AN ENHANCED SMOOTHING SCHEME FOR MPEG VIDEO STREAMS TRANSMISSION

Joelson Tadeu Vendramin, Keiko Verônica Ono Fonseca
Centro Federal de Educação Tecnológica do Paraná – CEFET-PR,
Programa de Pós Graduação em Engenharia Elétrica e Informática Industrial – CPGEI,
Av. Sete de Setembro, 3165, Curitiba-PR, Brasil

Keywords: video smoothing, MPEG video transmission, video trace files.

Abstract: Compressed video transmission over UDP/IP-based networks leads some challenging studies. One of them refers to the problem of minimizing the burstiness of video compressed traffic as it leads to poor network bandwidth utilization. Video smoothing has been proposed as one solution to this problem. This paper presents a smoothing algorithm to reduce such burstiness. The algorithm can bring good results in terms of peak bandwidth reduction while keeps a simple implementation if compared to other smoothing schemes.

1 INTRODUCTION

Today, with the development of Internet and the introduction of new network architectures, it is possible to transmit multimedia real-time traffic that have stringent Quality of Service (QoS) requirements (Bakiras, 2002). That made services such as Video-on-Demand (VoD), videophone, and TV-quality videoconferences feasible. In order to optimize storage and/or network resource capacity, video compression has been largely applied. The basic idea of video compression is to explore its spatial and temporal redundancies. One widely used family of open compression standards suitable for transmitting video frames over UDP/IP-based computer networks is that defined by the Motion Picture Experts Group (MPEG).

Video compression, however, can present a bursty nature and its traffic can lead to poor bandwidth utilization. Therefore, effective transmission schemes are required to transmit streamed video across the network. In a client-server scenario these schemes are known as smoothing algorithms, as they provide a new and less bursty transmission plan at the server side.

In that perspective, this paper presents a smoothing algorithm developed to achieve higher video quality under a given network capacity condition, while making an efficient bandwidth utilization. The algorithm uses the properties of a compressed video in the MPEG format. This paper is organized as follows. Section 2 describes the MPEG video standard; section 3, presents concepts of smoothing a video stream and section 4 the algorithm implementation itself is described. Section 5 brings the experiments developed and the simulation scenario. Finally, section 6 addresses the main results and section 7 concludes the paper and presents some future directions.

2 MPEG VIDEO STANDARD

Basically MPEG codes the video in a layered structure where the most important element is the frame (essentially the screen area). There are three kind of frames in MPEG: Intracoded (I), predictive (P) and bidirectional (B). Stand-alone static pictures, coded with JPEG (Joint Pictures Expert Group) algorithms, compose I-frames. P-frames are composed with the block to block differences with the last I or last P-frames. Finally B-frames are predicted from nearest past I or P- frame and future I or P-frame. These three frame types are arranged in a pattern called Group of Pictures (GoP), which is unique and repeats itself along the duration of the video stream (see figure 1).

The MPEG video stream over an IP-based computer network will be seen as a quasi-periodic traffic, representing the GoP structures, with high network utilization during the transmission of I

Tadeu Vendramin J. and Veronica Ono Fonseca K. (2004).
AN ENHANCED SMOOTHING SCHEME FOR MPEG VIDEO STREAMS TRANSMISSION.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 379-384
DOI: 10.5220/0001393103790384

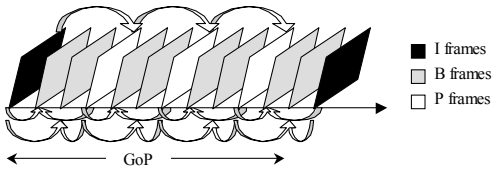

Figure 1: An MPEG GoP pattern and the three frame types.

frames and low-to-moderate network utilization during the transmission of P and B-frames. Such traffic generates Variable Bit Rates (VBR) that can degrade the network bandwidth utilization if not properly controlled. Hence, this justifies the importance of study smoothing mechanisms (Zhang, 1997).

