it doesn't guarantee the optimal solution due to the discrete unit (frame) and variable frame sizes.

An alternative is to use a Pruned-Tree-Search (PTS) algorithm, which can find the optimal solution more efficiently than an exhaustive enumeration. The idea behind PTS is to reduce the search space by identifying subspaces that cannot contain the optimal solution. A tree structure is employed where every node represents a possible solution. Tree depth corresponds to number of frames and branches represent channel assignment. Constraints, as given in (6), are associated with each node. Because of the linearity and nonnegativity of terms in (5), the search space can be greatly reduced by evaluating constraints starting from the root and pruning impossible sub-trees. Theoretically, the PTS algorithm has a worst-case complexity comparable to that of a complete search, i.e. $O(N^Q)$. However, experiments indicate that it finds the optimal solution at a significant speed-up due to the data profile of typical video traces.

4 Experiments and Results

A number of experiments have been performed to evaluate the performance of the proposed model under various channel conditions. Real-life MPEG video traces varying in content including Movies (*Star Wars*), Sports (*Soccer*), *News* and *MTV* were used in the performance evaluation. The frame size traces were extracted from MPEG-1 sequences (384×288, 25 frames/s, $40,000$ frames) encoded with the Berkeley MPEG-encoder (version 1.3) with the following parameters [1]: YUV(4:1:1, 8bits), Quantization value (I=10, P=14, B=180), “Logarithmic” / “Simple” motion vector search, 1 slice and half pel motion estimation. The GOP's are of pattern IBBPBBPBBPBB with fixed size 12. [9]

In the following experiments, $N = 3$ and channel 3 is the most reliable while channel 1 is the least reliable. The redundancy factors are taken as ($R_1^{(\mathrm{I})} = 0.48$, $R_1^{(\mathrm{P})} = 0.16$, $R_1^{(\mathrm{B})} = 0.03$), ($R_2^{(\mathrm{I})} = 0.35$, $R_2^{(\mathrm{P})} = 0.10$, $R_2^{(\mathrm{B})} = 0$), ($R_3^{(\mathrm{I})} = 0.24$, $R_3^{(\mathrm{P})} = 0.08$, $R_3^{(\mathrm{B})} = 0$), respectively, based on observations in [10, 11].

It occurs occasionally that none of the frames in a GOP can be transmitted while the total bandwidth is sufficient for several frames. This is due to the fact that some I-frame sizes are too large, commonly containing 40-60% of the bits in the entire GOP; none of the channels can transmit these large I-frames. The problem can be fixed if the I-frames are allowed to be further segmented into smaller parts, each of which is considered the same way as a frame. Each segment introduces additional overhead; therefore, the number of segments should be kept as low as possible. Based on experiments considering the number of channels, their bandwidths, and the ratio of I-frame size to that of the entire GOP, the number of segments is selected to be 3.

In the following, the frame rates achieved by the two algorithms are first compared. Then the effects of channel bandwidth distribution, parameter mismatch and varying video content are evaluated. When the optimal partitioning solution is not unique, one optimal solution is arbitrarily chosen.

---

[1] MPEG traces available via anonymous FTP from ftp-info3.informatik.uni-wuerzburg.de/pub/MPEG/

4.1 Comparison of Pruned-Tree-Search and Greedy Partitioning

When the partitioning is performed in a greedy manner, the optimal solution is not guaranteed because earlier assignments cannot be adjusted according to upcoming frame sizes. If it is performed in normal frame order, some received frames may not be decodable because other frames needed to decode them are not transmitted. Both result in a reduced frame rate on average compared to PTS with frame reordering, denoted as "PTS-reorder" below. In the following, PTS-reorder is compared with a greedy algorithm both without and with frame reordering.

A. Comparison of PTS-reorder and Greedy Algorithm without Frame Reordering

A comparison of decodable frame rate achieved by the PTS-reorder scheme and a greedy algorithm without frame reordering is performed with the total available bandwidth ranging from 70% to 150% of the average bit rate. The three channels are assumed to each have one third of the total bandwidth. The best results obtained with the greedy algorithm are when the frames are allocated to the channels cyclically. The PTS-reorder scheme slightly outperforms the greedy partition without reordering in maximizing the average number of decodable frames per GOP, with the average difference within 1 frame. The difference decreases at the two extremes of available bandwidth. At low bandwidths, only I-frames are transmitted for most GOPs in both schemes. At high bandwidths, all frames are transmitted in most GOPs, and the PTS-reorder scheme performs in a greedy manner due to sufficient available bandwidth.

