
ROYAL INSTITUTE
OF TECHNOLOGY

# Traffic Control for VBR Video in Packet Switched Networks

GYÖRGY DÁN

Licentiate Thesis
Stockholm, Sweden, 2004

TRITA-IMIT-LCN AVH 04:01
ISSN 1651-4106
ISRN KTH/IMIT/LCN/AVH-04/01—SE

Department of Microelectronics
and Information Technology
KTH, Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungl Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknologie licentiatexamen fredagen den 5 mars 2004 i Sal 5, Forum, KTH, Kista.



Tryck: Universitetsservice US AB

## Abstract

The best effort service offered by the Internet is not satisfactory for the transmission of loss and delay sensitive data, such as real-time voice and video. In order to provide service guarantees at reasonably high load levels additional control functions have to be employed in the network. The introduction of these functions is a critical issue as several aspects have to be taken into account, like scalability, economic feasibility and compliance with the end-to-end argument, one of the basic principles underlying the current Internet architecture. To fulfill these requirements control functions should be put at the network edge, while the core of the network should be kept simple to maintain the flexibility of the network.

The endpoint measurement-based admission control schemes proposed recently follow this principle. Most of these admission control schemes, however, suffer from limited granularity, namely, the QoS guarantees (packet loss, delay and delay jitter) within a service class are the same for all streams. Quality differentiation thus either requires quality specific service classes, which is in contradiction to the end-to-end argument, or the use of additional traffic control functions at the network edge.

Such additional control functions can decrease the packet loss probability given by the admission control to a level required by the application by introducing additional delay. Delay limited shaping can be used to decrease the burstiness of the streams and thus their packet loss probability at the price of increased delay. Forward error correction can be used to recover from losses within the network while incurring increased delay and some overhead.

This thesis gives an evaluation of how the different control functions can work together to improve transmission quality, and thus the perceived visual quality. The effects of the joint use of the control functions shaping and forward error correction are evaluated, and the optimal allocation of the delay is investigated. An exact mathematical model of the packet loss process for multimedia traffic is presented, and its applicability is evaluated.

# Acknowledgments

First I would like to thank my advisor Viktória Fodor for her support, for the fruitful conversations and sometimes hard critics, and for introducing me into the world of scientific research. I would also like to thank my second advisor professor Gunnar Karlsson for his helpfulness and his comments on my work. Furthermore I would like to thank all the people at LCN for the nice atmosphere, which made it possible for me to concentrate my efforts on research.

Additionally I would like to express my gratitude to the organizations and entities that have founded my research, the AWSI project and the Graduate School in Telecommunications.

I would like to express my special thanks to all my friends in Stockholm who gave me support whenever I needed it, and helped me to spend two wonderful years in Sweden. I would also like to thank to my friends back home and abroad who, though being far away, were always there to help me and made my holidays an infinite source of energy. Last but not least, I would like to thank to my family for their understanding, and patience, and their continuous support.

# Contents

# Chapter 1

# Introduction

During its thirty years of history the Internet has had many "killer applications." Though originally the network was only meant for data transfer, it was soon being used as a medium for personal communications, such as sending e-mails. With the appearance of the different multimedia encoding standards the network could even be used for transmission of audiovisual information. Later, as bandwidth and the user base of the Internet grew, the support for machine-to-human and human-to-human communications became more evident. But, while e-mail could offer the reliability of the postal mail service without major architectural implications on the network, the Internet has no mechanisms to support the quality requirements of audiovisual communications that one is used to in the Public Switched Telephone Network (PSTN) and the Integrated Services Digital Network (ISDN). Even though quality of service is assured in many cases in form of a Service Level Agreement (SLA) between the network provider and the user, the guarantees given are not sufficient to enable good quality real-time audiovisual communications.

## 1.1 Background

Internet is now considered as the universal network for data, voice and video communications. It is recognized that the best effort service provided today is not satisfactory for delay and loss sensitive applications such as live voice and video. It is generally accepted that traffic control functions must be employed to guarantee these applications adequate quality at reasonably high network loads. The introduction of new control functions in the Internet is, however, a critical issue. First, a very high number of networking devices have to be updated or replaced. Second, the complexity of the control functions may limit the scalability and the transmission capacity of the network. Third, it

is in contradiction with one of the principal ideas used in the design of the Internet, the end-to-end argument [1]. Fourth, it might require new business models for the operators who need to charge extra for traffic with quality of service.

In the last years a variety of admission control schemes based on per–hop or end–to–end measurements have been published to provide admission control for delay and loss sensitive traffic with very little or no support in the routers. Measurement-based admission control (MBAC) schemes base the acceptance decision on per-hop real–time measurements of the aggregate traffic intensities [2]. Endpoint measurement-based admission control (EMBAC) schemes decrease the required router support further by involving only the end systems in the admission control process [3, 4, 5, 6].The idea behind these schemes is to probe the transmission path from the sender to the receiver to estimate the load level in the network and accept new streams only if the load is acceptable. In the solution proposed in [5] small buffers, packet scale buffering, are used to ensure low packet delays and probe based admission control [7] is used to limit the packet loss probability.

Most of these admission control solutions suffer from limited granularity, namely, the QoS guarantees (packet loss, delay and delay jitter) within a service class are the same for all streams [2, 4]. Service differentiation thus requires quality specific service classes, which is against the end-to-end argument, or additional traffic control functions at the network edge. This latter option is investigated in this thesis.

## 1.2 Network architecture and control functions

We consider the following networking scenario. On-line video streams have to be transmitted through a large packet switched network - specifically, the Internet. We assume that the network has differentiated services (DiffServ) [8] capabilities, and the transmission of video streams is not disturbed by best effort traffic. In particular the network architecture described here corresponds to the controlled-load service class of the DIY service architecture described in [9].

We assume that EMBAC is used for the video streams to provide performance guarantees. While packet loss is limited by the admission control, end to end delay is limited by the small buffers in the network nodes and by the control functions at the network edge.

The full system, shown in Figure 1.1, consists of the following components. The source and the destination of the video stream, the source coder and decoder (in our case following the MPEG standard). The transmission and

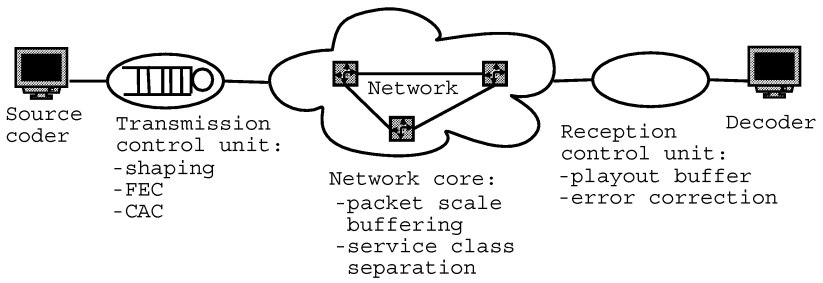


Figure 1.1: The considered network model with source coder and traffic control at the end nodes, and packet scale buffering inside the network.

reception control units, which can contain a source shaper, a playout buffer and error control coding and decoding. The goal of this thesis is to evaluate how the above system components can work together to improve transmission quality, and thus the perceived visual quality. In the following subsections we give a short overview of the system elements of interest.

## Packet scale buffering

In the considered network architecture, the buffers in the nodes provide buffering only for simultaneously arriving packets. In this case the buffer size is roughly equal to the ratio of the link capacity and the peak rate of the streams.

In this case of packet scale buffering the delay and the delay jitter are kept low, the loss process is less correlated, and as shown in [10] the queueing performance of long range dependent traffic, like VBR video, is dominated by the short range behavior.

As shown in [11] for short network paths the use of packet scale buffering combined with shaping results in higher packet loss probabilities than using burst scale buffering combined with per stream delay jitter compensation in the network. However, packet scale buffering outperforms burst scale buffering for long network paths. In[12]we compare the performance of the two solutions for the case of low end-to-end delays suitable for real-time communications. It is worth pointing out that even though packet losses can be more frequent when small buffers are used, the performance evaluation and thus the network dimensioning becomes easier to carry out.

## Delay limited source shaping

Shapers used at the sources decrease the frame to frame fluctuation of coded video streams. In the case of stored video, sufficient information is available a priori to achieve optimal shaping. But in the case of live video streams, like streaming and real-time communications, there is a trade-off between the delay introduced at the shaper and the effectiveness and complexity of shaping.

A solution for shaping with limited packet loss is proposed in [13]. The shaping is based on the BIND characterization of the stream to be shaped. The BIND parameters are continuously updated as the statistical properties of the stream change, which seems to be a rather complex process considering the real-time operation. In [14] the authors study statistically identical, peak rate controlled and leaky bucket shaped sources feeding a buffered multiplexer. The goal is to find the shaper rate that minimizes network resources like buffer space and bandwidth, while keeping the delay limited. The solution, however, can not handle multiplexed sources with different characteristics. Algorithms for lossless shaping of individual streams are presented in [15] and [16]. In these works the statistical quantities of the shaped streams are analyzed, but network scenarios are not considered. In [16] shaping with delays in the range of 2 to 30 seconds is proposed, an adequate solution for broadcasting applications. The algorithm shown in [15] works with shaper delays less than a second. To avoid the fluctuation of the shaper rate, the algorithm uses past frame sizes to predict future traffic intensity, and needs to know the length of a group of pictures (GOP) in advance.

In [17] two simple, low delay shaping algorithms are presented, and their efficiency in decreasing the packet loss probability is evaluated via mathematical analysis and simulations. The shapers used are based on a single buffer leaky bucket, as it is proved to be optimal for networks with small buffers [18]. Frames leaving the encoder are stored in the shaper buffer and are transmitted with a given transmission rate, which is dynamically adjusted to provide lossless shaping with bounded delay.

## Forward error correction

Forward error correction (FEC) has been proposed to recover from information losses in real-time applications, where the latency introduced by retransmission schemes is not acceptable. FEC increases the redundancy of the transmitted stream and recovers losses based on the redundant information. There are two main directions of FEC design to recover from packet losses. One solution, proposed by the IETF and implemented in Internet audio tools is to add a redundant copy of the original packet to one of the subsequent packets

[19]. The other set of solutions, which can be used as media independent FEC for VBR video, use block coding schemes based on algebraic coding, e.g. Reed-Solomon coding [20].

In the case of algebraic coding, a block of packets is collected and the coding is applied for each symbol position. This coding scheme introduces overhead and delay that depends on its parameters. The efficiency of the encoding depends on the average packet loss and the packet loss process, e.g. the burstiness of the loss process. Thus in order to be able to analyze the effects of the use of FEC it is important to have a suitable mathematical model of the loss process that can capture the behavior of different types of links, e.g. access links and the backbone.

The efficiency of FEC for video sources is analyzed using simulations in [21]. The author concludes that if only a part of the sources uses FEC, their loss probability can be decreased considerably, however the overall network performance can not be increased by using FEC for all streams. In [20] Kawahara et al. come to a similar conclusion using a discrete time analytical model of a multiplexer. They also conclude that the burstiness of a traffic source influences the gain it can achieve by using FEC. In [22] Cidon et al. analyse the loss process of bursty sources feeding a multiplexer with finite buffer capacity. They consider continuous and discrete time models, and conclude that the loss process in a multiplexer fed by bursty sources is correlated, which can influence FEC performance considerably. In [23] Altman et al. use the M/M/1 finite capacity queue to evaluate the gain achievable by FEC. They show that both under light and heavy traffic conditions, adding redundancy decreases the uncorrected packet loss probability. In [24] the optimal combination of FEC redundancy and source rate to maximize the perceived visual quality is evaluated using the Gilbert model as a loss model. The paper presents a dynamic rate allocation algorithm, which minimizes the perceptual quality degradation given a set of video and network parameters.

# Chapter 2

# Summary of original work

## 2.1 Paper A: On the Efficiency of Shaping Live Video Streams

*Gy. Dán, V. Fodor, "On the Efficiency of Shaping Live Video Streams", In Proceedings of SPECTS'02, July 2002, San Diego, CA, pp. 49-56.*

**Summary:** An analytical model has been developed and thorough simulations have been performed to evaluate the efficiency of delay limited source shaping for the transmission of live video streams. The goal of the paper is to show that even using computationally simple shaping algorithms significant gain can be reached. We consider MPEG coded video streams multiplexed at a single multiplexer.

*Delay limited shaping:* We propose two simple shaping algorithms for online shaping of MPEG video with a strict delay bound, one based on the residual transmission delays (RTD), and another, simpler one, which is based on the buffer content only (BC). The efficiency of the two solutions is evaluated by comparing the coefficient of variation (CoV) of the shaped streams. The calculations show that shaping with a delay bound as small as 40 ms results in a significant reduction of the CoV; the marginal gain decreases when increasing the shaping delay further. This gives prospect to the efficient use of traffic shaping in low delay applications. Though the RTD algorithm performs slightly better, we use the BC algorithm throughout our work to show that one can achieve considerable improvements even by using the simplest algorithm.

*Analytical model:* The analytical model presented in the paper is based on the theory of large deviations and makes it possible to calculate the average

loss probability as well as the relative loss probabilities of shaped and unshaped streams. Calculations are made using data derived from actual traces of MPEG videos, and validated using extensive simulations.

*Average loss probability:* Both the analytical model and simulations show that shaping half of the sources decreases the loss probability of all streams by roughly one order of magnitude. Furthermore streams applying shaping experience a loss probability 20 to 35 percent lower than the average at a load level where the average loss rate is $10^{-5}$. The achieved gain might make it desirable for individual users to apply shaping. The calculations show that for the considered traces shaping with a delay limit of 120 ms gives only minor improvement compared to shaping with a delay limit of 40 ms.

*Packet loss probabilities in I, P and B frames:* It is known that losses in the different types of frames in an MPEG video do not have the same effect on the perceived quality of the reconstructed video. Therefore it is interesting to evaluate the effects of shaping on the losses in the different frame types. The simulations have shown that if shaping is not used, I frames, which have the greatest influence on the percepted quality, have a loss probability which is up to 100 percent higher than the average loss probability experienced by the stream. However by using shaping the packet loss probability among the different frame types can be made equal, which may further improve the performance of the transmission.

**Contribution:** The original idea came from the second author of the paper. The author of this thesis performed the mathematical analysis based on results in the literature, and carried out the simulations to validate the mathematical model. The article was written in cooperation with the second author of the paper.

## 2.2 Paper B: Comparison of Shaping and Buffering for Video Transmission

*Gy. Dán, V. Fodor, "Comparison of Shaping and Buffering for Video Transmission", NTS 16, August 2002, Helsinki, pp. 78-87.*

**Summary:** In this paper we evaluate the efficiency and feasibility of packet scale buffering combined with delay limited shaping against delay limited buffering for the transmission of delay and loss sensitive video. We consider a transmission path with 10 hops. In the case of packet scale buffering most of the delay is spent on shaping, while in the case of delay limited buffer-

ing the available delay is split among the nodes on the transmission path for buffering.

*Average packet loss:* The simulations show that for delays in the range of 20 ms to 40 ms buffering outperforms shaping, even though the difference is less than one order of magnitude. Considering an acceptable packet loss probability of $10^{-5}$ the difference of the maximum network load with shaping and buffering is between 3 to 8 percent depending on the delay limit.

*Packet loss probabilities in I,P and B frames:* The simulations show that for a delay of 40 ms in case of delay limited buffering the I frames experience a relative loss probability up to 100 percent higher than in the case of shaping combined with packet scale buffering. The more favorable distribution of the losses among the frames improves the quality of packet scale buffering.

*Packet loss distribution:* Besides the average loss probability, it is important to investigate the probability of consecutive losses and the distribution of losses within a block of packets. According to the simulations in the case of packet scale buffering the consecutive loss probability is very small up to a load of 0.8, which corresponds to a packet loss probability of $10^{-5}$. In the case of delay limited buffering the probability of consecutive packet losses is high both in the low load and high load regions. The expected number of consecutive packet losses has a minimum at a load of 0.85, and it is always higher than in the case of packet scale buffering combined with shaping.

The burstiness of the packet loss process influences the applicability of control functions such as forward error correction. For this reason we have investigated the probability of single packet losses in a block of packets with the following results. In the case of packet scale buffering the probability of having a single packet loss in a block of packets decreases as the average load increases, with an initial value close to one when the average load is around 0.7. In the case of delay limited buffering the probability of multiple losses in a block is orders of magnitude higher than in the case of packet scale buffering. The low probability of multiple losses within a block of packets can make the use of FEC beneficial in the case of packet scale buffering combined with delay limited shaping.

**Contribution:** The original idea came from the second author of the paper. The author of this thesis has performed the simulations and analyzed the data. The article was written in cooperation with the second author of the paper.

## 2.3 Paper C: Quality Differentiation with Source Shaping and Forward Error Correction

*Gy. Dán, V. Fodor, "Quality Differentiation with Source Shaping and Forward Error Correction", MIPS 2003, November 2003, Naples, pp. 222-233.*

**Summary:** In this paper the joint use of FEC and delay limited shaping is investigated with respect to delay and loss sensitive video transmission. The results presented in this paper are based on simulations performed with traces of MPEG videos on a single multiplexer.

*FEC and shaping for differentiation:* The simulation results show that FEC can reduce the loss probability by one to two orders of magnitude depending on the amount of redundancy (overhead) used if only a small portion of the sources applies it. Shaping further decreases the loss probability by 25 to 50 percent. The reason for this is twofold: shaped streams experience a lower loss probability, and the loss process of shaped streams is less correlated which makes FEC more effective.

*Optimal allocation of delay between FEC and shaping:* Both FEC and delay limited shaping introduce some delay and improve the performance in terms of packet loss as a function of the delay but with a decreasing marginal gain. Finding the right allocation of delay between the two control functions is an optimization problem. The results indicate that even though the lowest average loss probability can be achieved by using FEC only, if we consider the impact of losses in the different frame types on the perceived quality, it is worthwhile to spend a fraction of the available delay on shaping.

*FEC redundancy and shaping delay:* Given a fixed bandwidth, the use of FEC decreases the effective load due to the overhead imposed. Since shaping and FEC have similar effects it is worthwhile to investigate if shaping can compensate for decreased FEC redundancy. The performed simulations showed that by introducing an additional delay of 60 ms the same loss probability can be achieved at a 33 percent lower redundancy level.

*Sensitivity analysis:* A weakness of quality differentiation using FEC is its sensitivity to the background traffic characteristics. The considered simulation scenario showed that even though the loss probability of all the streams depends on the burstiness of the background traffic, the difference between the loss experienced by streams with and without FEC is almost constant. By combining FEC with source shaping the robustness of the quality differentiation can be improved further.

**Contribution:** The original idea came from the second author of the paper. The author of this thesis has carried out the simulations and evaluated the results. The paper was written in cooperation with the second author of the paper.

## 2.4 Paper D: Analysis of the Packet Loss Process for Multimedia Traffic

*Gy. Dán, V. Fodor, G. Karlsson, "Analysis of the Packet Loss Process for Multimedia Traffic", Technical Report, TRITA-IMIT-LCN R 04:01, ISSN 1651-7717, ISRN KTH/IMIT/LCN/R 04/01–SE, Submitted to Networking 2004.*

**Summary:** In the case of multimedia traffic, like VBR video, the average loss probability is not sufficient to investigate the effects of loss on perceived visual quality. In this paper a mathematical model is presented to calculate the probability of $j$ losses in a block of $n$ packets for a bursty stream multiplexed with background traffic with constant packet size.

*Loss model:* The multimedia traffic is modeled by an L-state Markov-modulated Poisson process (MMPP), and the background traffic is represented by a Poisson process. Formulae to calculate the loss probability in a block of packets in the resulting MMPP+M/D/1/K queue are derived and a numerical method to calculate the probabilities is given.

*Model validation:* The results given by the model are validated against simulations of the same system, and the comparison shows that our model is accurate. The results of the model are compared with the results given by the model of the MMPP+M/M/1/K queue.

*FEC performance:* The results given by the model are compared to the results of simulations performed with real MPEG traces, and the capability of the widely used Gilbert model to capture the loss process of the considered MMPP+M/D/1/K queue is investigated. The derived formulae make it possible to parametrize the Gilbert model based on the block loss probabilities, and to compare its performance to that of our model.

