Astrid Undheim

# Characterization and Modeling of Slice-based Video Traffic

Thesis for the degree of Philosophiae Doctor

Trondheim, May 2009

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics

**NTNU**
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Telematics



ISBN 978-82-471-1564-0 (printed ver.)
ISBN 978-82-471-1565-7 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2009:92

Printed by NTNU-trykk

# Abstract

The interest in watching real-time video content transmitted over packet-based communication networks such as the Internet is growing. When resources are restricted, quality of service support and provisioning of service guarantees are needed in the network to ensure a satisfactory user experience. In addition, the video content, different choices made in the video encoding, the bitrate characteristics of the resulting video stream, and the network performance will have an influence on the perceived quality. Regarding the network performance, the packet loss ratio and packet loss distribution are identified as important performance parameters for real-time video, and are of particular interest. Estimating these parameters is a step towards assessing the perceived quality of service for a video transmission.

This thesis addresses issues related to video transmission over the Internet. In particular, new methods to characterize and analyze video traffic in a network perspective are proposed in order to estimate some key network performance parameters. This requires models of the traffic and the network elements. Video encoded using a newly developed slice-based H.264/AVC scheme is studied. This scheme intends to give less bursty video traffic, and will hence be favorable for encoding video to be transmitted over a resource constrained network.

Video clips encoded using this slice-based scheme are characterized using two different approaches. First using the correlation and distribution functions and second using a token bucket traffic model. The characterization gives statistical information about the video traffic and is a prerequisite for developing traffic models. Both of these issues are important since the slice-based video encoding produces a new type of video traffic.

The frame sequence of a slice-based encoded video clip is divided into sections that are classified using non-parametric methods. This classification is useful and necessary since a video stream in general is non-homogeneous and non-stationary. The classes can then be analyzed separately, and different models can be used for the classes. The distribution and dependence structures in the classes are studied. Next, a new approach for estimating loss is proposed using the classification of the video frames, giving the average loss as well as information about the clustering of the losses for the different classes. It is shown that losses over high thresholds are independent or weakly dependent, and the upper bounds of losses can be estimated using high quantiles. These quantiles give statistical guarantees for the amount of loss.

Next, a Gaussian model is developed for the video traffic. This model is advantageous since it incorporates the correlation functions of real video traces. Also, because of the additive properties of Gaussian processes, properties for an aggregated traffic stream can be deduced from the single streams. The packet loss for a video stream is defined using the exceedances of the video frame sizes over a threshold. Characteristics of a loss period, in terms of length and loss volume, are then found. These further give the loss ratio and loss distribution in a bufferless model as well as for a small buffer. Such results are important since the perceived quality depends on both the total amount of loss as well as the distribution of the losses.

For real-time applications, service guarantees are needed to ensure a satisfactory quality level for the users. These service guarantees are specified by network calculus server models. An approach to parameter estimation for important server models is proposed using external measurements on a network router. The obtained results are compared to the theoretical values, and the cause of the discrepancies is identified.

# Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU).

The PhD study has been conducted in the period August 2004 to January 2009. During the study period, I have been hosted and funded by the Centre for Quantifiable Quality of Service in Communication System, Centre of Excellence (Q2S). Q2S is funded by the Norwegian Research Council, NTNU and Uninett. The PhD study was formally conducted at the Department of Telematics, NTNU. In addition to the research work, it included mandatory courses corresponding to one semester of full-time studies, and one year of teaching assistance, funded by the Department of Telematics. Professor Peder J. Emstad has been the supervisor of this work.

During my four years at Q2S, there are several people I would like to thank. First and foremost, Professor Peder J. Emstad for accepting me as a PhD-student and for being my supervisor through these four years. I have appreciated all discussions and feedback.

Thanks also to the great researchers whom I have been lucky to work with: Natalia Markovich, co-author of two of my papers, Yuming Jiang for introducing me to the topic of network calculus, Yuan Lin for doing the encoding work for my video clips, and finally Arne Øslebø at Uninett for setting up the measurement equipment.

I thank everyone working at Q2S Centre, these four years would not have been the same without you. In particular, I thank Tor Kjetil Moseng who has been my room mate throughout our thesis works and Erik Hellerud for helping me out with latex related questions. I also want to thank the administrative staff Anniken, Hans, and Mette for being very helpful with everything and for creating a pleasant working environment.

Finally, thanks to my dear family and friends, my parents Helga and Bernt and my sisters Karin and Brit. Thank you for all your support and encouragement.

# Table of Contents

# List of Abbreviations

| | |
|---|---|
| ACF | Autocorrelation Function |
| AF | Assured Forwarding |
| AR | Autoregressive |
| ARMA | Autoregressive Moving Average |
| AVC | Advanced Video Coding |
| BA | Behavior Aggregate |
| BB | Bandwidth Broker |
| CBR | Constant Bit Rate |
| cdf | cumulative distribution function |
| CF | Correlation Function |
| D-BMAP | Discrete Batch Markovian Arrival Process |
| DAR | Discrete Autoregressive |
| DCT | Discrete Cosine Transform |
| df | distribution function |
| DiffServ | Differentiated Services |
| DPCM | Differential Pulse-Code Modulation |
| DRR | Deficit Round Robin |
| DSCP | DiffServ Code Point |
| DSL | Digital Subscriber Line |
| EF | Expedited Forwarding |
| EVI | Extreme Value Index |
| FARIMA | Fractional Autoregressive Integrated Moving Average |
| FBM | Fractional Brownian Motion |
| FEC | Forward Error Correction |
| FGN | Fractional Gaussian Noise |
| FIFO | First In-First Out |
| FMO | Flexible Macroblock Ordering |

| | |
|---|---|
| FR | Full Reference |
| GAR | Gamma Autoregressive |
| GBAR | Gamma Beta Autoregressive |
| GOP | Group of Pictures |
| GoS | Grade of Service |
| GR | Guaranteed Rate |
| IETF | Internet Engineering Task Force |
| iid | independent and identically distributed |
| IntServ | Integrated Services |
| IP | Internet Protocol |
| ISDN | Integrated Services Digital Network |
| ITU | International Telecommunication Union |
| LR | Latency Rate |
| LR-WSG | Latency Rate Worst-case Service Guarantee |
| LRD | Long-Range Dependent |
| MAC | Medium Access Control |
| MMFF | Markov Modulated Fluid Flow |
| MOS | Mean Opinion Score |
| MPEG | Moving Pictures Experts Group |
| MSE | Mean Squared Error |
| MTU | Maximum Transport Unit |
| MVNI | Multivariate Normal Integral |
| NAL | Network Abstraction Layer |
| NALU | NAL Unit |
| ned | negative exponential distribution |
| NR | No-Reference |
| NRK | Norwegian Broadcasting Corporation |
| P-MMBBP | Periodic Markov Modulated Batch Bernoulli Process |
| PFT | Packet Scale Rate Guarantee Virtual Finish Time |
| PHB | Per Hop Behavior |
| PLR | Packet Loss Ratio |
| PQoS | Perceived Quality of Service |
| PSNR | Peak Signal to Noise Ratio |
| PSRG | Packet Scale Rate Guarantee |
| QoE | Quality of Experience |

| | |
|---|---|
| QoS | Quality of Service |
| QQ | Quantile-Quantile |
| RED | Random Early Detection |
| RR | Reduced Reference |
| RSpec | Reservation Specification |
| RSVP | Resource Reservation Protocol |
| RTCP | Real Time Control Protocol |
| RTP | Real-Time Transport Protocol |
| RTSP | Real Time Streaming Protocol |
| SBBP | Switched Batch Bernoulli Process |
| SLA | Service Level Agreement |
| SLS | Service Level Specification |
| SRD | Short-Range Dependent |
| SSIM | Structural Similarity Index Measurement |
| SSQ | Single Server Queue |
| TCP | Transmission Control Protocol |
| TES | Transform Expand Sample |
| TOS | Type of Service |
| TSpec | Traffic Specification |
| UDP | User Datagram Protocol |
| UMTS | Universal Mobile Telecommunications System |
| VBR | Variable Bit Rate |
| VCEG | Video Coding Experts Group |
| VCL | Video Coding Layer |
| VFT | Virtual Finish Time |
| VoD | Video-on-Demand |
| VoIP | Voice over IP |
| VQEG | Video Quality Experts Group |
| WFQ | Weighted Fair Queueing |

# Part I

# Thesis Introduction

# Chapter 1

# Introduction

This thesis studies encoded video in a network perspective, first through general characterization of the video, next using different types of models both for the video traffic and the network in order to analyze the Quality of Service (QoS). The focus is on some key network performance parameters accessible at the network egress. Knowledge of these parameters is a step towards assessing the QoS perceived by a user watching the transmitted video. Video encoded using the recently developed slice-based video encoding scheme is studied. This scheme was developed in order to produce less bursty video traffic, and hence results in lower loss and delay for an encoded video stream transmitted through a network.

This chapter serves as an introduction to the thesis. Section 1.1 gives the motivation for the research conducted. The outline of the thesis is given in Section 1.2 and the main results and contributions are described in Section 1.3. The papers written as part of the thesis are listed in Section 1.4.

## 1.1 Motivation

Originally, the Internet was designed to provide best-effort data delivery [1]. With the introduction of Voice over IP (VoIP) and streaming video over IP, stricter requirements are imposed on the network since these real-time services have stringent constraints regarding the key network performance parameters: throughput, delay, delay jitter, and packet loss. The amount of video traffic transmitted over the Internet has increased tremendously lately [2], partly due to the growing popularity of web-based video streaming services such as YouTube [3]. Initially, YouTube provided only low quality video, but was recently updated to allow for high quality video and audio. Video clips are now shown in a widescreen high-definition format, using the latest standard for video coding, H.264/Advanced Video Coding (AVC) [4]. The Norwegian Broadcasting Corporation (NRK) has also had great success in offering real-time streaming of news etc., and recently also in providing streaming from the Olympic games in Beijing. In addition, mobile providers have put attention on video streaming for mobile devices, especially focusing on big sporting events such as football championships and the Olympics

to attract users. This video is mostly real-time, which put demands on the network in order to satisfy the quality requirements of the users. QoS support is then needed when the network resources are restricted.

QoS is defined by ITU-T in the recommendation E.800 [5] as follows: "The collective effect of service performance which determine the degree of satisfaction of a user of the service". As this definition shows, the user perspective is important when evaluating QoS and this has led to the introduction of the terms Perceived QoS (PQoS) and Quality of Experience (QoE) [6]. These terms link the user perception and expectations for QoS to the quantitative performance parameters accessible at the network boundaries, the most important being: throughput, packet delay, delay jitter, and packet loss.

To support end-to-end QoS for real-time traffic over the Internet, two different QoS architectures, namely the Integrated Services (IntServ) [7] and the Differentiated Services (DiffServ) [8] have been proposed by the Internet Engineering Task Force (IETF). Both of these architectures have the objective of minimizing delay, delay jitter, and loss for real-time applications, using a reservation-based approach and a class-based approach, respectively. To analyze the service guarantees specified by these models, a set of traffic and server models are defined under the name of Network Calculus, see e.g., [9] and [10].

Most of the video transmitted over the Internet is Variable Bit Rate (VBR) encoded video, which is often preferred over Constant Bit Rate (CBR) encoded video because of the constant end-user quality and higher compression efficiency for the former. H.264/AVC [4] is the latest video coding standard, and provides a considerable improvement in compression efficiency compared to earlier standards. A high compression gain is achieved by removing temporal and spatial redundancy. The present frame is encoded using previous or consecutive frame(s) as reference, taking advantage of the temporal redundancy in consecutive frames belonging to the same scene, while spatial redundancy is removed by transform coding. In addition, intra coded frames are inserted periodically to prevent error propagation. The size of the encoded frames is decided by the rate-distortion target for VBR coding, giving variable frame sizes. The resulting video stream is therefore bursty, which can cause network delay and hence packet loss due to late arriving packets. With this in mind, the explicit slice-based video encoding scheme was developed, originally described in [11], using the H.264/AVC standard. This new scheme has no Group of Picture (GOP) structure and the large intra coded frames are avoided. Video encoded using the slice-based scheme is hence smoother than regular frame-based H.264/AVC encoded video, while retaining the constant end-user quality and error resilience of the latter.

This thesis addresses issues related to video transmission over the Internet, focusing on slice-based encoded video. Traffic characterization, modeling, and analyzes are performed, with the objective of estimating the network performance parameters (i.e., the packet loss) when a traffic stream is transmitted through a given network. To be able to predict the QoS delivered to the users, models of the traffic and the network are needed. In particular, models for aggregated traffic are important, to reflect the fact that video traffic in the Internet is seen as aggregates of single video streams.

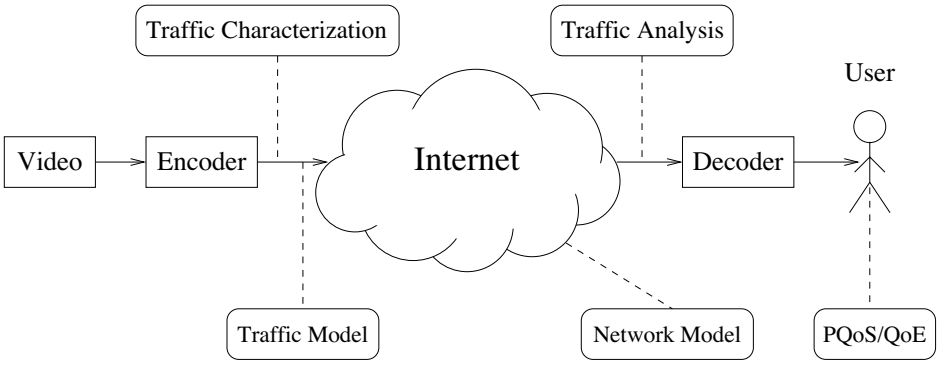



**Figure 1.1:** Main areas and focus in this thesis, concerning video transmission over the Internet.

The main areas and focus of this thesis shown in Figure 1.1 are traffic characterization, traffic analysis, traffic models, network models, and PQoS/QoE assessment. The latter is only discussed in terms of network performance, i.e., performance parameter estimation targeted at giving information about the PQoS is pursued. The next section gives the outline of the thesis, showing how these areas are covered.

## 1.2 Thesis Outline

This thesis is organized as follows. Part I continues with Chapter 2 which provides a background for the most important topics addressed in the thesis. Chapter 3 gives an overview of the H.264/AVC standard and in particular the explicit slice-based video encoding scheme. The two encoded video clips that are studied in this thesis are also described.

Following this, the main body of research work is divided into four parts. These parts are organized as follows:

**Part II: Characterization of Slice-based H.264/AVC Encoded Video Traffic.** This part of the thesis mainly focuses on characterization of slice-based encoded video. Studying the statistical properties of the slice-based video is interesting because this new encoding scheme produces video traffic with different characteristics from regular frame-based video. Characterization is also an important prerequisite for developing traffic models. Part II is divided into two chapters. In Chapter 4, one video clip encoded using the slice-based video encoding scheme is characterized with respect to the distribution functions and the correlation functions of the scene lengths and frame sizes. In addition, simulations are performed to compare the performance of a slice-based encoded stream to that of a regular frame-based encoded stream. In Chapter 5, token bucket characterization of the slice-

based encoded video is pursued. The token bucket traffic model is important in the Internet since it is used for resource reservation in IntServ and for analyzing service guarantees using network calculus both in IntServ and DiffServ.

**Part III: Non-parametric Analysis of Slice-based H.264/AVC Encoded Video Traffic.** This part of the thesis describes a non-parametric approach for classification and analysis of slice-based encoded video. Applying non-parametric methods presents a new approach to traffic analysis, where no information about the distribution functions is needed. This part of the thesis is divided into two chapters. In Chapter 6, sections of the slice-based encoded video stream are classified by the average frame size. A new method for scene change detection for the individual classes is presented, and the resulting scenes are checked for dependence. Also, characteristics of the classes of video data are studied, such as the mean excess function and the tail index, showing the distribution structure in the classes. In Chapter 7, the results from Chapter 6 are exploited for estimation of loss using exceedances of frame sizes over a high threshold. The high quantiles for the amount of loss are also found.

**Part IV: Characterization of Loss for Aggregated Video Using a Gaussian Model.** This part of the thesis is devoted to video traffic modeling based on the results from Part II, and estimation of loss using the resulting model. In Chapter 8, a Gaussian model is proposed for the slice-based encoded video. This model is advantageous since it incorporates the correlation functions of real video traces, while still being a simple, parsimonious model. Also, because of the additive properties of Gaussian processes, properties for an aggregated traffic stream can be deduced from single streams. It is found that the exceedances of frame sizes over a threshold for an aggregated stream are related to the exceedances of frame sizes over a threshold for a single video stream. In Chapter 9, these exceedances are analyzed, constituting loss periods in a bufferless model. The moments of the length and loss volume of a loss period are found numerically using the correlation functions of the slice-based encoded video traces. In addition, relations between the distributions of the length and loss volume of a loss period are exploited.

**Part V: Router Models for Quality of Service Assessment.** This part of the thesis focuses on using measurements for router model parameterization. In Chapter 10, network calculus server models are parameterized using external measurements on the input and output links of a router. The approach is to estimate the required parameters using measurements results from burst and backlog periods and to use known results to derive the results for Guaranteed Rate (GR) and Packet Scale Rate Guarantee (PSRG) server models. These models are used to analyze service guarantees in IntServ and DiffServ, respectively.

**Part VI: Concluding Remarks.** This part of the thesis contains a summary of the main results and conclusions of the thesis. Topics of future work are also identified and described.

## 1.3 Contributions

The main contributions of Part II are:

- Video traffic encoded using the slice-based H.264/AVC video encoding scheme is characterized, using distribution functions and correlation functions of the scenes and frame sizes. The results show that there is only negligible autocorrelation for the scene lengths, and for the size of the frames in different scenes. However, there is non-negligible correlation between the scene change frame at the beginning of a scene and the average frame size in the scene. Also the frames inside a scene exhibit non-negligible correlation.
- The packet loss and delay through a bottleneck node for the slice-based encoded video are compared to those for regular frame-based encoded video, using network simulations. The results show that the slice-based encoding is advantageous compared to frame-based encoding when the buffer size is small.
- Lossless and loss bounded token bucket and leaky bucket traffic models are parameterized for different slice-based encoded video streams. The results show that the slice-based encoded video can tolerate fewer reserved resources than the frame-based encoded video while still fulfilling the same loss and delay requirements. For all streams, the token bucket parameters are significantly reduced by introducing a small data buffer for input traffic queueing.

The main contributions of Part III are:

- Sections of a highly variable slice-based encoded video stream are classified according to the average frame size. From the resulting classes, a non-parametric method is proposed for scene change detection. The scenes are checked for dependence, both using regular dependence measures such as the Autocorrelation Function (ACF) and the Ljung-Box test, and using long-range dependence measures. From the ACFs and the Ljung-Box statistics only non-negligible correlation is found for scenes inside the classes, but the Hurst parameter estimates show signs of long-range dependence for the scenes in the considered classes.
- The distributions of the frame sizes in the classes are estimated by the mean excess function. The results show that all classes contain mixtures of heavy- and light-tailed distributed frame sizes. The number of finite moments for the frame size distributions is estimated using the Hill's estimate.
- The expected loss for the classified video stream when transmitted over a bottleneck link is estimated. Two non-parametric functions, the mean excess function and the extremal index are used. In addition, the high quantiles for

the losses are estimated, showing the upper bound for the amount of loss that can occur with a given probability.

The main contributions of Part IV are:

- A discrete, multivariate Gaussian model is proposed for the slice-based encoded video, taking the correlation between consecutive frames into account. Relations between single and aggregated streams are found for the exceedances of the frames sizes over a threshold.
- The relations between single and aggregated streams are exploited for calculating the moments of the length of a loss period for the aggregated stream based on the moments of the length for individual streams. Numerical results are given for correlation functions from different video traces, showing higher first and second moments for the length of a loss period when the correlation between consecutive frames is increased.
- The moments of the loss volume of a loss period are estimated using a similar numerical approach as for the length of a loss period. The results are compared to the loss volume of a loss period for a continuous Gaussian process and show satisfactory agreement for high thresholds. The loss volume is used for estimating packet loss in a bottleneck node with a small buffer, in addition to giving the loss directly in the bufferless case.
- A relation between the distribution of the length and loss volume of a loss period for the continuous process is recognized. This relation is shown to be valid for the length and loss volume of a loss period for the discrete process as well. In addition, the first moment of the length of a loss period is shown to agree with the first moment found using Little's formula.

The main contributions of Part V are:

- The parameters of network calculus server models, in particular GR and PSRG, are estimated using external measurements on a network router. The parameters are estimated directly from the measurement results. In addition, a new approach using burst and backlog period statistics and their relations to the GR and PSRG server models is developed. With the latter method, the evaluation of the delay for every packet is avoided.
- The measurement results are used for estimation of the router processing time. The minimum processing time is shown to be equal to the difference between the value of the theoretical server model rate parameters and the value of the measured rate parameter. Furthermore, the values of the error parameters from the measurements are higher than the values of the theoretical error parameters due to processing times being higher than the minimum processing time.
- The results from the server modeling of a router can be used to give delay bounds for token bucket constrained traffic flows. An example is given for the token bucket characteristics of the slice-based encoded streams from Chapter 5.

## 1.4 Publications

All papers are written under supervision and in cooperation with Professor Peder J. Emstad. Details of the contributions from the other co-authors are given under each publication.

### Papers Included in the Thesis

[A] Astrid Undheim, Yuan Lin, and Peder J. Emstad. "Characterization of Slice-based H.264/AVC Encoded Video Traffic." In *Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN)*, Toulouse, France, February 2007.

[B] Astrid Undheim, Yuming Jiang, and Peder J. Emstad. "Network Calculus Approach to Router Modeling Using External Measurements." In *Proceedings of the International Conference on Communications and Networking in China (ChinaCom)*, Shanghai, China, August 2007.

[C] Natalia Markovich, Astrid Undheim, and Peder J. Emstad. "Slice-based VBR Video Traffic-Estimation of Link Loss by Exceedance." In *Proceedings of the 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (QoS-IP)*, Venice, Italy, February 2008.

[D] Astrid Undheim and Peder J. Emstad. "Distribution of Loss Periods for Aggregated Video Traffic." In *Proceedings of the ITC Specialist Seminar 18 (ITCSS'18)*, Karlskrona, Sweden, May 2008.

[E] Astrid Undheim and Peder J. Emstad. "Characterization of Slice-based H.264/AVC Encoded Video Traffic Using Token Buckets." *Telecommunication Systems, Springer*, 39(2), October 2008.

[F] Natalia Markovich, Astrid Undheim, and Peder J. Emstad. "Classification of Slice-based VBR Video Traffic and Estimation of Link Loss by Exceedance." *Computer Networks, Elsevier*, 53(7), May 2009.

[G] Astrid Undheim and Peder J. Emstad. "Distribution of Loss Volume and Estimation of Loss for Aggregated Video Traffic." *Submitted for publication, 2009.*

[A]: Yuan Lin encoded the video clips used in this paper and the rest of the thesis and contributed in writing the description of the slice-based video encoding scheme in this paper.
[B]: Yuming Jiang proposed to use the network calculus server models for the router modeling, gave guidance during the work and the writing of the paper. The author performed the measurements, did the analysis and wrote the paper.
[C, F]: Natalia Markovich proposed new statistical non-parametric methods for the analysis of slice-based encoded video traffic. The author contributed in developing the ideas, applying the non-parametric methods to the video data, performing the computational analysis in Mathematica, and in writing the paper.

### Other Papers by the Author

[H] Astrid Undheim, Yuming Jiang, and Peder J. Emstad. "Network Calculus Approach to Router Modeling Using External Measurements." In *Proceedings of the 3rd EuroNGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements*, Turin, Italy, June 2006.
- This paper is an early version of Paper [B].

[I] Jan Erik Voldhaug, Erik Hellerud, Astrid Undheim, Erling Austreim, U. Peter Svensson, and Peder J. Emstad. "Effects of Network Architecture on Perceived Audio Quality." In *Proceedings of the 2nd ISCA Tutorial Research Workshop on Perceptual Quality of Systems*. Berlin, Germany, September, 2006.
- This paper is an early version of Paper [K].

[J] Astrid Undheim and Peder J. Emstad. "Characterization of Slice-based H.264/AVC Encoded Video Using Token Buckets." In *Proceeding of the Euro-FGI Workshop on New Trends in Modelling, Quantitative Methods and Measurements.* Gent, Belgium, May-June 2007.
- This paper is an early version of Paper [E].

[K] Jan Erik Voldhaug, Erik Hellerud, Astrid Undheim, Erling Austreim, U. Peter Svensson, and Peder J. Emstad. "Influence of Sender Parameters and Network Architecture on Perceived Audio Quality." *Acta Acustica United with Acustica*, 94(1), 2008.

[I, K]: The author contributed to ideas in this paper, performed the network simulations together with Erling Austreim and contributed to the writing of the paper.

# Chapter 2

# Background

This chapter gives a brief background on topics of particular relevance to this thesis. Section 2.1 gives an overview of QoS provisioning in the Internet, with special focus on QoS architectures for the Internet. QoS for video transmission over the Internet is discussed in Section 2.2. Section 2.3 summerizes the most important aspects of network calculus relevant for this thesis, including server models and traffic models.

## 2.1 QoS Provisioning in the Internet

This section gives an introduction to QoS provisioning in today's Internet, starting with an overview of the topic and continuing with the IntServ and DiffServ architectures.

### 2.1.1 Introduction

"The Holy Grail of computer networking is to design a network that has the flexibility and low cost of the Internet, yet offers the end-to-end quality-of-service guarantees of the telephone network" [12]. This quotation from the late 90's illustrates the ultimate goal of QoS provisioning in the Internet. Achieving this goal is difficult because of the differences between these two networks. The telephone network is connection-oriented and provides reserved resources as soon as the connection is set up. QoS in the telephone network is therefore defined as the call blocking probability as seen by the users and the term Grade of Service (GoS) is defined as the general quality, including user and service provider aspects [13]. The ITU-T recommendation E.800 [5] defines QoS for the telephone network and ISDN as follows:

**Quality of Service (ITU-T)**:

> "The collective effect of service performance which determine the degree of satisfaction of a user of the service."

According to E.800, QoS depends on the service performance, which is divided into support, operability, serveability, and security. The service performance then again relies on the network performance characteristics such as transmission performance and availability.

In the more recent ITU-T Recommendation G.1000 [14], new definitions for QoS terms are given in order to have a set of consistent definitions. G.1000 gives four different viewpoints for the QoS. These are:

1. QoS requirements of user/customer
2. QoS offered/planned by provider
3. QoS delivered/achieved by provider
4. QoS perceived by user/customer

In addition to putting more attention on the user, these new definitions are more targeted to the use in Internet, where the diversity of applications and services calls for new approaches to QoS compared to the telephone network. In this sense, the fourth viewpoint also resembles the term PQoS which is discussed in more details in Section 2.2.3.

The Internet, in contrast to the telephone network, was designed to offer connection-less, best-effort data delivery and had no focus on QoS initially [1]. It worked this way until the introduction of delay sensitive applications on top of IP. QoS was then also defined by the IETF in RFC 2216 [15] as follows:

**Quality of Service (IETF)** refers to:

> "the nature of the packet delivery service provided, as described by parameters such as achieved bandwidth, packet delay, and packet loss rates."

This definition focuses only on the network performance parameters, and does not take the user aspect into account at all.

With the introduction of real-time services over IP and the focus on QoS, the need for service differentiation mechanisms in the Internet became clear, leading to heavy research on the area. The research conducted mainly led to two different proposals for support of QoS in the Internet, developed by the IETF. These are the Integrated Service (IntServ) architecture [7] that offers absolute QoS and the Differentiated Service (DiffServ) architecture [8] that provides relative QoS. IntServ and DiffServ are described next.

### 2.1.2 IntServ

Observing that real-time applications did not perform well across the Internet due to variable queueing delays and congestion losses, the IETF proposed IntServ as the inital QoS architecture. IntServ was originally designed to be able to control the end-to-end packet delay and provide bandwidth sharing [7]. It is a per-flow based service differentiation scheme, focusing on resource reservation and

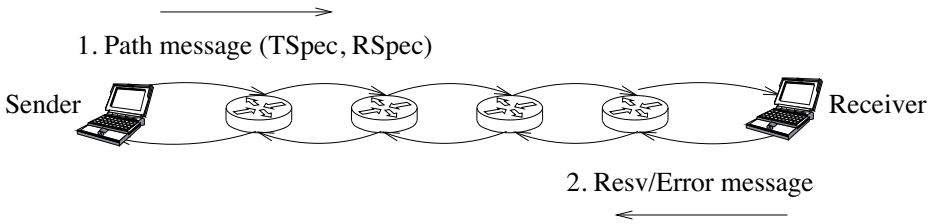



**Figure 2.1:** Resource reservation in IntServ using RSVP.

admission control on a per-flow basis for providing service guarantees. As such, state information for each IntServ flow is needed in every router on the path from source to destination, which makes the scheme unscalable in a large network.