There are GOPs where a simple greedy scheme achieves a slightly higher frame rate than the PTS-reorder scheme, as indicated by the negative numbers in Table 1. This occurs when the channel bandwidths are lower and P-frame sizes happen to be larger. In the reordering scheme, frames with larger sizes (I- or P-) are reordered to early positions. At lower channel bandwidths, the greedy algorithm may achieve a higher frame rate as shown by $IBBP_1$ compared with IP_1P_2, for example. However, PTS-reorder provides an evenly distributed spacing of frames within a GOP due to the delivery of P-frames, especially at lower frame rate. An even spacing can reduce the effect of time aliasing when frame rate up-conversion is performed at the receiver.

B. Comparison of PTS-reorder and Greedy Algorithm with Frame Reordering

A comparison of decodable frame rate achieved by the PTS and greedy algorithms with frame reordering is performed under the same channel bandwidth conditions as in the above experiment. When the frames are reordered in both algorithms as proposed, the frame rate difference decreases. However, PTS guarantees the optimality of the solution and thus outperforms greedy algorithm in all GOPs.

Table 1 lists differences in average GOP frame rates achieved by the PTS-reorder approach and greedy partitions in both cases. An average GOP frame rate was computed over 100 GOPs for each sequence with the total available bandwidth varying over the range [70%, 150%]; the table provides the maximum and minimum differences between the two partitions over the bandwidth range.

When varying bit rate video is transmitted over constant rate channels, there are segments with low (or high) video data rate in which both algorithms easily achieve full (or zero) frame rate and contribute zero difference. Therefore, enormous differences (possibly positive and negative) are averaged out over 100 GOPs. A comparison between the two algorithms removing this effect is performed where channel bandwidth-time products are assumed as a fixed factor of GOP data amount. Despite the unrealistic nature of the assumption, it provides useful insight into the algorithm performance. Table 2 lists the difference in GOP frame rate between the two algorithms with frame reordering under this assumption. It shows that the greedy algorithm, despite its suboptimality, performs as well as PTS in more than half of the GOPs tested.

Table 1: Frame rate difference between PTS and greedy partitions with channel bandwidth varying over [70%, 150%] of video average bit rate. Positive numbers indicate that PTS-reorder achieves a higher average frame rate. B-frames are sorted as discussed in section 3.1 in frame reordering. Frame rates are averaged over 100 GOPs for each bandwidth; minimum/ maximum are then taken over the bandwidth range.

Video clips (bits/s)	Max average difference (frames/GOP)		Min average difference (frames/GOP)	
	without reord.	with reord.	without reord.	with reord.
Star Wars (233k)	0.51	0.38	-0.1	0.04
Soccer (678k)	0.70	0.27	0.0	0.12
MTV (615k)	0.58	0.32	-0.09	0.0
News (517k)	0.70	0.21	0.0	0.0

Table 2: Frame rate difference between PTS and greedy algorithms with channel bandwidth-time product given by a factor (0.9) of current GOP data amount. The three channels have equal bandwidth. B-frames are sorted as discussed in section 3.1. Results are provided for the same frame reorder. Numbers in the brackets refer to the number of GOPs with the difference value (out of 100 GOPs).

Video clips (bits/s)	Frame rate difference (frames/GOP)		
	Max	Min	Mean
Star Wars (233k)	2	0 (61)	0.40
Soccer (678k)	2	0 (57)	0.47
MTV (615k)	6	0 (60)	0.45
News (517k)	2	0 (78)	0.23

C. Computational Complexity

The greedy algorithm has $O(Q)$ running time. The PTS algorithm has dramatically varying running time ranging from milliseconds to seconds depending on the pruned-tree size.

Despite its exponential running time, the PTS-reorder scheme is used in the following experiments for its optimality in order to concentrate on the effects of other factors. The greedy algorithm is expected to achieve results that are slightly sub-optimal.

4.2 Performance Effects of Differing Channel Bandwidth Distributions

The total bandwidth will in general not be equally distributed among the available channels. The PTS-reorder scheme is evaluated with the total available bandwidth ranging from 50%-200% of the video average bit rates, and with three different distributions of this bandwidth to the three channels. Let β_i ($1 \leq i \leq 3$) be factors representing the normalized bandwidth of the channels, the sum of which is equal to 1; the three distributions are "equal" ($\beta_i = 1/3$, $i = 1, 2, 3$), "half" ($\beta_1 = \beta_2 = 0.25$, $\beta_3 = 0.5$), and "skew" ($\beta_1 = \beta_2 = 0.1$, $\beta_3 = 0.8$).