3 SMOOTHING OF THE MPEG STREAM

The video transmission rate variability is a major challenge in achieving optimal bandwidth utilization. Considering the transmission of stored video from a server to a client across the network, a smoothing procedure tries to make the traffic less bursty, thus avoiding the waste of network resources. Also, as a general real-time design principle, less bursty (or *smoother*) traffic is easier to manage (Salehi, 1998), for example, when multiplexing several streams.

Basically, smoothing techniques can be divided into three categories: Smoothing by *temporal multiplexing*, smoothing by *stream aggregation* and smoothing by *work-ahead* (Salehi, 1998).

Smoothing by *temporal multiplexing* is achieved when a buffer is introduced somewhere in the path between the video server and the client. This can smooth the stream's peak rate but introduces extra delay (Wrege, 1996). The *stream aggregation* smoothing is statistical based. By the Central Limit Theorem, when several independent streams share a given resource their aggregate bandwidth requirements converge to a Normal distribution, provided that they have service times with finite variance (Salehi, 1998). Finally, the *work-ahead* smoothing is achieved when the server sends the video data ahead of schedule. The constraints are that (i) the data is available to be sent, and (ii) the client has sufficient buffer space to receive it (Reibman, 1992).

The *work-ahead* smoothing algorithms are often referred as *off-line* smoothing in the sense that all the work is done prior any transmission.

Another way to classify these algorithms is based on some optimization criteria. For example the *critical bandwidth allocation* (CBA) tries to minimize the number of bandwidth increases (Feng, 1995; Sechrest, 1995), while the *minimum changes bandwidth allocation* (MCBA) tries to minimize the number of rate decreases (Jahanian, 1995). Another optimization criterion (Salehi, 1998) is to achieve the greatest possible reduction in rate variability.

A *work-ahead* smoothing algorithm, called Enhanced Piecewise Constant Rate Transmission and Transport (e-PCRTT) is particularly important, since it was used as the "starting point" of our algorithm. The e-PCRTT algorithm divides the video stream into fixed size time intervals and then computes, for each of them, a transmission rate that will not overflow nor underflow the client's buffer, thus creating a transmission plan. As it knows, a priori, all frame lengths and the client buffer capacity, it can significantly reduce the traffic burstiness (Hadar, 2001).

4 ALGORITHM PROPOSAL

We propose an extension (modification) of the e-PCRTT algorithm. Instead of working with fixed size time intervals, our algorithm tries to extend as much as possible the calculated transmission rate, always trying to fit the transmission plan adjacent to the middle of the buffer's limits, as will be seen later.

(Feng, 1997) analyses several smoothing algorithms and concludes that the PCRTT is somewhat unstable under some conditions mainly because of the partitioning of frames at fixed intervals.

By removing this fixed size dependence and maintaining the e-PCRTT's simplicity as compared to other smoothing schemes (like CBA or MCBA), our algorithm can bring good results in the peak bandwidth reduction.

The algorithm was developed using C language. As input, it takes a text file (representing a original video trace data[1]) and the client's buffer size. Its output is a new text file modified to reflect the new transmission plan.

Figure 2 shows how the algorithm works. The *L* curve represents the accumulated bytes sent to the client during the video stream and the *U* curve is equal to *L* curve plus the client's buffer size. The transmission plan is the traced line that must stay in the middle path between L and U, or in other words, the transmission plan must be feasible.

[1] A video trace data is, basically, a text file that carries video information such as: frame size, type, a sequence number and its time.

As can be observed, the transmission plan is composed by successive constant bit rate runs. These bit rates are incremented (points a and c) or decremented (points b and d), thus preventing buffer underflow or overflow, respectively.

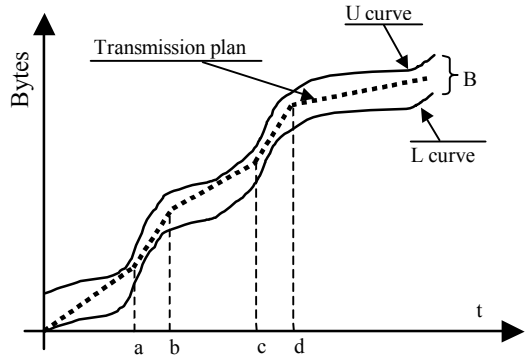

Figure 2: A smoothing transmission plan.