**Contribution:** The idea of developing a mathematical model came from the second author of the paper. The author of this thesis has developed the mathematical model based on results found in the literature, implemented, and carried out the simulations, and analyzed the resulting data. The formulae to calculate the performance measures were derived by the author. The article was written under the supervision of the two co-authors.

## 2.5 Other papers

Gy. Dán, V. Fodor, "The Effectiveness of Traffic Shaping in Networks with Small Buffers", RVK'02, June 2002, Stockholm (Not included in the thesis).

# Chapter 3

# Conclusions and future work

This thesis presents an evaluation of how different traffic control functions can improve the quality of real-time audiovisual communication. We consider a network scenario where small buffers are used within the network to keep delay low, and control functions are placed at the edge nodes to control the loss probability. Loss guarantees are given by endpoint measurement-based admission control schemes, providing the same loss and delay guarantees for all accepted streams. Additional traffic control functions are used to decrease the packet loss probability at the price of increased delay: delay limited shaping can be used to decrease the burstiness of a source and thereby increase the network's multiplexing performance; forward error correction can be used to recover from packet losses.

The network model employing small buffers and delay limited shaping as a traffic control function is compared to a network employing delay limited buffering combined with per stream delay jitter control. Simulations performed for a multiplexer of a multihop network show that the loss probability in the case of small buffers and shaping is about one order of magnitude higher than in the case of buffering. The analysis of the packet loss process shows that losses are less correlated under all traffic conditions in the case of small buffers, which makes the use of forward error correction possible. Even though the average packet loss probability can be higher in the case of packet scale buffering, its scalability and the favorable loss process make it a viable solution for large-scale deployment and we focus our work on networks employing packet scale buffering.

Both the mathematical model, and the simulations presented show that by using traffic shaping the loss probability of the individual streams decreases significantly, by up to thirty percent. The achieved gain is directly proportional to the introduced delay with a decreasing marginal gain. The simulations show that losses are more evenly distributed between the different frame

types of MPEG coded video as an effect of using shaping. By taking the influence of the packet losses on the perceived quality into account the use of shaping can bring considerable gains for the individual users. The network operators can benefit from the gradual introduction of shaping in the end nodes as well. The simulations show that if half of the users employ traffic shaping, the average packet loss at a given network utilization can decrease by up to one order of magnitude. In other words, the same average loss probability can be achieved at up to ten percent higher network utilization.

Simulations indicate that shaping can improve the efficiency of FEC considerably. The presented results show that by combining FEC and shaping the packet loss can be decreased by up to half an order of magnitude. Considering limited end-to-end delay we show under various scenarios that there exists an optimal combination of shaping and FEC which minimizes the loss of information. The simulations show that combining the two control functions increases the robustness of quality differentiation, which is an important issue when one is about to give guarantees on quality of service. The results also show that by introducing additional delay for shaping the average network load can be decreased, as less FEC overhead can be sufficient in order to reach a certain loss probability.

Since the performance of FEC and the error resilient features of MPEG encoded video are sensitive to the distribution of packet losses, an accurate model of the loss process is important to evaluate the possible effects of the control functions on the perceived visual quality. The validation of the presented mathematical model shows that the widely used Gilbert model is not always good enough to predict the performance of traffic control functions. Comparing the Gilbert model to our MMPP+M/D/1/K model we show that the Gilbert model is quite accurate at moderate load levels when the level of statistical multiplexing is high, but it fails to model the loss process accurately if losses are highly correlated, for example on an access link. The validation of our model via simulations shows that the model presented in this thesis captures the loss process accurately, and thus can be used to evaluate the performance of the control functions under diverse traffic conditions. The results indicate that FEC is a viable solution to decrease the loss probability experienced by individual streams. However, it is clear that the achievable gain depends on the average network load and the level of statistical multiplexing.

The results presented in this thesis give an insight into the possible effects of shaping and FEC when applied to real-time VBR video. They enable the design of an algorithm to tune the parameters of these control functions to achieve the best possible performance, and show that EMBAC schemes together with traffic control functions at the network edge can provide diverse quality of service guarantees with minimal changes in the existing network

infrastructure. There are however many open questions, which can be subject of further research.

- The optimal selection of FEC and shaping has been investigated, but the effects of adjusting the source rate has not been considered.
- Further properties of the loss process, like the loss correlation of a multiplexer fed by multimedia traffic, can be evaluated using the mathematical model with application to various channel coding solutions other than the Reed-Solomon code.
- The increasing importance of wireless communications in IP traffic calls for the analysis of heterogeneous network scenarios, where apart from losses due to congestion, bit errors and random losses can occur due to fading, or other degradation in channel quality.

# Bibliography


[1] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, pp. 277–288, November 1984.

[2] L. Breslau, S. Jamin, and S. Shenker, "Comments on the performance of measurement-based admission control algorithms," in *Proc. of IEEE INFOCOM*, pp. 1233–1242, March 2000.

[3] G. Bianchi, A. Capone, and C. Pertioli, "Throughput analysis of end–to–end measurement-based admission control in IP," in *Proc. of IEEE INFOCOM*, pp. 1461–1470, March 2000.

[4] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "End-point admission control: Architectural issues and performance," in *Proc. of ACM SIGCOMM*, pp. 57–69, August 2000.

[5] V. Fodor (née Elek), G. Karlsson, and R. Rönngren, "Admission control based on end-to-end measurements," in *Proc. of IEEE INFOCOM*, pp. 623–630, March 2000.

[6] R. B. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," in *Proc. of the 16th International Teletraffic Congress*, pp. 941–952, June 1999.

[7] I. Más Ivars and G. Karlsson, "PBAC: Probe based admission control," in *Proc. of QofIS, COST 263*, pp. 97–109, September 2001.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.

[9] G. Karlsson and F. Orava, "The DIY approach to QoS," in *Proc. of IWQoS*, pp. 6–8, May 1999.

[10] B. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: Myths and realities," in *Proc. of ACM SIGCOMM*, pp. 3–14, 1996.

[11] T. Wu and E. Knightly, "Buffering vs. smoothing for end-to-end QoS: Fundamental issues and comparison," in *Proc. of IEEE Performance'99*, August 1999.

[12] G. Dán and V. Fodor, "Comparison of shaping and buffering for video transmission," in *Proc. of NTS 16*, August 2002.

[13] H. Zhang and E. Knightly, "RED-VBR: A renegotiation-based approach to support delay-sensitive VBR video," *ACM Multimedia Systems Journal*, vol. 5, pp. 164–176, May 1997.

[14] N. Modani, P. Dube, and A. Kumar, "Measurement based optimal source shaping with a shaping+multiplexing delay constraint," in *Proc. of IEEE INFOCOM*, pp. 1807–1816, March 2000.

[15] S. S. Lam, S. Chow, and D. K. Y. Yau, "An algorithm for lossless smoothing of MPEG video," in *Proc. of ACM SIGCOMM*, pp. 281–293, 1994.

[16] J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video," in *Proc. of NOSSDAV*, pp. 249–257, May 1997.

[17] G. Dán and V. Fodor, "On the efficiency of shaping live video streams," in *Proc. of SPECTS'02*, pp. 49–56, July 2002.

[18] M. Reisslein, K. Ross, and S. Rajagopal, "Guaranteeing statistical QoS to regulated traffic: The multiple node case," in *Proc. of IEEE Decision & Control*, pp. 531–538, 1998.

[19] P. Dube and E. Altman, "Utility analysis of simple FEC schemes for VoIP," in *Proc. of Networking 2002*, May 2002.

[20] K. Kawahara, K. Kumazoe, T. Takine, and Y. Oie, "Forward error correction in ATM networks: An analysis of cell loss distribution in a block," in *Proc. of IEEE INFOCOM*, pp. 1150–1159, June 1994.

[21] E. Biersack, "Performance evaluation of forward error correction in ATM networks," in *Proc. of ACM SIGCOMM*, pp. 248–257, August 1992.

[22] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high speed networks," *IEEE Transactions on Information Theory*, vol. IT-39, pp. 98–108, January 1993.

[23] E. Altman and A. Jean-Marie, "Loss probabilities for messages with redundant packets feeding a finite buffer," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 779–787, 1998.

[24] P. Frossard and O. Verscheure, "Joint source/FEC rate selection for quality-optimal MPEG-2 video delivery," *IEEE Transactions on Image Processing*, vol. 10, no. 12, pp. 1815–1825, 2001.

# On the Efficiency of Shaping Live Video Streams

Gy. Dán and V. Fodor
*In Proceedings of the 2002 International Symposium on Performance Evaluation of Computer and Telecommunications Systems (SPECTS '02),*
July 2002, San Diego, California

# On the Efficiency of Shaping Live Video Streams

György Dán and Viktoria Fodor
KTH, Royal Institute of Technology
Electrum 229, 16440 Kista, Sweden
{gyuri,viktoria}@it.kth.se

## Abstract

In this work the efficiency of shaping live video streams is considered. We propose low complexity shaping algorithms adequate for real-time operation and supporting applications with a wide range of delay tolerance. The effect of shaping is investigated considering video streams multiplexed at an output link with a small buffer to absorb packet scale congestion. The advantage of using small buffers when transmitting video streams is the limited delay and delay variation. Consequently, we concentrate on the loss characteristics to evaluate the performance of the proposed solutions. We present mathematical analysis based on fluid flow modeling and the theory of large deviations and confirm the results with simulation.

**Keywords:** quality of service, live video transmission, source shaping, packet scale buffering, large deviation theory

## 1 INTRODUCTION

The transmission of live video traffic over the Internet is a fundamental problem of network design, since many video applications require limited end to end packet loss, delay and delay variation. It is generally accepted that traffic control functions must be employed to guarantee these service requirements at a reasonably high network load. The introduction of new control functions in the Internet, however, is a critical issue. First, a very high number of networking devices has to be updated or replaced, second, the complexity of the control functions may limit the span and the transmission capacity of the network.

Recently research efforts address this question by proposing probe based endpoint admission control (PBAC) solutions [6, 3] to provide quality of service guarantees in a way that the functionalities of the routers are kept simple and traffic control functions are placed into the hosts – or edge gateways – only. In the PBAC schemes a host, before transmitting traffic with QoS requirements, probes the network's transmission capability by sending a sequence of probe packets and decides about the transmission based on the statistical quantities of the probing process.

While the admission control ensures that the load of the network stays reasonably bounded, additional control functions can be applied at the hosts to increase the acceptable load, like traffic shaping to decrease the burstiness of the traffic streams and thus decrease the packet loss at the multiplexing nodes and forward-error correction to recover from packet losses.

In this work we investigate how source shaping can increase the efficiency of live video transmission considering MPEG video streams multiplexed at an output link with small buffer. The advantage of using small buffers is that the delay and delay variation is strictly limited by the buffer size an thus only the packet loss has to be controlled. Source shaping provides the following favorable properties: i) it does not require global decision or any modification in the network, ii) can improve the service quality of streams with different QoS requirements and traffic characteristics and iii) can be introduced in the network gradually.

We propose two solutions to shape video streams without packet loss and with given delay bound. The first scheme follows the ideas presented in [10], and determines the shaper rate considering the delay bounds of all the frames waiting in the shaper buffer. The second scheme considers only the delay bound of the last frame in the buffer, thus provides a solution with very low computational complexity.

We evaluate the performance of the proposed schemes with simulation and with analytical methods based on fluid flow approximation applying the theory of large deviations [1]. As the delay and delay variation is limited by the buffer size, the analysis focuses on the packet loss characteristics, as average loss, the loss of shaped and unshaped streams and the distribution of packet losses among the frame types of the MPEG stream.

The paper is organized as follows. In the next section we discuss related works and results. Section 3 describes the system model with the sources, the shapers and the multiplexer. Section 4 explains the two shaper algorithms we propose, and Section 5 presents the analytical model to evaluate the efficiency of source shaping. In Section 6 we present and discuss numerical results and in Section 7 we conclude our work.

## 2 PREVIOUS WORK

In this section we survey previous research results that inspired our work on shaping and transmission of on-line video

traffic over the Internet.

Traditional solutions to provide QoS guarantees in packet switched networks are based on per flow reservation messages and capacity reservations (e.g., RSVP), and as a consequence, suffer from scalability limitations [17]. To circumvent these scalability problems, several recent works have proposed some form of probe based endpoint admission control (PBAC). In these solutions the hosts (endpoints) send a sequence of probe packets before user data transmission, to detect the level of congestion in the network. The level of congestion and thus the possibility of user data transmission with the required QoS parameters is determined from the statistical quantities of the probe transmission process, like probe loss probability [6], delay and delay variation [2] or packet marking probability [5]. An excellent evaluation of the various designs can be found in [3]. The PBAC schemes provide QoS guarantees without the need of control functions inside the network, support QoS requirements depending on the users needs, and do not require the complex description of the traffic streams. Our work follows the basic idea of these solutions by investigating how additional control functions at the host can increase network efficiency.

The transmission of video streams requires limited packet loss, end to end delay and delay variation. Obviously, these values depend on the size of the buffers at the routers. Depending on the size of the buffer one can differentiate between packet scale buffering and burst scale buffering. In the first case only a small buffer is provided to absorb packets arriving simultaneously, thus the packet loss probability might be high while the delay is strictly limited. In the second case the buffer provides enough space to absorb larger bursts and consequently, limits the loss probability while the control of delay and delay variation becomes a complex issue [14]. The use of packet scale buffering for transmitting delay and loss sensitive data like coded video streams have been proposed in [6, 15, 16], showing that low packet loss probability and high network utilization can be achieved if the peak rate of the streams is low compared to the link capacities. In [15, 16] packet scale buffering is proposed together with source shaping. It is proved that a single buffer leaky bucket is an optimal shaper in this scenario.

The shaping of stored variable bit rate video streams is widely analyzed in the literature. Recent solutions are based on network calculus e.g., [4, 11] or the bounding interval dependent (BIND) characterization of the streams [7, 8]. Only a few works address the shaping of live video streams. The main questions to face in this case are the tradeoff between the delay introduced at the shaper and the available information on the traffic to be transmitted and the effectiveness and the complexity of traffic prediction. A solution for shaping with limited packet loss is proposed in [9]. The shaping is based on the BIND characterization. The BIND parameters are continuously updated as the statistical properties of the stream change, which seems to be a rather complex process considering the real-time operation. In [12] the authors study statistically identical, peak rate controlled and leaky bucket shaped sources feeding a buffered multiplexer. The goal is to find the shaper rate that minimizes network resources like buffer and bandwidth, while keeping the delay limited. The solution, however, can not handle multiplexed sources with different characteristics. Algorithms for lossless shaping of individual streams are presented in [10] and [13]. In these works the statistical quantities of the shaped streams are analyzed, but network scenarios are not considered. In [13] shaping with delays in the range of 2-30 seconds is proposed, an adequate solution for broadcasting applications. The algorithm shown in [10] works with shaper delays less than a second. To avoid the fluctuation of the shaper rate, the algorithm uses past frame sizes to predict future traffic intensity, and needs to know the length of a GOP in advance.

Our contribution to this line of works is the definition of shaping algorithms with very low complexity that support the transmission of live video streams with a wide range of acceptable shaper delay and the analysis of the effect of shaping when the video streams are multiplexed at an output link with small buffer. To the best of our knowledge, such results have not yet been presented in the literature.

# 3 MODEL DESCRIPTION

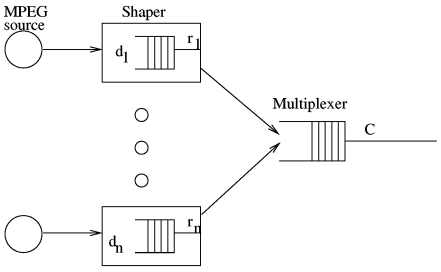


**Figure 1:** The considered system with MPEG sources, source shapers and a multiplexer.

The system model considered in this paper is shown in figure 1. The system includes traffic sources, source shapers and a multiplexing node with a single output link.

The sources generate MPEG coded streams, the most commonly used encoding scheme for the storage and transmission of video information. MPEG is a family of standards used for coding visual information in a digital compressed format. It has been designed to support a broad range of transmission rates and hence a broad range of visual quality. In an MPEG stream information is stored as a sequence of frames, corresponding to a sequence of pictures in a video, generated with fix time intervals. Compression is achieved by eliminating the spatial and temporal redundancy of the information in the

frames. Spatial redundancy is decreased by intraframe coding of the individual frames, while temporal redundancy is reduced by interframe coding between subsequent frames. Thus the sequence of frames consists of intraframe coded frames (I frames), and interframe coded predicted (P frames) and bidirectionally predicted frames (B frames). The subsequent frames between two consecutive I frames form a group of picture (GOP). The GOP structure of the streams can be different, depending on the required quality. A typical example for the sequence of frames is IBBPBBPBBPBB. As a consequence of the coding scheme, information loss in the three frame types has different effect on the percepted visual quality. The loss of data in an I frame propagates forward through the next GOP and backward to the last P frame (affecting up to 14 frames if the number of frames in an open GOP is 12). Meanwhile, the loss of data in a B frame only affects that particular frame.

The intra- and iterframe coding results in the fluctuation of the frame sizes on two timescales. The intraframe coding compresses complex scenes with less efficiency, and consequently, the frame sizes change on the long term at the scene changes. The interframe coding leads to short term frame size fluctuation, since I frames are usually significantly larger than P frames, and P frames are larger than B frames. The scale of the fluctuation is about a factor of 3 on the long, and a factor of 10 on the short term.

Shapers are used at the sources to decrease the frame to frame fluctuation of the coded video stream. The shaper we use in this work is a single buffer leaky bucket, as it is proved to be optimal for networks with small buffers [15]. Frames leaving the encoder are stored in the shaper buffer and are transmitted with a given transmission rate. The shaper is designed to introduce limited delay and provide lossless transmission, that is, no data can be lost due to buffer overflow or delay limit violation. To achieve this, the shaper transmission rate has to be adjusted depending on the size of the arriving frames and the buffer size has to be large enough to store the frames waiting for transmission. If the frames arrive at a regular basis, the maximum number of frames in the buffer can be bounded by the ratio of the delay limit to the frame interarrival time.

The shaped video streams are multiplexed at a network node with a single output link. Since the arrival rate of the streams can temporarily exceed the capacity of the outgoing link, the node is equipped with a buffer to store arriving data. In this work we consider packet scale buffering, the size of the buffer is in the order of the ratio of the output link transmission capacity to the peak rate of the video streams.

# 4 SOURCE SHAPING ALGORITHMS FOR LIVE VIDEO STREAMS

In this section two algorithms are proposed to control the rate $r$ of the source shaper when transmitting live MPEG video streams. Both aim to minimize the maximum and the variance of the transmission rate, and fulfill the following requirements:

i) The transmission delay in the shaper does not exceed the predefined maximum shaper delay and the shaping is lossless.

ii) The algorithms are simple in terms of the complexity of the shaper rate calculation and the amount of information considered, a requirement to assist real-time operation.

iii) The algorithms provide efficient solution for shaping with a large range of shaper delays.

iv) The algorithms do not require any apriori information on the encoding scheme of the video, i.e., the number and sequence of I, P and B frames in a GOP.

Both of the algorithms assume that the shaper can detect the type of the arriving frame. In general, the following rules apply to select the shaper rate $r$:

1. The shaper rate can be changed at any time $t$ when a new frame is generated and placed into the shaper buffer.

2. The shaper rate is increased if the new frame can not be transmitted within the delay limit $d$.

3. The shaper rate is decreased if the new frame is of type I, and all the traffic in the shaper buffer can still be transmitted within the delay limit. This rule is based on the assumption that small P or B frames do not indicate intensity change in the video stream. To decrease fluctuation, the new shaper rate is calculated as the average of the current rate and the minimum rate allowed by the delay limit.

4. P and B frames entering the shaper when the buffer is empty are transmitted with a rate such that the frame leaves the buffer before the new frame arrives, i.e., in one frame time, in order to prevent the shaper from keeping data before larger I and P frames arrive.

The two proposed algorithms described in the following differ in applying rule 3. The first algorithm is optimal in the sense, that the residual acceptable delay is considered for all the frames stored in the shaper buffer to determine the minimum shaper rate. The second, simplified algorithm does not follow the delays of the individual frames in the buffer, and calculates the shaper rate based on the buffer content only.