The Resource ReSerVation Protocol (RSVP) [16] is used as signaling protocol for resource reservation in IntServ, requesting resources along a path as shown in Figure 2.1. The reservation is receiver initiated, but the sender will signal the receiver to initiate the reservation. The reservation procedure is then as follows: 1) During the signaling phase, a path message with a Traffic Specification (TSpec) [17] and a Reservation Specification (RSpec) is sent from the sender to the receiver. The TSpec specifies the traffic characteristics of the flow with parameters such as token rate, bucket depth, peak rate, and maximum packet size while the RSpec defines the level of service required, such as delay guarantees and bandwidth requirements. 2) An admission control scheme is needed in the routers to determine whether a router should accept the flow or not. If accepted, the router records the traffic characteristics contained in the path message before forwarding it to the next router on the path. 3) The receiver responds to the path message by sending a reservation (Resv) message in the opposite direction along the same route as the path message. If the request is rejected, an error message is sent back to the sender. 4) If every router on the path accepts the resource request based on the TSpec and RSpec contained in the path message, bandwidth and buffer space are allocated and flow-specific state information is stored in the routers. Every router on the path must participate in the resource reservation process meaning that partial deployment of IntServ is not feasible.

IntServ introduces two service classes in addition to the best-effort service class. A deterministic Guaranteed Service [18], which gives an upper bound on the end-to-end delay and a stochastic Controlled Load Service [19], which provides a QoS to a flow approximately equal to the QoS that the flow would receive from an unloaded network element.

IntServ has not been a great success, mainly due to the per-flow resource reservation and the succeeding per-flow processing and per-flow state in the routers. These, together with the violation of the end-to-end design principle of the Internet (see e.g., [20] for a discussion on this issue), are some of the main reasons why IntServ has never been deployed. Different approaches have been proposed to solve the scalability problem. In [21], it is proposed to use IntServ

over DiffServ, where a DiffServ domain serves as a network element in IntServ and participates in the end-to-end resource reservation as a single network element. Furthermore, in [22] it is recommended to enhance the RSVP to perform resource reservation on classes of aggregates, where an aggregate consists of a number of flows with shared ingress and egress routers through an aggregate network. In this case, per-flow resource reservation is needed only at the edge of the network and per-aggregate resource reservation is performed in the aggregate network.

### 2.1.3 DiffServ

Two main problems with IntServ governed the need for another approach to QoS provisioning in IP networks. First and foremost, with each router maintaining per-flow state, IntServ could not scale well in a large network. In addition, only two service classes were specified and the flexibility of differentiation between flows of the same service class was lacking. With this in mind, the IETF proposed the DiffServ architecture [8]. DiffServ is therefore not a per-flow based scheme but a per-aggregate-class based service differentiation scheme, where the traffic is divided into a small number of Behavior Aggregates (BA). DiffServ also allows for different drop precedence levels for different flows, and even packets, belonging to the same class.

The classification into BAs is done at the ingress of the network and involves the DiffServ edge router assigning a DiffServ Code Point (DSCP) value to the Type of Service (TOS) byte in each IP-packet. This DSCP value specifies which BA the packet belongs to and decides the treatment that the packet receives from the core routers. Routers in the core of the network provide service guarantees to aggregates, using a variety of scheduling and queue management procedures.

The externally observable forwarding behavior in the routers for each BA is called Per Hop Behavior (PHB), where each PHB maps to a DSCP value. DiffServ defines two PHBs in addition to the default best-effort PHB, which are the Expedited Forwarding (EF) PHB [23] and the Assured Forwarding (AF) PHB group [24]. The EF PHB is supposed to provide low delay, low jitter, and low packet loss by ensuring a configured service rate to the EF aggregate, as well as a bounded deviation from this configured rate. Each node that provides EF service should then comply with the Packet Scale Rate Guarantee (PSRG) server model [25], as described in Section 2.3.1. The AF PHB group consists of four different AF classes, each of which is allocated an amount of buffer space and bandwidth in the nodes. Within each AF class, there are three different levels of drop precedence, thereby providing differentiated treatment within each class. The packets with the highest drop precedence value are dropped first in the case of congestion, using an active queue management scheme such as Random Early Detection (RED) [26].

Service Level Agreements (SLAs) define a service contract for the provisioning of service guarantees, and are used between the customers and their source DiffServ domain, as well as between different DiffServ domains [8]. These SLAs contain a Service Level Specification (SLS), which specifies the traffic characteristics of an aggregate as well as the PHB. The traffic characteristics are often defined using

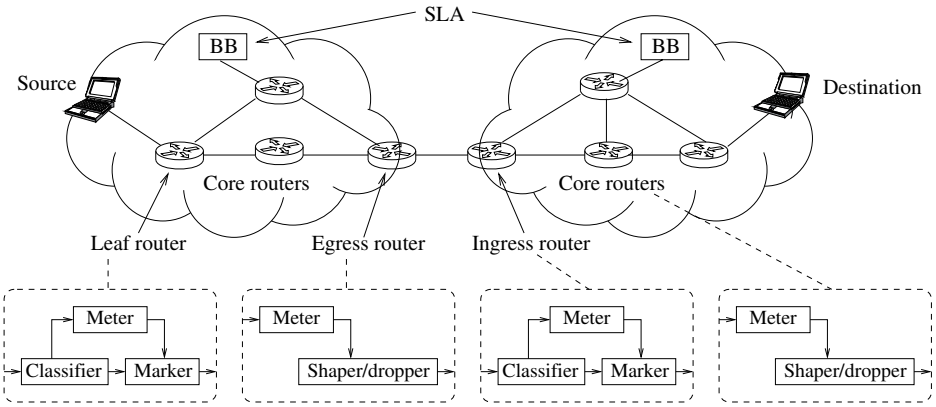



**Figure 2.2:** A simple DiffServ architecture (from [27]).

token bucket models [9] as described in Section 2.3.2. A Bandwidth Broker (BB) function is then needed in each DiffServ domain to perform admission control, manage network resources, etc., based on the SLAs.

A simple DiffServ architecture is shown in Figure 2.2. For the given source, the first capable downlink router in the source domain (leaf router) will perform per-flow classification, metering, and marking [27]. In addition, ingress and egress routers need traffic conditioning capabilities. At an egress router, DiffServ aggregates are shaped to conform to their profile, before being sent to another DiffServ domain. At the ingress routers, classification and marking of aggregates are performed, possibly after metering. In addition, core routers may include traffic conditioning capabilities such as metering and shaping (or dropping) of packets that are out-of-profile.

The use of DiffServ for transmission of layered MPEG-2 encoded video is demonstrated in [28], where different video coding layers are put in different DiffServ classes. It showed that the perceived quality of the the transmitted video is highly dependent on how the layers are created, and the layered video can tolerate a higher network load than regular video while achieving the same quality target.

An experimental DiffServ network was developed for the Internet2 Qbone project [27], but even though the project successfully demonstrated DiffServ in a test network, DiffServ has not gotten the deployment that was hoped for. Some people talk about overprovisioning instead of QoS support, arguing that DiffServ will not the be widely deployed because of too high costs relative to the benefits [29]. While overprovisioning may be an option in the backbone network, the wireless access links still have limited resources. Hence, in [30] it is argued that QoS mechanisms are indeed needed in today's network and that QoS support is well taken care of by the DiffServ architecture. Furthermore, the main obstacles to a wide deployment of DiffServ are seen as mainly business related. Hence,

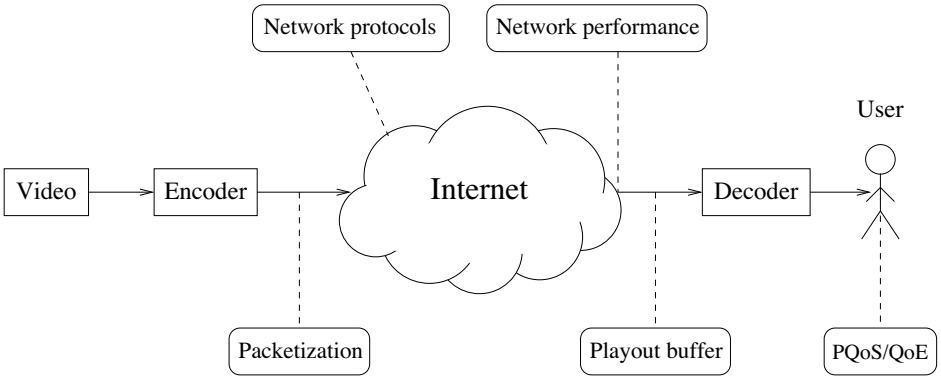



**Figure 2.3:** Multimedia and network aspects influencing the PQoS for video transmissions over the Internet.

with an increase in the amount of traffic transmitted over the Internet, service differentiation is likely to become a value-added feature in the near future.

## 2.2 QoS for Video Transmission over the Internet

In this section, QoS issues related to video transmission over the Internet are discussed. Different types of video applications that have divergent QoS requirements are described. Finally, challenges related to assessing the end-user perceived QoS for a video transmission over the Internet are discussed.

### 2.2.1 Video Transmission over the Internet

Video transmitted over the Internet is often real-time video, which means that requirements for low delay and loss are imposed in order to provide satisfactory end-user quality [31]. Because of these requirements and the high and variable bitrate of video traffic, various challenges arise for video transmissions over the Internet. Traditionally, network performance metrics such as throughput, delay, delay jitter, and packet loss probability were used for estimating the QoS for a video transmission, in accordance with IETFs QoS definition. However, with new advances in video coding and application level support for QoS, several other aspects influence the QoS perceived by the end-users. An overview of multimedia and network aspects that influence the perceived QoS is given next. These are also shown in Figure 2.3.

#### Video Content

The characteristics of the video content is the first aspect affecting the resulting quality. In particular, the type of video application is important. Video conference streams on one hand are often quite static and have a high degree of spatial

and temporal dependencies, making it relatively easy to compress. The resulting bitrate is therefore low and has a low variability. Action movies on the other hand have frequent scene shifts and higher motion within scenes, as well as a large amount of details, resulting in less spatial and temporal dependencies. This type of video will hence require more bits to encode, i.e., the rate-distortion function is larger for a high activity scene than for a low activity scene with the same distortion [32]. The resulting bitrate will probably also have higher variability because of some low activity scenes. The influence of the video content on the perceived quality is investigated in [33], where it is shown that the spatial and temporal complexity influence the perceived quality and should be taken into account in a model for predicting perceived quality.

The bitrate characteristics of slice-based encoded video are studied in Chapter 4 and 6. The bitrate reflects the video content after encoding and is important to study since the bitrate characteristics of the video will influence the network performance experienced.

### Encoding

In order to transport a video stream over the Internet, compression is needed to reduce the bitrate of the stream. H.264/AVC [4] is the latest standard for video coding and is studied in this thesis. The encoding process and the encoding parameters for H.264/AVC are targeted to the application type, the video content, the underlying network, and possible feedback from the network. Application layer techniques including error control and congestion control are used for coping with variable network conditions. Error control comprise Forward Error Correction (FEC), retransmissions, and error resilience [34]. For H.264/AVC, there has been much focus on error resilience tools for transmission in lossy network environments. These tools include picture segmentation, data partitioning, reference picture selection, flexible macroblock ordering etc. The effect of a lost packet on the distortion is highly dependent on which error resilience techniques are employed, and this is studied using simulations in [35].

### Packetization

After encoding, the video frames are divided into packets in a process called packetization. For H.264/AVC, the Network Abstraction Layer (NAL) was included to provide coding transparency towards the transmission medium. Simple packetization then involves putting a NAL Unit (NALU), containing a slice of a frame, into the payload of a Real-Time Transport Protocol (RTP) packet [36]. In order to have fully decodable packets at the receiver, it is advantageous to keep the slice size smaller than the Maximum Transport Unit (MTU) of the underlying network. Hence, being able to decode all packets that are successfully transmitted. Otherwise, the RTP packets are fragmented on the IP-layer to resemble the MTU of the underlying network [35]. Video frames are typically encoded at a rate around 30 frames per second and are usually much larger than typical MTU sizes.

This results in bursts of packets carrying the size of the frames being sent to the network. So, video traffic is generally bursty.

### Network Protocols

The transmission medium and the network protocols above are taken into account in the encoding process. The physical transmission medium will influence the FEC and error resilience tools applied to the encoded video [37]. The packetization process described above is also tailored to the underlying network, and the MTU of the Medium Access Control (MAC) layer will decide the optimal NALU size. For video traffic transmission over the Internet, the Internet Protocol (IP) [38] is the obvious choice at the network layer. For the transport layer, the Transmission Control Protocol (TCP) [39, 40] is currently employed for most non-real time video content. TCP is connection-oriented and ensures the delivery of the video packets without loss. This is accomplished by using retransmission of lost packets and hence delay is traded for zero loss. Some streaming services also use TCP as transport protocol, buffering a large amount of packets at the client side before starting the playback. The buffering will account for variable network delay and even give enough time to retransmit lost packets. For real-time video, there are more strict time constraints because of a maximum allowable delay. Packets arriving too late for decoding are of no use, while a low packet loss probability in the order of a few percents is usually considered acceptable. The User Datagram Protocol (UDP) [41] is therefore the preferred choice for transmission of real-time video such as conversational services and real-time streaming. In addition, the RTP [42] is usually employed to add support for real-time audio and video services on top of UDP. RTP includes sequence numbers, facilitating detection of lost packets. Finally, the Real Time Control Protocol (RTCP) is used together with RTP to monitor the QoS of the session [42] and the Real Time Streaming Protocol (RTSP) [43] is used for streaming video.

### Network Performance

In addition to the different choices made for video transmission, the network performance plays a significant role for the final perceived quality. The most important network performance parameters that can be evaluated at the network egress are:

- Throughput; defined as the number of bits successfully transmitted and received by a source destination pair in a time interval, divided by the time interval. IP link capacity using IP-layer bits is defined similarly in [44].

- Delay; defined as one-way delay [45] and round-trip delay [46].

- Delay variation (jitter); defined as the differences in the one-way delay of packets belonging to one stream [47].

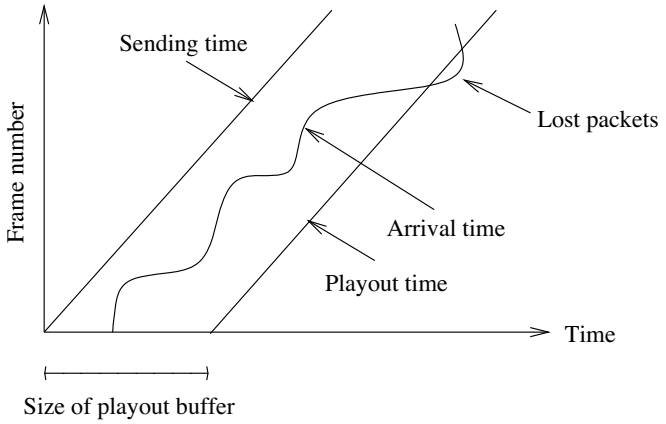



**Figure 2.4:** A video playout buffer for streaming and conversational video.

- Packet Loss Rate/Ratio (PLR); defined as the ratio of the number of lost packets to the number of transmitted packets between a source destination pair [48].

- Packet loss pattern; defined using the distance between consecutive losses and the length of a loss period [49].

An overview of the requirements to the performance parameters for different video applications is given in Section 2.2.2. Ultimately, these parameters should be used for assessing the QoS perceived by the end-users. This is discussed in more details in Section 2.2.3,

### Playout Buffer

For streaming video and conversational video as defined in the next section, an application specific playout buffer (or jitter buffer) is needed at the receiver side for absorbing variable network delays and to allow for retransmission of lost packets. The size of the playout buffer decides the maximum allowable variation in network delay and also the minimum time needed for buffering before the video playback is started. A too big buffer then means unnecessary delay while a too small buffer causes excessive packet loss [35]. A packet arriving at the playout buffer after its scheduled playout time is considered lost. This is shown in Figure 2.4. For real-time streaming video and conversational video, the size of the playout buffer should be minimized to ensure low end-to-end delay. For non real-time video streaming, the playout buffer can be large to account for large network delays as well as packet retransmissions and will hence provide a low packet loss probability.

#### Decoding

The decoder reconstructs the original video streams. In case of missing frames or parts of frames, the decoder performs error concealment, e.g., using prediction from previous frames or neighboring macroblocks [37]. The degree to which this is successful depends on the video content, the error resilience tools applied at the encoder, and also which parts of a frame and how much is lost. As described in [50], basic MPEG-2 systems do not decode a frame with a lost packet, but discard the entire frame and insert the previous frame instead. With H.264/AVC on the other hand, the error concealment is usually more sophisticated and a lost packet should optimally only result in one lost slice. This assumes that the slices are small enough to avoid fragmentation on the IP layer, as described under packetization. Hence, lost slices are recovered using error concealment tools corresponding to the error resilience added at the encoder.

#### Perceived QoS

After decoding, the video stream is played to the end-user. The final video quality perceived by the user is then denoted by the Perceived QoS (PQoS). As can be seen from this discussion, the perceived quality depends on the video clip, the encoding, the packetization, the network protocols, the network performance, the playout buffer, and finally the decoding. Adding the user expectation brings on the term Quality of Experience (QoE). The final quality experienced by the users is hard to predict and analyze without the use of subjective tests. PQoS and QoE for video transmissions over the Internet are discussed in more details in Section 2.2.3, together with an overview of important results for estimating the PQoS from the network performance parameters and the multimedia imposed impairments.

### 2.2.2 Video Applications

Applications used over the Internet can be divided into elastic and non-elastic applications [51]. Traditional Internet applications such as email, file transfer, and web-surfing are elastic, meaning that they can tolerate delay and losses, in addition to being able to decrease and increase their transmission rate depending on the network conditions. These applications typically use TCP. Real-time video applications on the other hand are non-elastic, meaning that they are less tolerant to packet loss and variations in delay, and they require a minimum capacity equal to the bitrate of the stream and cannot benefit from a higher available capacity. These applications typically use UDP.

A classification of video into download, streaming, and conversational video is common [31], where only download video is elastic. In addition, the Video-on-Demand (VoD) sub-class of streaming video is called semi-elastic in this thesis, since it can use TCP. These classes of video applications have divergent requirements for throughput, delay, delay jitter, and packet loss. This is also taken into account in the video encoder, where the encoder can be optimized for low-latency or coding efficiency, depending on the application [35].

**Table 2.1:** Network performance requirements for classes of video applications.

| Class | Application | Bandwidth | Delay/Jitter | Loss |
|---|---|---|---|---|
| Download | Video download | Elastic | <15 seconds | Zero |
| Streaming Video | Video-on-Demand | Semi-elastic | <10 seconds | <1% |
| | Live streaming | Non-elastic | <2 seconds | <1% |
| Conversational | Video conferencing | Non-elastic | <150 ms | <1% |
| | Video telephony | Non-elastic | <150 ms | <1% |

The least demanding video application in terms of delay is the downloading of a video for later replay. This application is elastic and can increase and decrease the bitrate according to the available bandwidth, as well as being tolerant to variations in the network delay. Downloading of video requires a lossless transmission, however this is taken care of by the TCP protocol.

Streaming video includes all types of video transmissions where the playback starts before the transmission of the video is finished [34]. Streaming video can be VoD, where the video typically is pre-recorded and stored at a streaming server, or live (real-time) streaming, which is available only in real-time. An example of the former is YouTube, while live-streaming from football matches is an example of the latter. These two types of streaming applications differ in their requirements to the playout delay. Although both VoD and live-streaming can tolerate some buffering, the latter typically has a shorter playout buffer and therefore lower maximum delay, but also higher loss probability. Both of them have stringent requirements for the packet loss probability, even as low as 1% [31].

Finally, the term conversational video is used, covering video conferencing, video telephony, and other applications that are two-way/multi-way. These applications have more stringent requirements for the network performance in terms of maximum end-to-end delay and loss probability. In addition, the applications are non-elastic and the bandwidth requirements are therefore stringent. Loss and delay critical applications are the focus in this thesis, and real-time streaming and conversational video are used.

The requirements to the network performance parameters for different video applications, based on the segmentation from the ITU-T Recommendation G.1010 [31] are summarized in Table 2.1. Live-streaming is not explicitly addressed in [31], but its delay requirement is set to a typical delay of two seconds here to distinguish it from VoD.

### 2.2.3 Perceived QoS and Quality of Experience

While QoS proposals for the Internet focus on network performance measures such as throughput, delay, delay jitter, and packet loss, the terms PQoS or QoE have gained more importance for multimedia applications. In this thesis, PQoS is used for the QoS as perceived subjectively by the end-users, while for QoE also user expectations and economical aspects are included. The former is therefore most important for the work in this thesis.

Focus on the QoE conforms with the G.1000 recommendation which clearly indicates the user perspective. A new definition for QoE is included as an appendix in the ITU-T Recommendation P.10 [52].

**Quality of Experience**:

> "A measure of the overall acceptability of an application or service, as perceived subjectively by the end-user."

This definition makes a clear separation between the QoS defined in E.800 [5] and in RFC 2216 [15] as concerned with the network performance and the QoE as the quality subjectively perceived by the users. However, this QoE definition better resembles the fourth viewpoint in G.1000 [14].

For evaluating the quality of a video stream as perceived by the users, actual users must be tested. A common measure of the perceived QoS is then the subjective Mean Opinion Score (MOS) [53], where the opinion score is a measure of the perceived quality as seen by test subjects. The evaluation is done by setting up a viewing test as described in [53] and letting a group of people evaluate different video clips, ranging them from 1 (bad) to 5 (excellent). The MOS result is then given as the average of the individual opinion scores. This approach is expensive and time-consuming. Objective tests are therefore frequently used instead. The goal of the objective tests is then to give results that correlate well with the MOS results. Three different classes of tests are used: Full Reference (FR), Reduced Reference (RR), and No-Reference (NR) tests. These are distinguished by the availability of the complete video stream or some simple statistics of the original video stream at the receiver, for comparison with the altered stream.

For FR methods, the original video stream is required for comparison with the distorted stream, which may not always be feasible. The Peak Signal to Noise Ratio (PSNR), which uses the Mean Squared Error (MSE) of the two streams, is a very common FR method, although being criticized for low correlation with subjective results [54]. Another simple FR method is the Structural Similarity Index Measurement (SSIM) described in [55]. SSIM focuses on measuring the structural information change in the distorted stream compared to the original stream in order to assess the image distortion. For RR methods, some statistics of the original video stream must be available at the receiver for evaluation of the quality. In [56], a combined RR/FR method is proposed, using wavelets. Finally, for the NR method, no information about the original stream is available at the receiver. Hence, the distorted video stream must be evaluated for estimating the quality. One approach is to measure the block-edge impairments as described in [57].

The Video Quality Experts Group (VQEG) has led the work on objective tests for assessing multimedia quality. The results reported in [58] showed satisfactory results for two FR algorithms and one RR algorithm, leading to two new ITU-T Recommendations in 2008. These are J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference [59] and J.246: Perceptual audiovisual quality measurement techniques for multimedia services

over digital cable television networks in the presence of a reduced bandwidth reference [60].

### 2.2.4 Evaluation of PQoS using Network Performance Parameters

Ultimately, a mapping between the network QoS parameters such as throughput, delay, delay jitter, and packet loss and the PQoS is the goal. This is also identified in a recent paper on video quality assessment [54], where packet loss based metrics are seen as a good solution to the PQoS assessment, due to the low computational complexity compared to evaluation of the fully decoded video stream. Having the discussion from the previous sections in mind, this may look as a difficult task. However, in particular the packet loss burstiness has been identified as an important metric for the PQoS, both for speech, audio, and video. This is especially important for these applications since decoders in general have more difficulties with concealing the effect of consecutive packet losses than single losses, as discussed e.g., in [50].

#### Speech

For speech, perceived QoS can be assessed using the E-model [61], which is an additive impairment model. Hence, impairments due to SNR, coding, transmission delay etc., are added to give a rating factor that is converted to a MOS value. The E-model has been updated recently, first to account for random packet loss based on results published in [62] and next to account for arbitrary loss distributions based on results published in [63]. The loss distribution is accounted for using the packet loss burst ratio, given as the ratio of the first moment of the length of a loss period to the first moment of the length of a loss period for random losses.

#### Audio

For audio, network simulations and subjective tests are used for evaluating the effect of packet loss burtiness on the perceived quality in [64]. The distribution of packet loss for music streams transmitted over a network is modeled using results from simulations. Both best-effort and DiffServ nodes are simulated and the differences in the packet loss burstiness resulting from these setups are investigated. The best-effort case showed higher burstiness compared to the DiffServ case because of RED active queue management for the latter. The same packet loss ratio then resulted in a higher MOS value for the DiffServ case than for the best-effort case for acceptable loss ratios.

#### Video

For video, several approaches have been proposed to assess the perceived QoS depending on the loss process. However, most of these approaches use the PSNR/MSE to estimate the distortion, instead of using subjective tests to estimate the PQoS. The effect of the burst loss on the distortion is modeled and compared

to simulations in [65]. The results show that the burst length of the loss process is important for estimating the distortion and that loss occurring in bursts affects the distortion more than single losses of the same amount. In [66], three different methods are presented for evaluation of the quality of distorted video using the MSE. The video is encoded using MPEG-2 and is transmitted over a packet network. The NoParse method uses measures of the packet loss rate only, while the QuickParse and FullParse methods also incorporates the impact of losses. Next, an approach to real-time assessment of the video quality is described in [50]. Here, a loss-distortion model is developed, using both multimedia and network aspects. For estimation of the distortion, the video content, type of video codec, packetization, loss recovery mechanisms, and the amount of loss are taken into account, the latter through the average number of packets between losses. A mapping between the distortion and the PSNR is given. However, an additive impairment model is used for the losses, and the more severe effects on the distortion when the losses occur in bursts are not taken into account. Finally, a hybrid metric for evaluation of perceived QoS is described in [54]. This model takes the network impairments and information about the video stream into account. Loss of intra or predictive coded slices give different impairments and the video coding layer complexity, including the content charateristics, the amount of scene changes, and the quantization level, are included to give the final MOS value.

Subjective test results on the effect of consecutive packet losses are given e.g., in [67]. Here, a random neural network model trained with results from subjective tests is used for evaluating the effect of both coding parameters and network QoS parameters on the perceived quality, for H.263 encoded video. In particular, it is found that increasing the number of consecutive lost packets while keeping the loss ratio constant leads to better quality because of fewer deteriorated frames. This is explained by the high frame rate (30 frames per second is used) and thereby difficulty of detecting a distorted frame.

In [68], a similar approach as for the E-model is pursued for video traffic. A parametric video quality model is proposed, with additive impairment factors calculated from the source quality, video coding, transmission impairments etc. The transmission impairments include the packet loss as well as the packet loss concealment. Only the packet loss ratio is investigated, however non-uniform distributed packet loss is expected to be included in a future model. A strong point is the use of subjective tests, and the ultimate goal is a comprehensive model for the evaluation of video quality comparable to the E-model for speech. The quality evaluation is more complicated for video compared to for speech, because of the content dependencies on the perceived quality as shown in [33]. The effect of the spatial and temporal complexity in the video sequences on the perceived quality are investigated for inclusion in the model from [68].

The extent to which losses are concealed is highly dependent on the error resilience tools, as shown using simulations in [35]. This also means that results for the effect of bursty losses can be taken into account when applying these tools. As this discussion shows, the effect of bursty losses is highly dependent on the decoding. With a decoding scheme that discards the whole frame in the case of a lost packet, bursty losses should improve the perceived quality compared to

random losses. However, a lost packet should optimally only result in the loss of one slice, and bursty losses will therefore most often decrease the perceived quality compared to random losses, since bursty losses are more difficult to conceal [50].

In this thesis, two approaches to the estimation of the loss are pursued. First, in Chapter 7, a non-parametric approach to the estimation of loss is proposed. This approach gives the amount of loss as well as some information about the clustering of the losses for a video trace. Second, in Chapter 9, the loss distribution is estimated using a Gaussian model for the video traffic. This approach gives the moments of the length and loss volume of a loss period for video traffic specified by its mean and covariance function.