Figure 3 brings the algorithm's pseudocode. Observe that it can be considered an hybrid version of e-PCRTT and MCBA.

5 EXPERIMENTS AND SIMULATION MODEL

To evaluate the algorithm two sets of experiments were conducted. At the first experiment set, we evaluate the mean bit rate and the peak bit rate, using different trace files and different buffer sizes. The following video trace files, obtained from (Fitzek, 2001) were used: Jurassic Park (a film); Silence of the Lambs (another film); ARD Talk (an speech) and *Susi und Strolch* (a cartoon).

These files are coded with MPEG-4 format but use the entire scene as a unique video object, so they can be considered, without leak of accuracy, as MPEG-2 streams. We adopted the same trace files for sake of result comparison.

The duration of all trace files were truncated to 1800 seconds (half an hour), than each of them was used as input to our algorithm with five different buffer sizes: 256 k, 512 k, 1024 k, 2048 k and 3072 kbytes.

Finally, for each trace file (the original and smoothed) the mean and the peak bit rates were extracted. The peak-to-mean ratio, as will be described, is assumed to be a performance metric of the smoothing algorithm.

At the second experiment set, we evaluate the performance of the smoothed stream in a client-server scenario where a video source competes for bandwidth with a concurrent traffic, i.e, any network application that shares a common link with the video stream. A typical example is a File Transfer Protocol (FTP) session.

A six-node NS-2 (Network Simulator) (Fall, 1999) scenario was built, implementing the client-server video model and a client-server competing FTP traffic. This model is depicted in figure 4.

All the network elements are identified as nodes, and at each node, an agent can be attached. Here s_1 and s_2 represent the video server and the competing traffic source, respectively. Their destinations nodes are d_1 and d_2. Node pairs define the network links. Each of them has its own bandwidth, delay and packet discard discipline parameters. The bottleneck link is r_1 - r_2; if it has not enough bandwidth to deal with the aggregate traffic, packet losses will occur and will be registered at an NS' trace output file. Node s_2 represents a FTP traffic source (Fall, 1999).

The following points were considered during the experiment 2:

- The trace file used in all simulations was Jurassic Park (and its 1024 k, 2048 k and 3072 kbytes smoothed versions);
- The smoothed trace files for buffers of 256 k and 512 kbytes were not used in the simulations;
- The simulation has a duration of 1800 s;
- The FTP source act three times during the simulation: At 100, 500 and 1500 s, having the following durations: 100, 500 and 100 seconds, respectively.

The next sections bring the performance metrics used in the simulation as well its results.

6 RESULTS

Figure 5 brings the peak-to-mean ratio as function of the client's buffer size for the four trace files. In this graph, a zero buffer means no smoothing (that is, the original trace file).

The lower this ratio is, less bursty will be the traffic. As it can be seen, better results are from the ARD Talk trace: With a client's buffer of 3072 kbytes the peak-to-mean ratio is almost 1. The *Susi und Strolch* trace also brings good results, but for buffer sizes upper to 1024 kbytes. Jurassic Park has almost a linear behavior as the buffer increases and, finally, the worst performance was observed in Silence of the Lambs.

```
// Part 1: Reads the input (file and buffer) and computes L/U curves
Read (Original_trace_file, Buffer_size)
For each frame in (Original_trace_file) do
     Computes L[i] (sum of past frame sizes until frame i)
     Computes U[i] = L[i] + Buffer_size
End

// Part 2: Computes the transmission rates inside each smoothing interval
"start_point" of transmission plan begins in L[1]
While exist frames in (Original_trace_file) do
     Increments frames by 1
     NewIncrement = True
     While (NewIncrement = True) do
          Computes "next_point" in transm.plan as (L[i]+U[i])/2
          Computes "transm.rate" based on "next_point" and "start_point"
          If ("transm.rate" upper U[i] inside interval) then // overflow
              NewIncrement = False
          If ("transm.rate" lower L[i] inside interval) then // underflow
              NewIncrement = False
     End
     Defines the smoothing interval as ("start_point", "next_point")
     Associates the "transm.rate" with the smoothing interval
     Define a new "start_point" based on "next_point"
     // Loop to start a new interval…
End

// Part 3: Adjusting the frame_time and frame_size
//          for each video frame arrival time
For each video frame arrival time in (Original_trace_file)
     Defines new (frame_time, frame_size) based on transm.plan
     Write (Smoothed_trace_file)
End
```