### Shaper rate control based on residual transmission delays

The algorithm based on residual transmission delays (RTD) works as follows. For every frame entering the shaper, the size of the frame $f_i$ and its latest departure time $t_i = t + d$ is recorded. The minimum shaper rate $r_{min}(t)$, allowed by the delay limit is calculated as

$$r_{min}(t) = \max_{n<N} \frac{f_0(t) + \sum_{j=1}^{n-1} f_{i-N+j+1}}{t_{i-N+j+1} - t}, \qquad (1)$$

where $N$ is the number of frames in the shaper at time $t$ and $f_0(t)$ is the residual size of the first frame in the buffer at time $t$. The residual size is less than the original frame size if the transmission of the frame has already started.

The complexity of the shaper rate calculation is $O(N^2)$ additions and $O(N)$ divisions, where the value of $N$ is bounded by $d/T_{frame}$. In addition to the actual shaper rate, the shaper has to remember the size and the arrival time of the frames waiting for transmission. A system clock has to be maintained and read at each frame arrival.

**Shaper rate control based on the buffer content**

This solution does not record the residual acceptable transmission delay for the frames waiting in the shaper buffer, the shaper rate calculation is based on the buffer content (BC).

When frame $i$ arrives to the shaper, its size is added to the amount of data in the shaper $b(t) = b'(t) + f_i(t)$, denoting the buffer occupancy before the frame arrival as $b'(t)$. The minimum shaper rate is calculated considering the buffer occupancy at the time of the new frame arrival:

$$r_{min}(t) = \frac{b(t)}{d}, \tag{2}$$

To avoid delay bound violation for frames stored in the buffer, the shaper rate can be decreased only if the buffer is empty before the new frame arrival (i.e., b'(t)=0), a significant constraint on rule 3 above.

As a consequence, this simplified BC algorithm follows the decreasing intensity of the stream with some delay compared to the RTD solution. The complexity of the BC algorithm, however, is very low (one addition and one division at each frame arrival), there is no need for system clock information and only the number of bytes waiting in the shaper buffer has to be stored.

# 5 ANALYTICAL MODEL

In this section we present an analytical method to calculate the overall packet loss probability and the distribution of the packet losses among sources at a multiplexer performing packet scale buffering.

The analysis is based on the fluid flow modeling of the traffic streams and uses results of the theory of large deviations to approximate probabilities of rare events. A short summary of the basic ideas behind the large deviation theory is presented in [1], Chapter 14.3.

The long term overall packet loss probability $P_{loss}$ is the ratio of the average packet loss rate to the average packet arrival rate:

$$P_{loss} = \frac{1}{m} E\{(\lambda_t - c)^+)\} = \frac{1}{m} \int_{\lambda_t > c} (\lambda_t - c) dP, \tag{3}$$

where $m$ is the mean rate of the multiplexed flows, $\lambda_t$ is the instantaneous arrival rate, $P$ is the probability distribution of the instantaneous arrival rate and $c$ denotes the link capacity. We can express $P_{loss}$ with the instantaneous loss probability $p_t$ as

$$P_{loss} = E\{p_t \lambda_t\}, \quad \text{where} \quad p_t = \frac{(\lambda_t - c)^+}{\lambda_t}. \tag{4}$$

Large deviation theory provides a way to approximate tail probabilities like $P\{\lambda_t > c\}$ and thus the loss probability.

First we introduce $P_\beta$, the shifted probability measure of $\lambda_t$, such that

$$dP_\beta = \frac{e^{\beta\lambda_t}}{\psi(\beta)} dP, \quad \text{where} \quad \psi(\beta) = E\{e^{\beta\lambda_t}\}, \tag{5}$$

and $\mu(\beta)$, the cumulant generating function as

$$\mu(\beta) = ln\psi(\beta). \tag{6}$$

From this, the original probability can be expressed as

$$dP = e^{-\beta\lambda_t} \psi(\beta) dP_\beta. \tag{7}$$

This shifted distribution can be accurately approximated around its mean $m(\beta) = E_\beta\{\lambda_t\}$ by a normal distribution with the same mean. Since $\beta$ is a free parameter $m(\beta)$ can be moved to the value of interest for the tail probability, in our case to $c$. The corresponding value of $\beta$, denoted by $\beta^*$ is given by

$$m(\beta^*) = c. \tag{8}$$

From the definition in Eq. 6 $m(\beta) = \mu'(\beta) = E_\beta\{\lambda_t\}$, and $\sigma^2(\beta) = m'(\beta) = \mu''(\beta)$ are the expected value and variance of the shifted distribution $P_\beta$. Consequently, if $\lambda_t$ is not constant (its variance, $\sigma^2(\beta)$ is positive), then $m(\beta)$ is strictly increasing where $\mu(\lambda_t)$ is finite and equation 8 has a unique solution.

The evaluation of the integral in Eq. 3 leads to [1]

$$P_{loss} \approx \frac{1}{\sqrt{2\pi} m \beta^{*2} \sigma(\beta^*)} e^{-\beta^* c + \mu(\beta^*)}. \tag{9}$$

The above calculated overall loss probability gives also the loss probability of the individual streams if they have the same characteristics. However, the loss distribution among streams with different characteristics will be uneven. Assume, that packets arriving in overload periods have the same loss probability independently of the source of the packets. Still, sources send a different proportion of packets during these periods. For bursty streams the bursts are correlated to overload periods as they are causing the overload themself. As a result, bursty streams experience higher loss probability than smooth ones.

The loss distribution among sources in the case of burst scale overflow can be estimated as described in [1]. Similarly to Eq. 3, the loss probability of the individual stream $i$ is equal to

$$P_{loss}^{(i)} = \frac{1}{m_i} E\{p_t \lambda_t^{(i)}\}, \tag{10}$$

where $m_i$ is the mean rate of the stream, $\lambda_t^{(i)}$ is the instantaneous rate, and $p_t$ is the loss probability at time $t$.

The probability shift method can be applied in this case as well. Assuming, that the probability that $\lambda_t$ significantly exceeds $c$ is very small, $P_{loss}^{(i)}$ can be approximated as

$$\frac{P_{loss}^{(i)}}{P_{loss}} \approx \frac{m}{c}\frac{m_i(\beta^*)}{m_i}. \tag{11}$$

To calculate the overall loss probability using Eq. 9 and the loss distribution among the sources using Eq. 11 the value of $\beta^*$, $m_i(\beta^*)$, $\sigma_i(\beta^*)$ and $m(\beta^*)$ has to be derived. These values can be expressed in closed form if $\lambda_t^{(i)}$ and $\lambda_t$ have some standard distribution (e.g., for normal distribution). In the case of real sources,however, they have to be calculated numerically. Assuming, that the distributions of the individual streams are known from measurements, and the multiplexed streams are independent, the following system of equations has to be solved

$$m(\beta^*) = c \tag{12}$$

$$m(\beta) = \sum_i m_{(i)}(\beta) \tag{13}$$

$$\sigma^2(\beta) = \sum_i {\sigma_{(i)}}^2(\beta) \tag{14}$$

$$m_{(i)}(\beta) = \frac{d}{d\beta} lnE\{e^{\beta\lambda_t^{(i)}}\} = \frac{\frac{d}{d\beta}E\{e^{\beta\lambda_t^{(i)}}\}}{E\{e^{\beta\lambda_t^{(i)}}\}} = \frac{E\{\lambda_t^{(i)}e^{\beta\lambda_t^{(i)}}\}}{E\{e^{\beta\lambda_t^{(i)}}\}} \tag{15}$$

$$\sigma_{(i)}^2(\beta) = \frac{d^2}{d\beta^2} lnE\{e^{\beta\lambda_t^{(i)}}\} = \frac{E\{{\lambda_t^{(i)}}^2 e^{\beta\lambda_t^{(i)}}\}E\{e^{\beta\lambda_t^{(i)}}\} - E\{\lambda_t^{(i)}e^{\beta\lambda_t^{(i)}}\}^2}{E\{e^{\beta\lambda_t^{(i)}}\}^2} \tag{16}$$

# 6 PERFORMANCE EVALUATION

In order to assess the effectiveness of the proposed source shaping solutions we consider the statistical quantities of the shaped video traces and packet loss statistics in case of multiplexing video streams at a single node with packet scale buffering. The presented results are based on the analytical method described in section 5 and on simulations using ns-2.

The considered scenario is shown in figure 1. It consists of $n$ independent sources generating MPEG video streams, single buffer leaky buckets as source shapers and a multiplexing node. Each stream is shaped with some delay constraint and then multiplexed at the node with a small buffer to resolve packet scale congestion.

We present results for two MPEG-4 video traces, a soccer game with an average bit rate of 1.1 Mbps and a talk show with an average rate of 540 kbps. The traces are approximately 3600 seconds, thus 90000 frames, and 2700 seconds, thus 67000 frames long. The frames of the MPEG traces are packetized to 188 bytes, as given for the transport stream in the MPEG-2 standard [IEC61883].

Throughout the simulations we consider a single outgoing link at the multiplexing node with a capacity of 45 Mbps in the case of the soccer game and of 22.5 Mbps in the case of the talk show. We choose the link capacities proportionally to the average rate of the streams. This solution allows us to compare results at the same link utilization and level of statistical multiplexing. The multiplexing buffer can store up to 15 packets.

### Trace statistics

First we consider the statistical properties of a single video stream before and after shaping for different values of maximum shaper delay $d$.

Figures 2 and 3 show the number of transmitted bits in a frame time for the original and shaped trace of the soccer game and the talk show respectively. The maximum shaper delays are 40 ms and 120 ms, the shaping is performed using the BC shaper algorithm. Even the relatively small shaper delay of 40 ms allows a significant reduction of the rate fluctuation. The maximum transmission rate is decreased from 3.6Mbps to 3Mbps for the soccer game trace and from 3.1Mbps to 2Mbps for the talk show trace.

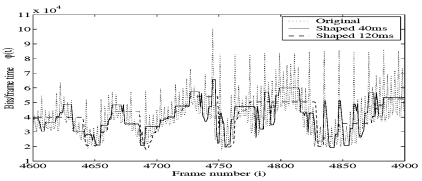


**Figure 2:** Number of bits transmitted in a frame time for the soccer game trace without shaping and with shaping for $d$=40 ms and $d$=120 ms, using the BC shaper algorithm.

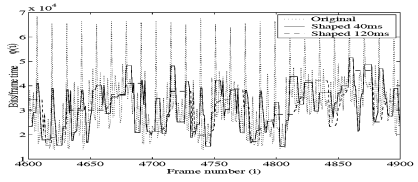


**Figure 3:** Number of bits transmitted in a frame time for the talk show trace without shaping and with shaping for $d$=40 ms and $d$=120 ms, using the BC shaper algorithm.

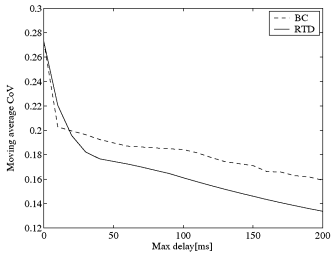


**Figure 4:** CoV of the shaped soccer trace versus maximum shaper delay $d$, considering the BC and RTD shaping algorithms.

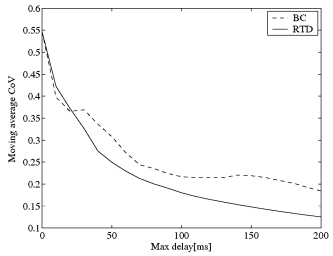


**Figure 5:** CoV of the shaped talk show trace versus maximum shaper delay $d$, considering the BC and RTD shaping algorithms.

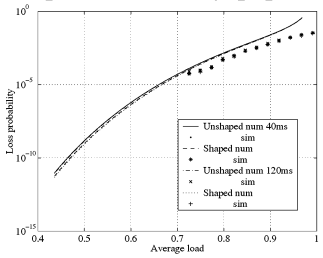


**Figure 6:** Loss probability of shaped and unshaped streams for 1 shaped stream. The talk show trace and the BC shaping algorithm is considered. Simulation validates the analytical results.

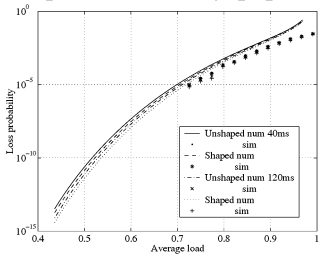


**Figure 7:** Loss probability of shaped and unshaped streams for half of the streams shaped. The talk show trace and the BC shaping algorithm is considered. Simulation validates the analytical results.

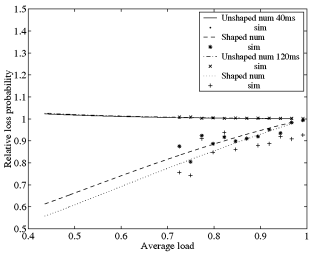


**Figure 8:** Relative loss probability of shaped and unshaped streams for 1 shaped stream. The talk show trace and the BC shaping algorithm is considered. Simulation validates the analytical results.

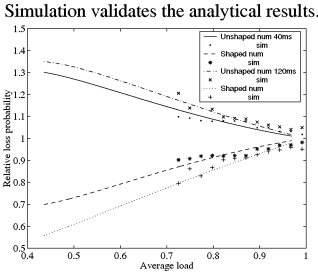


**Figure 9:** Relative loss probability of shaped and unshaped streams for and half of the streams shaped. The talk show trace and the BC shaping algorithm is considered. Simulation validates the analytical results.

Figures 4 and 5 show the coefficient of variation (CoV) of the traces, defined as

$$CoV = \sqrt{\frac{1}{N_{tr}} * \sum_{i=1}^{N_{tr}} (\varphi(i) - \frac{\sum_{j=i-N_{GOP}}^{i-1} \varphi(j)}{N_{GOP}})^2} * \frac{1}{E\{\varphi\}} \quad (17)$$

where $N_{tr}$ is the number of frames in the trace, $\varphi(i)$ is the number of bits transmitted in the $i$th frame time and $E\{\varphi\} = E\{f\}$ is the average number of bits transmitted in one frame time. The CoV is calculated using the moving average over one GOP time as mean value. This way the CoV reflects the frame to frame rate fluctuations without the rate variation due to the scene changes in the trace. The two curves in the figure show the CoV values for the two proposed shaping solutions.

The graphs show that for small values of shaper delay $d$ the CoV decreases very rapidly, reflecting that a delay of a couple of frame times (20-80 ms) allows the smoothing of the transmission rates of consecutive I, P and B frames. At larger delays the marginal gain decreases significantly.

The two shaping methods result in similar changes in the CoV values. As expected, the RTD method decreases the rate fluctuations better. For large values of $d$ the difference is around 20% in the terms of CoV reduction, since the RTD method can adjust the shaper rate more precisely based on the information maintained in the buffer, while the BC method over- and underestimates the shaper rate and has to make corrections later. As the difference is not significant, due to its simplicity we focus on the BC method in the followings.

### Packet loss probabilities

In this part we investigate the average packet loss probability of multiplexed video streams as a function of the average load, defined by the ratio of the sum of the mean rates of the streams to the link transmission capacity. The presented results are based on mathematical analysis, simulation results are shown to demonstrate the accuracy of the method. Simulations were run 20000 to 100000 seconds to have enough loss events even in the case of loss probabilities in the order $10^{-5}$.

We show results for shaping the talk show trace with the BC method. To see the effect of introducing shaping gradually at the sources two scenarios are considered. In the first one only one stream is shaped, while all the other multiplexed streams are transmitted unshaped. In the second scenario half of the streams are shaped at the source.

Figure 6 shows the loss probability of the shaped and unshaped sources for shaper delays of 40 ms and 120 ms in the case of 1 shaped source. Figure 7 shows the results for the scenario where half of the sources are shaped. Figures 8 and 9 show the relative loss probabilities of the shaped and unshaped sources compared to the average loss probability.

The numerical results are validated by simulation. The results reflect, that the mathematical analysis works well for losses up to $10^{-2}$, then slightly overestimates the loss probability as a consequence of the large deviation approximation.

Comparing figures 6 and 7 it can be seen that shaping half of the sources decreases the overall loss probability by roughly one order of magnitude. The results with different shaper delays show that in the case of the considered talk show trace, shaping with a delay of 40 ms is almost as efficient as shaping with a delay of 120 ms in the terms of reducing the loss probability. For the soccer trace shaping with a delay of 120 ms has a slightly bigger effect. This is due to the lower ratio of temporal redundancy which induces lower peak to mean ratio.

The results show in Figures 8 and 9 that the decrease in the loss probablity achieved by shaping the sources increases as the average load, and thus the loss rate decreases. Since the desired loss probability of video streams is in the order $10^{-5}$, the difference can be up to 35%, even if only one stream is shaped. The gain achieved depends on the stream characteristics, for the soccer trace the experienced gain was less, around 20% at a loss probability around $10^{-5}$.

### Packet loss probabilities in I, P and B frames

In addition to the average packet loss probability of the streams it is worthwhile to evaluate the packet loss probability in individual frame types, since it affects the percepted visual quality.

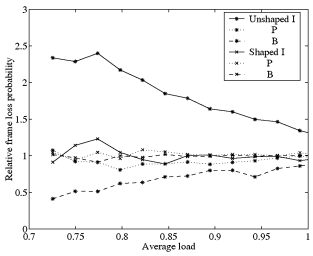


**Figure 10:** Relative packet loss probability in I,P,B frames of the shaped and unshaped streams for $d$ = 120 ms, half of the streams shaped. The talk show trace and the RTD shaping algorithm is considered. Simulation results.

Figure 10 shows the packet loss probability in I, P and B frames relative to the average loss probability for the shaped and unshaped sources for the scenario when half of the sources is shaped with a maximum shaper delay of 120 ms. The figure shows that while in the case of unshaped sources the loss probability in the I frames is the highest, up to 100% above the average loss probability and that in the B frames is the lowest, in the case of shaped sources the loss probabilities in the individual frame types are roughly the same. In the I frames the decrease of loss probability is around 60%. Consequently, as losses in the I frame have a significant effect on the visual quality, the positive effects of the shaping include not only lower loss probability but also the improved distribution of these losses among the frame types.

The presented results show the following effects of source shaping. Considering the trace statistics, shaper delays in the 20-40 ms range decrease the CoV of the stream significantly, higher delays introduce decreasing marginal gains. Comparing the two proposed shaping algorithms, the simple BC algorithm works rather well, especially at small shaper delays. Results on multiplexing the video streams at a multiplexer with small buffer show that the shaped streams experience lower loss probabilities than the unshaped ones, the difference is about 30%. The gradual introduction of the source shaping in the network has a significant effect, the loss probabilities decrease with one order of magnitude if half of the sources adopt shaping. The positive effect of shaping is reflected by the distribution of losses among the different frame types in the video stream. Multiplexing unshaped streams results high loss probability for the I frames, this loss probability decreases significantly if the stream is shaped.

# 7 CONCLUSION

In this paper we proposed and evaluated solutions that assist live video transmission over the Internet. Specifically, we considered the scenario, when the MPEG coded video streams are shaped at the source host and network routers provide small buffers to resolve packet scale congestion, motivated by the current trends of designing traffic control solutions, where the main idea is to add functions to the hosts and keep the operation of the network routers simple.

We proposed computationally simple shaping algorithms that are adequate to shape live video streams with a wide range of delay tolerance at the shaper and provided analytical method to evaluate the efficiency of the proposed solutions.

The analytical and simulation-based performance evaluation proved that i) a simple shaper algorithm based on buffer occupancy and delay limit results in efficient shaping in many cases; ii) even shaping with very low delay bound, adequate for real-time applications, improves the performance in terms of packet loss probability and the packet loss distribution among the different frame types; and iii) shaping provides a means to improve the quality of individual video transmissions even if not all the hosts shape their traffic.

Finally, as ongoing work we further investigate the methods to assist video transmission over the Internet. Specifically, we are interested in the efficiency of source shaping versus buffering at the routers and buffering versus forward error correction.

# References


[1] "Broadband network teletraffic, Final report of action COST 242," Springer, 1996.

[2] G. Bianchi, A. Capone, Ch. Pertioli, "Throughput Analysis of End–to–End Measurement-Based Admission Control in IP," in *Proc. of IEEE INFOCOM 2000,*, March 26–30, 2000, pp. 1461–1470.