The partitioning yields nearly the same average frame rate for different bandwidth distributions over the bandwidth range tested. "Skew" achieves a slightly higher frame rate (an increase of less than 0.2

frames/GOP) at bandwidths below 100% for *Star Wars* and *Soccer* traces; it provides a larger bandwidth in the most reliable channel and thus is efficient in error coding redundancy. For *News*, however, "skew" provides the lowest frame rate (the difference can reach 1 frame/GOP). In this case a large I-frame is allocated to channel 3 and the bandwidths of the other two channels are low relative to the P- or B-frame size and they are therefore frequently not used. The frame rate variations are slightly larger using the "skew" distribution, followed by "half" and "equal". In terms of computation speed, "skew" is twice as fast as "half" and about 10 times faster than "equal". Because more bandwidth is available in one of the channels, the pruned-tree is biased to certain branches and the optimal solution approaches that of a single channel.

4.3 Performance Effects of Channel Parameter Mismatch

The performance of the PTS-reorder scheme depends on the accuracy of channel parameters provided. When the parameters are inaccurate, the achieved GOP frame rate is not what was expected. Furthermore, the solution based on assumed parameters must adapt to actual channel conditions by dropping some allocated frames when the actual bandwidth is lower, or by transmitting remaining frames in the GOP in a greedy manner when more bandwidth is available (further partitioning of the remaining frames is disabled due to the lack of accurate channel parameters). Here, all channels are assumed to have the same mismatch for simplicity, though in practice parameters associated with the available channels are independent of one another if the channels are disjoint. The mismatch range considered is between -15% and $+5\%$, with negative difference indicating that the actual bandwidth is lower than assumed. A mismatch of 0% is equivalent to accurate bandwidth information. Based on the results described in the previous section, only the "equal" distribution of channel bandwidths is considered.

The average frame rate drops by 6-7 frames/GOP at -15% mismatch, and increases by 1-2 frames/GOP at $+5\%$, depending on video content and the ratio of assumed bandwidth to video average bit rate. Table 3 lists the difference between the expected frame rate based on assumed parameters and the actual frame rate when all three channels have -5% bandwidth mismatch. The mismatch performance depends on the specific partitioning solution. There are optimal solutions in which the I- or more P-frames are allocated to one of the channels without any B-frames in the same channel. In this case all the dependent frames in other channels must be dropped when the I- or one of P-frames is dropped due to reduced actual channel bandwidth. If mismatch is common, modifications to the partitioning must be made, trading off robustness to mismatch and optimal utilization of bandwidth. For example, to allocate one or more B-frames to each channel will increase the robustness of the partition to bandwidth mismatch; however, this may not result in the maximum frame rate.

Table 3: Differences between expected frame rate and actual frame rate at -5% bandwidth mismatch for various video traces. The total bandwidth is 130% of video average bit rate. Results are provided for 100 GOPs using PTS-reorder scheme.

Video clips (bits/s)	Ave. difference (frames/GOP)	Max difference (frames/GOP)	Standard deviation (frames/GOP)
Star Wars (233k)	3.67	12	3.96
Soccer (678k)	2.21	12	2.97
MTV (615k)	1.26	11	2.44
News (517k)	3.28	11	4.05

The mismatch performance of PTS is higher than that of the greedy algorithm on average when actual channel bandwidth is within about 5% of provided parameters. With larger mismatch, their performances are nearly identical. This is because the solutions provided by the two algorithms typically differ in very few frames. When mismatch increases, both algorithms result in significant loss of frames and give similar performance on average.

4.4 Effects of Differing Video Content

Various video content generates different patterns of frame size traces. For example, *News* has a large variation across both frames and GOPs. There are multiple segments (over multiple GOPs) in which the average bit rate is significantly lower than the average bit rate of the entire sequence. In contrast, the variation of *Star Wars* is small and the average bit rate per GOP is closer to the average bit rate of the entire sequence. These characteristics impact the achieved frame rates when the total available bandwidth varies. Figure 2 illustrates their average decodable frame rate over 100 GOPs when the total available bandwidth ranges from 50%-200% of the video bit rate. Again, the channels have equal bandwidth distribution. *News* achieves the highest average frame rate when the available bandwidth is greater than or equal to its average bit rate. At lower bandwidths, however, the average frame rate drops significantly because *News* has very large I-frames in most GOPs which cannot be transmitted. Additionally, when the available bandwidth drops to 50% of video bit rates, almost no frames per GOP can be transmitted for *Star Wars* and *News* but 2 frames/GOP can be transmitted for *Soccer* and more than 5 frames/GOP are transmitted for *MTV*. When the total bandwidth exceeds 150% of the video average rates, over 10 frames/GOP on average can be transmitted for all four video traces.