Figure 3: Smoothing algorithm pseudocode

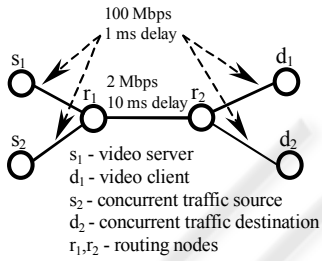

Figure 4: The simulation scenario for experiment 2

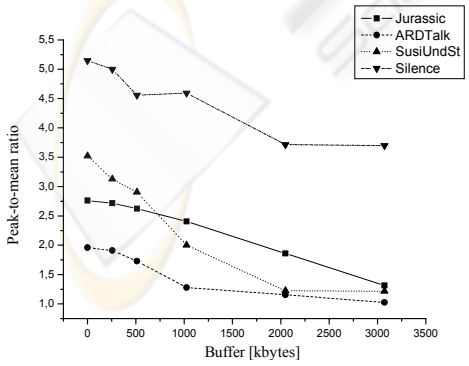

Figure 5: The peak-to-mean ratio.

For the second set of experiments, we are basically interested in the following performance metrics: (1) The use of bandwidth in the link between the routers $r_1 - r_2$ (see figure 4); (2) The client buffer instant occupancy; (3) The end to end delay; (4) If there are lost packets, the effect of this in the video frames.

As proposed in (Ito, 2002), we adopted two performance metrics: the *Frame Error Ratio* (FER), defined as the fraction of frames in error in each video stream, and the network *Useless Output* (UO), defined as the fraction of useless or unusable video data from all video streams in a multiplexed scenario. Frames in error are considered frames with a lost fragment and all its propagated frames (P and/or B), while the unusable video data is the data received by users as part of an unrecoverable frame.

The use of the bandwidth in the link between the routers for the Jurassic Park transmission as a function of time is depicted in figure 6. In figure 6(a) we have the original trace file and in figures 6(b), (c) and (d), we have smoothed files for buffers of 1024 k, 2048 k and 3072 k, respectively. In all plots the solid line refers to the traffic due to video only and the dotted line is the total traffic (video plus FTP). Observe that when FTP is present it takes all the available bandwidth (2 Mbps) and, when not present, the two lines are coincident.