[3] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," in *Proc. of ACM SIGCOMM 2000*, 28 Aug. – 1. Sept., 2000, pp. 57-69.

[4] R. Cruz, "A Calculus for Network Delay, Part I: Network Elements in Isolation," *IEEE Transactions on Information Theory,* vol. 37, no. 1, Jan. 1991, pp. 114-131.

[5] R. B. Gibbens and F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," in *Proc. of the 16th International Teletraffic Congress*, June 7–11, 1999, pp. 941-952.

[6] V. Elek, G. Karlsson, R. Ronngren, "Admission Control Based on End-to-end Measurements," in *Proc. of IEEE INFOCOM 2000,*, March 26–30, 2000, pp. 623–630.

[7] E. W. Knightly, "H-BIND: a New Approach to Providing Statistical Performance Guarantees to VBR Traffic," *IEEE Infocom'96,* March 1996.

[8] E. W. Knightly and H. Zhang, "D-BIND: an Accurate Traffic Model for Providing QoS Guarantees to VBR Traffic," *IEEE/ACM Transactions on Networking,* vol.5, no.2, April 1997, pp.219-231.

[9] H. Zhang and E. W. Knightly, "RED-VBR: A Renegotiation-based Approach to Support Delay-Sensitive VBR Video," *ACM Multimedia system Journal,* May 1997.

[10] Simon S. Lam, Simon Chow, David K. Y. Yau,"An Algorithm for Lossless Smoothing of MPEG video," *ACM SIGCOMM Computer Communication Review,* vol. 24, no. 4, Oct. 1994, pp. 281-293

[11] J-Y. Le Boudec and O. Verscheure, "Optimal Smoothing for Guaranteed Service," *IEEE Transactions on Networking,* vol.8, no.10, Dec. 2000.

[12] N. Modani, P. Dube and A. Kumar, "Measurement Based Optimal Source Shaping with a Shaping+Multiplexing Delay Constraint" *Proc. of IEEE Infocom 2000*, March 26–30, 2000,pp. 1807-1816.

[13] J. Rexford, S. Sen, J Dey, W. Feng, J. Kurose, J. Stankovic and D. Towsley, "Online Smoothing of Live, Variable-bit-rate Video" *Proc. of International Workshop on Network and Operating Systems Support for Digital Audio and Video*, May 1997, pp. 249-257

[14] J. W. Roberts, "Traffic Theory and the Internet," *IEEE Communications Magazine,* Jan. 2001., pp.94-99.

[15] M. Reisslein, K. W. Ross, S. Rajagopal, "Guaranteeing Statistical QoS to Regulated Traffic: The Multiple Node Case," in Proc. *IEEE Decision & Control'98,* pp. 531-538, 1998.

[16] M. Reisslein, K. W. Ross, S. Rajagopal, "Guaranteeing Statistical QoS to Regulated Traffic: The Single Node Case," in Proc. *IEEE Infocom'99,* pp. 1061-1072, 1999.

[17] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'dell, A. Romanow, A. Weiurib, and L. Zhang, "Resource ReSerVation Protocol - Version 1 Applicability Statement Some Guidelines on Deployment. RFC 2208," September 1997.

# Comparison of Shaping and Buffering for Video Transmission

Gy. Dán and V. Fodor
*In Proceedings of the 16th Nordic Teletraffic Seminar (NTS 16),*
August 2002, Espoo, Finland

# Comparison of Shaping and Buffering for Video Transmission

György Dán and Viktória Fodor
Royal Institute of Technology,
Department of Microelectronics and Information Technology
P.O.Box Electrum 229, SE-16440 Kista, Sweden
{gyuri,viktoria}@it.kth.se

**Abstract**

Video communication over the Internet requires performance guarantees in terms of limited packet loss probability and end to end delay. This paper compares two possible network scenarios for transmitting video streams, source shaping combined with small buffers at the network nodes and delay limited buffering in the network. It is shown that due to its simplicity and performance comparable to buffering source shaping can provide the solution for efficient video transmission in the Internet.

## 1 Introduction

The transmission of video traffic over large packet switched networks like the Internet is still a fundamental problem of network design. First, video applications require quality of service guarantees from the network, in terms of limited packet loss, end to end delay and delay variation. The acceptable packet loss probability depends on the video coding scheme and is in the range of $10^{-5}$–$10^{-3}$. The delay limitation depends on the application. While in the case of computer to human applications delay can be up to several seconds, in the case of human to human applications delay should be kept below 150 ms. Second, the provisioning of the required quality at a network utilization acceptable for the operator is a complex issue due to the characteristics of coded video streams. Considering the widely used MPEG coding, the transmission rate is changing in short and long time scales. Moreover, as a result of data compression the distribution of the packet losses affects the percepted visual quality: multiple packet loss in a single frame decreases the visual quality of the corresponding picture, and the effect of a packet loss in an I frame propagates to P and B frames.

Research and developement efforts today address the problem of finding efficient traffic control solutions that support visual communication and can be introduced with acceptable cost [3, 7]. One of the essential questions is whether to design networks with large buffers

at the network nodes or rather use small buffers and source shaping when transmitting delay and loss sensitive video traffic (E.g., [8, 9]). Intuitively, using large buffers low packet loss probability can be provided even at high network utilization, but the control of end to end delay and delay variation has to be solved. On the other hand, source shaping with small buffers at the nodes bounds the delays but at the price of increased packet loss probability.

Considering the feasibility of the two solutions, shaping has its advantages, as it can be introduced gradually, according to the individual applications' needs and does not require any support from the network. On the other hand, in networks with large buffers delay aware scheduling and/or jitter compensation has to be introduced at all the network nodes to limit the end to end delay and avoid the increase of burstiness of the traffic streams.

In [9] the performance of source shaping and buffering is compared for networks providing strict end to end delay bounds. The paper compares two solutions. In one of them, source shaping with the maximum acceptable delay is applied and nodes are equipped with small buffers, performing so called packet scale buffering [1]. In the other solution the maximum acceptable delay is divided among the network nodes thus nodes perform burst scale buffering with buffer size defined by the per node maximum delay. Nodes apply jitter compensation for each video stream to control the burstiness of the stream. It is shown that in the terms of packet loss traffic shaping is less efficient in the case of a single node multiplexer, but in the case of long transmission paths, when the maximum end to end delay has to be split among many nodes, traffic shaping outperforms buffering.

In this paper we further evaluate the performance of these two solutions, comparing the distribution of losses and their effect on the visual quality of the received video stream. The paper is organized as follows. The next section describes the two network scenarios considered, section 3 presents simulation results and discusses the loss characteristics of the scenarios, finally in section 4 we conclude our work.

# 2 System description

We consider the following networking scenario. On-line video streams have to be transmitted through a large packet switched network - specifically, the Internet. We assume, that the network has DiffServ [2] capabilities, and the transmission of video streams is not disturbed by best effort traffic.

We assume that call admission control is introduced for the video streams to provide performance guarantees. Measurement based end-node call admission control solutions that limit the packet loss probability of the accepted streams without any network signaling or per stream processing at the nodes are proposed in e.g.,[3, 5].

While packet loss is limited by the admission control, end to end delay is limited by the network architecture, specifically by the buffer sizes at the source shapers and network nodes.

The considered system is shown in figure 1, consisting of the source and the destination of the video stream, the source coder and decoder, the transmission and reception control units that can contain source shaper and playout buffer and error control coding and decoding.

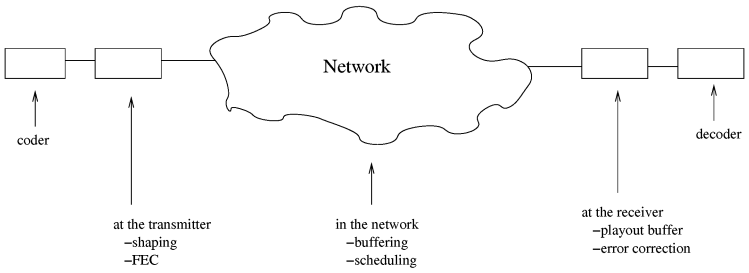


Figure 1: Considered network model

### Source coding

The sources generate MPEG coded streams, the most commonly used encoding scheme for the storage and transmission of video information. In the MPEG stream information is stored as a sequence of frames, corresponding to a sequence of pictures in a video, generated with fix time intervals. Compression is achieved by eliminating the spatial and temporal redundancy of the information in the frames. Spatial redundancy is decreased by intraframe coding of the individual frames, while temporal redundancy is reduced by interframe coding between subsequent frames. Thus the sequence of frames consists of intraframe coded frames (I frames), and interframe coded predicted (P frames) and bidirectionally predicted frames (B frames). The subsequent frames between two consecutive I frames form a group of picture (GOP). The GOP structure of the streams can be different, depending on the required quality. A typical example for the sequence of frames is IBBPBBPBBPBB.

The stream is highly bursty, with fluctuations on two timescales. The intraframe coding compresses complex scenes with less efficiency, and consequently, the frame sizes change on the long term at the scene changes. The interframe coding leads to short term frame size fluctuation, since I frames are usually significantly larger than P frames, and P frames are larger than B frames. The scale of the fluctuation is about a factor of 3 on the long, and a factor of 10 on the short term.

As a consequence of the coding scheme, information loss in the three frame types has different effect on the percepted visual quality. The loss of data in an I frame propagates forward through the next GOP and backward to the last P frame (affecting up to 14 frames if the number of frames in an open GOP is 12). Meanwhile, the loss of data in a B frame only affects that particular frame.

### Source shaping

Shapers used at the sources decrease the frame to frame fluctuation of the coded video stream. The shaper we use in this work is a single buffer leaky bucket, as it is proved to be optimal for networks with small buffers [8]. Frames leaving the encoder are stored in the shaper buffer and are transmitted with a given transmission rate, which is adjusted to provide lossless, delay limited shaping. Shaper algorithms for on-line video streams are proposed in [6]. In this paper, we apply a low complexity solution based on the shaper buffer content, as described in [4].

The algorithm to control the shaper transmission rate $r$ aims to minimize the maximum and the variance of the transmission rate, is efficient for large range of end to end delay limit and assists real time operation as it has low computational complexity and does not require knowledge on the GOP structure.

The algorithm assumes that the shaper can detect the type of the arriving frame and applies the following rules to set the shaper rate.

The shaper rate can be changed at any frame arrival $n$. The lowest acceptable shaper rate $r_{min}(n)$ is calculated based on the amount of data in the buffer, $b(n)$ and the delay limit $d$, as:

$$r_{min}(n) = \frac{b(n)}{d}.$$

The shaper rate is increased at any frame arrival if $r_{min}(n) > r(n-1)$. The shaper rate is decreased if $r_{min}(n) < r(n-1)$, the new frame is of type I and the buffer is empty before the frame arrival. This rule is based on the assumption that small P or B frames do not indicate intensity change in the video stream and is restricted since the delay limit of individual frames in the buffer can not be ensured. To decrease fluctuation, the shaper rate is decreased gently,

$$r(n) = \frac{r_{min}(n) + r(n-1)}{2}.$$

P and B frames entering the shaper when the buffer is empty are transmitted with a rate such that the frame leaves the buffer before the new frame arrives, i.e., in one frame time, in order to prevent the shaper from keeping data before larger I and P frames arrive.

If shaping is not applied the source sends the individual frames smoothed over one frame time i.e., 40 ms if the frame rate is 25 frames per second.

**Node architecture**

Packet streams leaving the source nodes are multiplexed at the network nodes. Two different node architectures are considered.

1. The output buffers at the nodes provide buffering for simultaneously arriving packets only (packet scale buffering). In this case the buffer size is in the range of the ratio of the link speed and the peak rate of the streams.

2. The output buffers are large to provide buffering for bursts (burst scale buffering). The buffer size in this case is limited by the maximum acceptable end to end delay and the number of hops on the transmission path. Specifically, it means maximizing the buffer capacities by $B_i = \frac{d}{N} c_i$ where $d$ is the maximum end to end delay, $N$ the number of nodes on the transmission path, and $c_i$ the link capacity at node $i$ [9]. It is assumed that nodes apply jitter compensation, which means that the delay at the networking nodes is exactly $d/N$.

Shaping of the sources has some advantages compared to buffering inside the network. *i*) Shaping can be introduced gradually according to the applications' needs and nodes inside the network are not affected. Buffering with delay limit requires updating of all the network nodes. *ii*) Furthermore, buffering with delay limit requires per stream delay and jitter control at the nodes, with stream specific limits, an thus is a source of network scalability problems. In the case of source shaping the end nodes keep track of the delay control and network nodes do not perform per stream processing.

# 3 Performance evaluation

To compare shaping and delay limited buffering we compare the average packet loss probabilities, the packet loss distribution among I, P and B frames, the probability of consecutive packet losses and losses in small blocks of packets. The presented results are simulation results, the simulation time was between 20000 to 60000 seconds to have enough loss events even in the case of loss probabilities in the order $10^{-5}$.

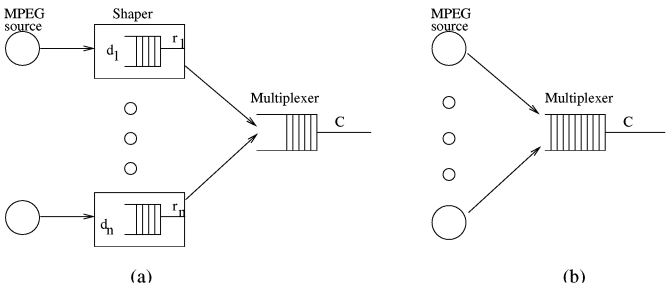


Figure 2: The considered network model, (a) with shapers, (b) with buffering

The simulated network model is shown in figure 2. The system includes traffic sources, source shapers and a multiplexing node with a single output link. Two scenarios are considered. In scenario $(a)$ the nodes apply source shaping with delay limited by the maximum acceptable end to end delay and the multiplexer provides a small buffer for packet scale buffering. In scenario $(b)$ the sources are not shaped and the acceptable per node delay is allocated to the multiplexer buffer.

We present results for an MPEG-4 video trace, a talk show with an average rate of 540 kbps. The trace is approximately 2700 seconds, thus 67000 frames long. The frames of the MPEG trace are packetized to 188 bytes, as given for the transport stream in the MPEG-2 standard [IEC61883]. The capacity of the output link is 22.5 Mbps. The size of the shaper buffer is determined by the considered end to end delay limits, 20 ms and 40 ms, acceptable for real-time communication. When shaping is used the buffer at the multiplexer stores up to 10 packets to provide packet scale buffering. When, instead, delay limited buffering is applied at the network nodes, we assume a transmission path length of 10 nodes, resulting buffer capacity for 38 and 66 packets for delays of 20 ms and 40 ms respectively.

### Average packet loss probabilities

First we investigate the average packet loss probability of the multiplexed streams as a function of the average load at the multiplexer, defined by the ratio of the sum of the mean rates of the streams to the link transmission capacity.

Figure 3 shows the average packet loss probability as a function of the load for end to end delays of 20 ms and 40 ms for both the shaped and the buffered traces. Buffering results in a lower average packet loss probability, the difference is less than one order of magnitude. Considering an acceptable packet loss probability of $10^{-5}$, the difference of the maximum network load with shaping and buffering is 3–8% depending on the delay limit.

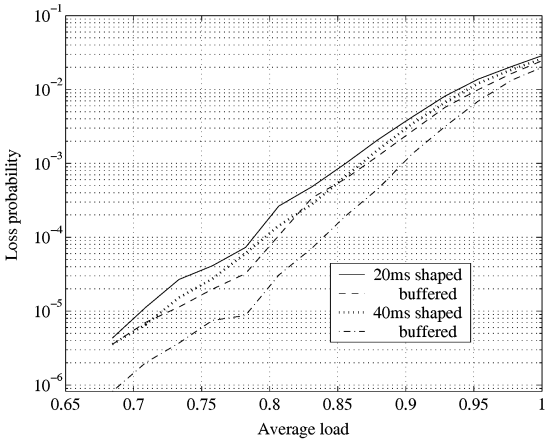


Figure 3: Average packet loss probability of the shaped and buffered streams.

**Packet loss probabilities in I, P and B frames**

In addition to the average packet loss probability of the streams it is worthwhile to evaluate the packet loss probability in individual frame types, since, as a consequence of the coding scheme it affects the percepted visual quality.

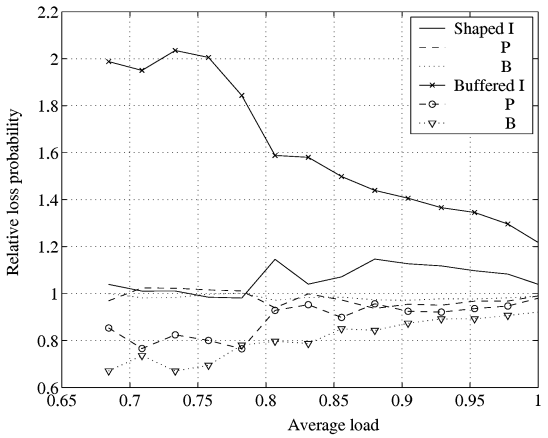


Figure 4: Relative packet loss probability in I,P,B frames of the shaped and buffered streams for $d$ = 40 ms.

Figure 4 shows the packet loss probability in I, P and B frames relative to the average loss probability for the shaped and buffered sources for an end to end delay of 40 ms. The figure shows that while in the case of unshaped sources the loss probability in the I frames is the highest, up to 100% above the average loss probability and that in the B frames is the lowest,

in the case of shaped sources the loss probabilities in the individual frame types are roughly the same. In the I frames the decrease of loss probability is around 60%. Consequently, as losses in the I frame have a significant effect on the visual quality, traffic shaping improves the distribution of the losses among the frame types.

### Consecutive packet loss

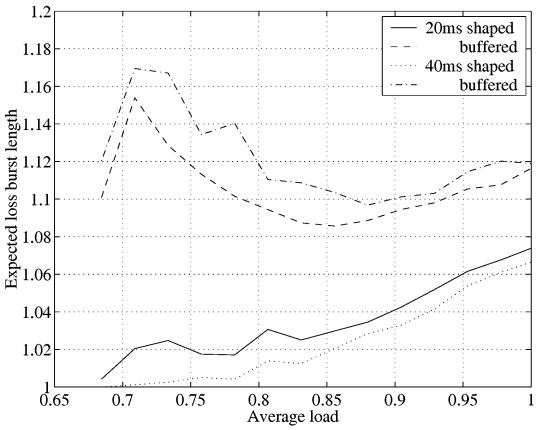


Figure 5: Expected number of consecutive packet losses.

Consecutive packet losses might have a degrading effect on the visual quality, as larger parts of a picture have to be recovered. Figure 5 shows the expected number of consecutively lost packets. In the case of the shaped sources the expected loss burst length is 1 for a load up to 0.7-0.8, meaning, that single packet losses happen in most of the cases, and increases as the load increases. In the case of buffered sources the graph of the expected burst length has a U shape, reaching a minimum of 1.08 consecutive packet losses at an average load of approximately 0.85. The shape of the graphs for the buffered case can be explained by the burstiness of the sources. In the case of low average load a bursty stream sending at a high bitrate is probable to cause a congestion period itself, and loose all its packets during that period, thus losses tend to occur in bursts. In the high load region the congestion periods get longer what explains the increase in the expected loss burst length for both the shaped and the buffered streams.

Considering the probability of consecutive packet losses, the result indicates that in the case of buffering the probability of two or more consecutive packet losses is orders of magnitude higher than in the case of shaping at low or modest average load.

### Losses in packet blocks

This part evaluates the average number of packets lost in small blocks of packets. The results help to investigate whether forward error correction solutions, based on block coding of a number of packets can improve the quality of the transmission.

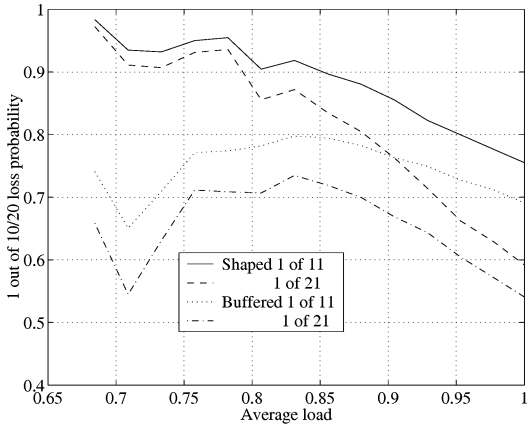


Figure 6: Probability of 1 packet loss out of 10 and 20 packets for $d$ = 20 ms.