## 2.3 Network Calculus

In the recent decades, several works analyzing service guarantees in the Internet have emerged under the name of network calculus, see e.g., [9, 10, 69, 70]. Network calculus is called the system theory for computer networks. In contrast to linear system theory where regular convolutions are used, network calculus frequently uses the theory of min-plus convolution [10]. Namely:

$$(f \otimes g)(t) := \inf_{0 \leq \tau \leq t} \{f(\tau) + g(t - \tau)\} \tag{2.1}$$

and the de-convolution:

$$(f \oslash g)(t) := \sup_{\tau \geq 0} \{f(t + \tau) - g(\tau)\} \ . \tag{2.2}$$

In general, network calculus defines bounding functions or envelopes for the amount of traffic arriving in a time period, called arrival curves. Also, bounding functions are defined for the service elements, called service curves. These curves are defined using the convolution operation for the service curve and the de-convolution for the arrival curve respectively, as described in the following.

Deterministic network calculus was defined first [9, 10, 69], followed by the probabilistic version called stochastic network calculus [70]. The former is employed in this thesis. Different types of traffic and server models defined under deterministic network calculus are then described next.

### 2.3.1 Server Models

The general service curve model is described first, followed by the Latency Rate (LR) server model which is a special case of the service curve model. In addition, under the two Internet QoS architectures, the Guaranteed Rate (GR) and Packet Scale Rate Guarantee (PSRG) server models are used to respectively define the Guaranteed Service of an IntServ router and the Expedited Forwarding PHB of a DiffServ router. Known relationships exist between these two models and the service curve model. These relationships are employed in Chapter 10 for estimating parameters for the GR and PRSG server models using external measurements on a router.

### Service Curve

A server offers a service curve $\beta(t)$ to a flow if and only if there is a $t^0 \geq 0$ with $t^0 \leq t$ such that the amount of service received by the flow in $[0, t]$, $W(t)$ and the amount of traffic generated by the flow in the time period $[0, t^0]$, $A(t^0)$ satisfies [10]:

$$W(t) \geq A(t^0) + \beta(t - t^0) \tag{2.3}$$

The adaptive service curve is a variant of the service curve model, and has a close connection to the PSRG server model [10].

### Latency Rate (LR)

The LR server model was defined in [71] for describing the worst-case behavior of a scheduler, covering a broad range of scheduling algorithms. The LR server model uses the concept of burst period (busy period is used in [71]), and the amount of service received by a flow in the burst period. A router burst period is defined as a time period $[t^0, t^*]$ where the arrival rate at time $t$ in $[t^0, t^*]$ is always at or above the reserved rate, $r$ [71]. That is:

$$A(t^0, t) \geq r(t - t^0) \ , \tag{2.4}$$

where $A(s, t)$ denotes the amount of traffic arrived in the time interval $[s, t]$.

The LR server model is defined using the burst period concept and uses two parameters, the allocated rate, $r$, and the latency term, $\Theta$. A server is then a LR server if and only if for each $t$ in the time period $[t^0, t^*]$, it is guaranteed that the amount of service received by a flow in this time period satisfies [71]:

$$W(t^0, t) \geq r(t - t^0 - \Theta) \ , \tag{2.5}$$

where $t^0$ is the starting time of the burst period and $t^*$ is the time instant when the last packet which arrived during the burst period leaves the server. The server is then a LR server with rate $r$ and latency $\Theta$.

As seen from this definition and the definition in Equation 2.3, the LR server model is a special case of the general service curve model where the service curve $\beta(t - t^0)$ is equal to $r(t - t^0 - \Theta)$. In the same way, the Latency Rate Worst-case Service Guarantee (LR-WSG) [72] is a special case of the adaptive service curve model, where the service curve is equal to $r(t - t^0 - \Theta)$, but the service guarantee is fulfilled for all $t$ in a backlog period.

### Guaranteed Rate (GR)

In contrast to the service curve and LR server models, the GR server model defines a deadline guarantee for all packets with regard to the service in a Single Server Queue (SSQ) with the same rate. For a GR server with rate $r$, it is then guaranteed that the $j$th arriving packet is transmitted by time [73]:

$$d^j \leq VFT^j + E \ , \tag{2.6}$$

where $d^j$ denotes the departure time of the $j$th packet from the server, $E$ is an error term and the Virtual Finish Time (VFT), which denotes the departure time in the SSQ with rate $r$, is iteratively defined for $j \geq 1$:

$$VFT^j = max\{a^j, VFT^{j-1}\} + \frac{l^j}{r} , \tag{2.7}$$

where $a^j$ is the arrival time of packet $j$ at the server, $l^j$ is the length of the $j$th packet and $VFT^0 = 0$. The error term $E$ captures how much a node may be late with respect to the ideal SSQ with constant service rate $r$ and is typically dependent on the scheduling algorithm employed.

The GR server model specifies the service of a Guaranteed Service node in IntServ. Scheduling algorithms conforming to the GR (and LR) server model includes First In-First Out (FIFO), Weighted Fair Queueing (WFQ), and Deficit Round Robin (DRR) as shown in [74].

#### Packet Scale Rate Guarantee (PSRG)

In the PSRG server model, a server may also be early compared to the ideal constant rate server [25, 72], thereby also providing a bound on the router jitter. The delay guarantee for the PSRG server model is similar to that of the GR server model, except for a modified VFT function, the PSRG VFT (PFT):

$$d^j \leq PFT^j + E , \tag{2.8}$$

where PFT is defined for $j \geq 1$:

$$PFT^j = \max\{a^j, \min\{PFT^{j-1}, d^{j-1}\}\} + \frac{l^j}{r} , \tag{2.9}$$

where $d^0 = 0$ and $PFT^0 = 0$.

It is easy to verify that PSRG implies GR since for any packet $j$, the inequality $PFT^j \leq VFT^j$ always holds. In other words, if a server provides PSRG with rate $r$ and error term $E$, it also provides GR with the same rate and the same error term.

Also, known relationships exist between the service curve model and the GR server model as well as between the adaptive service curve model and the PSRG server model. These relationships are exploited in Chapter 10 for estimating the rate and error parameters for the GR and PSRG server models using external measurements on a router.

### 2.3.2 Traffic Models

The network calculus traffic models are special in the sense that they provide deterministic or stochastic upper bounds on the amount of traffic generated in a time period. A deterministic traffic bound gives an absolute bound on the amount of traffic arriving in a time period:

$$Pr[\text{Amount of traffic arriving in a time period} \leq \alpha] = 1$$

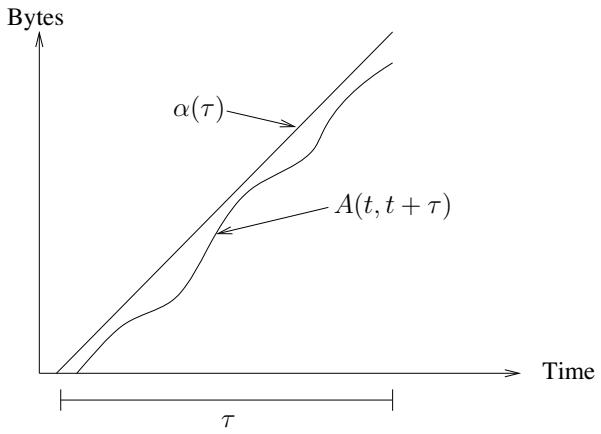



**Figure 2.5:** The traffic flow $A(t, t+\tau)$ constrained by the arrival curve $\alpha(\tau)$.

while a stochastic traffic bound gives a probabilistic bound on the amount of traffic arriving in a time period:

$$Pr[\text{Amount of traffic arriving in a time period} \geq \alpha] \leq p$$

In this thesis, the focus is on different variants of the token bucket traffic model, which is a deterministic traffic model. The token bucket model is of particular importance since its variant, the dual token bucket, is employed for the TSpec in IntServ, as well as for the SLS in DiffServ. The token bucket is also used for traffic shaping at a DiffServ node if an aggregate is out of profile. In addition, delay guarantees for the LR, GR, and PSRG server models are defined when the token bucket parameters of the input flows are known [75].

The general arrival curve is defined first, followed by the regular token bucket traffic model and the leaky bucket traffic model. The token bucket model is a special case of the arrival curve.

### Arrival Curve

The general deterministic arrival curve is defined as follows [10]:

A flow is said to be constrained by the function $\alpha(\cdot)$, and hence has an arrival curve $\alpha(\cdot)$, if and only if for each $t \geq 0$,

$$A(t, t+\tau) \leq \alpha(\tau) \ , \tag{2.10}$$

where $A(t, t+\tau)$ is the amount of traffic generated by the flow in the time period $[t, t+\tau]$.

The arrival curve $\alpha(\tau)$ and a traffic flow $A(t, t+\tau)$ constrained by the arrival curve are shown in Figure 2.5.

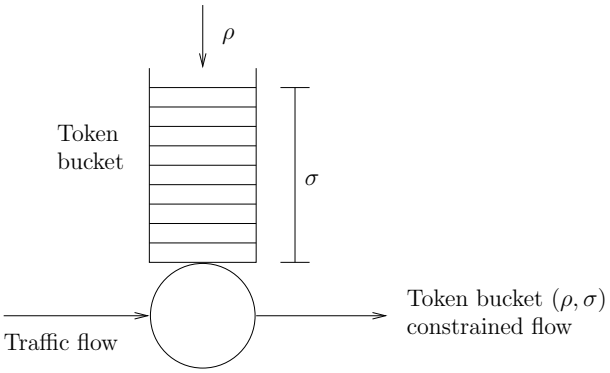



**Figure 2.6:** A token bucket with token generation rate $\rho$ and token bucket size $\sigma$.

### Token Bucket Traffic Model

The token bucket traffic model is a deterministic traffic model and is defined as follows [9]:

A flow is said to be token bucket $(\rho, \sigma)$-constrained if and only if for each $t \geq 0$,

$$A(t, t+\tau) \leq \rho\tau + \sigma \ , \tag{2.11}$$

where $A(t, t+\tau)$ is the amount of traffic generated by the flow in the time period $[t, t+\tau]$, $\rho$ is the token generation rate and $\sigma$ is the token bucket size which defines the maximum burst size to be transmitted.

The operation of the simple token bucket is shown in Figure 2.6. Tokens are generated with a rate $\rho$ until the token bucket reaches its maximum size, $\sigma$. A traffic flow under study arrives over a link and acquires an amount of tokens equal to the packet sizes. A packet finding fewer tokens than the packet size in the token bucket upon arrival is dropped. The objective of traffic characterization using token buckets is to find the parameters $\rho$ and $\sigma$ that are exactly large enough for each packet to find enough tokens in the token bucket upon arrival. The output stream from this token bucket will then be token bucket $(\rho, \sigma)$-constrained. Video traffic characterization using token bucket traffic models is performed in Chapter 5, both using simulations and using an analytical approach.

The dual token bucket, employed for the TSpec [17] in IntServ, is defined in a similar way. Namely, a flow is said to be dual token bucket $(\rho, \sigma; p, L)$ constrained if and only if for each $t \geq 0$:

$$A(t, t+\tau) \leq \min\{\rho\tau + \sigma, p\tau + M\} \ , \tag{2.12}$$

where $p$ is the peak rate of the flow and $M$ is the maximum packet size.

Additionally, a loss bounded token bucket model can be defined, with a bound on the probability of packet loss due to an empty token bucket at arrival.

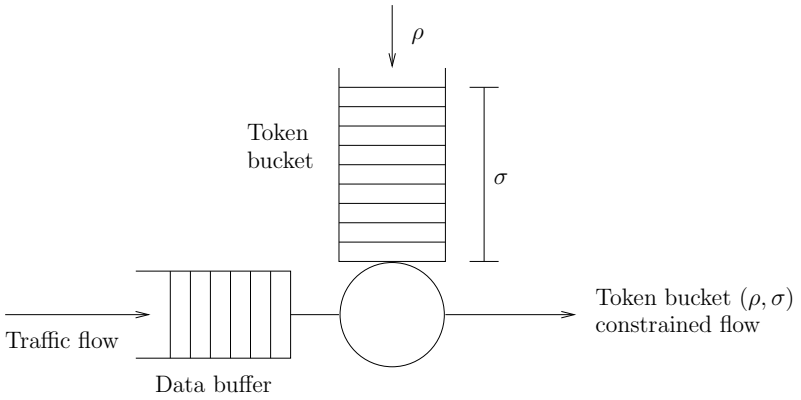



**Figure 2.7:** A leaky bucket with token generation rate $\rho$ and token bucket size $\sigma$.

### Leaky Bucket Traffic Model

The leaky bucket traffic model used in this thesis has a data buffer in addition to the token bucket as can be seen in Figure 2.7. It should be noted that several different definitions exist for the leaky bucket. Here, the model addressed in [76,77] is denoted a leaky bucket. It works as follows. If a packet finds too few tokens in the token bucket upon arrival, it is put in the data buffer for queueing until enough tokens are present, hence smoothing the input stream by introducing an extra delay. It is shown in [76] that the loss probability of an input stream to a leaky bucket depends on the token bucket size and the data buffer size only through their sum. The leaky bucket is also a deterministic traffic model, since the output from the leaky bucket is token bucket constrained with parameters $\rho$ and $\sigma$. Hence, the arrival curve of a leaky bucket corresponds to that of the token bucket. However, for traffic characterization, the leaky bucket will reduce the token bucket parameters for a flow by introducing some delay in the data buffer. The leaky bucket is employed in addition to the token bucket traffic model for traffic characterization in Chapter 5.

# Chapter 3

# Slice-based H.264/AVC

This chapter describes the explicit slice-based mode type selection scheme developed using the H.264/AVC standard [4]. The scheme was first introduced in [11], with the objective of reducing the burstiness of standard frame-based H.264/AVC encoded video by using explicit mode type selection on the slice level, with one intra coded slice per frame. When transmitting a slice-based encoded stream through a network, this reduced burstiness should result in less network delay and loss compared to standard frame-based encoded video. The slice-based video encoding scheme has also been studied in [78–83].

Section 3.1 starts with an introduction to video coding and gives the motivation behind the slice-based video encoding scheme. Section 3.2 gives an overview of the H.264/AVC standard, while Section 3.3 describes the proposed slice-based scheme in details. Finally, Section 3.4 presents the two sample traces used in this thesis.

## 3.1 Introduction

Video compression is necessary for effective transport of video streams over a network. VBR video coding has gained much importance because of its improved end-user quality and higher compression efficiency compared to CBR video coding. The enhanced quality compared to CBR is achieved by a constant rate-distortion target and hence using a constant quantization parameter for the whole sequence. However, the constant quantization parameter and the periodic prediction scheme used in VBR coding, with one large intra (I) coded frame at the beginning of each Group of Picture (GOP) and small predicted (P) frames, result in a bursty traffic stream when transmitted over a network. The bitrate variations due to changes in the motion level in consecutive scenes are not easily removed while retaining the end-user quality. The bitrate variations due to the different size of the intra coded frames compared to the predicted frames are on the other hand mostly removed in the novel enhancement to the H.264/AVC standard [4] called the explicit slice-based mode type selection scheme [11]. Here, the GOP structure is broken up and each frame contains an intra coded slice at successive positions. This results in a smoother video stream, at the cost of a slightly increased average

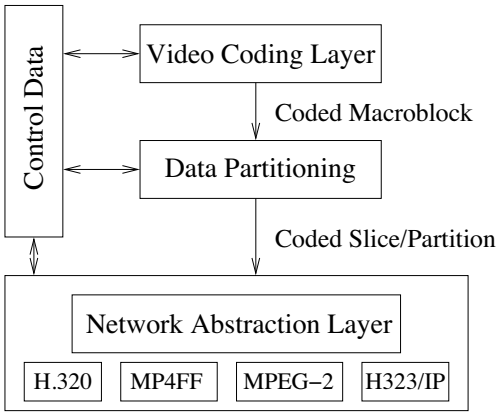



**Figure 3.1:** VCL and NAL in H.264/AVC (from [84]).

bitrate because of added redundancy in the form of partially overlapping I slices. The overlap is needed to prevent errors from propagating upwards into parts of the frame where it has already been removed by an I slice. The necessity of this overlap for error resilience is investigated in [79], and found to be dependent on the packet loss probability.

## 3.2 H.264/MPEG-4 Advanced Video Coding (AVC)

H.264/AVC is the latest standard for video coding developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Pictures Experts Group (MPEG) [4]. The main objectives with the new standard were to increase the coding efficiency compared to earlier standards and to provide coding transparency, followed by an adaptation to a wide diversity of underlying networks. Application areas include high definition TV broadcast as well as video transmission over media with lower data rates, such as DSL and UMTS [84].

In order to support the large diversity in applications and transmission media, the H.264/AVC design, in addition to the Video Coding Layer (VCL) for efficient video coding, included a Network Abstraction Layer (NAL). The NAL is employed for mapping the data from the VCL to a broad range of different transmission media, thereby providing transparent video coding on the VCL layer. The structure of the H.264/AVC encoder is shown in Figure 3.1. The coded macroblocks are assembled into slices before being handed over to the NAL, which maps the encoded video slices on to the transmission media. The VCL and NAL are described next.

### 3.2.1 Video Coding Layer (VCL)

H.264/AVC draws on previous ITU-T standards H.261, H.262 (MPEG-2) and H.263, and uses a similar block-based hybrid video coding as those [84]. Each picture is represented using variable-size macroblocks and inter-picture prediction

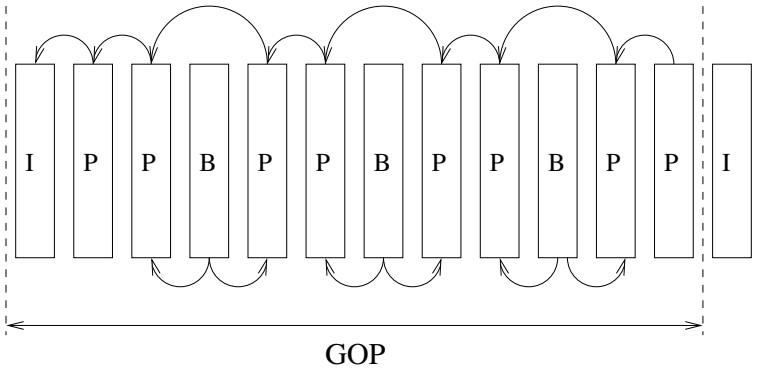



**Figure 3.2:** Example GOP structure with a regular, frame-based prediction scheme.

is employed for removing temporal dependencies between consecutive frames (i.e., motion compensation). Transform coding is then performed on the predicted residuals for removing spatial dependencies (i.e., dependencies within each frame). The name hybrid coding refers to these two steps for removing redundancies.

Macroblocks are assembled into slices, either sequentially or using Flexible Macroblock Ordering (FMO). With FMO, a slice consists of macroblocks from different parts of the picture, e.g., in a checker-board pattern. This means that macroblocks from a lost slice can be concealed using neighboring macroblocks belonging to successfully transmitted slices. This is shown to be beneficial e.g., for video conferencing applications [35].

Three main coding modes are used for the slices, in order to remove the temporal dependencies. Namely, intra coded (I), forward predictive coded (P), and bidirectional predictive coded (B) [84]. The prediction scheme for H.264/AVC is similar to previous standards, but is enhanced to allow for more flexible prediction. In contrast to earlier standards, the P slices can be predicted from a number of previous frames and the B slices from a number of previous and subsequent frames. In addition, it is now possible to use the B slices as a basis for prediction. An I frame is always inserted as the first frame in each GOP. In the simplest form, the same coding mode is applied for all slices of a frame for the rest of the frames in the GOP. These frames are then either P frames that are predicted from previous I or P frames, or B frames that are predicted from previous and subsequent I or P frames. The GOP structure can then look like: IPPBPPBPPBPP, given a GOP size of 12. This prediction scheme is shown in Figure 3.2. Throughout the rest of the thesis, this prediction scheme is called standard frame-based, reflecting on the use of the same encoding mode for all slices of a frame.

After removing temporal dependencies, transform coding is performed on the predicted residuals, as is also done in the previous ITU-T standards. However, the previous standards use a $8 \times 8$ Discrete Cosine Transform (DCT), while H.264/AVC primarily uses an integer transform based on the DCT. This transform operates

on $4 \times 4$ blocks instead of $8 \times 8$. Because of the improved prediction process in H.264/AVC compared to earlier standards, less spatial correlation exists to be removed by transform coding. Hence, the $4 \times 4$ transform is sufficient for removing correlation, in addition it gives less noise around edges compared to the $8 \times 8$ transform [84]. The transform coefficients are then quantized and entropy coding is performed on the quantized coefficients, with two different methods supported in H.264/AVC [84].

### 3.2.2 Network Abstraction Layer (NAL)

The H.264/AVC standard was designed to be efficient for a variety of applications and transmission systems. Examples of applications are: broadcasting, storage, conversational services, video-on-demand, and multimedia streaming. The transmission systems range from low bitrate, error prone wireless systems such as 802.11 and UMTS to high bitrate, error free fiber networks [84]. This is made effective by use of a NAL. Also, since the NAL adjusts the encoding process to fit the transmission system, better robustness against data errors and loss is achieved, since the error resilience tools can be customized to the transmission system.

The NAL divides the encoded data from the VCL into NAL Units (NALU) [35]. The NALUs are different for packet-based systems and byte-stream systems, where extra header information is needed for the latter to identify the NALUs within the stream, while this information is included in the data packet for packet-based systems. A set of NALUs are then assembled in NAL access units, where the decoding of a NAL access unit results in one video picture. In addition, non-VCL NALUs are employed for more efficient transmission of infrequently changing information needed for decoding [84].

For real-time transmission, the NALUs are put into the payload of an RTP packet. These NALUs should not be larger than the minimum MTU size of the transmission network, in order to avoid packet fragmentation. In this sense, each packet contains one slice of encoded video data and can be decoded individually [35].

## 3.3 Slice-based H.264/AVC Video Encoding

For a video stream transmitted over a network, the packet loss probability and queueing delay are of utmost importance. The objective of the explicit slice-based video encoding scheme proposed in [11] is therefore to reduce the burstiness of H.264/AVC encoded video and thereby reduce the packet loss and packet delay.

When applying the slice-based video encoding scheme, the frames of a GOP are decomposed into fixed size slices. A fixed pattern is then applied for the mode type selection (I or P) on the slice level as compared to the frame level for standard frame-based encoded video as illustrated in Figure 3.3.

The mode type selection is a property of the H.264/AVC standard, and mode types can be selected on the macroblock level. Instead of intra coding the first frame of each GOP as in standard frame-based video encoding shown in the upper part of Figure 3.3, a slice of each frame is intra coded in the slice-based scheme as

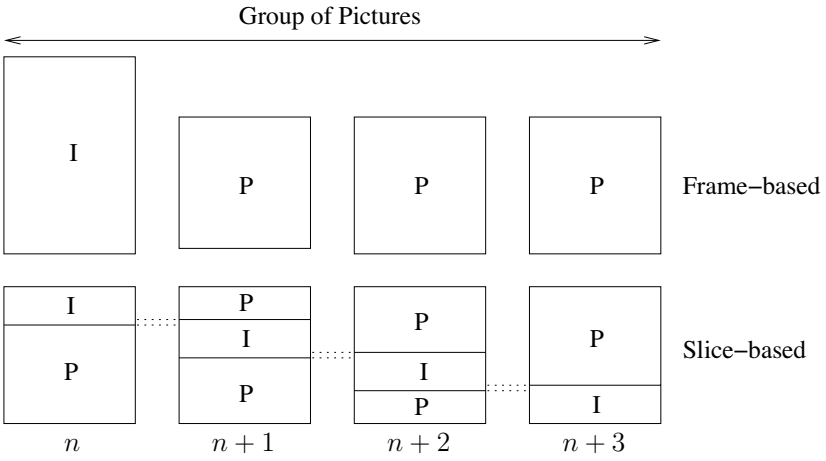


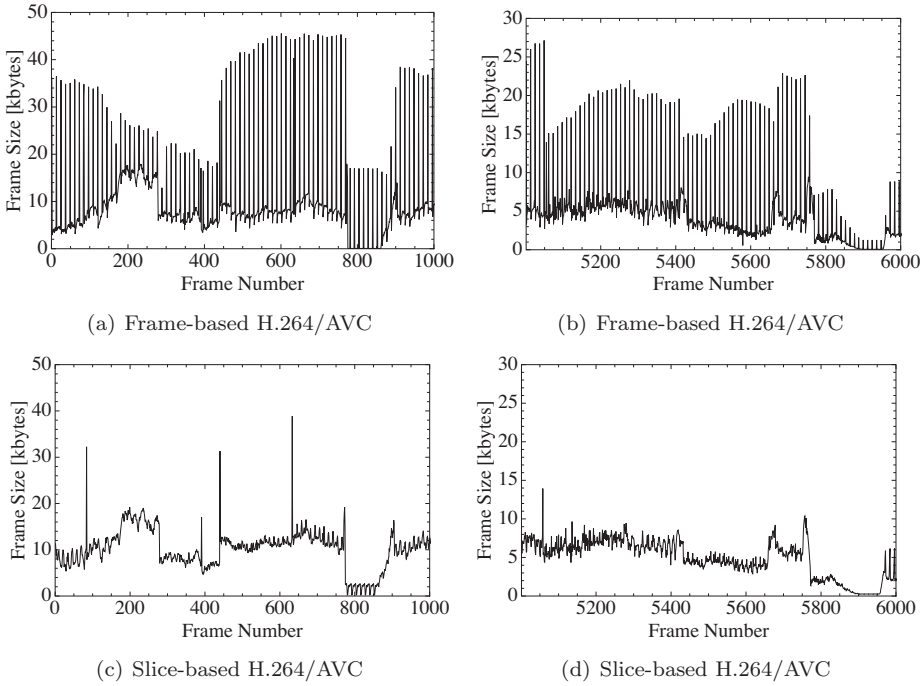

**Figure 3.3:** The mode type selection for the frame-based and slice-based video encoding schemes.

shown in the lower part of the figure. Consecutive frames belonging to the same scene will then have similar sizes, reducing the burstiness compared to frame-based encoding where the intra coded frames are significantly larger than the predicted frames. To prevent error propagation, a small overlap of the intra coded slices in consecutive frames is needed for the slice-based video encoding scheme, resulting in a slightly increased average bitrate compared to frame-based encoded video. The overlap is made equal to the maximum length of the motion vector in vertical direction to prevent errors from propagating upwards in the consecutive frames. The effect of the slice overlap for error robustness is described in more details in [11] and also investigated in more details in [79], where also the advantages of overlap versus no-overlap are studied. As expected the overlap option is most beneficial compared to no-overlap when the packet loss probability increases.

As explained earlier, the objective of the slice-based scheme is to reduce the burstiness of encoded video, and hence reduce the packet loss and delay. However, the increased bitrate for the slice-based encoded video due to the partially overlapping I slices will influence the gain of the scheme compared to regular frame-based encoded video. The network performance for slice-based encoded video compared to frame-based encoded video is investigated using network simulations in Chapter 4.

## 3.4 Description of the Sample Traces

The slice-based scheme was implemented in the H.264/AVC reference software version JM 10.1 [85] as described in [11]. The scheme does not require any modifications to the standard decoder and only the configuration of the encoder must be changed.




(a) Frame-based H.264/AVC

(b) Frame-based H.264/AVC

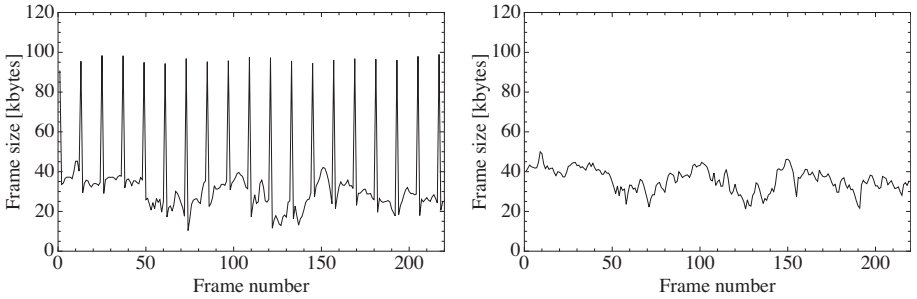
(c) Slice-based H.264/AVC

(d) Slice-based H.264/AVC

**Figure 3.4:** The generated frames for the StEM clip, standard frame-based H.264/AVC and slice-based H.264/AVC encoded video, respectively.

Throughout this work, two slice-based encoded test sequences are employed. The clips are encoded using intra and predictive coded slices. Part of the StEM [86] clip with frequent scene changes and large variations in the motion levels in consecutive scenes is used as one of the test sequences. The number of encoded bytes per frame for two sections of the video is shown in Figure 3.4.