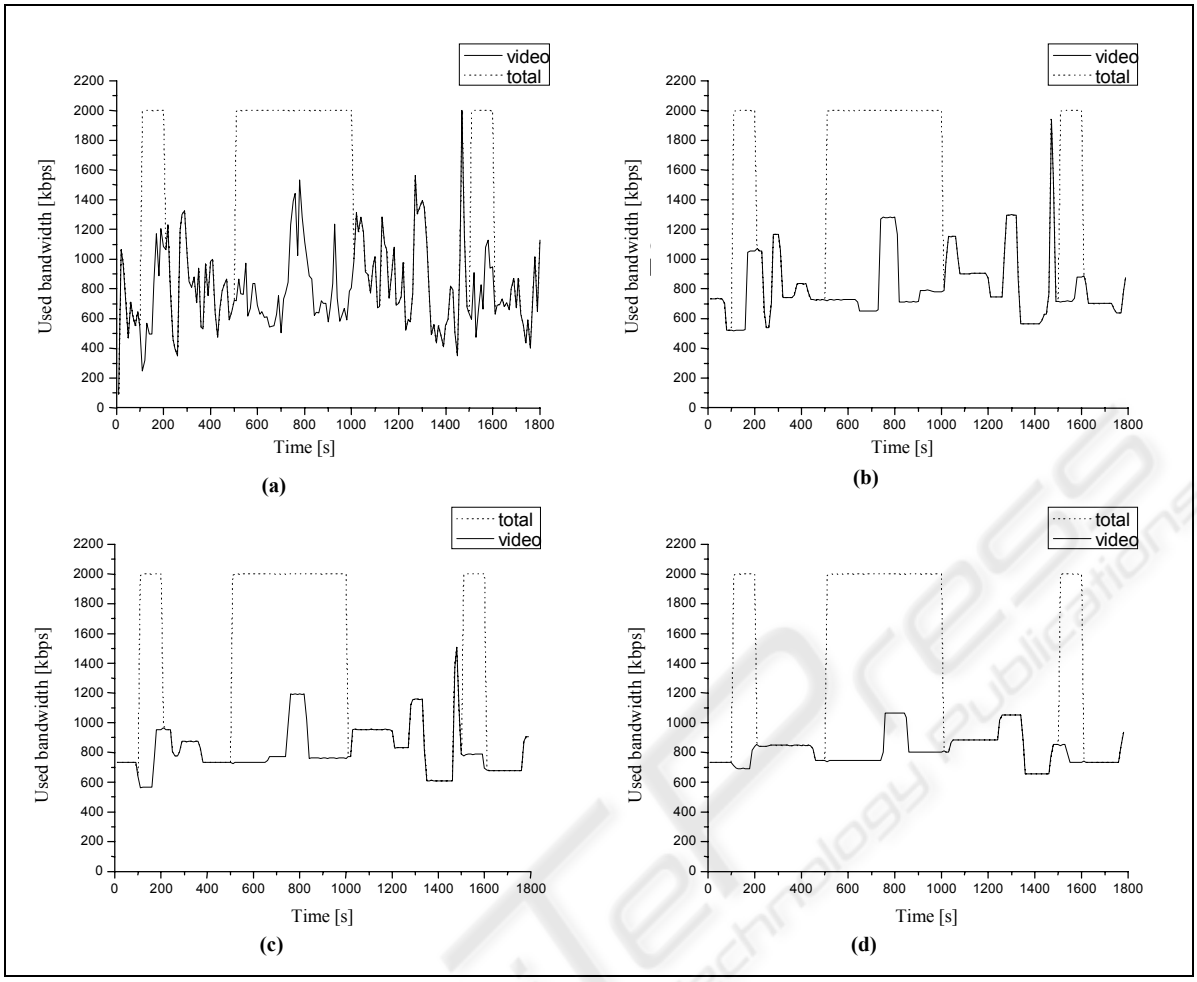

Figure 6: The used bandwidth between routers for the client's buffer size of: (a) no buffer; (b) 1024 kbytes; (c) 2048 kbytes; and (d) 3072 kbytes

The client buffer instant occupancy must show that the buffer never underflows nor overflows. As the use of the buffer depends fundamentally how the codec (present at video client) consumes the video frames, we made the following assumptions: (1) The buffer will increase as the video packets arrive from network; (2) The client will take from the buffer an entire GoP or, equivalently, 12 frames (see figure 1) every 480 ms; (3) The video playback at the client will start after two complete GoP are present in the buffer, this leads an initial playback delay of about 960 ms.

Considering those assumptions, we observed that, in some instants, the nominal buffer limit was reached. This can be easily overridden by increasing in 10% the real buffer limit, or, by specifying a lower value to the smoothing algorithm.

We can observe that in all cases, the end to end delay increment is lower (about 6 %), but the delay jitter has a higher increment (about 16 %). The delay's increment is because smoothing introduces extra packet fragmentation and the delay jitter's increment is a consequence of the intensive use of the client buffer.

The effect of the lost packets of video frames was observed during the original trace file transmission and in the trace smoothed for a 1024 kbytes of buffer. In all other cases, no losses were present. This is mainly because the smoothing has reduced the burstiness of the video traffic.

As explained before, FER is the fraction of frames in error in the video stream. For the original trace file transmission the calculated FER is 1,04 % and for the 1024 kbytes trace file, this metric is 0,03 %. If these values were estimated only by the packet loss ratio they would be 0,53 % and 0,01 %, respectively.

The useless output (UO) represents the fraction of unusable video frames that were delivered to the client. For the original video traffic, 3,83 % of all video data is unusable, while for the 1024 kbytes trace file, we have 0,34% as useless output.