Figure 6 shows the probability that out of 10 or 20 packets only 1 packet, if any, gets lost for an end to end delay of 20 ms. Figure 7 shows the same probability for an end to end delay of 40 ms. For the shaped sources the probability of loosing 1 packet out of 10 or 20 is close to 1 at a load up to 0.8, and decreases as the load increases. The graphs showing the 1 out of 10 and 20 packet loss probability for the buffered sources however have an upside down U shape, in accordance with figure 5. Thus the probability of loosing more than 1 packet out of 10 or 20 first increases as the average load increases to reach its maximum of 0.73 at an average load of 0.85, then decreases as the average load further increases.

The results show that with source shaping maximum 5% of the losses are multiple losses in packet blocks up to a load of 0.8, compared to 20–30% with buffering. It indicates that that while in the case of buffering the high probability of multiple losses makes error correction coding inefficient, in the case of shaping forward error correction might be useful to improve transmission quality in terms of residual packet loss probability.

# 4 Conclusion

In this paper we evaluated the feasibility and efficiency of source shaping versus delay limited buffering for the transmission of delay and loss sensitive video transmission over the internet.

The extensive performance analysis, based on simulation, provided the following results.

- The average packet loss probability can be up to one order of magnitude lower in the case of buffering at the network nodes.
- The packet loss distibution among frames is uneven if source shaping is not applied, with high packet loss probability in the I frames. Source shaping equalizes the per frame packet loss probabilities.
- The probability of consecutive packet losses can be orders of magnitude lower in the case of source shaping, depending on the average loss probability.

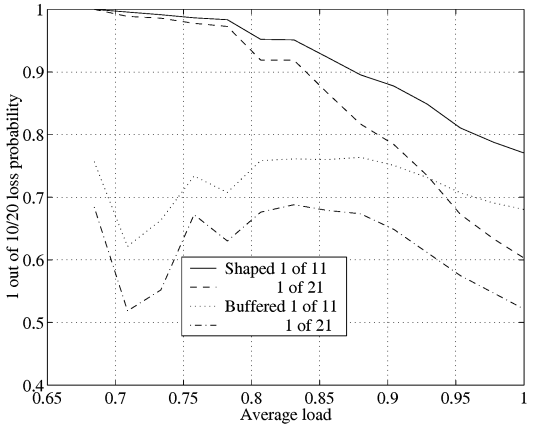


Figure 7: Probability of 1 packet loss out of 10 and 20 packets for $d$ = 40 ms.

- The probability of multiple losses in blocks of packets is 4–6 times lower with shaping than with buffering, giving a fair chance of efficient forward error correction.

The above results indicate that while the average packet loss might be higher in the case of packet scale buffering combined with traffic shaping, due to the change in the packet loss distribution, the percepted visual quality can be close to the one in networks with large buffers. Considering the feasibility of the two solutions, we believe that source shaping together with small buffers at the network nodes can provide the solution for transmission of delay sensitive video traffic in the Internet.

# References

[1] "Broadband Network Teletraffic, Final Report of Action COST 242," Springer, 1996.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services. RFC 2475", December 1998.

[3] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," in *Proc. of ACM SIGCOMM 2000*, 28 Aug. – 1. Sept., 2000, pp. 57-69.

[4] Gy. Dán, V. Fodor, "On The Efficiency of Shaping Live Video Streams", *SPECTS'02*, July 14–18, 2002.

[5] V. Elek, G. Karlsson, R. Ronngren, "Admission Control Based on End-to-end Measurements," in *Proc. of IEEE INFOCOM 2000,*, March 26–30, 2000, pp. 623–630.

[6] S. S. Lam, S Chow, D. K. Y. Yau, "An Algorithm for Lossless Smoothing of MPEG video," *ACM SIGCOMM Computer Communication Review,* vol. 24, no. 4, Oct. 1994, pp. 281-293

[7] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'dell, A. Romanow, A. Weiurib, L.Zhang, "Resource ReSerVation Protocol - Version 1 Applicability Statement Some Guidelines on Deployment. RFC 2208," September 1997.

[8] M. Reisslein, K. W. Ross, S. Rajagopal, "Guaranteeing Statistical QoS to Regulated Traffic: The Multiple Node Case," in Proc. *IEEE Decision & Control'98,* pp. 531-538, 1998.

[9] T. Wu, E. W. Knightly, "Buffering vs. Smoothing for end-to-end QoS: Fundamental issues and comparison," in Proc. *IEEE Performance'99* Aug. 1999.

# Quality Differentiation with Source Shaping and Forward Error Correction

Gy. Dán and V. Fodor
*In Proceedings of the 1st International Workshop on Multimedia Interactive Protocols and Systems (MIPS 2003),*
November 2003, Naples, Italy

# Quality Differentiation with Source Shaping and Forward Error Correction *

György Dán and Viktória Fodor

KTH, Royal Institute of Technology,
Department of Microelectronics and Information Technology,
{gyuri,viktoria}@imit.kth.se

**Abstract.** The transmission of video traffic over the Internet is a fundamental issue of network design. Video applications require quality of service guarantees from the network in terms of limited packet loss, end-to-end delay, and delay variation. The question of today's research and development is how to provide these guarantees considering the architecture of the present Internet. In the last years a variety of admission control schemes based on per-hop or end-to-end measurements has been suggested to control delay and loss sensitive streams with very little or no support at the routers. Most of these solutions, however, have to apply the same acceptance threshold for all streams, a significant limitation considering the diverse quality requirements of the applications. In this work we investigate how source shaping and forward error correction (FEC) can be used together to achieve application specific quality differentiation in terms of end-to-end delay and packet loss probability. While source shaping and FEC have been proposed independently to decrease the probability of packet loss due to buffer overflow, their joint use has not been studied before.
As the two control functions use the same scarce resource, end-node delay, and their efficiency to decrease loss probability is proportional to the introduced delay but with a decreasing marginal gain, combining the two a better performance can be achieved than by using only one of them.
The performance evaluation focuses on the optimal delay allocation for shaping and FEC, such that the loss probability is minimized. We investigate how shaping can be used to substitute FEC redundancy and the sensitivity of the quality differentiation to the background traffic characteristics.

**Keywords:** Quality of Service, source shaping, FEC

## 1 Introduction

Internet is now considered as the universal network for future data, voice and video communications. It is recognized, however, that the best effort service implemented today is not satisfactory for delay and loss sensitive applications such as voice and video. It is widely accepted that quality provisioning for these applications requires (i) transmission and scheduling solutions that process best effort and QoS sensitive traffic

in different ways, as reflected in both the IETF DiffServ and IntServ architectures; and (ii) call admission control for applications with strict QoS requirements, reflected in the controlled load and guaranteed service class proposed for the IntServ architecture. The question of today's research and development is how to implement these new functions through minimal changes in the architecture of the present Internet.

In the last years a variety of admission control schemes based on per-hop or end-to-end measurements has been published to provide admission control for delay and loss sensitive traffic with very little or no support at the routers. Measurement based admission control (MBAC) schemes base the acceptance decision on per-hop real-time measurements of the aggregate traffic intensities [1]. Endpoint admission control (EMBAC) schemes decrease the required router support even further, involving only the end systems in the admission control process [2–5]. The idea behind these schemes is to probe the transmission path from the sender to the receiver to experience the congestion level in the network and accept new streams only if the level of congestion is acceptable.

Most of these solutions, however, suffer from limited granularity, namely, the QoS guarantees (packet loss, delay and delay jitter) within a service class are the same for all streams [1, 3]. One way of quality differentiation is to define application specific service classes, but the management of a large number of classes would increase the complexity of the router operations. Instead, we believe that quality differentiation has to be achieved within one service class by using application specific traffic control functions at the end nodes.

In this paper we investigate how source shaping combined with forward error correction (FEC) can provide quality differentiation. Both of these functions exploit the streams' end-to-end delay limits looser than the one provided by the service class. Source shaping changes the traffic characteristics in a way that the expected packet loss probability of the stream decreases and the loss distribution becomes more even. FEC, in addition, recovers lost packets based on error coding, and consequently achieves lower perceived packet loss probability than the one ensured by the service class. The goal is then to share the delay available at the end-node – the difference between the acceptable end-to-end delay of the stream and the delay introduced by the network – between shaping and FEC such that the experienced information loss of the stream is minimized. The efficiency of source shaping and FEC for decreasing the probability of packet loss due to buffer overflow has been subject of extensive research, but the combined use of the two functions has not been investigated before.

In section 2 we discuss buffering strategies, explain the basic characteristics of source shaping and FEC and overview related work; in section 3 the combined use of shaping and FEC is described, in section 4 we evaluate the performance of combined source shaping and FEC and finally we conclude our work in section 5.

## 2 Network architecture and control functions

**Buffering** is the most straightforward solution to decrease packet loss probability in packet switched networks. Large buffers, however, introduce uncontrollable delay and delay variation and cause increasing burstiness on the transmission path. To utilize the

advantages of large buffers when delay sensitive traffic is transmitted in the network scheduling solutions with per stream delay and jitter control have to be applied.

On the other hand, the choice of using small buffers for transmitting delay sensitive traffic allows simple (e.g., FIFO) scheduling at the network nodes, since delay and jitter are limited by the maximum buffer sizes, and makes the network tractable as stream characteristics do not change significantly at the network nodes [6, 7]. The size of the buffers has to be selected in a way that the contention of simultaneously arriving packets is resolved (i.e., packet scale buffering is provided instead of burst scale buffering [8]), that means a buffer size in the range of $\min\{C/p, n\}$ packets, where $C$ is the transmission capacity of the link, $p$ is the maximum bitrate of the streams and $n$ is the number of input ports.

**Traffic shaping at the source node** is used to decrease the packet loss probability at the buffers inside the network by decreasing the burstiness of the traffic stream. As it is shown in e.g., [6, 9], (i) shaping even a part of the sources decreases average packet loss probability by orders of magnitude; (ii) shaped streams experience lower loss rates than unshaped ones, (iii) shaping, when applied to MPEG sources, decreases packet loss in loss sensitive I frames and (iv) makes the packet loss pattern more even as well, which in turn gives potential to FEC.

The performance of networks with source shaping and small buffers is analyzed in e.g., [6, 10, 11]. In [11] the performance of source shaping and buffering is compared for networks providing strict end-to-end delay bounds. The paper compares two solutions. In one of them, source shaping with the maximum acceptable delay is applied and nodes are equipped with small buffers, performing packet scale buffering. In the other solution the maximum acceptable delay is divided among the network nodes, thus nodes perform burst scale buffering with buffer size defined by the per node maximum delay. Nodes in this case apply jitter compensation. It is proved that source shaping outperforms buffering in the case of long transmission paths. In [10] the performance of these two solutions is compared considering video transmission, showing that shaping provides a visual quality similar to that of buffering even for short transmission paths. Source shaping in networks with small buffers is evaluated in [6] as well, proving that single buffer shapers are optimal in this case.

Proposals for source shaping algorithms address a variety of applications, like [12, 13] for streaming with known traffic pattern, [14] for lossy and [9, 15, 16] for lossless shaping for real-time traffic with unknown traffic pattern. The efficiency of shaping, in terms of decreasing the burstiness and consequently the packet loss probability depends significantly on the traffic stream itself. Considering MPEG coded video streams, shaping even with a very low, 20-40 ms delay is efficient, as it smoothes the data of large I and P frames. The efficiency increases with the introduced delay, but with decreasing marginal gain [9]. The above results motivate the use of source shaping combined with packet scale buffering for quality differentiation.

**Forward error correction** has been proposed to recover from information losses in real-time applications, where the latency introduced by retransmission schemes is not acceptable. FEC schemes increase the redundancy of the transmitted stream and recover losses based on the redundant information.

There are two main directions of FEC design to recover from packet losses due to buffer overflow. One solution, proposed by the IETF and implemented in Internet audio tools is to add a redundant copy of the original packet to one of the subsequent packets [17]. In the case of packet loss the information is regained from the redundant copy. This solution suits well interactive audio applications with low transmission rate and low delay limit. The efficiency of these schemes can be tuned by the number of redundant copies and the offset between the original packet and the redundant copy.

The other set of solutions uses block coding schemes based on, e.g., Reed-Solomon coding [18, 19]. In this case a block of packets is considered and error coding is applied for each bit position, generating a number of error correcting packets. The error correcting capability of Reed-Solomon codes with $k$ data packets and $c$ error coding packets is $c$ if data is lost, which is the case if coding is used to regenerate lost packets. FEC based on block codes introduces an overhead of $(c+k)/k$ percent. Delay is introduced at the receiver only, where the error correcting packets have to be received for packet regeneration. The decoding delay is $(c+k)t_p$, where $t_p$ is the packet interarrival time.

The error correcting capability of both classes of solutions increases with introduced decoding delay and overhead, with decreasing marginal gain.

The efficiency of FEC for correcting packet losses due to buffer overflow, however, is questionable due to the uneven distribution of packet losses and the additional load that FEC introduces in the network. Results considering different FEC schemes and based on analytical and simulation studies [17–19] show that the overall use of FEC does not always improve transmission quality, but FEC supports quality differentiation if only a part of the streams, requiring stringent QoS guarantees, applies error coding.

The above results indicate that both source shaping and FEC can be used for quality differentiation. The efficiency of the two functions is proportional to the introduced delay but with decreasing marginal gain. Consequently, combining shaping and FEC, by sharing the available end-node delay, a better performance may be achieved than by the use of only one of them.

## 3 Combined Source Shaping and FEC

In this work we propose the combined use of FEC and source shaping to support delay and loss sensitive transmission. If both functions are used at the end-nodes, the available end-node delay has to be split such that both functions can work efficiently. In addition, the two functions are not independent, as source shaping, by smoothing the packet losses in the stream, improves the packet loss correcting capability of FEC. It has to be noted that shaping achieves performance improvements without increasing the resource requirements of the streams, while FEC may introduce significant overhead. Thus, for some networking scenarios, FEC can prove to be an expensive control solution.

To evaluate the performance of combined source shaping and FEC we consider the following networking scenario.

We assume that the service class for loss and delay sensitive transmission uses dedicated buffer and link transmission capacities, and the applied call admission control together with FIFO scheduling at the routers provides the same bound on the packet

loss probability for all streams. We also assume that only small buffers are applied at the network nodes, providing buffering for simultaneously arriving packets only.

This system architecture thus provides the same, strict upper bound on the network delay and the same, stochastic upper bound on the average packet loss rate for all streams. Sources can then utilize the available end-node delay – the difference between their maximum acceptable end-to-end delay and the delay introduced by the network – to decrease the packet loss rate of the stream, using source shaping and FEC. Note, that packet loss happens due to buffer overflow only. All packets arrive within the defined delay limit to the destination due to the use of limited buffers at the network nodes and at the source shaper.

Given the stream specific end-node delay $D$, which is divided between shaping and FEC as $D = D_{sh} + D_{FEC}$, the parameters of the FEC and the shaper are calculated based on $c/k$, the required FEC redundancy and $m$, the mean transmission rate, including the redundant packets.

In the case of video transmission FEC blocks that are entirely within a video frame do not introduce any delay, since all packets of the frame have to be received to regenerate a picture, the ones that spread over more than one frame however do. The delay introduced by these blocks is the time between the arrival of the last packet from the frame where the FEC block started and the arrival of the last packet from the FEC block. Based on the delay assigned to FEC the maximum FEC block length $k+c$ is defined by $(k+c)/m < D_{FEC}$.

The shaper rate is adjusted as described in [9]. When frame $i$ arrives to the shaper, its size is added to the amount of data in the buffer $b(t) = b'(t) + f_i(t)$, where $b'(t)$ denotes the buffer occupancy before the frame arrival and $f_i(t)$ the size of the arriving frame. The shaper rate is then set to ensure that all data leave the buffer within the specified delay $D_{sh}$, thus $r(t) = b(t)/D_{sh}$. To avoid delay bound violation for frames stored in the buffer, the shaper rate can be decreased only if the buffer was empty before the arrival of the new frame.

Based on the aboves, the combined shaping and FEC algorithm works as follows. Frames generated by the source coder are put into the shaper buffer, the redundant packets according to the FEC scheme used are added and the shaper rate is adjusted. If the shaper rate during an FEC block transmission is lower than the average rate $m$, the FEC block is shortened by inserting an error correcting packet before schedule, to avoid the violation of the maximum FEC decoding delay $D_{FEC}$.

## 4 Performance Evaluation

In this section we evaluate how FEC combined with source shaping supports the transmission of delay and loss sensitive video streams.

The presented results are based on simulation. The simulated network model is shown in figure 1. The system includes traffic sources, channel coders doing FEC, source shapers and a multiplexing node with a single output link, modeling the transmission capacity dedicated for the controlled traffic. The multiplexing node performs simple FIFO queuing. We argue that results obtained with this simple model can be

extended to the multiple node case, based on the fact that the traffic characteristics of the streams do not change as they cross nodes with small buffers [20].

For the simulations we use an MPEG-4 coded talk show trace – since MPEG coding is often used to transmit video streams nowadays – with an average rate of 540 kbps and a peak rate of 2.5 Mbps. The trace is approximately 2700 seconds, thus 67000 frames long. The frames of the MPEG trace are packetized to 188 bytes, as given for the transport stream in the MPEG-2 standard [IEC61883]. The capacity of the output link is 22.5 Mbps. The buffer at the multiplexer can store up to 10 packets, which is the ratio of the output link capacity to the peak rate of the individual streams, thus the multiplexer provides packet scale buffering. At full utilization there are approximately 38 streams competing at the multiplexer, depending on the FEC schemes used. The considered available end-node delays run from 60 ms to 120 ms, where the lower delays correspond to conversational while the higher to on-line streaming applications. The confidence interval of the presented simulation results is 5% or less at 95% confidence level.

The performance analysis investigates how the packet loss probability depends on the applied control functions and on the network load. The network load is defined as the ratio of the sum of the mean rate of the streams including FEC redundancy, and the link transmission rate.

For the sake of simplicity we use the notation CF(k,c,d) for a control function with FEC of block length of $k$ data packets and $c$ redundant packets, and shaping with a delay of $d$ ms. For example, 80 ms end-node delay and a FEC scheme with $k$=20 and $c$=2 leave 20 ms delay for shaping for the considered stream mean rate and packet size. This control function is thus denoted as CF(20,2,20).

**Combined Source Shaping and FEC** To analyze the efficiency of FEC combined with source shaping we consider a scenario where the multiplexer serves a combination of traffic streams using a variety of control functions. The FEC redundancy is 10% or 20% and the available end-node delay is 80 ms or 120 ms. 14% of the multiplexed streams do not apply any control function (CF(1,0,0)), 14%-14% of them use a FEC scheme with 10% and 20% redundancy without shaping (CF(20,2,0) and CF(20,4,0)), then the same FEC schemes are used with 80 ms (CF(20,2,20) and CF(20,4,14.5)) and 120 ms (CF(20,2,60) and CF(20,4,54.5)) end-node delays. Figure 2 shows how the average uncorrected packet loss probability depends on the network load. The results show that FEC achieves loss differentiation of 1 to 2 orders of magnitude at the considered redundancy levels. Adding shaping, the loss probabilities further decrease with 25-50%. The reason for this improvement is twofold. First, the loss probability decreases as streams get smoother, second, for shaped streams the loss distribution becomes more even, increasing the efficiency of FEC.

**Optimal Delay Allocation** As the efficiency of both shaping and FEC is proportional to the introduced delay, splitting the available end-node delay between the two functions is an optimization problem. In this part we show some simple examples how different delay allocations affect the probability of uncorrected packet loss. Figure 3 shows a scenario where the link is shared between streams having an end-node delay limit of

90 ms, split between FEC and shaping. The overhead of the streams using FEC is 10%, the block length varies from $k = 10$ to $k = 30$, introducing different coding delays. Note, that for block length $k = 30$ no delay remains left for shaping (CF(30,3,0)). The figure shows that CF(10,1,60) is outperformed by those using larger blocks, however it is hard to distinguish between the streams using CF(20,2,30) and CF(30,3,0).