When comparing the slice-based encoded stream and the standard frame-based encoded stream, the intra coded frames dominate and cause burstiness for the frame-based encoded stream while the slice-based encoded video stream is much smoother. However, the average bitrate is higher for the latter, due to the overlapping of the intra coded slices as explained above. The slice-based encoded stream is dominated by large peaks in the frame sizes because of scene changes. For the scene changes, the encoder selects intra coded mode also for the predicted slices because of the prediction errors, resulting in a larger size (in terms of number of bytes) for the scene change frames. These scene change frames are present for the frame-based encoded video as well, as can be seen in Figure 3.4(a) and 3.4(b). However, they are similar in size to the I frames and therefore more difficult to detect. The occurrence of large frames only at scene changes for the slice-based encoded video simplifies the scene change detection for slice-based encoded video streams compared to frame-based encoded video streams, as is




(a) Frame-based H.264/AVC (b) Slice-based H.264/AVC

**Figure 3.5:** The generated frames for the Mobile clip, standard frame-based H.264/AVC and slice-based H.264/AVC encoded video, respectively.

**Table 3.1:** The encoding parameters for the StEM and Mobile clips.

| Parameter | StEM | Mobile |
|---|---|---|
| GOP size | 12 | 12 |
| Frame rate | 30 fps | 15 fps |
| Frame size | 720x576 | 720x576 |
| Slice size | 720x48 | 720x48 |
| I slice size | 720x64 | 720x64 |
| Macroblock size | 16 x 16 | 16 x 16 |
| Maximum length of motion vector | [-16;15.5],[-16;15.5] | [-16;15.5],[-16;15.5] |

explained in Chapter 4.

It is also obvious that the average frame size varies over consecutive scenes and shows signs of non-stationarity. Both the scene lengths and the average bitrate are different for the first 1000 frames compared to frames 5000-6000, with frequent scene changes in the first part of the sequence and more infrequent scene changes in the period between frame 5000 and 6000. Additionally, the dependence between the size of the scene change frames and the average frame size in the consecutive scene is visually observable.

Also, the well known "Mobile" video clip without scene changes is investigated and the number of encoded bytes per frame for the frame-based and slice-based streams is shown in Figure 3.5. As can be seen in Figure 3.5(b), there are no large frames for the slice-based stream and hence no scene changes exist for the Mobile clip. The bitrate is smooth for the slice-based stream compared to the frame-based stream where the large intra coded frames dominate. The higher average bitrate for the slice-based stream can also be seen in the figures.

Both of the clips are encoded with a GOP size of 12 frames while the frame rate is 30 and 15 frames/second respectively for the StEM and Mobile clips. The encoding parameters are summarized in Table 3.1, showing the different size of the regular slices and the intra coded slices because of the overlap.

**Table 3.2:** Statistics for the frames of the slice-based encoded StEM and Mobile clips (frame sizes in kbytes).

| Sequence | Sample Size | Minimum | Maximum | Average | StDev |
|---|---|---|---|---|---|
| StEM | 7190 frames | 0.181 | 50.294 | 8.764 | 7.038 |
| Mobile | 220 frames | 21.210 | 49.970 | 35.869 | 5.800 |

The statistical description of the frame sizes for the test sequences is given in Table 3.2, showing the main characteristics of the frames. The frames for the StEM clip are more variable in the number of bytes than the frames from the Mobile clip. This is reflected both in the minimum and maximum of the frame sizes as well as in the standard deviation.

# Part II

# Characterization of Slice-based H.264/AVC Encoded Video Traffic

The results in this part have been published as follows:

Astrid Undheim, Yuan Lin, and Peder J. Emstad. "Characterization of Slice-based H.264/AVC Encoded Video Traffic." In *Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN)*, Toulouse, France, February 2007.

Astrid Undheim and Peder J. Emstad. "Characterization of Slice-based H.264/AVC Encoded Video Traffic Using Token Buckets." *Telecommunication Systems, Springer*, 39(2), October 2008.

# Chapter 4

# Traditional Characterization

In this chapter, the slice-based encoded StEM clip as described in Section 3.4 is studied. This video clip is dominated by large frames caused by scene changes. The characteristics of the video stream divided into scenes using a scene change detection algorithm are then investigated, looking at the marginal distributions and the correlations for the scenes and frame sizes. A simulation study is also performed, to investigate the multiplexing properties and the buffer overflow probabilities for the slice-based encoded stream compared to the frame-based encoded stream.

The study shows that frame correlation is present only at low lags and the periodic correlation structure seen in frame-based encoded video is mostly removed for the stream under study. Results from simulations show that the slice-based encoded video stream performs better than the frame-based encoded video stream in terms of lower packet loss probabilities and lower average packet delay as long as the buffer size is small or the link utilization is moderate.

## 4.1 Introduction

Video characterization has been an important topic for a long time and is an important prerequisite for developing traffic models. Studying the statistical properties of video traffic is especially important here, since the slice-based video encoding scheme produces video traffic with different statistical characteristics than regular frame-based encoded video traffic. The most important characteristics for a video stream are the marginal distributions and the correlation functions since they influence the performance experienced when a video stream is sent through a network. The distribution of the scene lengths, the GOP sizes, and the frame sizes are required. The correlation should be estimated on different levels, the autocorrelation for the scene lengths and the frame sizes as well as the correlation between the first frame in each scene and the ordinary frames in the scenes are of interest. Slice-based H.264/AVC encoded video has some properties that are different from standard frame-based H.264/AVC encoded video and this calls for new approaches. The GOP structure is less dominant, since no entire

frames are intra coded. Also, the first frame in each scene is larger than the ordinary frames because of the prediction error as explained in Section 3.4 and must be handled separately. For the standard frame-based encoded video, these scene change frames are comparable in size to the I frames and are therefore not as prominent as they are for the slice-based encoded video.

### 4.1.1 Related Work

Scene changes are an important part of recorded video and several works use scene statistics to characterize the video traffic. Hence, several scene change detection algorithms have been introduced in the literature. In [87], the scene changes for video sequences encoded by an intrafield/interframe Differential Pulse-Code Modulation (DPCM) coding scheme are detected by identifying frames with a large number of bits compared to the previous and consecutive frame. However, this algorithm is not useful for video encoded using a GOP structure, where the large I frames will cause false positives. Therefore, the algorithm is refined in [88] to look at the size of the I frames only and it is used for scene change detection for MPEG-2 encoded video. Next, in [89], a similar scene change detection algorithm based on the size of the I frames is introduced for MPEG encoded video. Here, an increase in the I frame size over two consecutive I frames is needed to detect a scene change. This algorithm could also be further developed to look at all frame sizes instead of only the I frames. Finally, in [90], scene changes are detected based on changes in the GOP size for MPEG encoded video.

The distribution of the scene lengths has also been investigated in several other works. In [87], a large number of video clips is investigated and the scene length distributions are found to match one of the Gamma, Weibull, or the Generalized Pareto distributions. In [91], the scene length distribution in number of GOPs is modeled using a Geometric distribution based on results from references therein. Also in [88], a Geometric distribution is used for the scene lengths.

For the frame size distribution, it is usually distinguished between scene change frames and ordinary frames for non-GOP encoded video, and between I, P, and B frames for video encoded with a GOP structure. In [87], scene change frames are found to be closest to the Gamma or Weibull distributions while the regular frames in the scenes are Pareto distributed. In [89], the frame size distributions for all frame types are found to match a Lognormal distribution for MPEG video. In [92], both I, P, and B frames sizes as well as GOP sizes are found to match the Gamma and Lognormal distributions. In [88], the frame size distribution for MPEG video is modeled using a hybrid Gamma/Pareto distribution for all frame types, the scene change frames are similar in size to the I frames and are modeled as regular I frames. In [93] and [94], the size of the I, P, and B frames for MPEG video is modeled using a Gamma distribution and also in [95] it is found that both I, P, and B frames are Gamma distributed for H.264/AVC encoded video.

Scene lengths are generally believed to be uncorrelated, and independent, identically distributed (iid) scene lengths are modeled in [87, 96]. The correlation structure for non-GOP encoded video is studied in [87], and high correlation is found between the scene changes frames and the consecutive frame as well as

for the intra scene frames. For video encoded with a GOP structure, the frame correlation shows periodic peaks caused by the I frames, and the frame correlation is retained over scene boundaries as seen e.g., in [90, 97, 98].

It is widely known that bursty traffic may lead to buffer overflow and thereby packet loss and packet delay. This is shown using simulation of an Ethernet network in [99]. However, when a frame-based encoded stream experiences loss, the loss probability is higher for packets from I frames than for packets from P frames. This is obvious because of the higher number of bits for the I frames, and is also shown in [89]. The effect of packet losses from I, P, and B frames on the distortion is modeled in [54], with larger degradations from lost I slices than for P and B slices. For the slice-based encoded video there are no I frames, and the loss probability should be the same for all frames. It is therefore of great value to study the performance of the slice-based encoded video, to compare it with the frame-based video.

### 4.1.2 Chapter Outline

The rest of this chapter is organized as follows. The scene change detection for the slice-based encoded video stream is investigated in Section 4.2, using different algorithms from the literature. The marginal distributions for the scene length and the frame size are investigated in Section 4.3 while the correlation functions are investigated in Section 4.4. The simulation studies conducted to compare the network performance of the slice-based encoded video to that of standard frame-based encoded video are introduced in Section 4.5 and the results from the simulations are shown in Section 4.6. Finally, some conclusions are given in Section 4.7.

## 4.2 Scene Change Detection

For the StEM clip, the increased bitrate due to scene changes is well observable by peaks in the bitrate as can be seen in Figure 4.1. The stream is dominated by the scene properties, i.e., the bitrate is nearly constant within the scenes. Scene statistics are therefore used to analyze and characterize the video stream. In general, a scene is a portion of the video clip without shifts of view supplied by editing or sudden camera movement or zooming [100]. The scenes detected from the video trace should resemble the visually identifiable scenes. However, for video characterization, the interest is mainly in the changes in bitrate. Hence, some real scene changes may occur without a change in bitrate and bitrate changes may occur without a real scene change.

For the slice-based encoded video, no frames are entirely intra coded, except for the first frame in each sequence. This first frame must be intra coded since there are no previous frames to make predictions from. Also, the first frame in a scene will be large because of prediction errors when trying to predict the frame based on the previous frame which is from another scene. This simplifies the scene change detection, since the first frame in each scene is very large compared to the ordinary frames.

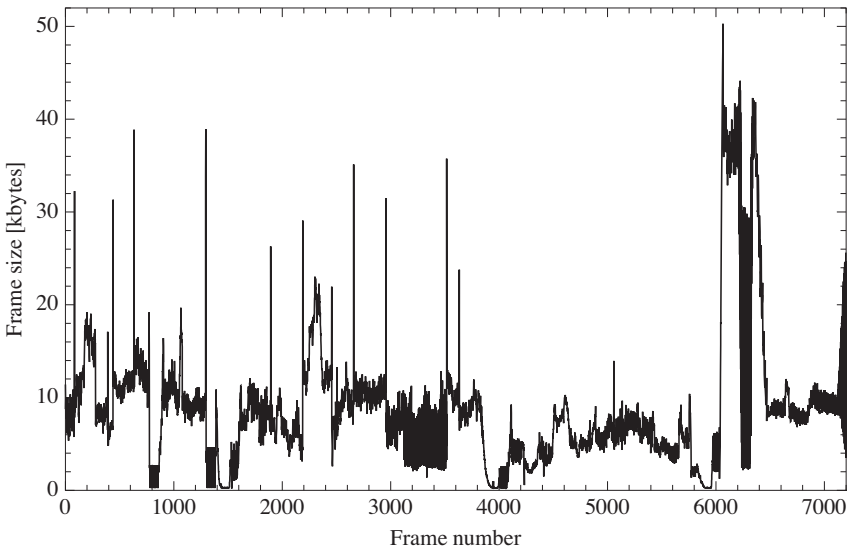



**Figure 4.1:** The frame sizes for the slice-based encoded StEM clip.

The scene change algorithms from the literature are investigated for the slice-based stream. The method from [87] can be used directly to identify the scene changes. This method locates frames that are large compared to the previous and consecutive frames and is hence applicable for scene change detection for the slice-based encoded video. When the method from [89] is applied to the slice-based encoded video, the results are unsatisfactory because an increase in frame size over two consecutive frames is needed to detect a scene change. For the slice-based encoded video, a single large frame indicates a scene change and the algorithm is unable to detect the visually observable scene changes in the video stream. Finally, the method from [90] can be applied for the slice-based stream without modifications. However, this method complicates the procedure since although the GOP number with the scene change is identified, the scene change frame is still unknown and requires further processing to locate. Because of the shortcomings of the I frame method from [89] and the GOP method from [90], the method from [87] is used to detect the scene changes in this chapter. Later, a new non-parametric method for scene change detection for slice-based encoded video is introduced in Chapter 6.

The scene changes for the slice-based encoded video are then detected by looking at the second difference in frame sizes divided by a number of previous frame sizes [87]:

$$\frac{[X(n+1)-X(n)]-[X(n)-X(n-1)]}{(1/6)\sum_{j=n-5}^{n}X(j)} < -\lambda \tag{4.1}$$

where $X(n)$ is the $n$th framesize and $\lambda$ is used as a threshold for the bitrate

**Table 4.1:** The results from the scene change detection, scene lengths given in number of frames.

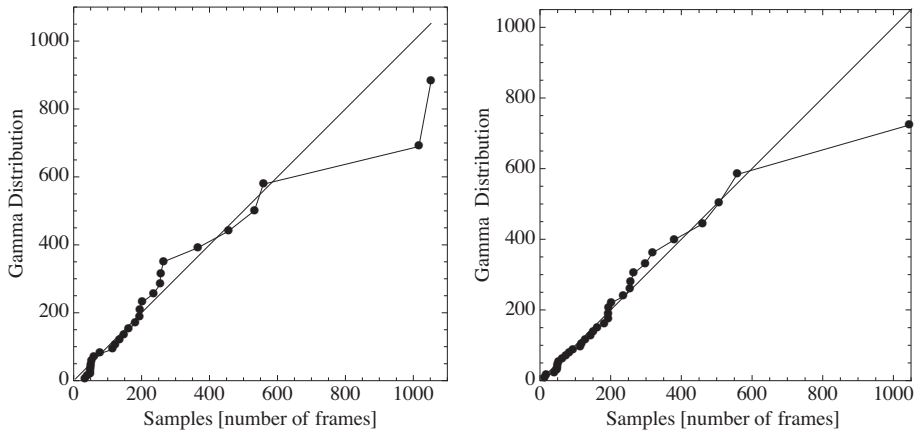
| $\lambda$ | Scene length | Averaging level | | |
|---|---|---|---|---|
| | | 6 | 12 | 24 |
| 0.4 | Sample Average | 215.8 | 197.8 | 182.5 |
| | Sample Variance | 67857.4 | 45603.6 | 40443.7 |
| 0.6 | Sample Average | 273.9 | 254.3 | 254.3 |
| | Sample Variance | 96079.6 | 90315.7 | 90433.0 |
| 0.8 | Sample Average | 274.2 | 264.0 | 264.0 |
| | Sample Variance | 96595.1 | 91942.6 | 91942.6 |
| 1.0 | Sample Average | 274.4 | 264.2 | 264.2 |
| | Sample Variance | 96855.7 | 92196.2 | 92196.2 |

increase needed to have a scene change.

The algorithm performs very satisfactory for the first 1000 frames of the sequence. However, in some other parts of the sequence, the motion is so high that prediction errors cause highly oscillating bitrate. For these parts, the algorithm will detect frequent scene changes that are false positives. Some constraints are therefore put on the scene change detection algorithm, saying that a scene change is detected only if no other scene changes were detected for the previous 12 frames. This means that the minimum scene length is set to 13 frames. In comparison, the minimum scene length is set to two GOPs in [89]. An additional problem becomes apparent with very small frames, where only a small increase in bitrate results in a scene change. The minimum scene change frame is therefore set to the average frame size, which is equal to 8.764 kbytes.

The two parameters for the scene change detection algorithm, $\lambda$ and the averaging level, were evaluated in [87]. A $\lambda$ value of 0.5 was used, and it was argued that averaging over the previous six frames is almost identical to averaging over 24 frames. For the slice-based stream, the algorithm is investigated for $\lambda$ values ranging from 0.4 to 1.0 and the averaging is performed over 6, 12 and 24 frames. The results from the scene change detection are shown in Table 4.1, where the average number of frames in the resulting scenes as well as the sample variance are shown.

The results are insensitive to the $\lambda$ value and the averaging level in some parts of the sequence, e.g., for the first 1000 frames where all scene changes are detected regardless of the parameter setting. It can also be seen that when an averaging over six previous frames is used, a $\lambda$ value ranging from 0.6 to 1.0 gives almost exactly the same average and variance. A $\lambda$ value of 0.4 is finally chosen and the averaging is done over 6 previous frames, based on the need to minimize the number of false positives while still detecting the most prominent scene changes.




(a) $\lambda = 0.4$, averaging 6 (b) $\lambda = 0.4$, averaging 24

**Figure 4.2:** QQ-Plot of the sample scene lengths versus the Gamma distribution.

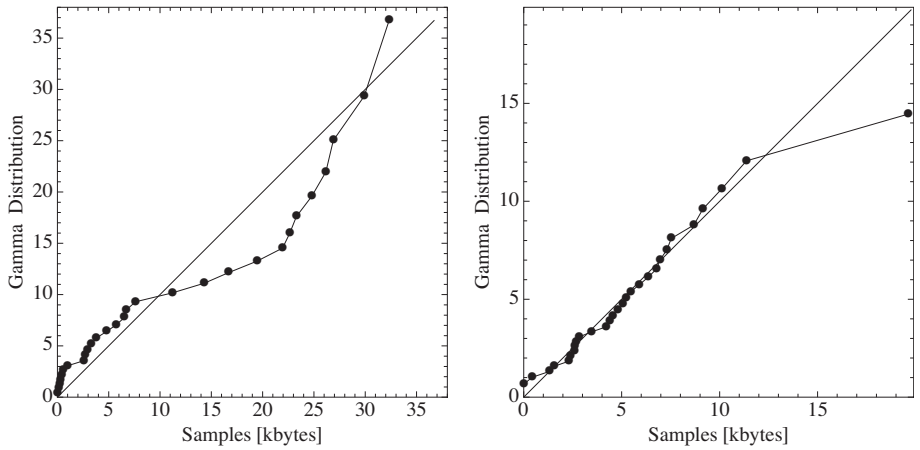
## 4.3 Marginal Distributions

The marginal distributions of the scene lengths and the frame sizes are investigated next.

### 4.3.1 Scene Length Distribution

The scenes are identified using the algorithm from Equation 4.1 with $\lambda = 0.4$ and the averaging is done over six frames as explained above. A Quantile-Quantile (QQ) plot, which is a common goodness-of-fit test, is used for analyzing the marginal distribution of the scene lengths. A QQ-plot shows the theoretical quantiles from a distribution with mean and variance equal to the sample average and sample variance against the samples. Distributional similarity between the two data sets is verified when the points fall on a straight line. In Figure 4.2, the sample scene lengths for different parameter settings are plotted against the Gamma distribution with mean and variance equal to the sample average and sample variance found in Table 4.1.

Since the test clip is short, the number of scene changes is small. This means that an insufficient number of samples exists to verify any distribution. However, the samples fall close to the straight line except for the largest samples, which are too large compared to the theoretical quantiles. These outliers are caused by the variations in the test stream, where most of the scene changes are frequent, but some parts of the stream have very few scene changes. This means that more samples are needed to verify the right tail of the distribution. The sample distribution has a longer right tail than the Gamma distribution, which means that the largest samples are larger than the corresponding theoretical quantiles and an additional distribution could be used for modeling the right tail.




(a) Sample scene change frame (b) Sample average frame size

**Figure 4.3:** QQ-Plot of the scene change frames and the average frame sizes in each scene versus the Gamma distribution.

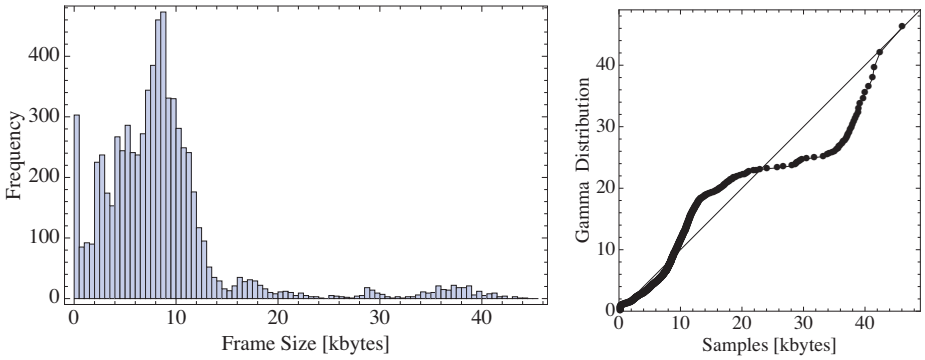
For the present samples, the Geometric distribution with parameter 0.05 gives the exact same match as the Gamma distribution and could be used for modeling the scene lengths as well. This corresponds to the results in [91], where the scene length in number of GOPs is modeled using a Geometric distribution.

### 4.3.2 Frame Size Distribution

The first frame in each scene is larger than the ordinary frames because of the prediction errors as explained earlier and is hereafter called a scene change frame. The marginal distribution of the scene change frames can also be estimated by the Gamma distribution and the QQ-plot is shown in Figure 4.3(a), where the minimum sample value is subtracted from the data. Although the samples do not match the Gamma distribution perfectly, the match is satisfactory in the right tail. For a video traffic model, it is especially important to match the right tail for a correct prediction of the buffer overflow probabilities.

The average frame size in each scene is calculated. The distribution of the average frame size is investigated and the samples with the minimum sample value subtracted are plotted against the Gamma quantiles in Figure 4.3(b). As for the scene lengths, the largest sample deviates from the theoretical quantiles and the right tail is heavier for the samples than for the theoretical distribution. This could be taken into account by using a second distribution to model the right tail. The outlier for the average frame size is caused by one particular part of the video stream where the bitrate suddenly increases over a short period equal to one scene, as can be seen in Figure 4.1. This part of the video stream is so different from the rest of the stream that it is difficult to match it to any distribution.

For the ordinary frames, the histogram of the frame sizes is shown in Figure




(a) Histogram (b) QQ-Plot

**Figure 4.4:** Histogram and QQ-Plot for the ordinary frames.

4.4(a) and the QQ-Plot in Figure 4.4(b). For the slice-based encoded video, each frame has an intra coded slice, and the distribution of the ordinary frames is different from the Gamma distribution which was found to match the P frames in [95]. Also the QQ-Plot shows large deviations from the straight line and then also the Gamma distribution. The single high bitrate scene is dominating here as well, with a local peak below 40 kbytes. In addition, there is a peak at low frame sizes, which reflects the highly variable bitrate for the video sequence under study. When the high and low bitrate frames are disregarded, the rest of the frames are closer to the Gamma distribution, supporting the view that the frame sizes are Gamma distributed for a video stream with a less variable bitrate than the stream under study.

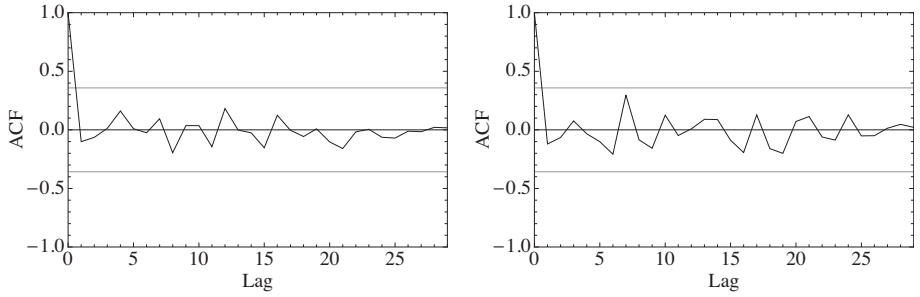
It is difficult to estimate the marginal distribution for both the scene change frames and the average frame size in the scenes, due to the non-stationarity of the stream. This is explored in Chapter 6, where the video frames are divided into classes in order to have more stationary frame sizes inside the classes.

## 4.4 Sample Correlations

The sample correlations for the video stream are derived on different levels to reflect the characteristics of the video stream. Both the Autocorrelation Functions (ACFs) and the Correlation Functions (CFs), e.g., between the scene change frame and the average frame size in a scene are estimated.

### 4.4.1 Scene Level

The ACFs for the length of the scenes as well as the size of the scene change frames are investigated. The sample autocorrelations of an iid sequence have the distribution $N(0, 1/n)$, where $n$ is the number of samples. This means that at least 95% of the sample autocorrelation values should be within the bounds




(a) ACF, length of scenes

(b) ACF, size scene change frame

**Figure 4.5:** The ACF for the length of the scenes and the size of the scene change frame.

$\pm 1.96/\sqrt{n}$ [101] for an iid sequence. This bound is used in the following to assess if the CFs are from independent samples or not. If the correlation is mainly within these bounds, it is regarded as a sign of independence. However, more tests are needed to prove that the samples are indeed independent.

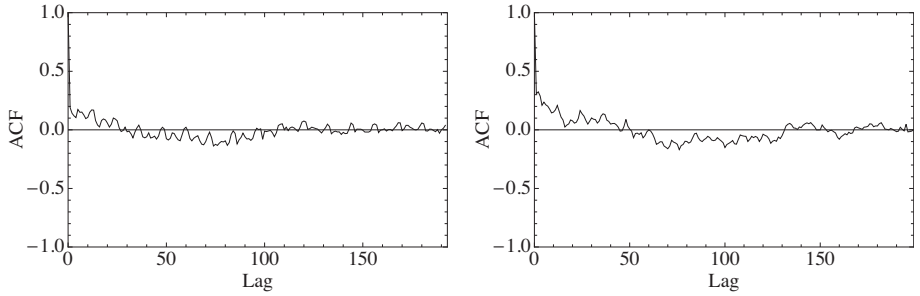
For the scene lengths, the ACF is well within the bounds $\pm 1.96/\sqrt{n}$ for lags $> 0$ as can be seen in Figure 4.5(a), which means that only negligible correlation is detected for the number of frames in consecutive scenes. For the size of the scene change frames, the ACF is shown in Figure 4.5(b). The size of the scene change frames shows signs of independence as well, based on the same arguments as above. Also, the CF between the length of the scenes and the size of the scene change frames is investigated and it is negligible on all lags. Hence, the iid assumption cannot be rejected for any of the scene level statistics. The marginal distributions of the scene lengths and the size of the scene change frames found in Section 4.3.1 are then the only statistics needed for modeling the scenes when developing a model of the video traffic.

### 4.4.2 GOP Level

For frame-based encoded video, the GOP correlation is dominating the ACF on the frame level, and this is also shown for standard frame-based H.264/AVC encoded video in [98]. For the slice-based H.264/AVC encoded video on the other hand, the GOP structure is imperceptible as can be seen for the StEM clip in Figure 4.1. This could also be seen in Figure 4.6, showing the ACFs for two different scenes. There is, as was expected, no distinctive correlation with a period of 12, which means that the GOP statistics can be omitted from the analysis.

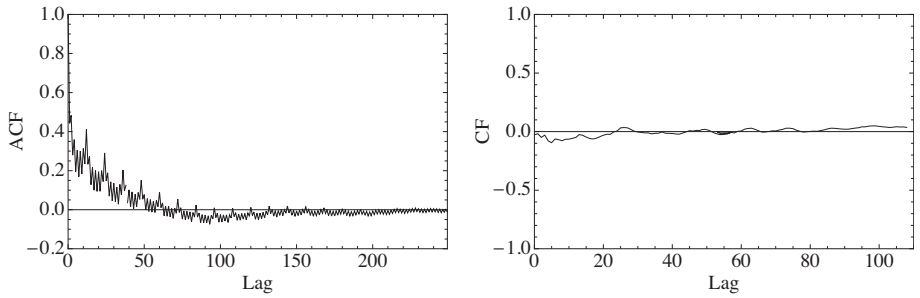
### 4.4.3 Frame Level

The overall ACF at the frame level is investigated by removing the scene change frame in each scene and looking at the ACF for the ordinary frames in each scene




(a) ACF, scene 6 (b) ACF, scene 21

**Figure 4.6:** The sample ACF for all frames in two particular scenes.



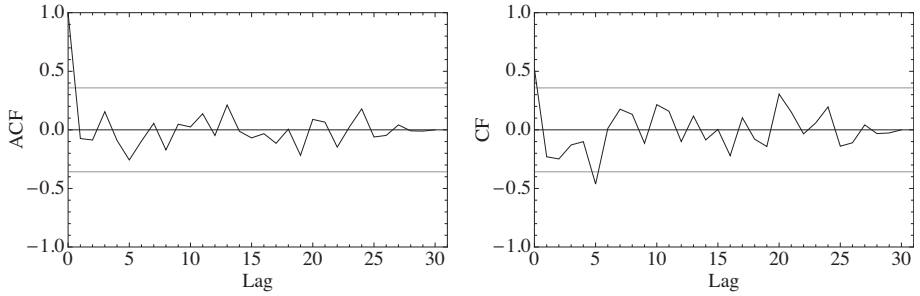

(a) The ACF for ordinary frames, averaged over the scenes (b) The CF for frames in different scenes

**Figure 4.7:** The correlation for ordinary frames.

and then averaging over the scenes, a method that was introduced in [87]. The overall ACF is then found by letting all scenes longer than $h$ contribute to the autocorrelation at lag $h$. The ACF is shown in Figure 4.7(a) and shows that the correlation is non-negligible at low lags, but is decreasing with increasing lags as was expected. A 12-frame periodicity in the ACF for low lags is also present; this is caused by some periodicity in individual scenes, but is not representative for the whole sequence.