7 CONCLUSION

The first set of experiments is related to the smoothing algorithm itself. By analyzing the smoothed video traces we can conclude that the efficiency of the algorithm depends on the type of the video. The *Susi und Strolch* trace file have the greatest reduction (about 65 %, based on the peak-to-mean ratio metric), the worst performance was observed in the Silence of the Lambs trace file (only 28 % of reduction). We think this happened because in the *Susi und Strolch* the instants of bursts are, predominantly, punctual, while in the Silence of the Lambs the burst is spread in time.

The smoothing algorithm proposed in this paper is based on two other smoothing algorithms: e-PCRTT and MCBA. We have showed that it can bring good results in terms of peak bandwidth reduction, keeping the simplicity of the e-PCRTT computation associated with the ability of extend the transmission rate runs as much as possible, which is a characteristic present in MCBA.

The second set of experiments explores the effect of transmitting such a smoothed video over a simple network configuration where video competes with a concurrent traffic (FTP). One characteristic of the FTP traffic is that it occupies all the available bandwidth when it is present. In this case, the smoothing contributed to optimize the use of the bandwidth by reducing the bursts.

The study of the client buffer occupancy is interesting as it ensures that the buffer will never overflow nor underflow, as this is the assumption of the smoothing algorithm.

The end to end delay in all smoothed trace files has almost the same value, which in turn, is higher than in the original trace file transmission. The delay's variation is more sensitive in the smoothed traces. This was expected by the principle of the smoothing: The new transmission plan imposes more or less delay once it changes the transmission rate (increasing or decreasing it).

Finally, the study of packet loss is very important in a video transmission since a single lost packet can degrade a whole GoP. We measured this effect through the FER and UO metrics, but since we are not dealing with packet discard discipline, the study is merely informative.

Some kind of intelligent packet discard discipline at the routers leading in a better network utilization can also be addressed.

REFERENCES

Bakiras, S., Li, V.O.K., 2002. Maximizing the Number of Users in an Interactive Video-on-Demand System. In *IEEE Transactions on Broadcasting*, Vol. 48, No. 4, 281-292.

Ito, M.R., Bay, Y., 2002. A Packet Discard Scheme for Loss Control in IP Networks with MPEG Video Traffic. In *8th IEEE International Conference on Communication Systems 2002*.

Zhang, Z., Kurose, J., Salehi, J.D., Towsley, D., 1997. Smoothing, Statistical Multiplexing and Call Admission Control for Stored Video. In *IEEE Journal on Selected Areas in Communications*, Vol. 15, 1148-1166.

Salehi, J.D., Zhang, Z. Kurose, J., Towsley, D., 1998. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. In *IEEE/ACM Trans. Networking*, Vol. 6, 397-410.

Wrege, D., Knightly, E., Liebeherr, J., Zhang H., 1996. Deterministic delay bounds for vbr video in packet-switching networks: Fundamental limits and practical tradeoffs. In *IEEE/ACM Trans. Networking*, Vol. 4, 352-362.

Reibman, A.R., Berger, A.W., 1992. Constraints on variable bit-rate video for ATM networks. In *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, 361-372.

Feng, W., Sechrest, S., 1995. Smoothing and buffering for delivery of prerecorded compressed video. In *Proc. of the IS&T/SPIE Symp. on Multimedia Comp. and Networking*, 234-242.

Sechrest, S., Feng, W., 1995. Critical bandwidth allocation for the delivery of compressed video. In *Comput. Communication*, Vol. 18, 709-717.

Jahanian, F., Feng, W., Sechrest, S., 1995. Optimal buffering for the delivery of compressed prerecorded video. In *Proc. of IASTED/ISMM International Conference on Networks*.

Hadar, O., Cohen, R., 2001. PCRTT Enhancement for Off-Line Video Smoothing. In *The Journal of Real Time Imaging*, Vol. 7, No. 3, 301-314.

Fall, K., Varadhan, K. (eds.), 1999. NS – Notes and Documentation, tech. rep., The Vint Project.

Fitzek, F.H.P., Reisslein, M., 2001. MPEG-4 and H.263 Video Traces for Network Performance Evaluation. In *IEEE Network*, Vol. 15, No. 16, 40-54.

Feng, W., Rexford, J., 1997. A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video. In *Proc. IEEE Infocom*, 58-66.