Considering MPEG coded streams, the distribution of losses in an MPEG stream may have significant influence on the percepted visual quality. Losses in I frames propagate forward to the next group of pictures, up to the next I frame and backwards to the previous P frame, losses in P frames propagate forward to the next I frame. Consequently, losses in I and P frames have increased effect on the percepted visual quality. Figure 4 shows the weighted loss probabilities [21] for the same FEC schemes as figure 3. The graph shows similar characteristics as figure 3, but here the CF(20,2,30), which leaves some delay available for source shaping, achieves the lowest loss probability (by a factor of up to 2), due to the more even loss distribution. Figures 5 and 6 show a similar scenario for a delay of 120 ms. The optimal delay allocation in figures 4 and 6 is consistent in the sense that it allocates almost one frame interarrival time for shaping while the rest for FEC. It allows efficient error control, while shaping makes losses for large I frames and small B frames even. On the other hand simulations run with an end-node delay of 60 ms give CF(20,2,0) as the optimal solution, showing that the difference between the efficiency of FEC with a block length of 20 and 10 is higher than what shaping with a delay of 30 ms can compensate for.

In addition, comparing the simulation results we see that the width of the confidence intervals depends on the control scheme. The 95% confidence interval for CF(30,3,30) streams is approximately one third of that of the streams using CF(40,4,0), that is, the difference between the loss probability of the streams applying the same combination of control functions is lower. It shows that shaping makes the performance of FEC more predictable.

Simulation results with increased FEC redundancy show similar characteristics, though the difference between the packet loss probability of the streams with and without FEC is higher.

**FEC Redundancy and Shaping Delay** The use of FEC may decrease the effective load due to the introduced overhead. Figure 7 shows the number of accepted streams with an admission control limiting the loss probability without FEC at 0.1%, for an increasing ratio of streams using FEC. Two cases are compared. In one of them the end-node delay is 90 ms, a part of the streams use this delay for shaping (CF(1,0,90)), a part of them for FEC with 10% redundancy (CF(30,3,0)). In the other case the end-node delay is 120 ms, the used control functions are CF(1,0,120) and CF(40,4,0). As shown on the figure, the effective load decreases as higher ratio of streams uses FEC. Increasing the end-node delay the number of accepted streams, thus the effective load becomes higher.

Since shaping and FEC have similar effects, it is worth investigating if shaping can compensate for decreased FEC redundancy. Figure 8 shows the weighted loss probabilities in a scenario where the bandwidth is shared among sources using FEC only and sources using FEC with a lower level of redundancy combined with shaping. Compar-

ing the streams using CF(20,2,60) and those using CF(20,3,0) indicates that by shaping the FEC redundancy can be decreased from 15% to 10% to achieve the same loss probabilities.

**Sensitivity Analysis** Finally, we evaluate the sensitivity of source shaping and FEC with respect to the background traffic characteristics at the multiplexer. The sensitivity analysis is important, since these functions themselves do not give loss guarantees. Guarantees are given only by the call admission process, while applications can have some expectations how the performance improves with additional control at the end-nodes. Figure 9 shows packet loss values for different background traffic characteristics. The background traffic characteristics are changed by shaping the background streams with delays up to 120 ms, resulting in smoother background traffic at high shaping delay values. FEC controlled streams have 120 ms end-node delay in all cases. Two scenarios are compared. In one of them 25% of the multiplexed streams use CF(30,3,30), in the other scenario 25% of the multiplexed streams use CF(40,4,0). For both cases 75% of the streams give the background traffic without FEC. The network load level is constant 0.82. The figure shows the uncorrected loss probabilities for the background traffic and the FEC controlled streams. The loss probability decreases for all traffic as the shaping delay of the background traffic increases, the gap between the background traffic and the FEC controlled traffic is 1.5 to 2 orders of magnitude, increasing slightly as the background traffic gets smoother. Figure 10 shows similar scenarios at a load level of 0.87. Comparing figures 9 and 10 we see that the gain achieved by FEC and shaping slightly decreases as the network load increases, but is still higher than one order of magnitude. These results indicate that sources can have some expectations on the minimum performance improvements without information on the network load and background traffic characteristics.

## 5 Conclusion

In this paper we examined how FEC combined with source shaping can decrease the uncorrected loss probability and thus add quality differentiation capability to admission control schemes that provide the same loss and delay thresholds for all the accepted streams. As both of these functions introduce end-node delay, the question is how to divide the delay between the two functions. The presented simulation based analysis, considering MPEG coded video streams provided the following results:

- Considering multiplexed streams applying different FEC schemes and allowing different end-node delays, FEC decreases the average loss probability with 1 to 2 orders of magnitude at reasonable network loads. Using shaping in addition to FEC, the loss probability is further decreased by 25-50%.
- By splitting the available delay between source shaping and FEC one can achieve better percepted quality than by applying FEC only. The optimal sharing of delay between the two functions depends on the available delay and the efficiency of shaping and FEC for the specific stream characteristics.

- Source shaping combined with FEC can reduce the level of FEC redundancy needed to achieve a given loss probability, thus contributing to higher effective network utilization.
- The gain achieved by using FEC and shaping does not considerably depend on the background traffic characteristics, rather on the average load.
- Source shaping makes the performance improvement due to FEC and thus the quality differentiation more predictable.

The above results indicate that admission control giving identical delay and loss guarantees for all streams combined with stream dependent source shaping and FEC provides a solution for transmitting audio-visual information with diverse quality requirements without introducing stream specific control functions inside the network. The results also assist to define an algorithmic solution for selecting the optimal FEC redundancy and assigning delay to shaping and FEC, which is subject of our further research.

## References


1. L. Breslau, S. Jamin, and S. Shenker, "Comments on the performance of measurement-based admission control algorithms," in *Proc. of IEEE INFOCOM 2000*, pp. 1233–1242, March 2000.
2. G. Bianchi, A. Capone, and C. Pertioli, "Throughput analysis of end–to–end measurement-based admission control in IP," in *Proc. of IEEE INFOCOM 2000*, pp. 1461–1470, March 2000.
3. L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: Architectural issues and performance," in *Proc. of ACM SIGCOMM*, pp. 57–69, August 2000.
4. V. Fodor (née Elek), G. Karlsson, and R. Ronngren, "Admission control based on end-to-end measurements," in *Proc. of IEEE INFOCOM 2000*, pp. 623–630, March 2000.
5. R. B. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," in *Proc. of the 16th International Teletraffic Congress*, pp. 941–952, June 1999.
6. M. Reisslein, K. Ross, and S. Rajagopal, "Guaranteeing statistical QoS to regulated traffic: The single node case," in *IEEE Infocom'99*, pp. 1061–1072, 1999.
7. J. Roberts, "Traffic theory and the internet," *IEEE Communications Magazine*, pp. 94–99, January 2001.
8. *Broadband Network Teletraffic, Final Report of Action COST 242*. Springer, 1996.
9. G. Dán and V. Fodor, "On the efficiency of shaping live video streams," in *Proc. of SPECTS'02*, pp. 49–56, July 2002.
10. G. Dán and V. Fodor, "Comparison of shaping and buffering for video transmission," in *Proc. of NTS 16*, August 2002.
11. T. Wu and E. Knightly, "Buffering vs. smoothing for end-to-end QoS: Fundamental issues and comparison," in *Proc. of IEEE Performance'99*, August 1999.
12. E. Knightly and H. Zhang, "D-BIND: an accurate traffic model for providing QoS guarantees to VBR traffic," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 219–231, April 1997.
13. J.-Y. Le Boudec and O. Verscheure, "Optimal smoothing for guaranteed service," *IEEE Transactions on Networking*, vol. 8, December 2000.

14. H. Zhang and E. Knightly, "RED-VBR: A renegotiation-based approach to support delay-sensitive VBR video," *ACM Multimedia Systems Journal*, vol. 5, pp. 164–176, May 1997.
15. S. S. Lam, S. Chow, and D. K. Y. Yau, "An algorithm for lossless smoothing of MPEG video," in *Proc. of ACM SIGCOMM*, pp. 281–293, 1994.
16. J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video," in *Proc. of NOSSDAV*, pp. 249–257, May 1997.
17. P. Dube and E. Altman, "Utility analysis of simple FEC schemes for VoIP," in *Proc. of Networking 2002*, May 2002.
18. I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high speed networks," *IEEE Transactions on Information Theory*, vol. IT-39, pp. 98–108, January 1993.
19. K. Kawahara, K. Kumazoe, T. Takine, and Y. Oie, "Forward error correction in ATM networks: An analysis of cell loss distribution in a block," in *Proc. of IEEE INFOCOM 1994*, pp. 1150–1159, June 1994.
20. M. Reisslein, K. Ross, and S. Rajagopal, "Guaranteeing statistical QoS to regulated traffic: The multiple node case," in *Proc. of IEEE Decision & Control'98*, pp. 531–538, 1998.
21. K. Mayer-Patel, L. Le, and G. Carle, "An MPEG performance model and its application to adaptive forward error correction," *ACM Multimedia*, December 2002.

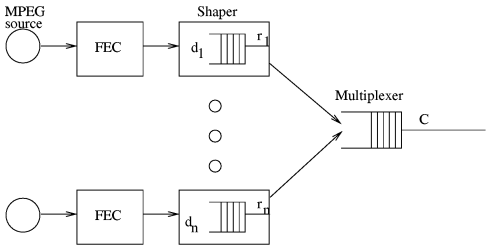


**Fig. 1.** The considered network model. MPEG source, FEC, delay limited shaping at the end nodes and packet scale buffering inside the network.

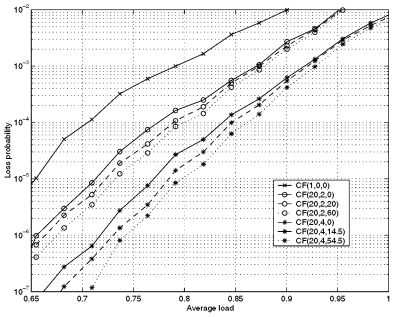


**Fig. 2.** Loss probability with FEC and source shaping, for FEC block size $k$=20, redundancies $c = 2,4$ and end-node delays 80-120 ms

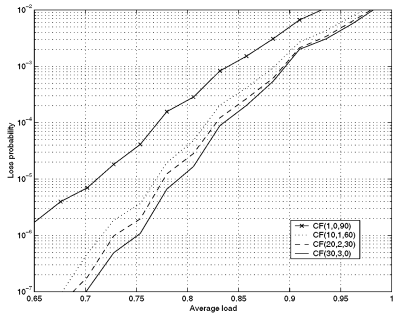


**Fig. 3.** Loss probability with FEC and source shaping for different FEC block sizes and 90 ms end-node delay

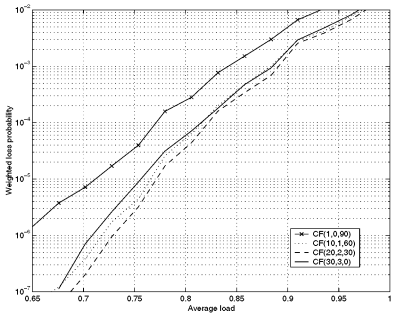


**Fig. 4.** Weighted loss probability with FEC and source shaping for different FEC block sizes and 90 ms end-node delay

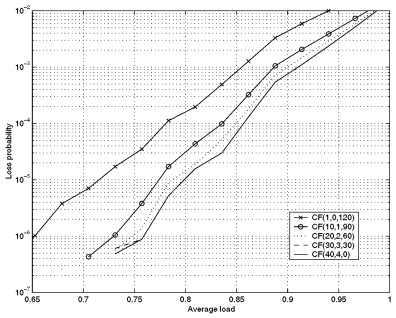


**Fig. 5.** Loss probability with FEC and source shaping for different FEC block sizes and 120 ms end-node delay

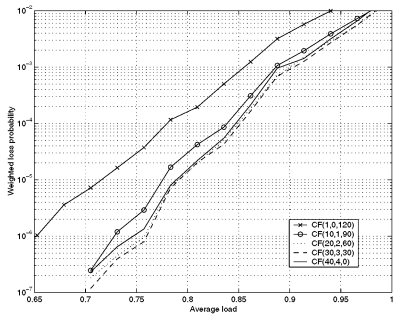


**Fig. 6.** Weighted loss probability with FEC and source shaping for different FEC block sizes and 120 ms end-node delay

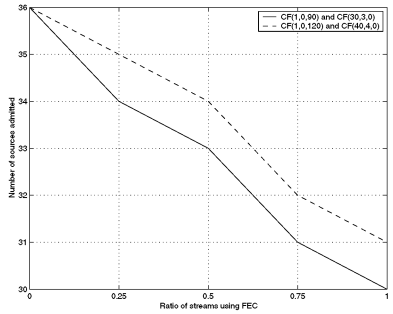


**Fig. 7.** Number of admitted sources vs the ratio of streams using FEC for different end-node delays

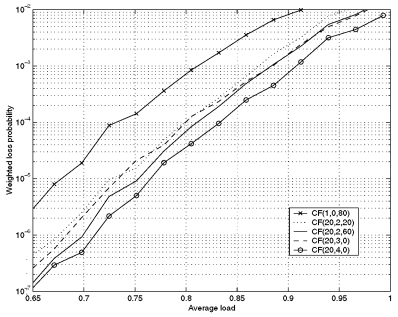


**Fig. 8.** Weighted loss probability for different combinations of redundancy and shaping delay

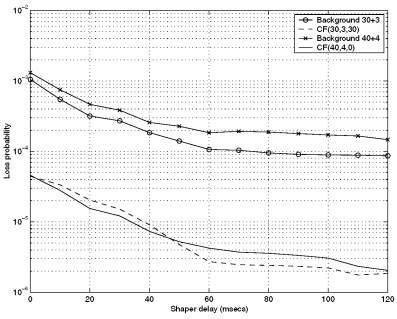


**Fig. 9.** Loss probability vs background traffic shaping delay for different FEC schemes, $\rho = 0.82$

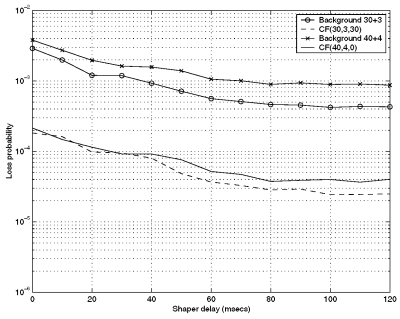


**Fig. 10.** Loss probability vs background traffic shaping delay for different FEC schemes, $\rho = 0.87$

# Analysis of the Packet Loss Process for Multimedia Traffic

Gy. Dán and V. Fodor and G. Karlsson
*Submitted to the 3rd IFIP-TC6 Networking Conference (Networking 2004),*
May 2004, Athens, Greece

# Analysis of the Packet Loss Process for Multimedia Traffic *

György Dán, Viktória Fodor and Gunnar Karlsson

KTH, Royal Institute of Technology,
Department of Microelectronics and Information Technology
{gyuri,viktoria,gk}@imit.kth.se

**Abstract.** In the case of multimedia traffic, like VBR video, the average loss probability is not sufficient to investigate the effects of loss on perceived visual quality, but it is difficult to analytically model the queuing behavior for such traffic. It has been shown that in the case of real-time communications, for which small buffers are used for delay reasons, short range dependence dominates the loss process and so the Markov-modulated Poisson process (MMPP) might be a reasonable source model. In this paper we present an exact mathematical model for the loss process of an MMPP+M/D/1/K queue; we validate it via simulations and compare it to other mathematical models, like the MMPP+M/M/1/K and the Gilbert model, and to simulations with real MPEG-4 video traces. We conclude that the other models give accurate results only in a small set of network scenarios, while our model can capture the loss process of VBR video sufficiently well in most cases. This makes it possible to analyze the effects of forward error correction on transmission quality in various network scenarios.

## 1 Introduction

In the case of flow-type multimedia communications, as opposed to elastic traffic, the average packet loss is not the only measure of interest. The burstiness of the loss process, the number of losses in a block of packets, has a great impact both on the user perceived visual quality and on the possible ways of improving it, for example by forward error correction or receiver-based error concealment.

In this paper we present a model to analyze the packet loss process of a bursty source, for example VBR video, multiplexed with background traffic in a single multiplexer with a finite queue and constant packet sizes. We model the bursty source by an L-state Markov-modulated Poisson process (MMPP) while the background traffic is governed by a Poisson process. We validate our model via simulations and compare the results to simulations made with real video traces.

* Technical report, TRITA-IMIT-LCN R 04:01 ISSN 1651-7717 ISRN KTH/IMIT/LCN/R 04/01–SE

It is well known that compressed multimedia, like VBR video, exhibits a self-similar nature [1]. Yoshihara et al. use the superposition of 2-state IPPs to model self-similar traffic in [2] and compare the loss probability of the resulting MMPP/D/1/K queue with simulations. They found that the approximation works well under heavy load conditions and gives an upper bound on the packet loss probabilities. Ryu and Elwalid [3] showed that short term correlations have dominant influence on the network performance under realistic scenarios of buffer sizes for real-time traffic. Thus the MMPP may be a practical model to derive approximate results for the queuing behavior of LRD traffic such as real-time VBR video, especially in the case of small buffer sizes. Recently Cao et al. [4] showed that the traffic generated by a large number of sources tends to Poisson as the load increases due to statistical multiplexing and hence justifying the Poisson model for the background traffic.

The paper is organized as follows. Section 2 gives an overview of the previous work on the modeling of the loss process of a single server queue. In Section 3 we describe our model to calculate the loss probability in a block of packets. In Section 4.1 we validate our model by simulations with MMPP and real VBR traffic sources and compare our results to those obtained for exponential service times. In Section 4.2 we use our model to evaluate the FEC performance for VBR video sources and finally in Section 5 we conclude our work.

## 2 Related Work

In [5], Cidon et al. present an exact analysis of the packet loss process in an M/M/1/K queue, that is the probability of losing $j$ packets in a block of $n$ packets, and show that the distribution of losses may be bursty compared to the assumption of independence. They also consider a discrete time system fed with a Bernoulli arrival process describing the behavior of an ATM multiplexer. In [6], Gurewitz et al. present explicit expressions for the above quantities of interest for the M/M/1/K queue. In [7] the multidimensional generating function of the probability of $j$ losses in a block of $n$ packets is obtained and an easy-to-calculate asymptotic result is given under the condition that $n \leq K + j + 1$.

The above models consider exponentially distributed service times. Most multimedia standards however use constant packet sizes for transmission, and real-time multimedia streams complying to the same standard are likely to share the same service class in the network (like for example the diffserv expedited forwarding). So the packet size distributions will differ significantly from exponential and tend to be rather deterministic. In this case the M/M/1/K queuing model overestimates the loss probability and has a different loss process.

Models with general and deterministic service times have been proposed for calculating various measures of queuing performance. In [8], Ait-Hellal et al. present an

asymptotic result for a system where the service times and the interarrival times are stationary ergodic, in particular they show that if the block lengths $k$ and redundancy $j$ is large enough, then the frame loss probabilities can be made arbitrarily small. In [9] the conditional loss probability (CLP) is derived for the N*IPP/D/1/K queue and it is shown that the CLP can be orders of magnitude higher than the loss probability.

The performance of an MMPP/G/1/K queue was evaluated in [10] considering the superposition of multimedia and data traffic at a single server queue, and the corresponding delay distribution was given. The waiting time and queue length distribution of the N/G/1/K queue (N stands for the Neuts process) was derived in [11] including the MMPP/G/1/K queue as a special case. Even though the waiting time and queue length distribution of the MMPP/G/1/K queue has been derived, a more thorough analysis of the packet loss process has not yet been done.

Another approach to calculate the probability of $j$ losses in a block of $n$ packets is to use a channel model, for example the Gilbert model, to describe the correlation of losses and derive the probability of losses in a block. This approach is followed in [12] to evaluate the efficiency of FEC for VBR video transmission. Though it is easy to calculate the block loss probabilities using this method, one has to choose the parameters of the loss model. A direct correlation between the source characteristics and the corresponding loss process can therefore not be investigated.