Second, the correlation between frames in consecutive scenes is investigated, where correlation at lag 0 is the correlation between the $i$th frame in two consecutive scenes. The overall CF is then found in the same way as for the ACF in a scene. The CF is shown in Figure 4.7(b) and shows that there is no correlation between frames of consecutive scenes. This indicates that the scenes are correctly identified, since a scene change indicates a noticeable bitrate change and hence low correlation between the frame sizes in the different scenes.

The ACF for the average frame size in consecutive scenes is shown in Figure 4.8(a). This ACF shows only negligible correlation as well. Finally, the CF for




(a) ACF average frame size

(b) CF for the size of the scene change frame and the average frame size

**Figure 4.8:** The ACF for the average frame size and the CF for the size of the scene change frame and the average frame size.

the size of the scene change frame and the average frame size in each scene is investigated. As was expected, this is non-negligible at lag 0 as can be seen in Figure 4.8(b). This means that the average frame size in a scene is dependent on the size of the scene change frame of the same scene, i.e., the first frame in the scene. At lags $> 0$, the CF is negligible, which strengthens the view that the scenes are independent, both with regard to the scene lengths and the size of the frames in each scene.

Having looked at the correlations on different levels, the correlation between the scene change frame and the average frame size in the same scene is the only non-negligible correlation in addition to the autocorrelation for the frames in the same scene. These correlations should be retained in a simulation model of slice-based H.264/AVC encoded video, e.g., using an Autoregressive (AR) model as has been done in [88] or alternatively a Gamma AR (GAR) model as in [102].

## 4.5 Network Simulations

Important information about a traffic stream can be found by looking at the characteristics of the stream, e.g., the marginal distributions and the correlation functions. However, when transmitting the stream over a network, other properties may be more important and should be investigated for both the frame-based and the slice-based encoded video using simulations. The Packet Loss Ratio (PLR), the packet delay and the distribution of the losses over I and P frames are of interest. In [89] it was shown that the loss probability is higher for I frames than for P and B frames for the same stream. In addition, the effect of a lost packet from an I frame can be more severe in terms of degraded quality than a lost packet from a P frame, since the former means that the picture will not be restored until the next GOP. This means that the performance with regard to error resilience may be higher for a slice-based encoded stream than for a frame-based encoded stream, even when the PLR is higher for the former. The error resilience for the

slice-based encoded video is also studied and compared to frame-based encoded video in [79].

The multiplexing gain for VBR video traffic is usually high and the gain for the slice-based video encoding in comparison to the frame-based video encoding should be studied. The multiplexing gain is defined in several different ways. In [103], the multiplexing gain is found as the ratio of the number of VBR sources to the number of CBR sources transmitted over the same link with the same PLR. A different definition is given in [89], where the multiplexing gain $\mu_p$ is defined as follows:

$$\mu_p = \frac{n \cdot \text{ utilization at } N = 1 \text{ and } PLR = p}{\text{utilization at } N = n \text{ and } PLR = p} \tag{4.2}$$

where $n$ is the number of streams in the aggregated stream and the utilization is defined as the ratio of the average bitrate of the video stream(s) to the capacity needed on the output link to achieve the required loss ratio.

However, this last definition is non-intuitive, because increasing the utilization for the multiplexed stream decreases the multiplexing gain, which is the opposite of what is expected. A new definition of the multiplexing gain is then proposed as follows:

$$\begin{aligned} \mu_p &= \frac{n \cdot \text{ capacity needed at } N = 1 \text{ and } PLR = p}{\text{capacity needed at } N = n \text{ and } PLR = p} \\ &= \frac{\text{utilization at } N = n \text{ and } PLR = p}{\text{utilization at } N = 1 \text{ and } PLR = p} \end{aligned} \tag{4.3}$$

These two expressions give the same results, however the former shows the multiplexing gain in terms of reduced capacity usage for the same PLR when sources are multiplexed. The latter expression gives a similar definition as Equation 4.2 in terms of the link utilization. The capacity needed to ensure the given loss ratio may also be termed the effective bandwidth [104].

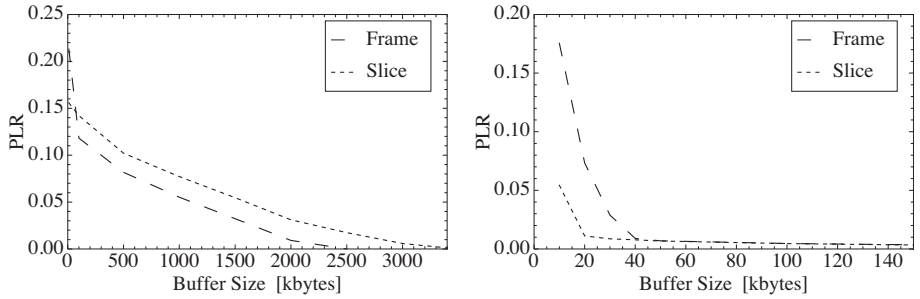
In [89], a fixed buffer size in bytes is used when calculating the multiplexing gain. Here, the multiplexing gain with a constant buffer size both in size and in terms of delay is evaluated. The latter means that the buffer size increases in accordance with the capacity to ensure a fixed maximum buffer delay.

## 4.6 Results from Simulations

The simulations are performed with a trace-based simulator implemented in Demos (Discrete Event Modelling on Simula) [105], implementing a single bottleneck link with a drop tail queue in the sender node. The queue length and the bottleneck capacity are varied.

### 4.6.1 Single Stream

The simulations investigating the PLR for a single stream are performed with two different capacities, 2000 kbps and 4000 kbps, both with varying buffer sizes. The average bitrate is approximately 6% higher for the slice-based encoded stream than




(a) PLR, capacity 2000 kbps

(b) PLR, capacity 4000 kbps

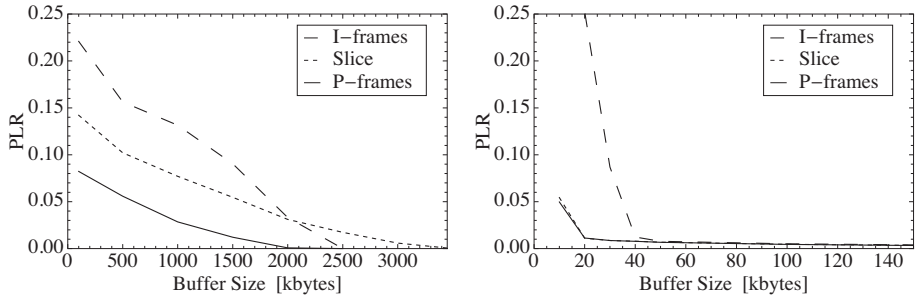
**Figure 4.9:** The PLR from simulations with a single stream over a bottleneck link with variable capacity and buffer size.

for the standard frame-based encoded stream. The results from the simulations are shown in Figure 4.9.

When the link is heavy loaded, there will almost always be packets under transmission and the PLR will be higher for the slice-based encoded video traffic because of the higher bitrate. However, when the link is less loaded, the peaks in the bitrate caused by the I frames for the standard frame-based encoded stream will fill up the buffer and cause packet loss. This will be most prominent for small buffer sizes. Therefore, the loss ratio for the standard frame-based encoded stream will be higher than for the slice-based encoded stream for small buffer sizes, as can be seen in Figure 4.9(b).

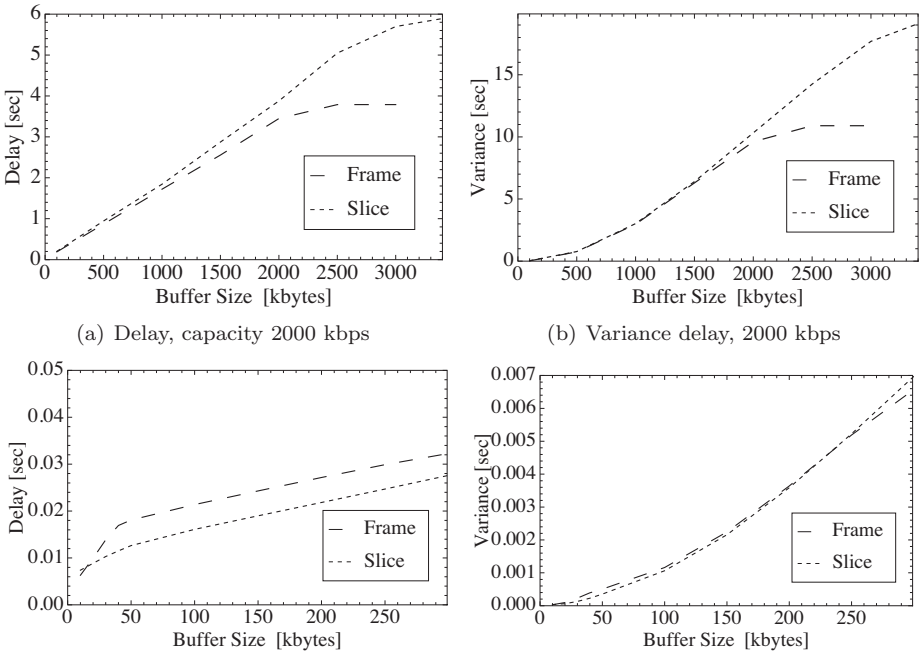
As mentioned earlier, the packet loss probability is different for packets belonging to I, P, and B frames [89]. This is so because these frames have different sizes and when congestion occurs, packets from I frames are more likely lost due to their larger sizes compared to B and P frames. For the slice-based encoded stream, the bitrate is more constant inside a scene and the packet loss probability should be similar for all packets. The scene change frames on the other hand are larger than the ordinary frames and are most likely lost in the case of network congestion. The PLRs for the packets from I and P frames are shown in Figure 4.10 for the single stream, indicating the same trends as in [89], with a higher loss ratio for the I frames compared to the P frames.

When the utilization on the link is high, as can be seen in Figure 4.10(a), the PLR for the packets from I frames is much higher than for the packets from P frames. For a lower utilization, as is shown in Figure 4.10(b), the same is true for low buffer sizes. This shows that for situations where loss occurs, either because of a low link capacity or a small buffer, packets belonging to I frames will experience a higher PLR than packets belonging to P frames. Even when the slice-based encoded stream experiences a higher loss ratio, the performance may be better than for the frame-based encoded stream because of the high number of lost packets from I frames for the latter. This effect is modeled in [54], where the loss from I, P and B slices have different impact on the distortion.




(a) PLR, capacity 2000 kbps (b) PLR, capacity 4000 kbps

**Figure 4.10:** The PLR for I and P frames for the frame-based scheme together with the PLR for the slice-based stream.




(a) Delay, capacity 2000 kbps (b) Variance delay, 2000 kbps

(c) Delay, capacity 4000 kbps (d) Variance delay, 4000 kbps

**Figure 4.11:** The average packet delay and the variance of the packet delay from simulations with a single stream on a bottleneck link.

The packet delays with different capacities on the link are also investigated. The average packet delay as well as the variance when the buffer size is varied are shown in Figure 4.11. When the capacity is low, the average delay is slightly higher for the slice-based encoded stream, coinciding with the results from the packet loss analysis. However, the frame-based encoded stream experiences a higher average packet delay than the slice-based encoded stream with a capacity of 4000 kbps, except for very low buffer sizes. The higher burstiness for the frame-based encoded stream is most prominent when the utilization on the link is moderate, as was also seen for the packet loss. The variance of the packet delay follows the same trends as the average delay, with the variance slightly higher for the slice-based encoded stream for low capacities and slightly higher for the frame-based encoded stream for high capacities. When the buffer size is large, both for low and high capacities, the packet loss ratio for the frame-based encoded stream is close to zero, and the variance is higher for the slice-based encoded stream.

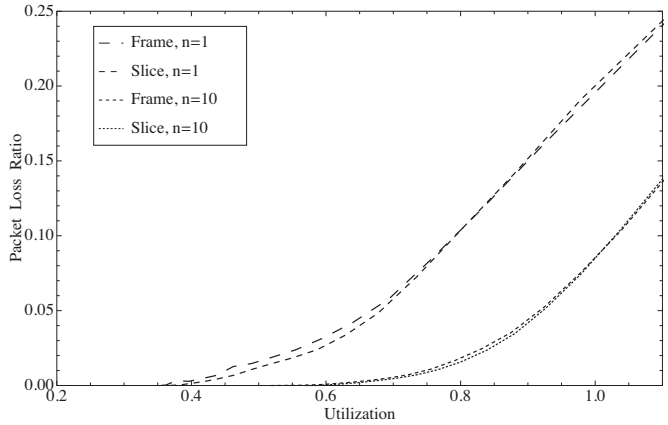
### 4.6.2 Aggregated Stream

The aggregation is performed over a single link and a total of ten identical streams are aggregated. The aggregation method from [103] is used, producing a circular list of the original video stream and choosing the start frame of each multiplexed stream uniformly over the total stream length. The starting time of each stream is also chosen uniformly over one inter frame period (33ms) to avoid synchronized sources. The PLRs from the simulations are shown versus the utilization on the link in Figures 4.12(a) and 4.12(b).

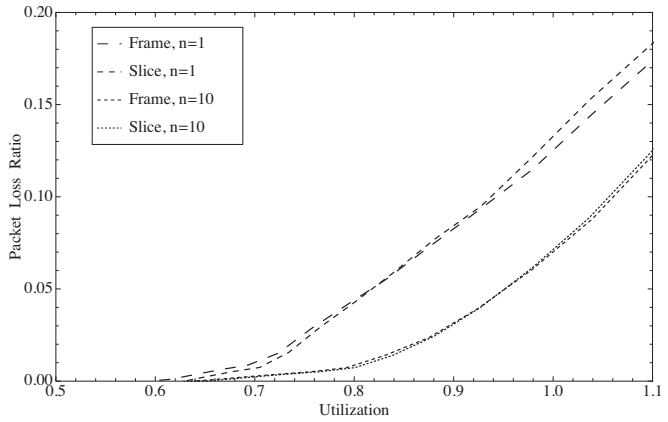
With a small buffer in the sender, the loss ratio for the slice-based encoded stream is lower than for the frame-based encoded stream as long as the utilization is kept moderate, as is seen in Figure 4.12(a). This also holds for the case with a larger buffer as can be seen in Figure 4.12(b), where the PLR for the slice-based encoded stream is lower than for the frame-based encoded stream when the utilization is below 0.7-0.8. However, the difference is smaller than for the case with a smaller buffer. When the buffer size is constant in delay as shown in Figure 4.12(c), the PLR for the single stream is lower for the slice-based stream for all levels of utilization. However, with multiplexed streams, the PLRs are almost the same for the slice-based and frame-based encoded video.

### 4.6.3 Multiplexing Gain

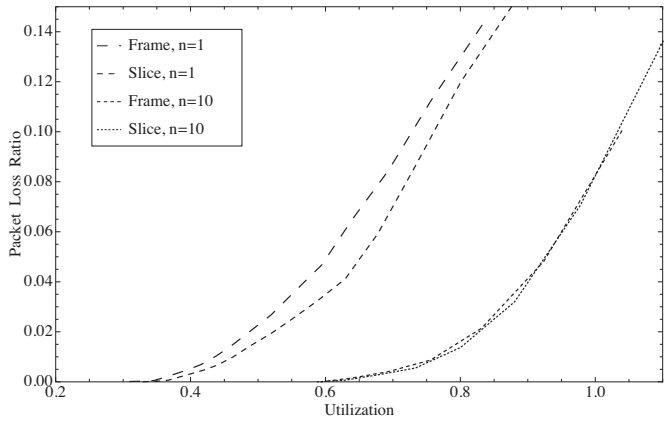
The multiplexing gain is obtained graphically for the aggregated streams in Figure 4.12(a)-4.12(c) and is shown in Table 4.2 for a PLR equal to 0.01. As is shown, the gain is higher for the frame-based encoded video, with all buffer sizes. However, the difference in percent is higher for the smallest buffer size. The multiplexing gain is one of the advantages with VBR encoded video. The multiplexing gain is high because of the highly variable bitrate, (see e.g., [89, 103]). The slice-based encoding scheme removes some of the variations in the bitrate, resulting in a lower multiplexing gain for the slice-based scheme compared to the frame-based scheme, at least when the number of multiplexed sources is high. A high number




(a) Buffer size 100 kbytes




(b) Buffer size 1000 kbytes




(c) Buffer size 100 ms

**Figure 4.12:** The PLR from simulations with aggregated streams and different buffer sizes.

**Table 4.2:** The multiplexing gain with 10 multiplexed sources.

| Slice-based | | | Frame-based | | |
|---|---|---|---|---|---|
| 100 kB | 1000 kB | 100 ms | 100 kB | 1000 kB | 100 ms |
| 1.582 | 1.147 | 1.680 | 1.670 | 1.156 | 1.857 |

of multiplexed streams is most common in the core network, where the capacity is high. For the access network on the other hand, the number of multiplexed streams will be lower and the slice-based scheme will be more favourable compared to the frame-based scheme.

## 4.7 Conclusion

A video clip encoded using the slice-based video encoding scheme is characterized. The scene changes are more dominant in the slice-based encoded stream than in the corresponding frame-based encoded stream since there are no intra coded frames between scene changes for the former. The slice-based encoded video is therefore characterized mainly on the scene level. This view is defended by the correlation analysis, showing that no dominating correlations exist on a larger scale than in a scene, where there is non-negligible correlation between the size of the first frame in each scene and the average frame size in the same scene. Also, it is shown that there is non-negligible correlation for frames inside one scene. However, there is no evidence of correlation at the GOP level and the GOP statistics can be omitted from the study. The marginal distributions of the scene lengths and the average frame sizes in the scenes are both found to be close to a Gamma distribution for the stream studied. However, because of the non-stationarity in the frame sizes, it is difficult to estimate the distributions.

The simulation results show that the slice-based stream performs better than the frame-based stream when the utilization is low or the buffer size is small. When looking at the PLR for packets belonging to I and P frames, it could be argued that the performance may be better for the slice-based scheme even with a higher loss ratio because of the high number of lost packet from I frames for the frame-based scheme. Also the packet delay for a given link capacity is higher for the frame-based encoded stream than for the slice-based encoded video stream due to higher burstiness for the former. Analysis of the multiplexing gain with ten multiplexed sources shows that the gain for the slice-based encoded video is reduced compared to the frame-based encoded video. This is explained by less burstiness. However, large gains are achieved by multiplexing slice-based encoded streams as well.

# Chapter 5

# Token Bucket Characterization

In this chapter, slice-based encoded video streams are characterized using the token bucket and leaky bucket traffic models and compared to standard frame-based encoded streams. Both lossless and loss bounded token bucket and leaky bucket models are investigated and the high quantiles are found for the amount of loss. It is shown that the reduced burstiness for the slice-based encoded video leads to lower token bucket parameters than for the frame-based encoded video for the Mobile clip without scene changes. For the StEM clip with scene changes, a larger reduction in the token bucket parameters compared to the frame-based encoded video is experienced when a small amount of delay or loss is allowed. Finally, an approach to estimate the parameters for the token bucket model using simple characteristics of the slice-based encoded StEM clip is developed.

## 5.1 Introduction

Traditional traffic characterization focuses on estimating specific properties of the traffic under study, such as marginal distributions and correlations, as seen in Chapter 4. These characteristics are often used for developing traffic models for different purposes. Traffic models needed for resource reservation require a bounding function for the stream under study and a different approach to the traffic characterization is needed. The token bucket traffic model is one such bounding function and its variant, the dual token bucket traffic model is used in the TSpec [17] of IntServ [7] for reserving resources for a given flow. In addition, the token bucket traffic model is employed in DiffServ [8] networks, where token bucket constrained flows can be given absolute delay bounds. Streaming video and video conferencing are delay critical services as shown in Section 2.2 and these are examples of applications that will benefit from resource reservation.

The token bucket traffic model is specified using the parameters $\rho$ and $\sigma$, where $\rho$ is the token generation rate and $\sigma$ is the token bucket size. Tokens are put into the token bucket at the rate $\rho$ until it reaches the maximum size $\sigma$. A traffic stream that is sent through a token bucket acquires a number of tokens equal to the packet sizes of the stream. For traffic characterization, the task is to find the

appropriate token bucket parameters ($\rho$, $\sigma$) such that all packets of the traffic stream can be sent without delay.

The token bucket traffic model is used in this work for comparing the resource reservation needs for slice-based and frame-based encoded video streams, where the token bucket parameters are found from simulations. The StEM and Mobile video clips described in Section 3.4 are studied. Real-time video can tolerate a certain amount of packet loss without degradation in quality, as discussed in Section 2.2. The token bucket parameters when fulfilling a maximum packet loss ratio for the frame-based and slice-based encoded streams are therefore also investigated. Additionally, a data buffer is included in the token bucket model for traffic shaping. High quantiles for the packet loss, given the token bucket parameters, are also investigated. Finally, statistics about the scene changes and the average frame sizes in the scenes are used to deduce the token bucket parameters analytically for the StEM clip.

### 5.1.1 Related Work

Due to the importance of the token bucket traffic model for admission control and resource reservation in the Internet, several techniques have been introduced for estimating the token bucket parameters analytically or experimentally. In [106], an equivalent queueing system is used for estimating the token bucket parameters for a voice over IP application when a stochastic model of the traffic flow exists. Token bucket parameter estimation for VBR MPEG encoded video is performed in [107] using a combined Markov chain and Discrete Autoregressive (DAR) model of the video traffic, in [108] using a Switched Batch Bernoulli Process (SBBP) as a model of the MPEG encoded video traffic, and in [109] using an analytical model with the average data rate as an estimate of the token generation rate. In [110] it is shown that long-range dependence affects the token bucket parameters and a Fractional Brownian Motion (FBM) model is used for modeling the long-range dependent traffic. For all of these methods, it is required that a stochastic model of the traffic under study exists, or that the traffic trace is available.

### 5.1.2 Chapter Outline

The rest of this chapter is organized as follows. In Section 5.2, the approach to token bucket parameter estimation using simulations is described, while the results are shown in Section 5.3. The token bucket curves for the slice-based and frame-based encoded streams are found through simulations and both lossless models and models with restricted loss probability are investigated. Furthermore, the token bucket parameters are found when the frames of the streams are reshuffled, in order to investigate the long- and short-term correlation. In Section 5.4, the token bucket parameters are found analytically for the slice-based encoded stream with scene changes. Finally, some conclusions are given in Section 5.5.

## 5.2 Token Bucket Parameter Estimation from Simulation

In this section, the method for estimating the token bucket parameters from simulations is described, as well as some proposals for choosing the optimal parameters from the token bucket curve. Two different traffic models, the token bucket traffic model and the leaky bucket traffic model as described in Section 2.3, are used in this work. The simple token bucket traffic model causes no traffic shaping and is therefore useful for characterizing the traffic, while the token bucket model with loss is used for traffic shaping needed for resource reservation. The leaky bucket has a data buffer in addition to the token bucket, thereby shaping the traffic stream by delaying some of the packets. Both models are parameterized for the studied slice-based and frame-based encoded video clips and lossless models and models allowing a certain percentage of packet loss are investigated as described next.

A property of token bucket constrained flows described in [9] can be used to estimate the token bucket parameters for a traffic flow using simulations. A bounding function $Q(t)$ is defined as follows:

$$Q(t) = \sup_{s \leq t} \{A(s,t) - \rho(t-s)\} \tag{5.1}$$

when this holds, $Q(t) \leq \sigma$ if and only if the flow defined by the arrival process $A$ is token bucket $(\rho, \sigma)$ constrained.

This means that the burst parameter $\sigma$ can be calculated for a given token generation rate by letting the flow under study, defined by the arrival process $A$, be input to a virtual Single Server Queue (SSQ) system with $\rho$ as the service rate and an infinite buffer size. $Q(t)$ is then the queue length at time $t$ and the maximum queue length throughout the simulation can be used as an estimate for $\sigma$. The simulations are performed with different values for the rate parameter, resulting in several sample pairs ($\rho$, $\sigma$).

The simulations give the curve of the token bucket parameters for the stream under study, the optimal parameter pair however, is an open question. A starting point, employed e.g., in [110], originally proposed in [12], is to identify a knee point at the curve outside of which either the token generation rate or the token bucket size increase heavily with only a small reduction of the other parameter. The token bucket parameters should then be located in the vicinity of this knee point.

Next, in [111], it is suggested to use the maximum allowable delay, $D_{max}$, to set the token bucket parameters, where $\sigma = D_{max}\rho$ for a LR server [71] when the output scheduler is WF$^2$Q+ and it is assumed that the latency parameter is negligible.

Finally, for a GR server [73] with the allocated rate $r$ and error term $E$ to a flow which is token bucket $(\rho, \sigma)$-constrained (where $\rho \leq r$) it is guaranteed that the delay of every packet of the flow is bounded by [75]:

$$D \leq \frac{\sigma}{r} + E \tag{5.2}$$

For a FIFO scheduler, the theoretical error term $E$ is zero [74] and a deterministic bound for the delay is provided for token bucket constrained flows:

$$D \leq \frac{\sigma}{r} \tag{5.3}$$

The worst case delay is then given when the token generation rate is equal to the reserved rate for the flow at the server.

The presented results give two corresponding relations for the delay bound and the token bucket parameters, valid when the network element is specified either as a LR or GR server and with the specified output schedulers. In the next section, the token bucket curves found from simulations are analyzed together with the delay curves estimated from Equation 5.3.

## 5.3 Results from Simulations

This section shows the results from the token bucket parameter estimation using simulations. Both lossless and loss bounded token bucket and leaky bucket models are investigated. The lossless token bucket model is also investigated after reshuffling the frames of the video streams. Finally, the high quantiles for the losses at a token bucket with given parameters are presented.

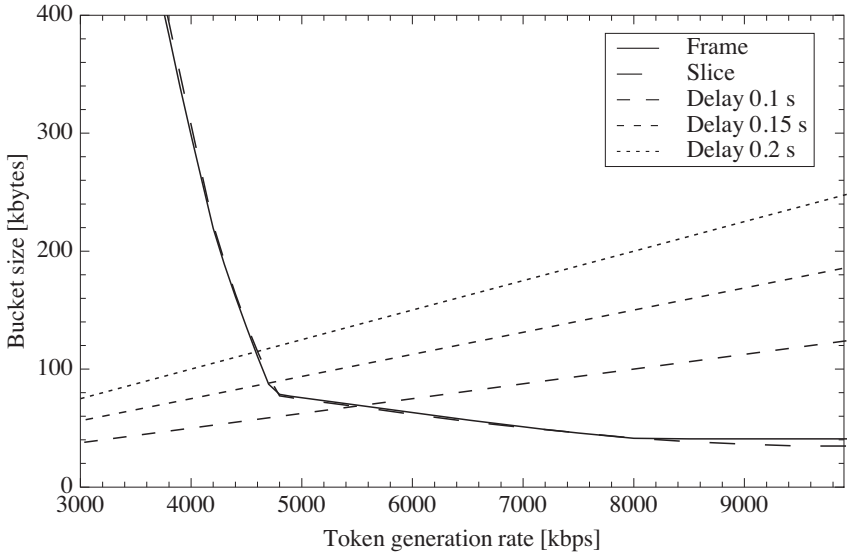
### 5.3.1 Lossless Token Bucket

The token bucket curves for the StEM clip and the Mobile clip found from simulations are shown in Figure 5.1(a) and Figure 5.1(b) respectively, together with the delay curves estimated from Equation 5.3, with $r = \rho$. The first 6000 frames of the StEM clip are used, to avoid the high bitrate scene described in Chapter 4.