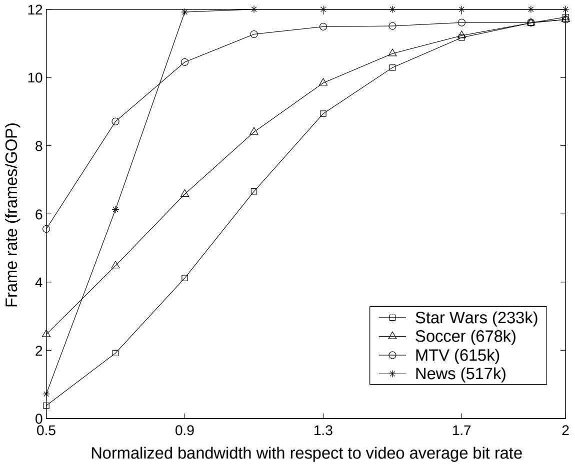

Figure 2: Average frame rate (frames/GOP) over 100 GOPs using PTS-reorder algorithm vs. total channel bandwidths (normalized with respect to video bit rate) for various video traces. The three channels have equal bandwidth.

5 Discussions and Conclusion

This paper has proposed a framework to efficiently partition pre-encoded video sequences for transmission over multiple channels with various qualities. The available bandwidths of the channels are the primary constraints in the partitioning process. The maximum decodable frame rate is achieved after the partition. The formulation is independent of the FEC mechanisms and can adapt to varying channel conditions.

The PTS algorithm guarantees the optimality of the partition solution for a given frame order, and along with the proposed frame reordering scheme it achieves a better average performance than a simple greedy approach. However, its exponential running time discourages its use for increased number of channels or frames in a GOP. In the situations with tight timing constraints the greedy algorithm is a fast approximation algorithm. The bandwidth distributions among multiple channels have little influence on the partitioning performance, but do affect the computational complexity. Parameter mismatch analysis demonstrates that if mismatch is common, modifications to the algorithm can be made at the expense of lower frame rate and bandwidth utilization.

The partitioning formulation demonstrated in this paper is basic, on which more complex video transmission schemes can be developed. It is invoked whenever the current available resource (bandwidth) drops below the current requirement (video data rate), despite the efforts of possibly limited adjustments of resource assignment (e.g. channel rate control). Exploiting the *a priori* knowledge associated with pre-encoded video sources allows multiple GOPs to be processed in advance of their transmission times when the channel parameters are relatively stable or can be pre-estimated using accurate channel models. This is expected to further improve the channel bandwidth utilization, and can facilitate multiple channel video streaming with dynamic partitioning.

References

[1] P. H. Fredette, "The past, present, and future of inverse multiplexing," *IEEE Comm. Magazine*, vol. 32, no. 4, pp. 42–6, April 1994.

[2] Y. Wang and Q. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–97, May 1998.

[3] E. W. Biersack, "Performance evaluation of forward error correction in an ATM environment," *IEEE JSAC*, vol. 11, no. 4, pp. 631–640, May 1993.

[4] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Comm.*, vol. 15, no. 1-2, pp. 77–94, Sept. 1999.

[5] R. Castagno et al., "A method for motion adaptive frame rate up-conversion," *IEEE Trans. CSVT*, vol. 6, no. 5, pp. 436–46, Oct. 1996.

[6] B. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Chapman & Hall, London, UK, 1997.

[7] Q. Zhang et al., "Resource allocation with adaptive QoS for multimedia transmission over W-CDMA channels," in *Proc. WCNC*, Chicago, IL, Sept. 2000, vol. 1, pp. 179–84.

[8] A. R. Reibman and B. G. Haskell, "Constraints on variable bit rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 4, pp. 361–72, Dec. 1992.

[9] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proc. of the 20th Conf. on Local Computer Networks*, Minneapolis, MN, Oct 1995, pp. 397–406.

[10] M. Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 357–372, Mar. 2001.

[11] F. Hartanto and H. R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks," in *Proc. of 10th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 1999)*, Sydney, Australia, Nov 1999, pp. 21–4.