## 3 Model description

We consider a system with fixed size packets having transmission time $D$. Packets arrive to the system from two sources, a Markov-modulated Poisson process (MMPP) and a Poisson process, representing the tagged source and the background traffic respectively. The packets are stored in a buffer that can host up to K packets, and are served according to a FIFO policy. Every $n$ consecutive packets from the tagged source form a block, and we are interested in the probability distribution of the number of lost packets in a block in the steady state of the system. Throughout the calculations we use notations similar the those in [5].

We assume that the sources feeding the system are independent. The MMPP is described by the infinitesimal generator $Q$ with elements $r_{ij}$ and the arrival rate matrix $\Lambda = diag\{\lambda_1, \ldots, \lambda_L\}$, where $\lambda_i$ is the average arrival rate while the underlying Markov chain is in state $i$. The Poisson process modeling the background traffic has average arrival rate $\lambda$. The superposition of the two sources can be described by a single MMPP with arrival rate matrix $\hat{\Lambda} = \Lambda \oplus \lambda = \Lambda + \lambda I = diag\{\hat{\lambda_1}, \ldots, \hat{\lambda_L}\}$, and infinitesimal generator $\hat{Q} = Q$, where $\oplus$ is the Kronecker sum. Packets arriving from both sources have the same length, and thus the same transmission time.

Our purpose is to calculate the probability $P(j,n), n \geq 1, 0 \leq j \leq n$ of $j$ losses in a block of $n$ packets. We define the probability $P^a_{x,l}(j,n), 0 \leq x \leq KD, l = 1 \ldots L, n \geq$

$1, 0 \leq j \leq n$ as the probability of $j$ losses in a block of $n$ packets, given that the remaining workload in the system is $x$ just before the arrival of the first packet in the block and the first packet of the block is generated in state $l$ of the MMPP. As the first packet in the block is arbitrary,

$$P(j,n) = \sum_{l=1}^{L} \int_{x=0}^{KD} V(x,l) P^a_{x,l}(j,n). \tag{1}$$

An approximation for $V(x,l)$, the workload distribution of the steady state queue as seen by an arriving packet can be given by the steady state distribution of the MMPP/D/1/K queue as outlined in Appendix A.

The probabilities $P^a_{x,l}(j,n)$ can be derived according to the following recursion. The recursion is initiated for n = 1 with the following relations

$$P^a_{x,l}(j,n) = \begin{cases} 1 & j = 0 \\ 0 & j \geq 1 \end{cases} \qquad x \leq (K-1)D,$$

$$P^a_{x,l}(j,n) = \begin{cases} 0 & j = 0, j \geq 2 \\ 1 & j = 1 \end{cases} \qquad (K-1)D < x. \tag{2}$$

Using the notation $p_m = \frac{\lambda_m}{\lambda_m + \lambda}$ and $\overline{p}_m = \frac{\lambda}{\lambda_m + \lambda}$, for $n \geq 2$ the following equations hold.

$$P^a_{x,l}(j,n) = \sum_{m=1}^{L} \int_0^{x+D} f_{lm}(t) \{ p_m P^a_{x+D-t,m}(j,n-1) + \overline{p}_m P^s_{x+D-t,m}(j,n-1) \} dt + \int_{x+D}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n-1) + \overline{p}_m P^s_{0,m}(j,n-1) \} dt$$

for $0 \leq x \leq (K-1)D$ and

$$P^a_{x,l}(j,n) = \sum_{m=1}^{L} \int_0^{x} f_{lm}(t) \{ p_m P^a_{x-t,m}(j-1,n-1) + \overline{p}_m P^s_{x-t,m}(j-1,n-1) \} dt + \int_{x}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j-1,n-1) + \overline{p}_m P^s_{0,m}(j-1,n-1) \} dt \tag{3}$$

for $(K-1)D < x$. $P^s_{x,l}(j,n)$ is given by

$$P^s_{x,l}(j,n) = \sum_{m=1}^{L} \int_0^{x+D} f_{lm}(t) \{ p_m P^a_{x+D-t,m}(j,n) + \overline{p}_m P^s_{x+D-t,m}(j,n) \} dt + \int_{x+D}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n) + \overline{p}_m P^s_{0,m}(j,n) \} dt \qquad 0 \leq x \leq (K-1)D$$

for $0 \leq x \leq (K-1)D$ and

$$P^s_{x,l}(j,n) = \sum_{m=1}^{L} \int_0^{x} f_{lm}(t) \{ p_m P^a_{x-t,m}(j,n) + \overline{p}_m P^s_{x-t,m}(j,n) \} dt + \int_{x}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n) + \overline{p}_m P^s_{0,m}(j,n) \} dt \tag{4}$$

for $(K-1)D < x$. The probability $P^s_{x,l}(j,n), 0 \le x \le KD, l = 1 \ldots L, n \ge 1, 0 \le j \le n$ is the probability of $j$ losses in a block of $n$ packets, given that the remaining workload in the system is $x$ just before the arrival of a packet from the background traffic and the MMPP is in state $l$. In (3) and (4) $f_{lm}(t)$ denotes the interarrival-time distribution of the joint arrival process and is given in Appendix B.

### 3.1 Numerical Evaluation

The above set of an infinite number of integral equations can be solved using numerical integration, so that the infinite number of integral equations is substituted by a finite number of linear equations. More precisely the finite integrals in equations (3) and (4) are calculated numerically while the infinite integrals - as the integrand only depends on $t$ in $f_{lm}(t)$ - can be evaluated analytically as shown in Appendix B (31). We introduce $\Delta$ the step size for the numerical integration such that $D = N\Delta$, and so instead of equations (2-4) we can write

$$P^a_{i,l}(j,n) = \begin{cases} 1 & j = 0 \\ 0 & j \ge 1 \end{cases} \quad i \le (K-1)N,$$
$$P^a_{i,l}(j,n) = \begin{cases} 0 & j = 0, j \ge 2 \\ 1 & j = 1 \end{cases} \quad (K-1)N < x. \tag{5}$$

For $n \ge 2$ the following recursive equations hold.

$$P^a_{i,l}(j,n) = \sum_{m=1}^{L} \sum_{\tau=0}^{i+N} f_{lm}(\tau\Delta) c_\tau^{i+N} \{ \frac{\lambda_m}{\lambda_m + \lambda} P^a_{i+N-\tau,m}(j,n-1) + \overline{p}_m P^s_{i+N-\tau,m}(j,n-1) \} + \int_{i\Delta+D}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n-1) + \overline{p}_m P^s_{0,m}(j,n-1) \} dt \tag{6}$$

for $0 \le i \le (K-1)N$ and

$$P^a_{i,l}(j,n) = \sum_{m=1}^{L} \sum_{\tau=0}^{i} f_{lm}(\tau\Delta) c_\tau^{i} \{ \frac{\lambda_m}{\lambda_m + \lambda} P^a_{i-\tau,m}(j-1,n-1) + \overline{p}_m P^s_{i-\tau,m}(j-1,n-1) \} + \int_{i\Delta}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j-1,n-1) + \overline{p}_m P^s_{0,m}(j-1,n-1) \} dt \tag{7}$$

for $(K-1)N < i$. $P^s_{i,l}(j,n)$ is given by

$$P^s_{i,l}(j,n) = \sum_{m=1}^{L} \sum_{\tau=0}^{i+N} f_{lm}(\tau\Delta) c_\tau^{i+N} \{ \frac{\lambda_m}{\lambda_m + \lambda} P^a_{i+N-\tau,m}(j,n) + \overline{p}_m P^s_{i+N-\tau,m}(j,n) \} + \int_{i\Delta+D}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n) + \overline{p}_m P^s_{0,m}(j,n) \} dt \tag{8}$$

for $0 \leq i \leq (K-1)N$ and

$$P^s_{i,l}(j,n) = \sum_{m=1}^{L} \sum_{\tau=0}^{i} f_{lm}(\tau\Delta) c^i_\tau \{ \frac{\lambda_m}{\lambda_m + \lambda} P^a_{i-\tau,m}(j,n) + \overline{p}_m P^s_{i-\tau,m}(j,n) \} + \int_{i\Delta}^{\infty} f_{lm}(t) \{ p_m P^a_{0,m}(j,n) + \overline{p}_m P^s_{0,m}(j,n) \} dt \quad (9)$$

for $(K-1)N < i$, where the coefficient $c^i_\tau$ is the $\tau^{th}$ weighting coefficient in the $i$ degree numerical integration. Through carefully choosing the numerical method by increasing $N$, the error induced by the numerical integration decreases at least proportional to $(\frac{1}{N})^5$.

The procedure of computing $P^a_{i,l}(j,n)$ is as follows. First we calculate $P^a_{i,l}(j,1), i = 0 \ldots KN$ from the initial conditions (5). Then in iteration $k$ we first calculate $P^s_{i,l}(j,k), k = 1 \ldots n-1$ using equations (8) and (9) and the probabilities $P^a_{i,l}(j,k)$, which have been calculated during iteration $k-1$. Then we calculate $P^a_{i,l}(j,k+1)$ using equations (6) and (7).

## 4 Performance Analysis

In this section we show results obtained with the model described in Section 3 for a 3-state MMPP. First we compare our model with simulation results and other mathematical models of loss processes. Then we use our model to assess the performance of FEC in a scenario where a multimedia stream (modeled by a 3-state MMPP) is multiplexed with background traffic (the superposition of a large number of possibly multimedia streams modeled by Poisson arrivals) in a multiplexer with a finite queue. We compare the results to the widely used Gilbert model and simulations with real MPEG traces. The simulations were performed in ns-2.

### 4.1 Model Evaluation

In the following section we compare our model to analytical models and two sets of simulations. We use the simulations to verify the accuracy of our model. In the first set of simulations we simulate an MMPP+M/D/1/K system. Both in our analytical model and in the simulation the MMPP has 3 states, with arrival intensities $\lambda_1 = 116/s, \lambda_2 = 274/s, \lambda_3 = 931/s$ and transition rates $r_{12} = 0.12594, r_{21} = 0.25, r_{23} = 1.97, r_{32} = 2$. The service time in the considered scenarios is 0.5 ms, 0.25 ms, 0.15 ms, 66.84 $\mu$s, 33.42 $\mu$s, and 9.7 $\mu$s. Considering a packet length of 188 bytes, the average bitrate of the MMPP is 540 kbps and the link speeds are 3 Mbps, 6 Mbps, 10 Mbps, 22.5 Mbps, 45 Mbps and 155 Mbps accordingly. For the sake of simplicity we will refer to the link speeds in the following. In the models as well as in the simulations we use the background process to change the average load $\rho$.

To check the accuracy of estimating the loss process of VBR video with an MMPP we have performed simulations with an MPEG-4 encoded video traffic trace multiplexed with a Poisson arrival process at a multiplexer with a finite queue. The MPEG-4 trace is approximately 2700 seconds, thus 67000 frames long, and was used to set the parameters of the 3-state MMPP. The frames of the MPEG stream are packetized to 188 bytes, as given for the transport stream in the MPEG-2 standard [13]. The link capacity in the different scenarios is 3 Mbps, 6 Mbps, 10 Mbps, 22.5 Mbps, 45 Mbps and 155 Mbps. The simulation time in both simulations was between 20 and 40 thousand seconds. The queueing delay is set to 0.5 ms in all cases, resulting in queue lengths from 2 to 60 packets depending on the link speed.

Besides the simulations we compare our model with deterministic service times to a finite queue fed by sources with the same characteristics as above but with exponentially distributed service times with mean values given above. The model for the resulting MMPP+M/M/1/K queue is a generalization of the multiple stream model in [5].

A widely used channel model for the evaluation of multimedia transmission schemes in error prone environments is the Gilbert-model [14], which is a two-state time discrete Markov model. State 0 corresponds to the reception of a packet, while state 1 to the loss of a packet. The distribution of the length of the error bursts $B$ is described by the transition rates $p$ and $q$ $(p+q \leq 1)$ as

$$P\{B=i\} = (1-q)^{i-1}q. \tag{10}$$

If $p+q=1$ then the model is called Bernoulli. The loss probability in the Gilbert model is given by $P_{loss} = \frac{p}{p+q}$. The parameters of the Gilbert model can be tuned in different ways, [14] uses a method based on the loss burst distribution. We use the following method based on the loss probabilities in a block of packets $P(j,n)$

$$p = q(1-P(0,1))/P(0,1), \qquad q = \sum_{i-1}^{n} P(i,i)/(\sum_{i=1}^{n} iP(i,i)). \tag{11}$$

In order to compare our model and the Gilbert model we derive formulas to calculate the expected burst length and the probabilities $P(j,n)$ of the loss process based on the Gilbert model. The expected loss burst length can be calculated based on the probabilities $P(j,n)$

$$E[B] = \sum_{n=1}^{\infty} P(B \geq n) = \sum_{n=1}^{\infty} P(n,n), \tag{12}$$

and using the Gilbert model

$$E[B_G] = \sum_{i=1}^{\infty} i(1-q)^{i-1}q = \frac{1}{q}. \tag{13}$$

We use the Gilbert model to calculate the probabilities $P(j,n)$ and compare them to the probabilities given by the MMPP+M/D/1/K model in order to verify if the Gilbert model can assess the loss process of the queue. The probabilities $P(j,n)$ for the Gilbert model can be calculated in a similar way to the one shown in Section 3 and are given by the following equation:

$$P(j,n) = \frac{q}{p+q}P_0(j,n) + \frac{p}{p+q}P_1(j,n), \tag{14}$$

where the probabilities $P_i(j,n)$ are the conditional probabilities of loosing $j$ packets in a block of $n$ packets given that the first packet arrives in state $i, (i \in \{0,1\})$, and thus will or will not be lost. $P_i(j,n)$ are given by the following recursive equations:

$$\begin{aligned} P_0(j,n) &= (1-p)P_0(j,n-1) + pP_1(j,n-1) \\ P_1(j,n) &= qP_0(j-1,n-1) + (1-q)P_1(j-1,n-1). \end{aligned} \tag{15}$$

Figures 1, 2 and 3 show the probability of loosing $j$ packets in a block of 22 packets ($n = 22$) at three different load levels on a 3 Mbps and a 22.5 Mbps link. The figures show that the results of the model with exponentially distributed service times differ significantly. Furthermore the Gilbert model can only capture the loss process at very low loss levels and a high level of statistical multiplexing, and for small block lengths but not for small values of $j$. The figures also show that the loss process in the multiplexer fed by real traces is rather similar to the one given by our model using an MMPP.

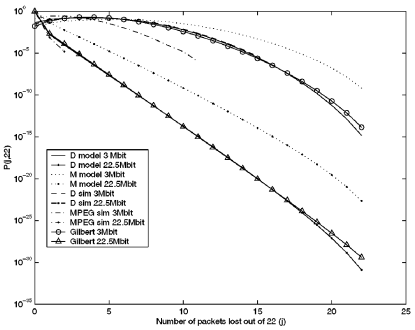


**Fig. 1.** Probability of $j$ packets lost out of 22 on a 3 Mbps and a 22.5 Mpbs link at ρ=0.59

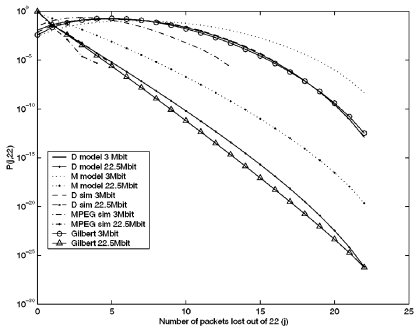


**Fig. 2.** Probability of $j$ packets lost out of 22 on a 3 Mbps and a 22.5 Mpbs link at ρ=0.75

Figure 4 shows the average loss run length as a function of the average load for two different link capacities. The figures shows that the Gilbert model can capture the burst

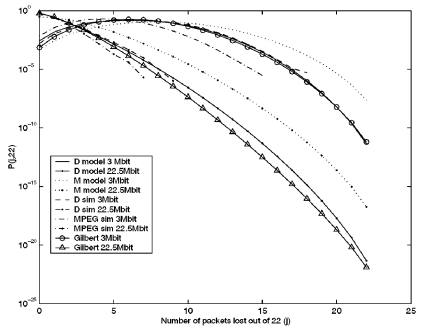


**Fig. 3.** Probability of $j$ packets lost out of 22 on a 3 Mbps and a 22.5 Mpbs link at ρ=0.91

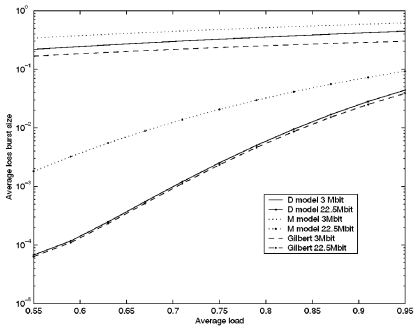


**Fig. 4.** Expected burst length on a 3 Mbps and a 22.5 Mbps link vs the average load.

size distribution if the level of statistical multiplexing is high. The results given by the MMPP+M/M/1/K model differ significantly from the loss process of the simulations.

### 4.2 FEC performance

Forward error correction (FEC) has been proposed to recover from information losses in real-time applications, where the latency introduced by retransmission schemes is not acceptable. FEC increases the redundancy of the transmitted stream and recovers losses based on the redundant information. There are two main directions of FEC design to recover from packet losses. One solution, proposed by the IETF and implemented in Internet audio tools is to add a redundant copy of the original packet to one of the subsequent packets[15]. The other set of solutions, considered in this paper, use block coding schemes based on algebraic coding, e.g. Reed-Solomon coding [16]. The error correcting capability of RS codes with $k$ data packets and $c$ redundant packets is $c$ if data is lost, which is the case if coding is used to recover packet losses.

The performance of an FEC scheme is largely affected by the distribution of the loss process, e.g. the probability of loosing more than $c$ packets in a block of $k+c$ packets. Given the probabilities $P(j,n)$ the uncorrected loss probability for an RS(k,c+k) scheme can be calculated as

$$P^*_{loss} = \frac{1}{c+k} \sum_{j=c+1}^{c+k} jP(j,c+k). \tag{16}$$

Figures 5, 6, 7 and 8 show the average loss probability, and the uncorrected loss probabilities for RS(10,11) and RS(20,22) codes for the MMPP+M/M/1/K, and the MMPP+M/D/1/K models and the simulation of the MMPP+M/D/1/K system for different link speeds. Both RS codes used introduce an overhead of 10%, so that the

sources, if they decide to use FEC and want to keep their bitrate unchanged, have to decrease the amount of useful information sent by 10%. As compensation they expect lower probability of incorrigible loss. In the case of VBR compressed video the decrease of the loss probability can compensate for a certain reduction in the source rate, and thus one can achieve better perceived visual quality.

We have chosen two different block lengths to illustrate the effect of the block length on the error correcting capability of the RS codes. The figures show that our model slightly underestimates the probability of uncorrected packet loss, which is due to the approximation of the workload distribution, but it models the packet loss distribution accurately under all load and loss conditions. Comparing the losses of streams using RS(10,11) and RS(20,22) it can be seen that although in general increasing the block size (c+k) results in much more efficient block codes, at high loss rates (e.g. higher than the overhead introduced by the FEC code) the contrary is true, shorter block codes are more efficient.

Figures 9, 10, 11 and 12 show the average loss probability, and the probability of uncorrected loss for RS(10,11) and RS(20,22) codes for different link speeds for the MMPP+M/D/1/K model, simulations with a real MPEG trace and the Gilbert model. Comparing the loss probabilities of the simulations with real MPEG traces and the results of the model we can see that as long as the multimedia source has a significant influence on the queuing behavior, the model overestimates the loss probabilities. However the gain of using different FEC schemes is similar both for the model and the simulation with the real traces. In figure 11 the tagged source achieves the loss probability of $10^{-4}$ at $\rho = 0.77$ by using RS(22,20) according to our model. Without FEC the same loss level is experienced for $\rho = 0.6$. From a different perspective, by using RS(22,20) the source can decrease the probability of uncorrected loss by 1 to 2 orders of magnitude at reasonable loss levels. In this scenario the perceived visual quality improves, if the decrease of two orders of magnitude in the loss probability compensates for 10% decrease of the effective bandwith. Similar behavior can be seen in the case of the MPEG stream. Thus, the model captures the loss process accurately.