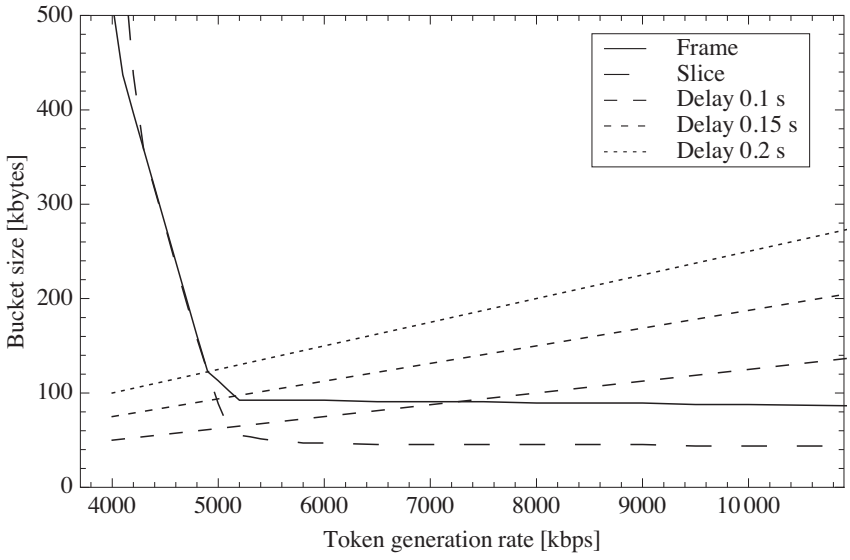
For the StEM video clip, a knee point is located between 4500 and 5000 kbps both for the slice-based and the frame-based stream, suggesting this as a starting point for the token generation rate. The token bucket size at the knee point is around 80 kbytes. The maximum average bitrate in a scene is approximately equal to 4000 kbps, which means that the token generation rate at the knee point is well above the maximum average bitrate in a scene.

To fulfill an end-to-end delay requirement of 150 ms for video conferencing [31], the network delay should be considerable lower than 150 ms, which justifies 100 ms as the delay bound. Figure 5.1(a) shows that in order to fulfill this delay requirement, a token generation rate slightly higher than the token generation rate at the knee point is needed for the StEM clip. The token bucket parameters for the slice-based stream are lower than the parameters for the frame-based stream for all token bucket pairs fulfilling this delay requirement. However, since the token bucket traffic model gives an upper bound on the traffic in a time period, the high bitrate scene change frames for the slice-based stream result in very similar token bucket curves for the slice-based and frame-based streams.

For a video stream with no scene changes on the other hand, the difference between the token bucket curves for the slice-based and frame-based streams is




(a) StEM clip




(b) Mobile clip

**Figure 5.1:** The token bucket curves together with the delay curves.

much larger, in favor of the slice-based scheme. This is illustrated in Figure 5.1(b), where the token bucket curves for the Mobile clip, without scene changes, are shown. As can be seen, the token bucket sizes for the slice-based stream for token generation rates higher than at the knee point are now significantly lower than those for the frame-based stream. For very low token generation rates however, the higher average bitrate of the slice-based stream implies that a larger token bucket size is needed than for the frame-based stream.

For the Mobile video clip, the knee point is located around 5000 kbps both for the frame-based and the slice-based streams. The average bitrate for the slice-based stream is 4300 kbps, which is again lower than the token generation rate at the knee point. To satisfy the delay constraint of 100 ms, a token generation rate slightly higher than the rate at the knee point is needed for the slice-based stream, while a token generation rate much higher than the rate at the knee point is needed for the frame-based stream.

### 5.3.2 Lossless Leaky Bucket

With a data buffer introduced, the video streams are smoothed at the token bucket, delaying those packets that arrive at an empty token bucket as shown in Section 2.3.2. The number of packets delayed in the data buffer as well as the average and sample variance of the delay experienced by these packets are studied. The results from the simulations with variable token generation rate and token bucket size for the frame-based and slice-based encoded streams, StEM and Mobile clips, are found in Table 5.1.

The results for the StEM clip show that because of the burstiness of the streams, introducing only a very small delay at the token bucket can significantly reduce the token bucket size. The total number of packets delayed for the slice-based stream is lower than for the frame-based stream because of the lower burstiness for the former, where the scene change frames are the main cause of burstiness. Without scene changes, as for the Mobile clip, the difference between the slice-based and frame-based streams is even larger and with a token bucket size of 40 kbytes, only a very few packets of the slice-based stream are delayed in the data buffer for the studied token generation rates.

The total end-to-end delay bounds for the video streams will be the same with the added data buffer, since the size of the data buffer and hence the maximum delay is equal to the reduction in the token bucket size. This holds since the loss probability at a leaky bucket is dependent on the token bucket size and the data buffer size only through their sum [76]. A smaller token bucket size then leads to a lower delay bound through the network, but with an added delay from the data buffer.

### 5.3.3 Loss Bounded Token Bucket

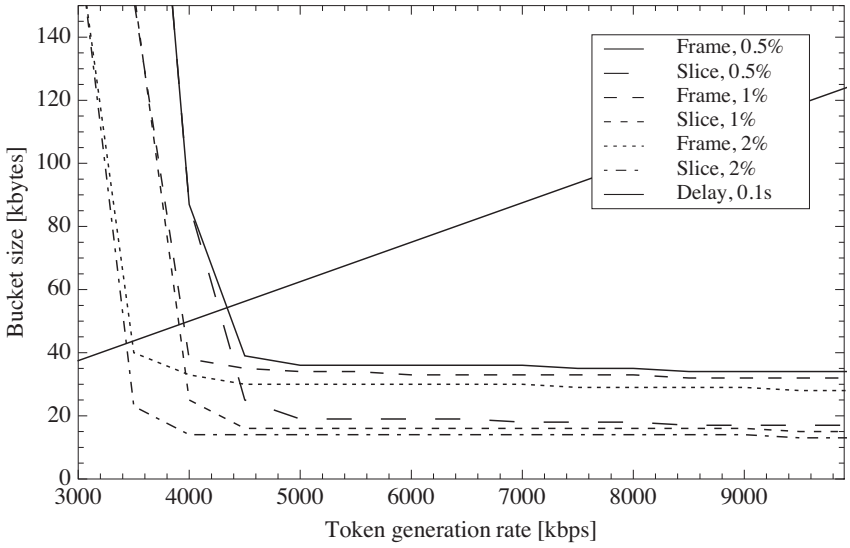
Real-time video traffic can tolerate a certain amount of packet loss without noticeable degradations in quality, see e.g., [31,67] and the discussion in Section 2.2. For resource reservation purposes, a token bucket model satisfying a predetermined

**Table 5.1:** Statistics for the data buffer.

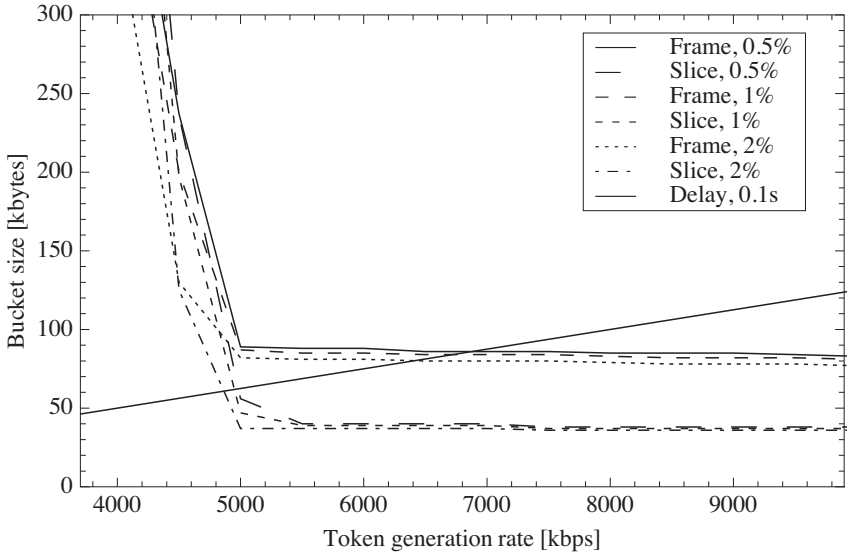
| | | Rate $\rho$ [kbps] | Size $\sigma$ [kbytes] | # delayed packets | Average delay [s] | Max delay [s] | Variance delay |
|---|---|---|---|---|---|---|---|
| StEM | Frame-based | 5000 | 20 | 2581 | 0.0154 | 0.0957 | $1.27{\cdot}10^{-4}$ |
| | | | 30 | 683 | 0.0120 | 0.0791 | $1.27{\cdot}10^{-4}$ |
| | | | 40 | 67 | 0.0213 | 0.0634 | $3.48{\cdot}10^{-4}$ |
| | | 6000 | 20 | 2039 | 0.0126 | 0.0627 | $6.90{\cdot}10^{-5}$ |
| | | | 30 | 579 | 0.0088 | 0.0489 | $4.83{\cdot}10^{-5}$ |
| | | | 40 | 51 | 0.0114 | 0.0358 | $9.78{\cdot}10^{-5}$ |
| | Slice-based | 5000 | 20 | 614 | 0.0196 | 0.0985 | $2.80{\cdot}10^{-4}$ |
| | | | 30 | 207 | 0.0172 | 0.0795 | $3.35{\cdot}10^{-4}$ |
| | | | 40 | 50 | 0.0276 | 0.0623 | $3.32{\cdot}10^{-4}$ |
| | | 6000 | 20 | 137 | 0.0198 | 0.0615 | $2.83{\cdot}10^{-4}$ |
| | | | 30 | 55 | 0.0215 | 0.0472 | $1.75{\cdot}10^{-4}$ |
| | | | 40 | 32 | 0.0154 | 0.0309 | $6.97{\cdot}10^{-5}$ |
| Mobile | Frame-based | 6000 | 20 | 2318 | 0.0343 | 0.1035 | $7.47{\cdot}10^{-4}$ |
| | | | 40 | 998 | 0.0325 | 0.0762 | $3.33{\cdot}10^{-4}$ |
| | | | 60 | 460 | 0.0202 | 0.0434 | $1.27{\cdot}10^{-4}$ |
| | | 8000 | 20 | 1965 | 0.0237 | 0.0757 | $3.99{\cdot}10^{-4}$ |
| | | | 40 | 723 | 0.0245 | 0.0521 | $1.90{\cdot}10^{-4}$ |
| | | | 60 | 342 | 0.0165 | 0.0306 | $5.54{\cdot}10^{-5}$ |
| | Slice-based | 6000 | 20 | 1925 | 0.0133 | 0.0372 | $4.54{\cdot}10^{-5}$ |
| | | | 40 | 21 | 0.0041 | 0.0096 | $4.84{\cdot}10^{-6}$ |
| | | | 60 | 0 | - | - | - |
| | | 8000 | 20 | 1840 | 0.0097 | 0.0269 | $2.36{\cdot}10^{-5}$ |
| | | | 40 | 8 | 0.0035 | 0.0053 | $1.58{\cdot}10^{-6}$ |
| | | | 60 | 0 | - | - | - |

ratio of lost packets is therefore of great value, reducing the resource reservation needs for a bursty stream. The simulation results for the cases with 0.5, 1, and 2% packet loss are shown in Figure 5.2(a) and 5.2(b) for the StEM clip and the Mobile clip, respectively. The token bucket parameters are found by simulating a number of token bucket sizes for each token generation rate and finding the lowest token bucket size that provide the required bound on the loss ratio.

For the StEM clip, the slice-based encoded video stream performs significantly better than the frame-based encoded stream for all packet loss ratios. For the highest packet loss ratio the token bucket size is almost twice as large for the frame-based stream as for the slice-based stream for token generation rates higher than at the knee point. The slice-based encoded stream has very few large frames, which is clearly appreciated when a small amount of packet loss is allowed. For the Mobile clip, the reduction in the token bucket parameters for the slice-based stream under the different packet loss ratios is lower compared to the no-loss case (see Figure 5.1(b)) than for the StEM clip because of the frequent scene change frames for the latter. However, the token bucket sizes for the slice-based stream




(a) StEM clip




(b) Mobile clip

**Figure 5.2:** The token bucket curves with packet loss ratio 0.5, 1, and 2%.

are still significantly lower than for the frame-based stream for the same token generation rates, which is favorable when reserving resources.

### 5.3.4 Loss Bounded Leaky Bucket

Finally, the token bucket parameters are investigated for a loss bounded leaky bucket, hence introducing both delay and loss. The results from the simulations with a data buffer of 10 ms and 0.5, 1 and 2% loss are shown in Figure 5.3(a) and 5.3(b) for the StEM clip and the Mobile clip, respectively. The same method as for the loss bounded token bucket is applied for finding the minimum token bucket size for each token generation rate.

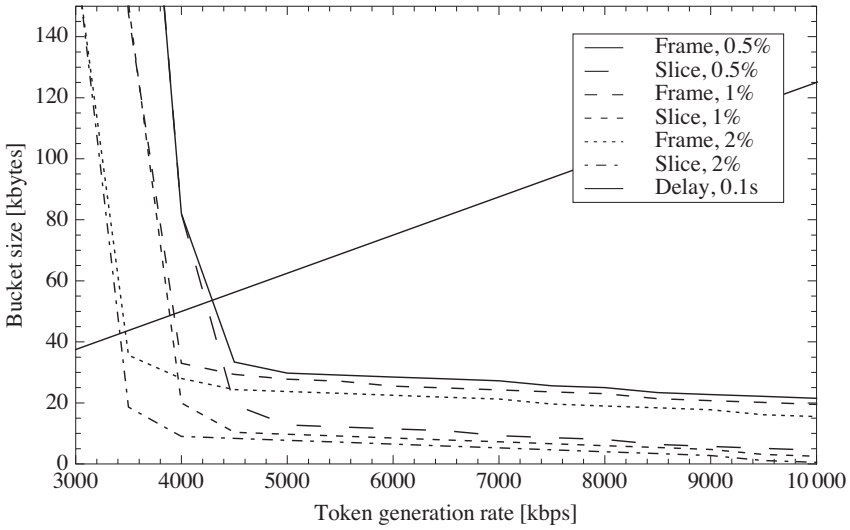
The figures show that by introducing a data buffer with 10 ms delay at the token bucket and allowing a small amount of packet loss, the token bucket parameters are significantly reduced. For the StEM clip, the token bucket size is reduced from approximately 60 kbytes for the simple token bucket to approximately 15 kbytes for the loss bounded leaky bucket for a token generation rate of 5000 kbps and a PLR of 2%, while for the Mobile clip a reduction from 88 kbytes to approximately 30 kbytes is experienced for the same token generation rate and PLR.

### 5.3.5 Token Bucket Curves after Reshuffling

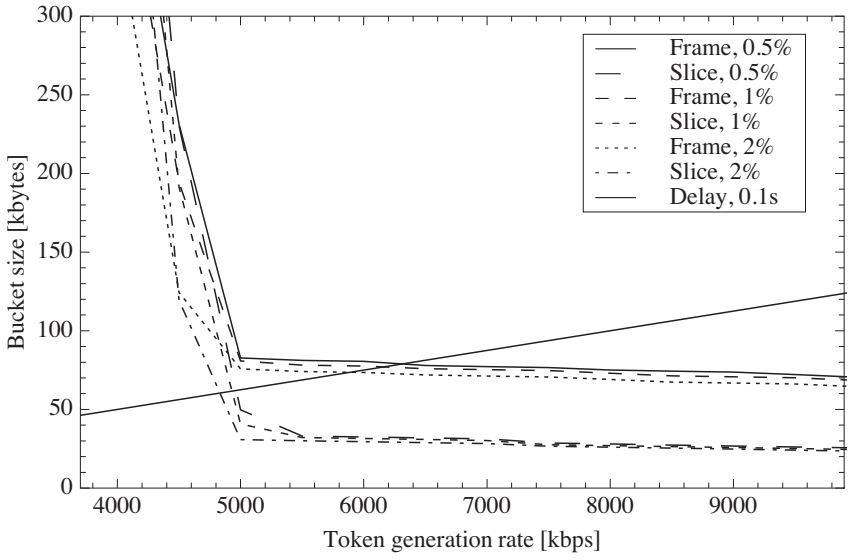
By removing the large intra coded frames for the slice-based encoded video, the correlation between frames of different scenes is reduced. In Chapter 4, only negligible correlation between the scenes is detected for the slice-based encoded StEM clip. For the frame-based stream on the other hand, the lag-12 correlation will remain positive and high across scene boundaries because of the large intra coded frames as shown in [90, 97, 98].

The effect of long-range dependence on the token bucket curve has previously been investigated, using reshuffling of the video stream on different time scales [110], not taking the scenes into account. Investigating the token bucket curves after reshuffling the frames of the StEM clip internally in the scenes as well as reshuffling the scenes, is of interest. The former reveals the effect of short-term correlation, while the latter shows the long-term correlation over the scenes. This analysis is interesting in order to see if there is differences in the correlation for the slice-based and frame-based encoded video streams which affect the token bucket curves in different ways.

The token bucket curves for the StEM clip after reshuffling are shown in Figure 5.4. Different random seeds are employed for the reshuffling, with similar results, hence only one of the result sets is presented here. The token bucket curve for the externally reshuffled stream is identical to the unshuffled stream both for the slice-based and frame-based encoded streams, except for low token generation rates where the token bucket size is so high that correlation between scenes influence the token bucket parameters. The internal reshuffled stream has lower token bucket parameters than the unshuffled stream for the slice-based stream, but slightly higher token bucket sizes for high token generation rates for the frame-based

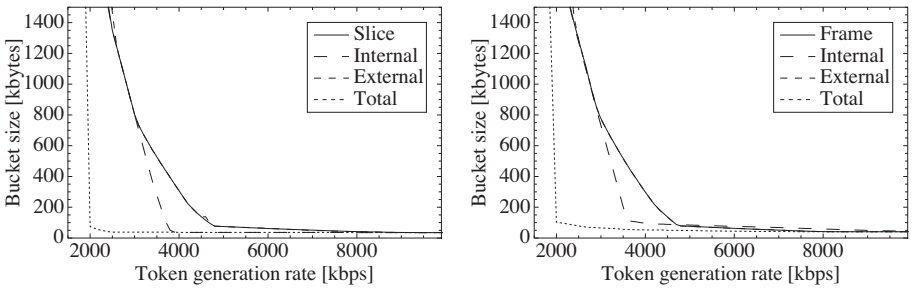



(a) StEM clip




(b) Mobile clip

**Figure 5.3:** The token bucket curves with 10 ms data buffer and packet loss ratios 0.5, 1, and 2%.




(a) StEM clip - Slice (b) StEM clip - Frames

**Figure 5.4:** The token bucket curves with reshuffling of the scenes (external), the frames inside the scenes (internal), and all frames (total) respectively.

stream. This indicates correlation between the frames in each scene, while the correlation between frames in different scenes is minimal. Furthermore, the internal reshuffled stream is identical to the total reshuffled stream for token generation rates above 4000 kbps and identical to the original stream for token generation rates lower than 3000 kbps for the slice-based stream. The token bucket sizes in these bounding points correspond to the size of the largest frame and to the length of two frame periods, respectively. For the slice-based stream, the internal reshuffled version is close to the unshuffled stream because the correlation is high inside a scene. For the frame-based stream, internal reshuffling may put two or more intra coded frames closer together, which causes higher token bucket sizes for the same token generation rates for the internal reshuffled stream compared to the unshuffled stream.

Since the token bucket curves for the regular stream and the external reshuffled stream are the same for the StEM clip, except for very low token generation rates, the scenes can be looked at independently when determining the token bucket parameters. This is exploited for the analytical token bucket estimation in Section 5.4.

For the Mobile clip there are no scene changes and only total reshuffling is performed. The results are shown in Figure 5.5. Similar results as for the StEM clip are observed, where the reshuffled stream has higher token bucket sizes than the unshuffled stream for high token generation rates for the frame-based stream. For the slice-based stream the curves are the same for the reshuffled and the unshuffled version except for low token generation rates.

### Test of Long-Range Dependence

Long-Range Dependence (LRD) indicates the degree to which an incident in a time series is dependent on an incident far away in time, see e.g., [112]. A common definition is that the ACF sums to infinity for a long-range dependent time series. The long-range dependence may be seen in comparison to the effect of reshuffling

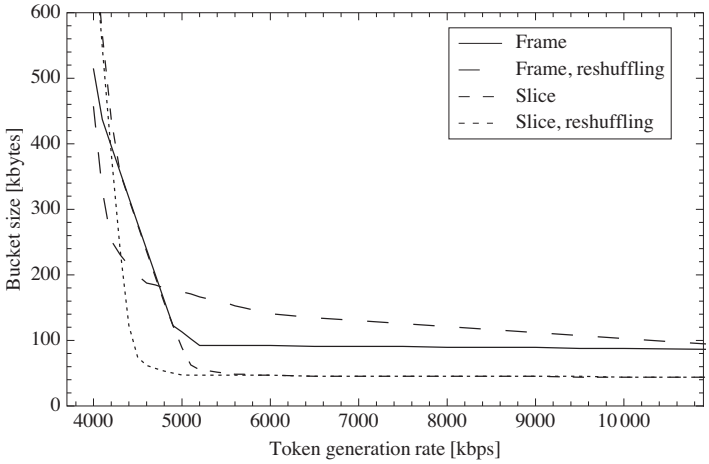



**Figure 5.5:** The token bucket curves for the Mobile clip after total reshuffling.

the frames. The Hurst parameter $H$ $(1/2 < H < 1)$ estimates the degree of long-range dependence for a time series, where $\rho(h) = c \cdot h^{2(H-1)}$ for a constant $c > 0$ and a large $h$ [112]. A value close to 0.5 indicates short-range dependence and a value close to 1 indicates long-range dependence. The Hurst parameters for the different reshuffled streams as well as the original streams are estimated using the method from [113], assuming that the processes are exactly second-order self-similar:

$$\hat{H}_n = \frac{1}{2}[1 + \log_2(1 + \hat{\rho}_n(1))] \tag{5.4}$$

where $\hat{\rho}_n$ is the sample ACF for the stream.

When $H$ is unknown, as in our case, the 95% confidence interval centered around $\hat{H}_n$ can be simply estimated as [113]:

$$w_n = \frac{5}{\sqrt{n}} \tag{5.5}$$

where $n$ is the number of samples.

The results in Table 5.2 show that the Hurst parameters are almost identical for the original stream and the externally reshuffled stream for both the slice-based and frame-based streams for the StEM clip. This is in agreement with the token bucket curves, which are also very close for the two streams. The internal reshuffled version for the slice-based stream has a lower Hurst parameter, showing less long-range dependence, while the total reshuffled stream has a Hurst parameter around 0.5, which reflects the short-range dependence of the totally randomized stream. However, the assumption that the processes are exactly second-order self-similar is not checked and the Hurst parameters should only be used to compare the reshuffled streams with the original stream.

**Table 5.2:** The Hurst parameter for the different streams.

| | | Stream | $\hat{\rho}_n(1)$ | $\hat{H}_n$ | 95% CI |
|---|---|---|---|---|---|
| StEM | Slice | Regular | 0.8725 | 0.9525 | [0.8879, 1.0170] |
| | | External reshuffling | 0.8738 | 0.9530 | [0.8884, 1.0175] |
| | | Internal reshuffling | 0.6026 | 0.8402 | [0.7757, 0.9048] |
| | | Total reshuffling | -0.0213 | 0.4845 | [0.4199, 0.5490] |
| | Frame | Regular | 0.1942 | 0.6280 | [0.6926, 0.5635] |
| | | External reshuffling | 0.1958 | 0.6290 | [0.5645, 0.6936] |
| | | Internal reshuffling | 0.1936 | 0.6277 | [0.5631, 0.6922] |
| | | Total reshuffling | -0.0243 | 0.4823 | [0.4177, 0.5468] |
| Mobile | Slice | Regular | 0.8866 | 0.9579 | [0.6200, 1.2958] |
| | | Total reshuffling | -0.1013 | 0.4230 | [0.0851, 0.7608] |
| | Frame | Regular | -0.0805 | 0.4394 | [0.1016, 0.7773] |
| | | Total reshuffling | -0.0205 | 0.4850 | [0.1472, 0.8229] |

### 5.3.6 High Quantiles for the Packet Loss

The high quantiles for the loss at the token bucket give the upper bounds for the amount of loss that can occur with a given probability. To estimate the high quantiles for the packet loss at the token bucket, the token bucket must be small enough and the token generation rate must be high enough to let the token bucket always be full at the arrival of a new video frame, in order to provide independent losses. This means that the token bucket should be filled up in a time period equal to a frame period minus the length of the maximum frame size, $s_{max}$, with a given token generation rate. The maximum token bucket size is then calculated as follows:

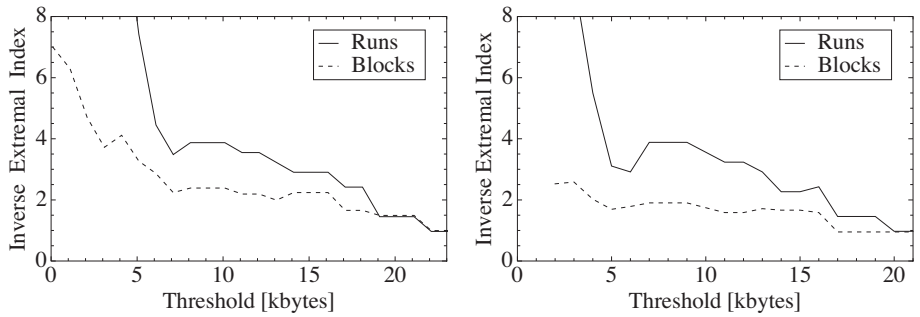
$$\sigma_{max} \leq \rho \cdot \Big(\frac{1}{f} - \frac{s_{max} \cdot 8}{c}\Big) \tag{5.6}$$

where $f$ is the number of frames per second and $c$ is the capacity on the incoming link. For all token buckets smaller than this maximum, the number of lost packets from a frame is only depending on the size of the frame and not the token bucket content at arrival.

The high quantiles of the packet loss for given token bucket parameters are of interest. The high quantiles for the specific token generation rate of 5000 kbps are investigated, giving a maximum token bucket size of 18.8 kbytes when $c$ is 100 Mbps, $s_{max}$ is 40.385 kbytes and $f$ is 30. Two different token bucket sizes, 14 and 16 kbytes are used, since these correspond to approximately 2 and 1% packet loss respectively for the slice-based encoded StEM clip as is shown in Figure 5.2(a). The loss samples, giving the amount of loss per frame, are then found from simulations.

The (1-$p$)th quantile for the packet loss is estimated using the Weissman's estimator [114]:

$$x_p^w = Y_{(n-k_0)}\Big(\frac{k_0+1}{(n+1)p}\Big)^{1/\hat{\alpha}} \tag{5.7}$$




(a) Buffer size 14 kbytes, $k$=31 (b) Buffer size 16 kbytes, $k$=33

**Figure 5.6:** The extremal index estimation by blocks and runs estimates.

where $Y_{(1)} \leq Y_{(2)} \leq ... \leq Y_{(n)}$ are the order statistics of the packet losses, $Y_1, ..., Y_n$ and $\hat{\alpha}$ is an estimate of the tail index $\alpha$. $\alpha = 1/\gamma$, where $\gamma$ defines the shape of the tail, and $\alpha$ is found e.g., using Hill's Estimator [115]:

$$\hat{\alpha}^H = \Big(\frac{1}{k_0}\sum_{i=1}^{k_0} \ln Y_{(n-i+1)} - \ln Y_{(n-k_0)}\Big)^{-1} \tag{5.8}$$

where the smoothing parameter $k_0$ is found using the bootstrap method as described in in more details in Chapter 6. The bootstrap method produces resamples of the initial data set and chooses the $k_0$ that gives the minimum Mean Squared Error (MSE).

The assumptions for using the Weissman's estimator are that the underlying data are stationary and independent, the latter checked by the extremal index. In addition, the data should have a positive tail index.

The extremal index is estimated using the blocks or runs estimators, also described in Chapter 6, where the sample is divided into fixed size blocks. The inverse extremal index then denotes the number of exceedances per cluster, where a cluster is a block with at least one exceedance over a given threshold. The extremal index should be close to 1 to assume asymptotic independence. The block and runs estimators for a fixed block size, $k$ and variable threshold are shown in Figure 5.6. For both buffer sizes, the extremal index is close to 1 for high thresholds. This information means that independence or weak-dependence of the losses can be assumed.

The Hill's estimates, which are higher than 1 for both buffer sizes, are shown in Table 5.3. Hence, both requirements for the Weissman's estimator are fulfilled and the estimator is used for estimating the high quantiles. The results are shown in Table 5.3. Loss above the high quantiles will only occur with a very low probability. For the token bucket size of 14 kbytes, the amount of loss from one frame will be within 42.049 kbytes with 99% probability.