Comparing the Gilbert model to the results of our model we can conclude that at low levels of statistical multiplexing and long block sizes ($n = 22$) the results of the Gilbert model are very inaccurate. It can only capture the loss process at very low loss levels and nearly independent losses (Fig. 12). It has one drawback: its parameters have to be set correctly. Our model can be used to set the parameters of the Gilbert model as described before. In other scenarios our model is the only model that can capture the loss process of a bursty source with constant packet sizes.

## 5 Conclusion

In this paper we presented a model to evaluate the probability of loosing $j$ packets in a block of $n$ packets in an MMPP+M/D/1/K queue. Via simulations we have shown that

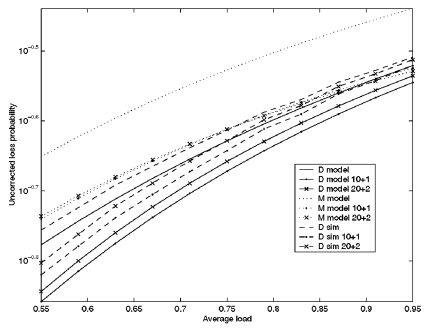


**Fig. 5.** Probability of uncorrected packet loss vs. average load on a 3 Mbps link,K=2

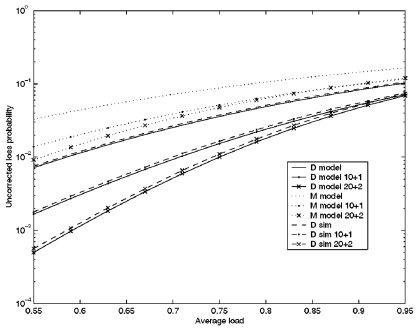


**Fig. 6.** Probability of uncorrected packet loss vs. average load on a 10 Mbps link, K=5

the model is accurate, and in the case of a high level of statistical multiplexing it can be used to model the behavior of VBR coded video streams. We have compared our model with an MMPP+M/M/1/K model and showed that the results differ significantly, both in the average packet loss, as well as in the packet loss process. Thus our model helps in getting further insight into the packet loss process of queues with deterministic service time and bursty input traffic. Using the model and simulations we have shown that the use of FEC can significantly decrease the probability of uncorrected packet loss at reasonable loss levels, and can increase perceived quality. We have compared our model with the Gilbert model, and came to the conclusion that the Gilbert model can only capture the loss process if the level of statistical multiplexing is very high, and the loss probability low. In this case the Gilbert model is easy to use to evaluate for example FEC performance. Our model is still needed to determine the parameters of the Gilbert model so that it can emulate a realistic loss process.

## 6 Appendices

## A Workload distribution

The Laplace transform of the virtual waiting time distribution of the MMPP/G/1/K queue is given in [11]. Following the arguments presented there one can derive the Laplace transform of the workload distribution

$$V(s) = \frac{1}{[\mu - \overline{\pi_0}(\hat{Q}-\hat{\Lambda})^{-1}\overline{e}]}\{\overline{\pi_0}[-s(sI-\hat{\Lambda}+\hat{Q})^{-1}(\hat{Q}-\hat{\Lambda})^{-1}$$

$$+\sum_{k=1}^{N-1} T[\hat{\Lambda}S]^{k-1}(sI+\hat{Q})S[G^*(s)]^k - [G^*(s)]^{N-1}\sum_{k=0}^{N-1}\overline{\pi_k}T[\hat{\Lambda}S]^{N-k-1}]]$$

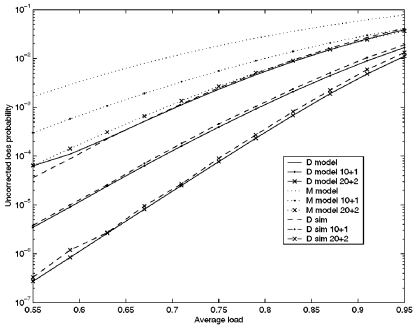


**Fig. 7.** Probability of uncorrected packet loss vs. average load on a 22.5 Mbps link,K=10

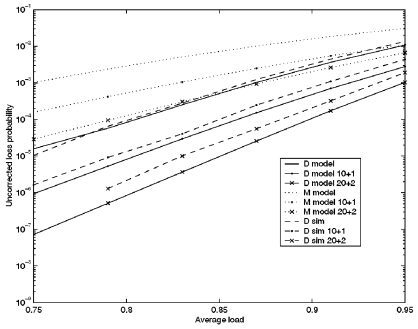


**Fig. 8.** Probability of uncorrected packet loss vs. average load on a 45 Mbps link, K=20

$$+[G^*(s)]^{N-1}\sum_{k=1}^{N-1}\overline{\pi_k}\sum_{j=N-k}^{\infty}\left[\sum_{k=0}^{j}A_kT[\hat{\Lambda}S]^{n-k}-G^*(s)T[\hat{\Lambda}S]^n\right]\},$$

where $S=(\hat{\Lambda}-sI-\hat{Q})^{-1}$, $T=(sI-\hat{\Lambda}+\hat{Q})^{-1}$, $G^*(s)$ is the Laplace transform of the service time distribution. $A_k$ is an $L\times L$ matrix whose $(l,m)$th element denotes the conditional probability of the MMPP reaching phase $m$ and having $k$ arrivals during a service time, starting from phase $l$. Instead of calculating the inverse Laplace transform of the above expression we use an approximation based on the steady state distribution of the queue length to calculate the workload distribution of the steady state system. A way to calculate the steady state queue length distribution, $\pi(i,l)$ $(0\leq i\leq K, 1\leq l\leq L)$, of an MMPP/G/1/K queue is described in [11]. The queue length distribution as seen by an arriving packet, $\Pi(i,l)$ $(0\leq i\leq K, 1\leq l\leq L)$, can be calculated from the steady state queue length distribution as

$$\Pi(i,l)=\frac{\pi(i,l)\lambda_l}{\sum_{l=1}^{L}\lambda_l\sum_{i=0}^{K}\pi(i,l)}. \tag{17}$$

Given the queue length distribution as seen by an arriving packet $\Pi(i,l)$, $0\leq i\leq K, 1\leq l\leq L$ the workload distribution $V(i,l)$ is approximated by

$$V(i,l)=\begin{cases}\Pi(0,l) & i=0\\ \Pi(\lceil i/N\rceil,l)/N & 0<i<=NK\end{cases}, \tag{18}$$

where $N$ is defined in Section 3.1.

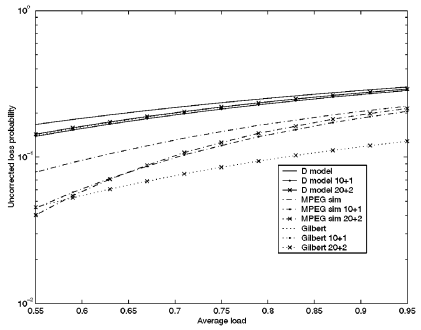


**Fig. 9.** Probability of uncorrected packet loss vs. average load on a 3 Mbps link,K=2

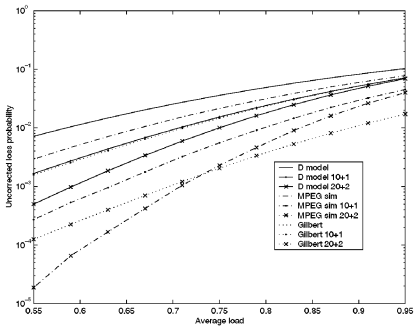


**Fig. 10.** Probability of uncorrected packet loss vs. average load on a 10 Mbps link, K=5

## B Interarrival-time distribution

The probability $f_{lm}(t)$ denotes the joint conditional probability that the time between two arrivals from the joint arrival stream is $X_k = t$ and the state of the MMPP at the moment of the arrival is $J_{k+1} = m$ given that at the time of the last arrival the MMPP was in state $J_k = l$. A straightforward way to calculate $f_{lm}(t)$ is using the equality

$$f_{lm}(t) = [e^{(\hat{Q}-\hat{\Lambda})t}\hat{\Lambda}]_{l,m}. \tag{19}$$

Instead we may calculate $f_{lm}(t)$ using the Laplace transform

$$f^*(s) = \mathcal{L}\left\{e^{(\hat{Q}-\hat{\Lambda})t}\hat{\Lambda}\right\} = [sI - \hat{Q} + \hat{\Lambda}]^{-1}\hat{\Lambda}. \tag{20}$$

By performing the matrix inversion and multiplication followed by the inverse Laplace transform we get that the interrarival time distribution is the weighted sum of exponentially distributed random variables

$$f_{lm}(t) = \sum_{i=1}^{L} A_{i,lm} e^{\alpha_i t}, \tag{21}$$

where $\alpha_i$ is the $i^{th}$ root of $det[sI - \hat{Q} + \hat{\Lambda}]$ and can be calculated analytically for $L \leq 4$.

In the following we show how the calculation proceeds for $L = 3$. To calculate the roots $\alpha_i$ we rewrite $t(s) = det[sI - \hat{Q} + \hat{\Lambda}]$ in (20) to the form

$$t(s) = a_3 s^3 + a_2 s^2 + a_1 s + a_0, \tag{22}$$

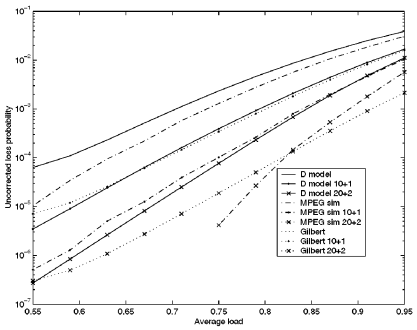


**Fig. 11.** Probability of uncorrected packet loss vs. average load on a 22.5 Mbps link,K=10

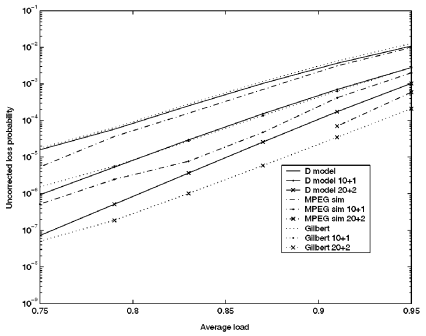


**Fig. 12.** Probability of uncorrected packet loss vs. average load on a 45 Mbps link, K=20

where

$$
\begin{aligned}
a_3 &= 1 \\
a_2 &= r_{12}+r_{13}+\lambda_1+r_{31}+r_{32}+\lambda_3+r_{21}+r_{23}+\lambda_2 \\
a_1 &= \lambda_2*r_{31}+r_{13}*\lambda_3-r_{21}^2+r_{13}*r_{23}+r_{13}*r_{32}+r_{13}*r_{21}+r_{21}*r_{31}+ \\
&\quad r_{21}*r_{32}+r_{21}*\lambda_3+r_{23}*r_{31}+r_{23}*\lambda_3+\lambda_2*r_{32}+\lambda_2*\lambda_3+r_{12}*r_{31}+ \\
&\quad r_{12}*r_{32}+r_{12}*\lambda_3+r_{12}*r_{21}+r_{12}*r_{23}+r_{12}*\lambda_2+r_{12}*\lambda_2+r_{13}*\lambda_2+ \\
&\quad \lambda_1*r_{31}+\lambda_1*r_{32}+\lambda_1*\lambda_3+\lambda_1*r_{21}+\lambda_1*r_{23}+\lambda_1*\lambda_2 \\
a_0 &= r_{12}*\lambda_2*r_{32}+r_{12}*\lambda_2*\lambda_3+r_{13}*r_{21}*\lambda_3-r_{21}^2*r_{31}-r_{21}^2*r_{32}-r_{21}^2*\lambda_3+ \\
&\quad r_{12}*r_{21}*r_{31}+r_{12}*r_{21}*r_{32}+r_{12}*r_{21}*\lambda_3+r_{12}*r_{23}*r_{31}+r_{12}*r_{23}*\lambda_3+ \\
&\quad r_{13}*r_{23}*\lambda_3+r_{13}*\lambda_2*r_{32}+r_{13}*\lambda_2*\lambda_3+\lambda_1*r_{21}*r_{31}+\lambda_1*r_{21}*r_{32}+ \\
&\quad \lambda_1*r_{21}*\lambda_3+\lambda_1*r_{23}*r_{31}+\lambda_1*r_{23}*\lambda_3+\lambda_1*\lambda_2*r_{31}+ \\
&\quad \lambda_1*\lambda_2*r_{32}+\lambda_1*\lambda_2*\lambda_3-r_{31}*r_{21}*r_{23}. \qquad (23)
\end{aligned}
$$

We denote the roots of (22) with $\alpha_i, i = 1,2,3$. Knowing $\alpha_i$ we can perform the partial fraction decomposition of (20) with respect to $s$

$$f^{l,m*}(s) = \sum_{i=1}^{L} \frac{A_i^{lm}}{s-\alpha_i}, \qquad (24)$$

where $A_i^{lm}$ can be calculated as

$$
\begin{aligned}
A_1^{lm} &= (c_2^{lm}*\alpha_1^2 - c_1^{lm}*\alpha_1 + c_0^{lm})/(\alpha_2-\alpha_1)/(\alpha_3-\alpha_1) \\
A_2^{lm} &= (c_2^{lm}*\alpha_2^2 - c_1^{lm}*\alpha_2 + c_0^{lm})/(\alpha_1-\alpha_2)/(\alpha_3-\alpha_2)
\end{aligned}
$$

$$A_3^{lm} = (c_2^{lm} * \alpha_3^2 - c_1^{lm} * \alpha_3 + c_0^{lm})/(\alpha_2 - \alpha_3)/(\alpha_1 - \alpha_3). \tag{25}$$

The coefficients $c_2^{lm}, c_1^{lm}, c_0^{lm}$ are the following:

$$\begin{aligned}
c_2^{11} &= \lambda_1 \\
c_1^{11} &= \lambda_1(r_{31} + r_{32} + \lambda_3 + r_{21} + r_{23} + \lambda_2) \\
c_0^{11} &= \lambda_1(r_{21}r_{31} + r_{21}r_{32} + r_{23}r_{31} + r_{21}\lambda_3 + r_{23}\lambda_3 + \lambda_2 r_{31} + \lambda_2 r_{32} r_{31} + r3 + \lambda_2\lambda_3) \\
c_2^{12} &= 0 \\
c_1^{12} &= \lambda_2 r_{21} \\
c_0^{12} &= \lambda_2(r_{21}r_{31} + r_{21}r_{32} + r1\lambda_3 + r_{13}r_{32}) \\
c_2^{13} &= 0 \\
c_1^{13} &= \lambda_3 r_{13} \\
c_0^{13} &= \lambda_3(r_{21}r_{23} + r_{13}r_{21} + r_{13}r_{23} + r_{13}\lambda_2) \\
c_2^{21} &= 0 \\
c_1^{21} &= \lambda_1 r_{21} \\
c_0^{21} &= \lambda_1(r_{21}r_{31} + r_{21}r_{32} + r_{21}\lambda_3 + r_{23}r_{31}) \\
c_2^{22} &= \lambda_2 \\
c_1^{22} &= \lambda_2(r_{31} + r_{12} + r_{13} + \lambda_1 + r_{32} + \lambda_3) \\
c_0^{22} &= \lambda_2(r_{12}r_{31} + r_{12}r_{32} + r_{12}\lambda_3 + r_{13}r_{32} + r_{13}\lambda_3 + \lambda_1 r_{31} + \lambda_1 r3 + \lambda_1\lambda_3) \\
c_2^{23} &= 0 \\
c_1^{23} &= \lambda_3 r_{23} \\
c_0^{23} &= \lambda_3(r_{23}r_{12} + r_{13}r_{23} + r_{23}\lambda_1 + r_{13}r_{21}) \\
c_2^{31} &= 0 \\
c_1^{31} &= \lambda_1 r_{31} \\
c_0^{31} &= \lambda_1(r_{21}r_{32} + r_{21}r_{31} + r_{23}r_{31} + \lambda_2 r_{31}) \\
c_2^{32} &= 0 \\
c_1^{32} &= \lambda_2 r_{32} \\
c_0^{32} &= \lambda_2(r_{32}r_{12} + r_{13}r_{32} + r_{32}\lambda_1 + r_{21}r_{31}) \\
c_2^{33} &= \lambda_3 \\
c_1^{33} &= \lambda_3(r_{13} + r_{12} + \lambda_1 + r_{21} + r_{23} + \lambda_2) \\
c_0^{33} &= \lambda_3(r_{13}r_{23} + r_{13}r_{21} - r_{21}^2 + r_{12}r_{21} + r_{12}r_{23} + r_{13}\lambda_2 + r_{12}\lambda_2 + \lambda_1 r_{21} + \lambda_1 r_{23} + \lambda_1\lambda_2).
\end{aligned} \tag{26}$$

Thus the Laplace transform of the conditional probability $f^{l,m}(t)$ has the form

$$f^{l,m}(s) = \sum_{i=1}^{L} A_i^{lm} \frac{1}{s - \alpha_i} \tag{27}$$

and $f^{l,m}(t)$ is

$$f^{l,m}(t) = \sum_{i=1}^{L} A_i^{lm} e^{\alpha_i t}. \tag{28}$$

By the definition of $f_{lm}(t)$

$$\sum_{m=1}^{L} f_{lm}(t) = P(X_{k+1} = t | J_k = l), \tag{29}$$

and

$$\int_0^{\infty} f_{lm}(t) = [(\hat{\Lambda} - \hat{Q})^{-1} \hat{\Lambda}]_{l,m}, \tag{30}$$

which accounts for the evolution of the underlying Markov chain. Based on (21) the infinite integrals in equations (6),(7),(8) and (9) can be calculated as

$$\int_x^{\infty} f_{lm}(t) dt = -\sum_{i=1}^{L} \frac{A_{i,lm}}{\alpha_i} e^{\alpha_i x}. \tag{31}$$

## References


1. J. Beran, R. Sherman, M. Taqqu, and W. Willinger, "Long-range dependence in variable-bit-rate video traffic," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1566–1579, 1995.
2. T. Yoshihara, S. Kasahara, and Y. Takahashi, "Practical time-scale fitting of self-similar traffic with markov-modulated poisson process," *Telecommunication Systems*, vol. 17, no. 1-2, pp. 185–211, 2001.
3. B. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: Myths and realities," in *Proc. of ACM SIGCOMM*, pp. 3–14, 1996.
4. J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun, "Internet traffic tends toward poisson and independent as the load increases," in *Nonlinear Estimation and Classification*, Springer, 2002.
5. I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high speed networks," *IEEE Transactions on Information Theory*, vol. IT-39, pp. 98–108, January 1993.
6. O. Gurewitz, M. Sidi, and M. Cidon, "The ballot theorem strikes again: Packet loss process distribution," *IEEE Transactions on Information Theory*, vol. IT-46, pp. 2599–2595, November 2000.
7. E. Altman and A. Jean-Marie, "Loss probabilities for messages with redundant packets feeding a finite buffer," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 779–787, 1998.
8. O. Ait-Hellal, E. Altman, A. Jean-Marie, and I. A. Kurkova, "On loss probabilities in presence of redundant packets and several traffic sources," *Performance Evaluation*, vol. 36-37, pp. 485–518, 1999.
9. H. Schulzrinne, J. Kurose, and D. Towsley, "Loss correlation for queues with bursty input streams," in *Proc. of IEEE ICC*, pp. 219–224, 1992.

10. H. Heffes and D. M. Lucantoni, "A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE Journal on Selected Areas in Communications*, vol. 4, September 1986.
11. C. Blondia, "The N/G/1 finite capacity queue," *Commun. Statist. - Stochastic Models*, vol. 5, no. 2, pp. 273–294, 1989.
12. P. Frossard, "FEC performances in multimedia streaming," *IEEE Comm. Letters*, vol. 5, no. 3, pp. 122–124, 2001.
13. D. Hoffman and G. Fernando, "RTP payload format for MPEG1/MPEG2 video," RFC 2250, January 1998.
14. W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proc. of NOSSDAV*, 2000.
15. P. Dube and E. Altman, "Utility analysis of simple FEC schemes for VoIP," in *Proc. of Networking 2002*, May 2002.
16. K. Kawahara, K. Kumazoe, T. Takine, and Y. Oie, "Forward error correction in ATM networks: An analysis of cell loss distribution in a block," in *Proc. of IEEE INFOCOM*, pp. 1150–1159, June 1994.