**Table 5.3:** High quantiles for the losses.

| Token Bucket | $\hat{\alpha}^H$ | 99% | 99.9% |
|---|---|---|---|
| 14 kbytes | 1.330 | 42.049 kbytes | 237.478 kbytes |
| 16 kbytes | 1.381 | 43.202 kbytes | 228.888 kbytes |

## 5.4 Analytical Estimation of the Token Bucket Parameters

In the previous section it was shown that simulations provide a straightforward method for deciding the token bucket parameters for a video stream. However, the trace of the video stream under study must be available for analysis. Also, several simulation runs are needed to find the token bucket curve. Estimating the token bucket parameters based only on some simple characteristics of the video stream may therefore be of great value. The StEM clip is investigated since the Mobile clip only has one scene.

As seen from the simulation results, the token generation rate should be higher than the maximum average bitrate in the scenes to fulfill the delay requirements and only this part of the token bucket curve is investigated. The analysis is divided into two different cases.

**Case 1: Minimum Token Bucket Size.**

The minimum token bucket size is determined by the number of tokens needed to transmit a maximum sized frame, which is the maximum sized scene change frame, $s_{max}$. The token bucket will be empty at the departure of the last packet from the frame, which happens a time $t_s$ after the arrival of the first packet of the frame at the token bucket. Time $t_s$ is then given by:

$$t_s = \frac{s_{max} \cdot 8}{c} \tag{5.9}$$

where $c$ [$bps$] is the capacity on the incoming link to the token bucket.

Next, the token generation rate must be sufficiently high so that enough tokens are generated for frames arriving after the maximum sized scene change frame to be transmitted without loss. The maximum size of the frame arriving after a scene change frame is given by the maximum sized ordinary frame, $o_{max}$. As was found in Chapter 4, the correlation is high between the size of the scene change frame and the size of the consecutive frames and estimating the consecutive frame by the maximum size of an ordinary frame is reasonable. The time available to generate tokens for the maximum sized ordinary frame arriving after the maximum sized scene change frame, $t_{min}$, is then:

$$t_{min} = \frac{1}{f} - t_s \tag{5.10}$$

where $f$ is the number of frames per second, which is 30 for the stream under study.

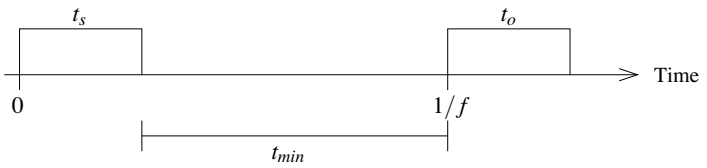



**Figure 5.7:** The time periods needed for Case 1.

The time from the arrival of the first packet of the maximum sized ordinary frame until the arrival of the last packet of the frame for generating tokens has not been included. This time is denoted as $t_o$, and is defined similarly to $t_s$:

$$t_o = \frac{o_{max} \cdot 8}{c} \tag{5.11}$$

These time periods are shown in Figure 5.7.

The minimum number of tokens that should be generated in the time period $t_{min}$ is then given by:

$$n_{min} = o_{max} \cdot 8 - (t_o \cdot \rho_{min}) \tag{5.12}$$

where $\rho_{min}$ is the minimum token generation rate.

From these equations, the minimum token generation rate can be calculated as:

$$\rho_{min} = \frac{n_{min}}{t_{min}} \tag{5.13}$$

And with the minimum token generation rate $\rho_{min}$, the minimum token bucket size is given by:

$$\sigma_{min} = s_{max} \cdot 8 - (t_s \cdot \rho_{min}) \tag{5.14}$$

where the number of tokens generated from the arrival of the first packet to the arrival of the last packet is subtracted from the maximum scene change frame to find the minimum token bucket size.

#### Case 2: Variable Token Bucket Size.

For a token bucket size larger than the minimum size, the size of the token bucket will again be decided by the rate needed for the token bucket to be full enough to transmit the number of consecutive large frames arriving a specific number of frame periods after the scene change frame, or at other periods in a scene. It is not enough to look at one maximum sized ordinary frame as for Case 1, since the maximum sized scene change frame will now leave behind a non-empty token bucket. The token generation rate needed for generating enough tokens for the first maximum sized ordinary frame is then lower than the token generation rate needed for the second maximum sized ordinary frame. The number of maximum sized ordinary frames taken into account is decided by the maximum number

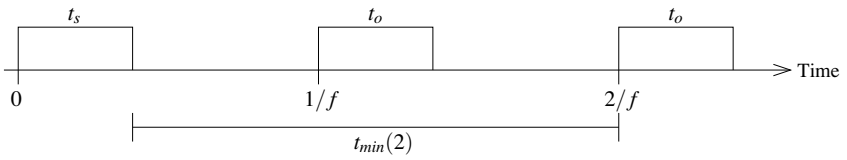



**Figure 5.8:** The time periods needed for Case 2.

of consecutive frames larger than the number of tokens generated in the same interval.

With a token generation rate higher than the maximum average bitrate in a scene, the token bucket is always full when a scene change frame arrives and a number of tokens are still present in the token bucket after the transmission of the scene change frame. The amount of tokens in the token bucket after the departure of the last packet of the scene change frame is then given by $\sigma - j(\rho)$ for the token generation rate $\rho$ where $\sigma$ is the token bucket size and $j(\rho)$ is the number of tokens required to transmit a maximum sized scene change frame when the token generation rate is $\rho$.

Now, $t_{min}(k)$ is the time available to generate tokens for $k$ maximum sized ordinary frames arriving after the maximum sized scene change frame, given by:

$$t_{min}(k) = \frac{k}{f} - t_s \tag{5.15}$$

This time period for $k = 2$ is shown in Figure 5.8 together with the time periods $t_s$ and $t_o$.

The minimum number of tokens that should be generated in the time period $t_{min}(k)$ is then given by:

$$n = k \cdot o_{max} \cdot 8 - t_o \cdot \rho - (\sigma - j(\rho)) \tag{5.16}$$

and the token generation rate for a given token bucket size is then given by:

$$\rho = \frac{n}{t_{min}(k)} \tag{5.17}$$

for a token bucket size $\sigma$, where $j(\rho)$ is given by:

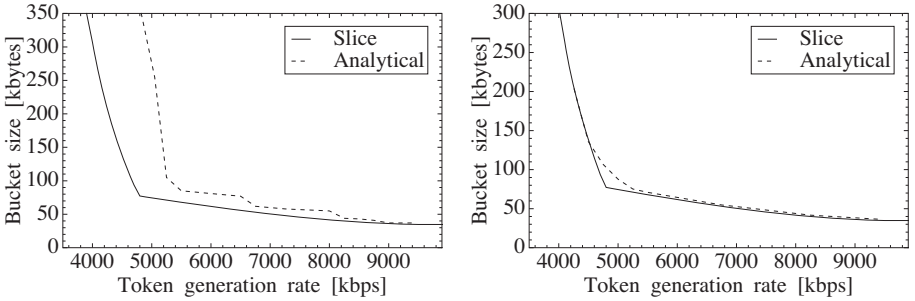
$$j(\rho) = s_{max} \cdot 8 - (t_s \cdot \rho) \tag{5.18}$$

The parameters for the slice-based video used to calculate the upper bound of the token bucket parameter curve are shown in Table 5.4.

The analytical curve found from these calculations is shown in Figure 5.9(a), together with the token bucket curve from simulations. As can be seen from the figure, when the token generation rate is close to the maximum average bitrate in a scene, the number of consecutive frames larger than the number of tokens generated in the same interval increases. This is what causes the knee point in the token bucket curve.

**Table 5.4:** Parameters for the slice-based video stream.

| Parameter | Value |
|---|---|
| $s_{max}$ | 40385 bytes |
| $o_{max}$ | 38930 bytes |
| $\overline{f}_{max}$ | 16638 bytes |
| $f$ | 30 fps |




(a) Approach 1 (b) Approach 2

**Figure 5.9:** The token bucket parameters found from analytical estimation.

In Equation 5.16, it is assumed that the size of all large frames is equal to the size of the maximum sized ordinary frame. This means that the token bucket size will be overestimated when the token generation rate is decreased, as can be seen in Figure 5.9(a). As a second approach, it is assumed that the size of the $k$ large frames is lowered in correspondence to the lower token generation rate. Hence, if ten frames are larger than the token generation rate at 5000 kbps, and 15 frames are larger at 4500 kbps, these five frames are set to 5000 kbps. The analytical curve from this calculation is plotted in Figure 5.9(b) together with the curve from simulations. The analytical curve gives an upper bound for the token bucket curve derived by simulations for all token bucket pairs with a token generation rate higher than the maximum average bitrate in a scene.

## 5.5 Conclusion

Token bucket traffic models for slice-based and frame-based encoded video streams are investigated through simulations and it is found that a high token generation rate is needed to fulfill the strict delay requirements imposed on real-time video traffic. Two video clips, the StEM clip and the Mobile clip, are studied. The StEM clip has frequent scene changes, resulting in large frames at the scene boundaries. For this stream, the token bucket curves are therefore almost identical for the slice-based and frame-based streams. However, for the Mobile clip without scene changes, the token bucket parameters are significantly lower for the slice-based

stream than for the frame-based stream. Hence, less resources need to be reserved for the slice-based stream than for the frame-based stream for the same delay guarantees. When a small amount of loss and/or delay is allowed, the token bucket parameters are significantly reduced for all of the streams, especially for the slice-based streams. Hence, the resource reservation needs are lowered and the utilization and maximum delay in the network is increased and decreased respectively. The high quantiles are estimated for the token bucket traffic model with given token bucket parameters. The high quantiles give valuable insight into the statistical loss guarantees.

The long-term correlation of the slice-based and frame-based streams are studied by reshuffling the frames of the streams and it is shown that only correlation inside the scenes affect the token bucket parameters, except for very low token generation rates. This means that the correlation between the scenes is small since it does not affect the token bucket curve.

Finally, the token bucket parameters are also investigated analytically for the slice-based encoded StEM video clip, with very satisfactory agreement for all token generation rates higher than at the knee point. This method requires only knowledge of the scene changes and the maximum number of consecutive large frames.

# Part III

# Non-parametric Analysis of Slice-based H.264/AVC Encoded Video Traffic

The results in this part have been published as follows:

Natalia Markovich, Astrid Undheim, and Peder J. Emstad. "Slice-based VBR Video Traffic-Estimation of Link Loss by Exceedance." In *Proceedings of the 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks (QoS-IP)*, Venice, Italy, February 2008.

Natalia Markovich, Astrid Undheim, and Peder J. Emstad. "Classification of Slice-based VBR Video Traffic and Estimation of Link Loss by Exceedance." *Computer Networks, Elsevier*, 53(7), May 2009.

# Chapter 6

# Classification of Slice-based Encoded Video Traffic

In this chapter, the slice-based encoded StEM clip as described in Section 3.4, is further studied. Due to the high variability, non-stationarity, and non-homogeneity in the underlying video data as identified in Chapter 4, the video trace is divided into sections that are classified according to their average frame size. A new approach to scene change detection is then developed, using the empirical quantiles of frame sizes within the classes. The dependence and distribution structure of the scenes within the obtained classes are investigated.

## 6.1 Introduction

Video traffic is in general non-stationary and non-homogeneous. This causes problems when developing models for the video traffic, and also when trying to apply known estimation methods for the analysis of the video data. Also the sequence of frame sizes from the StEM clip is non-homogeneous and non-stationary. As observed in Section 4.3, it is difficult to estimate the distribution functions of the frames because of the non-stationarity. Besides, due to the specific slice-based encoding scheme, a special kind of dependence among the frame sizes is observed in Section 4.4, different from regular frame-based encoded video where the GOP structure causes a periodic correlation structure. Hence, in the following, the sequence of frame sizes is divided into sections, which are classified according to the average frame size. This is done in order to obtain classes where the distribution and dependence structures are stationary.

In order to have homogeneous groups of frames and also to identify the large scene change frames, the entire video trace was divided into scenes in Section 4.2. When the classes are known, the scenes can be identified using a simple quantile estimation method, without any need for parameterization.

The dependence structure for scenes within the classes is estimated using the

classical dependence measures like the ACF and Ljung-Box test, while the cross correlation between scene lengths and scene change frames is estimated using the Kendall's $\tau$ and Spearman's $\rho$, in addition to linear correlation. The long-range dependence of scenes within the classes is also analyzed and quantified, using the Hurst parameter. In addition, the extremal index is calculated for the entire frames and the classes. The extremal index shows the asymptotical dependence in the time series [115–118] and is also employed for estimating the average loss over a threshold for the considered video stream in Chapter 7.

The classes obtained are also checked by the mean excess function and the tail index to find the type of distribution within each class. The entire trace contains a mixture of light- and heavy-tailed distributed frame sizes. A heavy tail means that the tail of the distribution goes to zero at infinity with a slower rate than the exponential one [119] and some moments of the distribution may not exist. The lack of the second or even the first moment implies that it is impossible to use the sample average and the sample variance to estimate them. The number of finite moments can be identified by the tail index, which reflects the shape of the tail of the distribution.

The proposed classification is useful when trying to specify a bounding function for the video stream, e.g., a token bucket specification as in Chapter 5, where different bounding functions can be specified for the different classes. Also, for video modeling the classes are useful and different models, e.g., Autoregressive Moving Average (ARMA) models or Gaussian models as in Chapter 8, can be specified for each class in case the whole video trace cannot be modeled by a single model. The classes can also be mapped to states in a Markov chain for modeling of the video traffic.

### 6.1.1 Related Work

The classification of video frames is an essential step for developing a Markov model for video traffic with variations in the bitrate. Each class then maps to a state in the Markov chain. This is investigated in [90], where MPEG encoded video is classified both on the GOP level and the scene level. Scenes are classified by the average GOP size, and the GOPs in each scene class are classified by the average frame size. A nested Markov chain is then developed for generation of synthetic video traffic.

Long-range dependence in VBR video traffic is extensively analyzed in [120], and a large number of video clips are used to prove the existence of LRDs in VBR video data. The confidence intervals for the Hurst parameter are above 0.5 for all video clips. Since then, the Hurst parameter has been estimated for a large number of video traces in different papers. MPEG-1 encoded video traces are studied in [92], and the Hurst parameter is estimated using R/S plots. A total number of 15 traces are analyzed, and the Hurst parameter is above 0.5 for all traces and above 0.8 for 11 of the traces, clearly indicating LRD for the traces under study. Also in [97], the Hurst parameter for the frame sizes of a long video clip is estimated using the R/S method, and a value of 0.995508 is found.

Long-Range Dependent (LRD) input traffic is shown to have a significant effect on the buffer overflow probability in [121]. For real-time traffic transmitted over a network with small buffers, it is found that the LRD does not have a large effect on the buffer overflow [122]. In [96], it is found that LRD traffic leads to higher buffer overflow probabilities compared to Short-Range Dependent (SRD) traffic when the buffer size is large. For small buffer sizes, the correlation persistence has less effect on the buffer overflow probabilities, in correspondence to [122].

### 6.1.2 Chapter Outline

The rest of this chapter is organized as follows. The classification of the video frames and a new procedure for the scene change detection are described in Section 6.2. The obtained classes are checked for dependence in Section 6.3. Estimation of the mean excess function and the tail index for the classes is considered in Section 6.4, giving information about the frame size distributions. Estimation of the extremal index is performed and subsequent classification using the extremal index is discussed in Section 6.5. Finally, the chapter is concluded in Section 6.6.

## 6.2 Scene Change Detection

In this section, the slice-based encoded StEM video trace is divided into sections of video frames which are classified by means of the average frame sizes. Next, a new scene change detection method is proposed, using the empirical quantiles within the classes. The short- and long-range dependence of the resulting scenes are investigated.

### 6.2.1 Classification of the Video Frames

Part of the StEM [86] clip, as presented in Section 3.4, with frequent scene changes and large variations in the motion levels in consecutive scenes is used as a test sequence. In Figure 6.1, the frame sizes of the trace are shown together with the thresholds 10.925 kbytes, 21.278 kbytes and 38.802 kbytes corresponding to the 80%, 95% and 99% empirical quantiles of the frame sizes, and the threshold 8.764 kbytes equal to the average frame size.

From the trace, it can be visually observed that the video trace can be divided into sections and the sections can be classified by the average frame size, giving four classes. Separating the frames into more than these four classes is difficult because there will not be enough observations for each class. The statistics for the classes and the entire trace are shown in Table 6.1. The upper and lower bounds of the classes are indicated by the frame numbers.

### 6.2.2 Definition of Scenes

The division of the highly variable StEM clip into scenes was motivated in Chapter 4. For the analysis in this chapter it is also important to identify scenes as

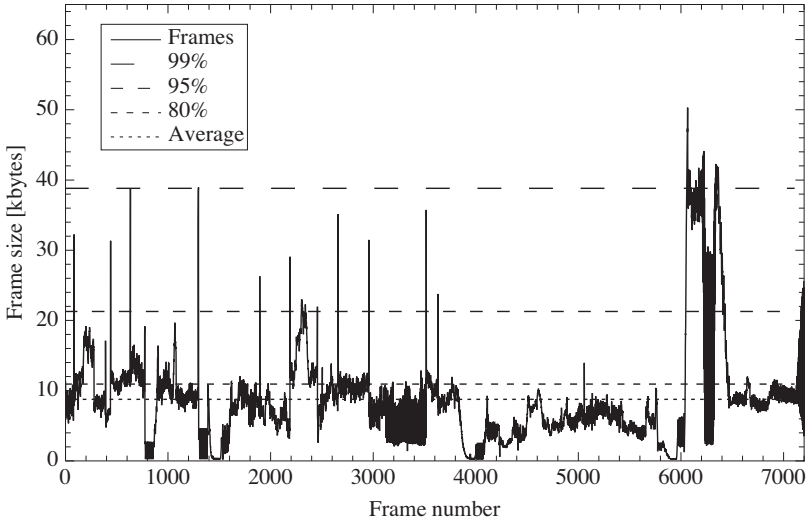



**Figure 6.1:** The frame sizes of the slice-based encoded video stream together with the average frame size 8.764 kbytes and the 80%, 95% and 99% empirical quantiles.

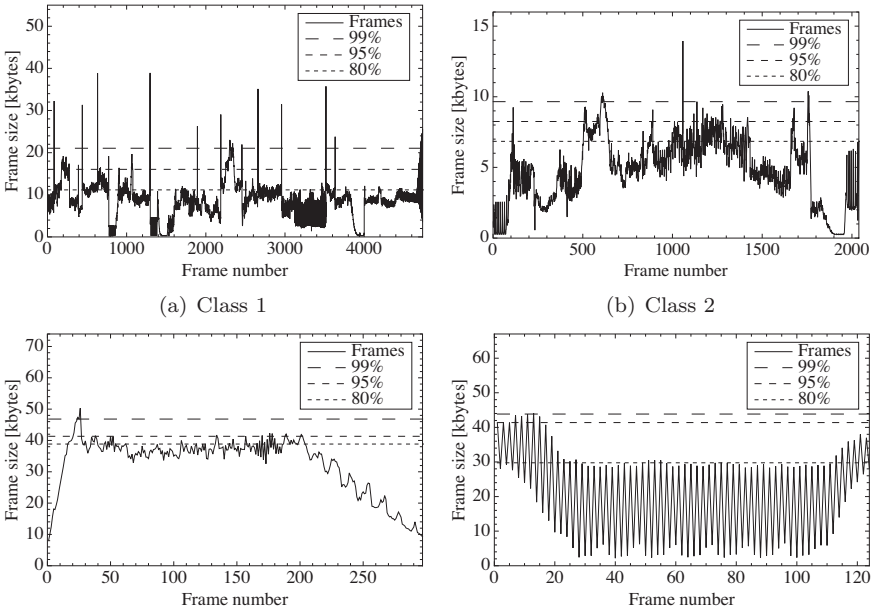
**Table 6.1:** Description of the video traffic data (frame size in kbytes).

| Class | Frame number | Samples | Min | Max | Average | StDev |
|---|---|---|---|---|---|---|
| All | $[0, 7198]$ | 7198 | 0.181 | 50.294 | 8.764 | 7.038 |
| Class 1 | $[0, 4000] \cup [6461, 7198]$ | 4738 | 0.181 | 38.930 | 8.680 | 4.167 |
| Class 2 | $[4001, 6039]$ | 2039 | 0.214 | 13.949 | 4.789 | 2.276 |
| Class 3 | $[6040, 6210] \cup [6335, 6460]$ | 297 | 8.226 | 50.294 | 32.561 | 9.006 |
| Class 4 | $[6211, 6334]$ | 124 | 2.175 | 44.143 | 20.339 | 13.572 |

independent blocks of data in order to estimate the extremal index in Section 6.5 and afterwards the average loss per cluster in Chapter 7.

The scene change detection for the slice-based encoded video traffic is completely different to that for the frame-based encoded video traffic. In Chapter 4, the algorithm for scene change detection proposed in [87] was used, with good results. However, using this algorithm requires parameter selection in order to work with the underlying data.

Here, an additional approach to the scene change detection is proposed, namely to use the quantiles of the frame sizes to identify the scene change frames. Because of the different average frame size and sample variance of the classes, the quantiles are investigated for the classes separately. The frame sizes in the different classes are shown together with the 80%, 95%, and 99% empirical quantiles in Figure 6.2. The empirical quantile $x_p$ of level $(1-p)\cdot 100\%$, $p \in (0,1)$, is determined by means of the empirical distribution function $F_n(x) = (1/n)\sum_{i=1}^{n} \mathbf{1}(x \geq X_i)$




(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

**Figure 6.2:** The 80%, 95%, and 99% empirical quantiles for the separate classes.

of frame sizes in such a way that $F_n(x_p) = (1 - p)$. Here $\mathbf{1}(A)$ is an indicator function of the event $A$.

As can be seen from the figures, occasionally there will be bursts of frames higher than the quantiles. In this case, the first frame in the burst will be chosen as the scene change frame. 13 frames is the minimum scene length as for the approach in Chapter 4. From the figures, it can be observed that very few scene changes are detected by the 99% quantiles. At the same time, the estimation of the extremal index in Section 6.5 requires a sufficiently large number of independent scenes. Hence, in the rest of this work the 95% and 80% quantiles will be used.

The sample average and sample variance of the scene lengths and sizes of the scene change frames from the quantile approach for the 80% and 95% quantiles are shown in Table 6.2 together with the results from the scene change detection method from Chapter 4 with $\lambda = 0.4$ (note that the scene lengths are given in terms of number of frames in Chapter 4). Statistics for the entire trace and the classes are shown. Class 3 and 4 have too few scenes to make any statistics for the original method and the 95% quantile method. Class 3 has rather constant bitrate apart from the frame numbers exceeding 200, while Class 4 is short and has a periodic structure of the frame sizes with high variability as can be seen in Figure 6.2.

The new quantile method for scene change detection has some important advantages. First, it does not require the selection of any parameters and selects

**Table 6.2:** Scene statistics for the classes.

| Class | Method | Number of Scenes | Scene Lengths (in Mbytes) | | Scene Change Frames (in kbytes) | |
|---|---|---|---|---|---|---|
| | | | Mean | Variance | Mean | Variance |
| All | 80% Quantile | 190 | 0.332 | 0.173 | 14.545 | 84.122 |
| | 95% Quantile | 73 | 0.864 | 1.160 | 19.473 | 131.855 |
| | Original | 33 | 1.912 | 4.372 | 18.761 | 115.690 |
| Class 1 | 80% Quantile | 121 | 0.340 | 0.179 | 14.241 | 27.012 |
| | 95% Quantile | 39 | 1.055 | 1.618 | 21.306 | 50.593 |
| | Original | 27 | 1.774 | 2.568 | 19.466 | 108.440 |
| Class 2 | 80% Quantile | 51 | 0.191 | 0.071 | 7.408 | 1.146 |
| | 95% Quantile | 24 | 1.197 | 15.450 | 8.662 | 3.916 |
| | Original | 6 | 1.628 | 3.411 | 8.446 | 169.351 |
| Class 3 | 80% Quantile | 13 | 0.744 | 0.350 | 37.342 | 77.427 |
| Class 4 | 80% Quantile | 5 | 0.504 | 0.020 | 35.294 | 40.171 |

the scene change frames from the distribution of the frame size only. Second, the quantile method provides a larger number of scenes than the original method, which is advantageous when estimating the extremal index. In the next section, the scenes from the 80% quantile method are checked for dependence.

## 6.3 Test of the Dependence of Scenes

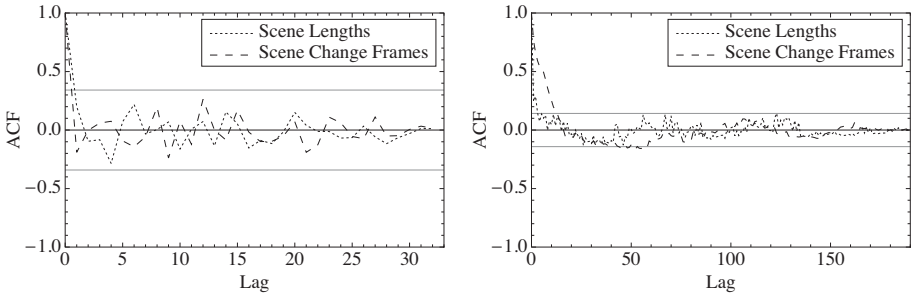
The scenes obtained using both scene change detection methods are checked for dependence, first using regular short-range dependence measures and next using long-range dependence measures.

### 6.3.1 Short-Range Dependence

The sample ACFs for the scene lengths and the size of the scene change frames are shown in Figures 6.3-6.5 for the original and the quantile scene change detection methods. For the latter method, the 80% quantiles in the respective classes are employed to find the scene changes. Class 3 and 4 are omitted because of the low number of scenes.

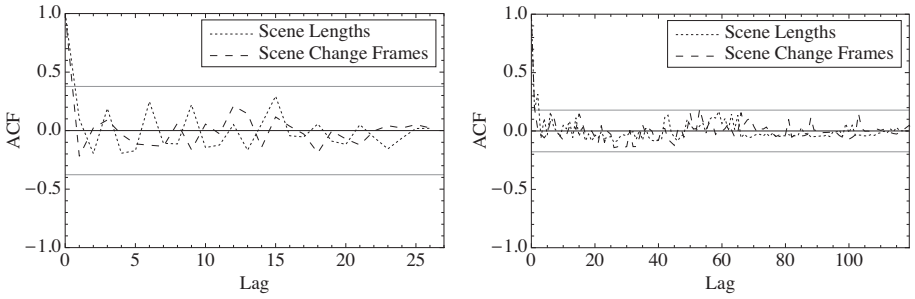
Since all ACFs for the classes are located inside the 95% confidence interval with the bounds $\pm 1.96/\sqrt{n}$, it can be assumed that both the scene lengths and the size of the scene change frames are independent for all the considered classes. For the scene change frames for the 80% quantile method on the entire trace, the ACF is non–negligible for low lags.

In addition, the Ljung-Box portmanteau test [123] is applied to the scene lengths and the size of the scene change frames to check the dependence. This test was originally proposed for checking the hypothesis of independent residuals after fitting an ARMA model to a time series. It tests the value of a sum of



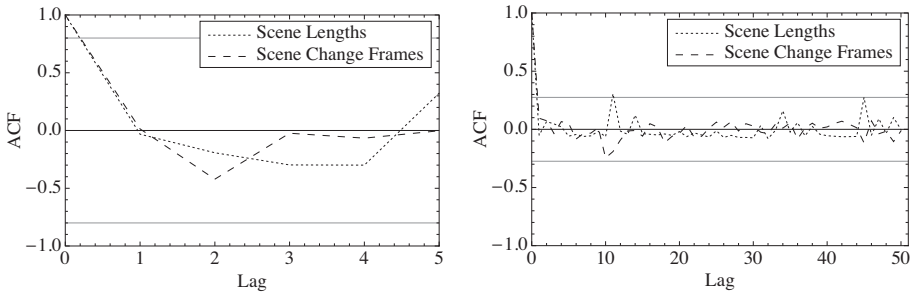

(a) Original scene change detection (b) 80% quantile scene change detection

**Figure 6.3:** The ACF for the scenes of the entire sample.




(a) Original scene change detection (b) 80% quantile scene change detection

**Figure 6.4:** The ACF for the scenes of Class 1.




(a) Original scene change detection (b) 80% quantile scene change detection

**Figure 6.5:** The ACF for the scenes of Class 2.

**Table 6.3:** Ljung-Box test results.

| Class | Method | Lags | Scene Lengths | | Scene Change Frames | |
|---|---|---|---|---|---|---|
| | | | Q | p-value | Q | p-value |
| All | 80% Quantile | 20 | 54.284 | 0.000 | 540.260 | 0.000 |
| | | 30 | 69.126 | 0.000 | 555.910 | 0.000 |
| | Original | 10 | 9.131 | 0.520 | 7.804 | 0.648 |
| | | 20 | 17.349 | 0.630 | 16.720 | 0.671 |
| Class 1 | 80% Quantile | 20 | 28.850 | 0.091 | 14.756 | 0.790 |
| | | 30 | 34.331 | 0.268 | 33.101 | 0.318 |
| | Original | 10 | 11.380 | 0.329 | 4.973 | 0.893 |
| | | 20 | 22.669 | 0.305 | 14.509 | 0.804 |
| Class 2 | 80% Quantile | 20 | 9.947 | 0.969 | 10.850 | 0.950 |
| | | 30 | 13.746 | 0.995 | 13.073 | 0.997 |
| | Original | 3 | 1.882 | 0.597 | 2.142 | 0.544 |

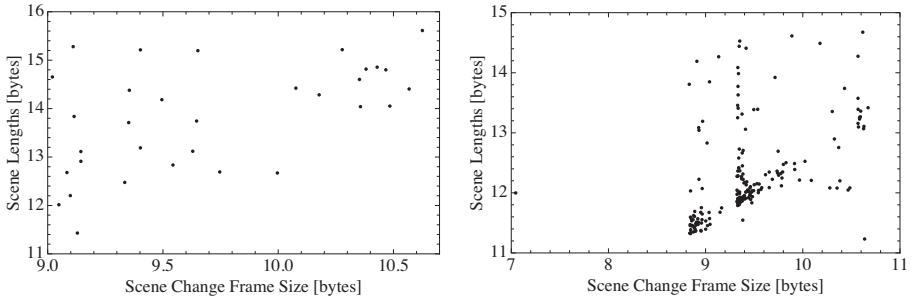
autocorrelations of a time series [101]. The statistic:

$$Q = n(n+2)\sum_{j=1}^{h} \hat{\rho}^2(j)/(n-j) \tag{6.1}$$

is then calculated. Its distribution may be approximated by the Chi-square distribution with $h$ degrees of freedom. Here, $\hat{\rho}(j)$ is the sample ACF at lag $j$ and $n$ is the number of samples. A large value of $Q$ suggests that the data are not independent and identically distributed (iid). The iid hypothesis is rejected at level $\alpha$ if $Q > \chi^2_\alpha(h)$, where $\chi^2_\alpha(h)$ is the $\alpha$ quantile of the Chi-square distribution with $h$ degrees of freedom, i.e., $Pr\{\chi^2 > \chi^2_\alpha(h)\} = \alpha$, $0 < \alpha < 1$.

The results of Ljung-Box test applied to the scene lengths and the scene change frames for the entire sample, Class 1, and Class 2 and for the two scene change detection methods are shown in Table 6.3. The iid hypothesis is rejected at level 0.05 if the corresponding p-value given in the table is less than 0.05. For the StEM clip, this means that the iid hypothesis is rejected for the quantile method employed on the entire trace, in correspondence with the results from the analysis of the ACFs. For the rest of the data the iid hypothesis cannot be rejected.

It is also interesting to investigate the dependence between the scene lengths and the size of the scene change frames by scatter plots. The scatter plots of the considered scene change detection methods for the whole trace are shown in Figure 6.6, and demonstrate dependence of different types. The original method provides nearly uniformly distributed points above the approximate line $y = 1.6x+11$. Such uniformness indicates rather weak dependence. In case of the quantile method, many points are located along the approximate line $y = x - 8.5$. The rest of the points are located above this line, but not uniformly. The natural logarithms of scene change frame sizes equal to the approximate values 8.9, 9.5 and 10.5 correspond to many different scene lengths. These three conglomerates correspond to the average scene change frame sizes of Class 2, Class 1, and both Class 3 and 4, respectively. Class 4 can be separated from Class 3 by the observation of the




(a) Original method (b) 80% quantile method

**Figure 6.6:** Scatter plot of the natural logarithm of the scene lengths in bytes versus the natural logarithm of the size of the scene change frames in bytes for the whole trace.

conglomerate with the center 10.47. Hence, the proposed quantile method for the scene change detection is more sensitive to the selection of the classes with different average frame sizes than the original method used before.

In addition, the Kendall's $\tau$ and Spearman's $\rho$ rank correlation coefficients which measure the degree of correspondence between two data sets are calculated for the scene lengths $\{L_i\}$ and the size of the scene change frames $\{S_i\}$, $i = 1, 2, ..., n$. The scenes are ranked in terms of the scene lengths according to the order statistics $L_{(1)} \leq L_{(2)} \leq ... \leq L_{(n)}$. $r_i$ is then the rank of the scene among $S_1, ..., S_n$ that has the length $L_{(i)}$. The coefficients $\tau$ and $\rho$ are calculated by the formulas

$$\rho = 1 - \frac{6S_\rho}{n^3 - n} \in [-1, 1], \qquad \text{where} \qquad S_\rho = \sum_{i=1}^{n} (r_i - i)^2 \tag{6.2}$$

and

$$\tau = \frac{2S_\tau}{n(n-1)} \in [-1, 1], \qquad \text{where} \qquad S_\tau = \sum_{i=1}^{n} \sum_{j=i+1}^{n} sign(r_j - r_i), \tag{6.3}$$

where $sign(x)$ is equal to 1 if $x > 0$ and to $-1$ if $x < 0$.

The linear dependence is also calculated. The results for both scene change detection methods are shown in Table 6.4. All the considered dependence measures give a value zero for independent random variables but the converse is not true. Since the dependence measures have mostly nonzero values, one may assume that the scene lengths and the size of the scene change frames are most probably dependent and this dependence is nonlinear. Similar to scatter plots, all dependence measures for the entire sample demonstrate weaker dependence for the original method than for the 80% and 95% quantile methods.

**Table 6.4:** Kendall's and Spearman's rank correlation coefficients for the scene lengths and the size of the scene change frames.

| Method | Class | Kendall's $\tau$ | Spearman's $\rho$ | Linear |
|---|---|---|---|---|
| 80% Quantile | All | 0.4377 | 0.5644 | 0.3071 |
| | Class 1 | 0.1703 | 0.2122 | 0.0653 |
| | Class 2 | 0.1969 | 0.2750 | 0.0337 |
| | Class 3 | -0.0256 | -0.0714 | 0.1848 |
| | Class 4 | 0.2000 | 0.3000 | 0.4103 |
| 95% Quantile | All | 0.2721 | 0.3764 | 0.3321 |
| | Class 1 | 0.1795 | 0.2391 | 0.2068 |
| | Class 2 | -0.0399 | -0.0300 | -0.0219 |
| Original Method | All | 0.2808 | 0.3660 | 0.2054 |
| | Class 1 | 0.2454 | 0.3160 | 0.1394 |
| | Class 2 | -0.6000 | -0.7143 | -0.7021 |

### 6.3.2 Long-Range Dependence

To check the long-range dependence of the scenes the Hurst parameter $H$, $0.5 < H < 1$ within each class is calculated by the aggregated variance method ($A/V$) [124], the rescaled adjusted range method ($R/S$) [125], and the Abry-Veitch wavelets estimator [126].

The $A/V$ and $R/S$ methods are based on the properties of self-similar processes. Namely, $Var(X^{(m)}(k)) \sim m^{2H-2}Var(X_k)$ and $E(R(l,r)/S(l,r)) \sim a_1 r^H$, respectively, where $a_1$ is a positive, finite constant [124, 127]. According to the A/V method, one plots the logarithm of $\widehat{Var}X^{(m)}$ versus $\log(m)$. A straight regression line approximating the points has the slope $\beta = 2H - 2$, $-1 \leq \beta < 0$. According to the $R/S$ method the estimate of $H$ is given by the slope of the statistics $\log(R(l_i,r)/S(l_i,r))$ against $\log(r)$, where $r$ denotes a range.

In order to check the self-similarity, Higuchi's method [128] is used. Using a given time series $X_1, X_2, ..., X_n$, one first constructs a new time series $X_k^m$ which is defined as follows:

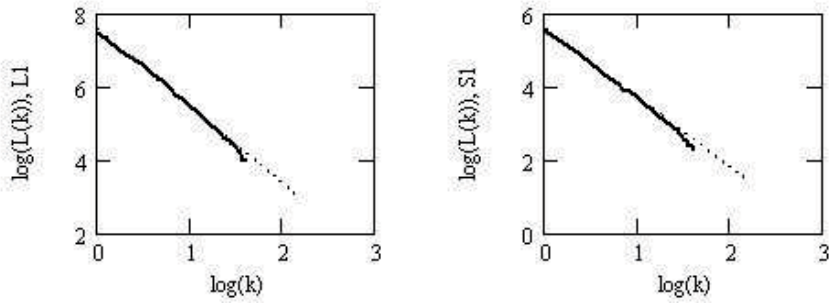
$$X_k^m : X_m, X_{m+k}, X_{m+2k}, ..., X_{m+[(n-m)/k]k}, \qquad m = 1, 2, ..., k. \tag{6.4}$$

Then one calculates

$$L_m(k) = \frac{n-1}{k^2[(n-m)/k]} \sum_{i=1}^{[(n-m)/k]} |X_{m+ik} - X_{m+(i-1)k}|, \tag{6.5}$$

and computes a log-log plot of the statistic $\overline{L(k)}$, which is the average value over $k$ sets of $L_m(k)$, versus $k$. A constant slope $D$ in $\overline{L(k)} \propto k^{-D}$ indicates self-similarity.

It was found that the scene lengths and scene change frame sizes of the four underlying classes are almost self-similar processes, since the slopes of the log-log plots of the statistic $\overline{L(k)}$ against $k$ are approximately constant. The results for




(a) Scene lengths (b) Scene change frame size

**Figure 6.7:** The log-log plot of the statistic $\overline{L(k)}$ versus $k$ for the scene lengths and the scene change frame sizes of Class 1.

**Table 6.5:** Estimation of the Hurst parameter of the scenes obtained by the 80% Quantile method.

| Class | Scene Lengths | | | Scene Change Frame | | |
|---|---|---|---|---|---|---|
| | $R/S$ | $A/V$ | Abry-Veitch | $R/S$ | $A/V$ | Abry-Veitch |
| All | 0.60 | 0.65 | 0.80 [0.66, 0.95] | 0.62 | 0.55 | 0.78 [0.60, 0.96] |
| Class 1 | 0.55 | 0.65 | 0.63 [0.25, 1.01] | 0.65 | 0.55 | 0.70 [0.44, 0.96] |
| Class 2 | 0.80 | 0.55 | - | 0.85 | 0.50 | - |
| Class 3 | 0.70 | 0.58 | - | 0.60 | 0.50 | - |
| Class 4 | 0.90 | -0.65 | - | 0.96 | 1.08 | - |

Class 1 is shown as an example in Figure 6.7. This implies that the A/V and R/S methods can be applied to estimate the Hurst parameter $H$ of the scene lengths and scene change frame sizes.

An $\hat{H}_n$ close to 1 indicates possible long-range dependence. It implies that the dependence in the time series is kept over an unusually long period of time. The results of the calculations for the $A/V$ and $R/S$ methods given in Table 6.5 lead to the conclusion that the scene lengths and the scene change frames of all classes have a moderate amount of long-range dependence. The results for Class 2-4 are not reliable due to the small number of observations ($n \in \{51, 13, 5\}$).

The Abry-Veitch estimator is a wavelet based estimator for the Hurst parameter, shown to be unbiased under very general conditions [126]. The results from applying the Abry-Veitch estimator show a bit higher Hurst parameter estimates for both the scene lengths and the scene change frames compared to the $A/V$ and $R/S$ methods. Nevertheless, since the confidence intervals for the Abry-Veitch estimates include values around 0.6, the hypothesis that independence or short-range dependence also occurs cannot be rejected.

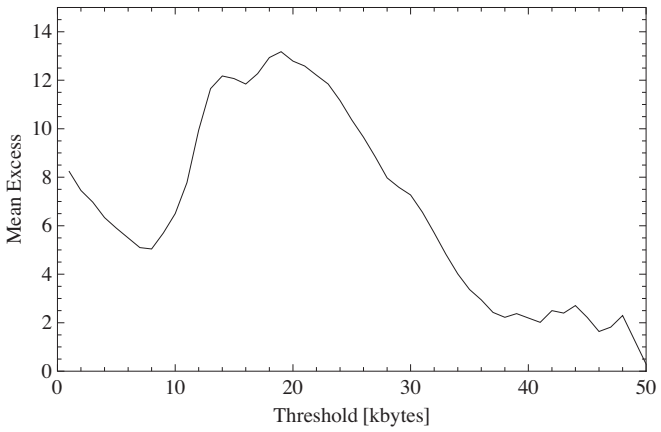



**Figure 6.8:** The sample mean excess function $\hat{e}(u)$ for the whole trace against the threshold $u$.

## 6.4 Estimation of the Mean Excess and Tail Index

The mean excess function is a simple test to detect visually whether a distribution has a light or a heavy tail [119]. The mean excess function $e(u) = E(X - u|X > u)$ or its empirical analogue, the sample mean excess function:

$$\hat{e}(u) = \sum_{i=1}^{n} (X_i - u)\mathbf{1}\{X_i > u\} / \sum_{i=1}^{n} \mathbf{1}\{X_i > u\} \tag{6.6}$$

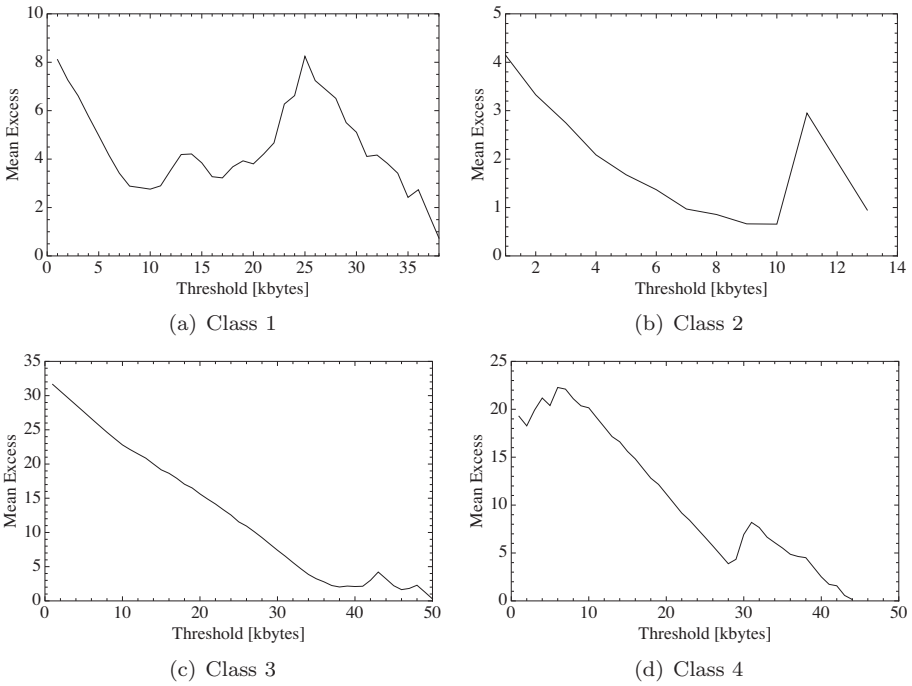
determines the average bit loss over a number $n$ of frame sizes $\{X_i\}$ under investigation within a corresponding time $t$, where $t = n/30$, because 30 frames are formed each second. $u$ is a threshold given in kbytes. $\hat{e}(u)$ is plotted for the whole trace in Figure 6.8 and shows that the frame size data are non-homogeneous in the following sense. Generally an increasing (or decreasing) plot indicates that the data are distributed with a heavy (or light) tail, a linear mean excess plot corresponds to Pareto-type distributed data, while a constant $\hat{e}(u)$ corresponds to an Exponential distribution.

The non-homogeneity in the frame sizes reflects another class structure of the frame size data regarding the threshold value $u$. The frame sizes that exceed the values $u \in [0, 8]$, $u \in (8, 14]$, $u \in (14, 15]$, $u \in (15, 21]$, $u \in (21, 37]$, $u \in (37, 44]$ and $u \in (44, 50]$ kbytes correspond to one of seven classes as shown in Table 6.6. At the interval $(37, 44]$, $\hat{e}(u)$ tends to increase not significantly. For a large $u$, the $\hat{e}(u)$ is not quite reliable since there are not enough observations beyond such $u$. Hence, it is difficult to make precise conclusions regarding the interval $(44, 50]$.

Since the frame size data from the whole trace follow a mixture of distributions, classification by the mean excess function concerns the distributions of component classes of the mixture. For example, if $u$=7 kbytes or $u$=41 kbytes, the frame

**Table 6.6:** Classification by the mean excess value for a given threshold $u$.

| Contributor classes | Threshold interval | Distribution type |
|---|---|---|
| $1-4$ | $u \in [0, 8]$ | Light-tailed |
| $1-4$ | $u \in (8, 14]$ | Pareto-like |
| $1, 3, 4$ | $u \in (14, 15]$ | Light-tailed |
| $1, 3, 4$ | $u \in (15, 21]$ | Pareto-like |
| $1, 3, 4$ | $u \in (21, 37]$ | Light-tailed |
| $1, 3, 4$ | $u \in (37, 44]$ | Heavy-tailed |
| $3, 4$ | $u \in (44, 50]$ | Light-tailed |




(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

**Figure 6.9:** The sample mean excess function for the individual classes.

sizes from all classes and the frame sizes from only the third and fourth class respectively, contribute to the mean excess value.

Figure 6.9 shows the $\hat{e}(u)$ for each class. The estimates indicate that all classes contain mixtures of heavy- and light-tailed distributed frame sizes. This is very typical for telecommunication data, where none of the classical distribution models fits the data [119]. Indeed, the heavy-tailed components dominate the shape of the tails and determine the heaviness of tails in the mixtures.

The distribution type of the frame sizes can also be checked by estimation of the

tail index. For the wide class of heavy-tailed distributions called distributions with regularly varying tail, the tail distribution function is determined by $1 - F(x) = x^{-\alpha} \cdot \ell(x)$, where $\ell(x)$ is a slowly varying function, i.e., $\lim_{x\to\infty} \frac{\ell(tx)}{\ell(x)} = 1$ for any positive $t$. Positive constants and $\ln(x)$ provide examples of $\ell(x)$ [115, 116, 119]. The tail index $\alpha$ and the Extreme Value Index (EVI) $\gamma = \alpha^{-1}$ indicate the shape of the distribution tail. The smaller the value of $\alpha$, the heavier is the tail. A positive sign of $\alpha$ indicates that the distribution is heavy-tailed. The value of $\alpha$ shows the number of finite moments for the regularly varying distributions. Namely, the moment is finite, i.e., $E[X^\beta] < \infty$ if $\beta < \alpha$ and infinite, $E[X^\beta] = \infty$, if $\beta > \alpha$.

Since Pareto-like regularly varying components are present in all four classes and determine the heaviness of tails of the classes one can assume that the distribution of the frame size is regularly varying. To estimate $\gamma$ for the trace under study the popular Hill's estimator, which is only valid for positive EVIs, is used:

$$\widehat{\gamma}^H(n, k_0) = \frac{1}{k_0} \sum_{i=1}^{k_0} \ln X_{(n-i+1)} - \ln X_{(n-k_0)}. \tag{6.7}$$

This estimator can be applied to dependent data under mild mixing conditions [129]. This is a big advantage for the analysis of our data. Here, $X_{(1)} \le X_{(2)} \le \ldots \le X_{(n)}$ are the order statistics of the sample $X^n = \{X_1, X_2, \ldots, X_n\}$ and $k_0$ is a smoothing parameter. $k_0$ is selected by the bootstrap method [119, 130]. The idea of the bootstrap method is to minimize the bootstrap estimate of the Mean Squared Error (MSE), MSE=$E(\hat{\gamma}^H(n, k_0) - \gamma)^2$ by $k_0$. Since $\gamma$ is unknown, it must be replaced by the bootstrap estimate $\hat{\gamma}^B$. The latter is an average of estimates $\hat{\gamma}$ which are built by re-samples $\{X_1^*, X_2^*, ..., X_{n_1}^*\}$. These re-samples are drawn from the initial sample $\{X_1, X_2, ..., X_n\}$ with replacement. The re-sample size $n_1 = n^\beta$, $0 < \beta < 1$ is smaller than the size of the initial sample $n$.

Let $\hat{\gamma}_i^H(n_1, k_1)$ denote the Hill's estimate of the tail index constructed by the $i$th of $B$ bootstrap re-samples. Then one finds the $k_1 \in [1, ..., n_1 - 1]$ that provides the minimum empirical bootstrap MSE:

$$\begin{aligned}\widehat{MSE}(n_1, k_1) =& \Big(\frac{1}{B}\sum_{i=1}^{B} \hat{\gamma}_i^H(n_1, k_1) - \hat{\gamma}^H(n, k_0)\Big)^2 \\ &+ \frac{1}{B-1}\sum_{i=1}^{B}\Big(\hat{\gamma}_i^H(n_1, k_1) - \frac{1}{B}\sum_{i=1}^{B}\hat{\gamma}_i^H(n_1, k_1)\Big)^2\end{aligned} \tag{6.8}$$

The relation between $k_0$ and $k_1$ is given by:

$$k_0 = k_1 \cdot \Big(\frac{n}{n_1}\Big)^\nu, 0 < \nu < 1. \tag{6.9}$$

As it is proven in [130], $\beta = \frac{1}{2}$ and $\nu = \frac{2}{3}$ provide a consistent bootstrap Hill's estimate, i.e., for a sufficiently large sample size $n$ the bootstrap estimate $\frac{1}{B}\sum_{i=1}^{B}\hat{\gamma}_i^H(n_1, k_1)$ converges to $\gamma$.

**Table 6.7:** Hill's estimate of the tail index $\alpha$ of frame sizes for the selected classes.

| Class number | Hill's estimate |
|---|---|
| Class 1 | 4.1248 |
| Class 2 | 11.1003 |
| Class 3 | 18.4467 |
| Class 4 | 5.8846 |

The results from the estimation of the tail index $\alpha$ is shown in Table 6.7 for the selected classes, with $B = 200$. The values of the Hill's estimate indicate that Class 1 has the heaviest tail in the sense that it has four finite moments. In contrast, Class 3 has the lightest tail with 18 finite moments. This follows from the common property of the class of the regularly varying distributions mentioned before. The values of the Hill's estimate correspond to the mean excess plots in Figure 6.9, where Class 1 and 4 have large intervals with an increase in the mean excess, corresponding to heavy-tail distributed frames sizes.

## 6.5 Estimation of the Extremal Index

In this section, the extremal index is estimated and used for classification of the video stream. Subsequently, the extremal index is employed for estimating the average loss per cluster in Chapter 7.

Let $X_i$, $i = 1, 2, ..., n$ be a stationary process with a marginal distribution function (df) $F(x)$ and $\widetilde{X}_i$, $i = 1, 2, ...n$ be an associated iid sequence with the same df. According to the theory of extremal values, for large $n$ and $u_n$, typically

$$P\{\max(X_1, ..., X_n) \leq u_n\} \approx P^{\theta}\{\max\left(\widetilde{X}_1, ..., \widetilde{X}_n\right) \leq u_n\} = F^{n\theta}(u_n) \quad (6.10)$$

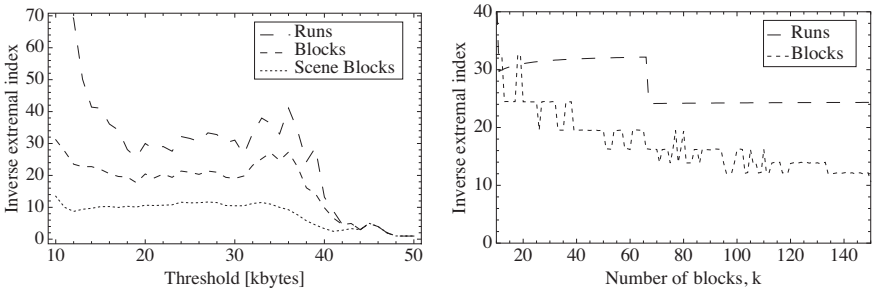
holds, where $\theta \in [0, 1]$ is a constant known as the extremal index [115, 117, 118]. For iid sequences, $\theta = 1$ holds. The extremal index characterizes the change in the limiting distribution of the sample maxima due to dependence in the sequence. Estimators of $\theta$ are distinguished by the different definitions of a cluster.

Regarding the blocks estimator, a cluster is defined as a block of data with at least one exceedance over a threshold. The blocks estimator for the threshold $u$ is calculated by the formula:

$$\overline{\theta}^{B}(u) = \frac{k^{-1}\sum_{j=1}^{k}\mathbf{1}(M_{(j-1)r,jr} > u)}{rn^{-1}\sum_{i=1}^{n}\mathbf{1}(X_i > u)}, \quad (6.11)$$

where $M_{i,j} = \max(X_{i+1}, ..., X_j)$, $k$ is the number of blocks, and $r = [n/k]$ is the number of observations in each block, where $[\cdot]$ denotes the integer part of the number.

For the runs estimator, a cluster is defined as a block of data with some number of exceedances over a threshold and at the same time the subsequent $r$ observations are all below the threshold. The runs estimator for the threshold $u$




(a) $k = 100$, $r = 71$

(b) Threshold $u = 38$ kbytes

(c) Threshold $u = 50$ kbytes

**Figure 6.10:** Blocks, scene blocks, and runs inverse estimates $1/\overline{\theta}^B$, $1/\overline{\theta}^B_S$ and $1/\overline{\theta}^R$ for the whole trace.

is calculated by the formula:

$$\overline{\theta}^R(u) = \frac{(n-r)^{-1}\sum_{i=1}^{n-r}\mathbf{1}(X_i > u, M_{i,i+r} \leq u)}{n^{-1}\sum_{i=1}^{n}\mathbf{1}(X_i > u)} \tag{6.12}$$

The runs estimate does not require any block structure and has a better asymptotic bias than the blocks estimate [118]. $1/\overline{\theta}^B(u)$ and $1/\overline{\theta}^R(u)$ have a simple interpretation. Both are the ratio of the number of observations that exceed the threshold $u$ to the number of clusters and show the average number of exceedances in a cluster.

The selection of the threshold $u$ and the number of blocks $k$ (or $r$ in the runs estimator), driven by the sample, is still an open problem. The idea is to select parameters that make the clusters independent [116]. Very roughly, they should be sufficiently far away from each other. The latter may provide some mixing conditions which are necessary to satisfy Equation 6.10 [115].

The simplest way is to use a value that corresponds to a stable interval of the plot $(1/\overline{\theta}(u), u)$ over a range of thresholds for a fixed parameter $k$ (or $r$) as an estimate for $\theta$. The reason is that both of the considered estimates are consistent, i.e., $\theta = \lim_{n\to\infty}\overline{\theta}$. In Figure 6.10(a), $\overline{\theta}^B = 0.052$ and $\overline{\theta}^R = 0.036$ are the average values over $u \in \{15, 16, ..., 38\}$ kbytes and $u \in \{20, 21, ..., 38\}$ kbytes for the blocks

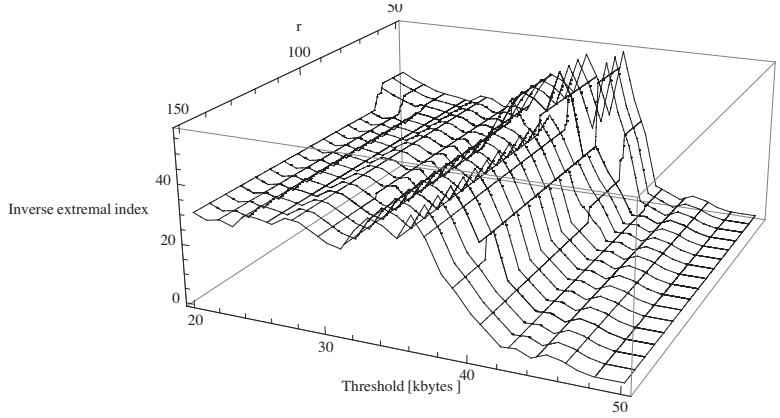



**Figure 6.11:** Inverse runs estimate for variable thresholds $u$ and the runs parameter $r$.

and runs estimates, respectively. The sudden drop in the inverse extremal index for thresholds above 38 kbytes is due to the higher bitrate of Class 3 and 4. Both estimates of $\theta$ are close to 1 for $u = 50$ kbytes. This means that the exceedances over 50 kbytes are well approximated by overshooting arising from a stationary process of iid random variables $X_i$ with the same marginal distribution. From Figures 6.10(b) and 6.10(c) it can be seen that beyond $k = 100$ both plots reach stability. However, the behavior of the inverse extremal index in the case of independent clusters in Figure 6.10(c) is different from the dependent case in Figure 6.10(b).

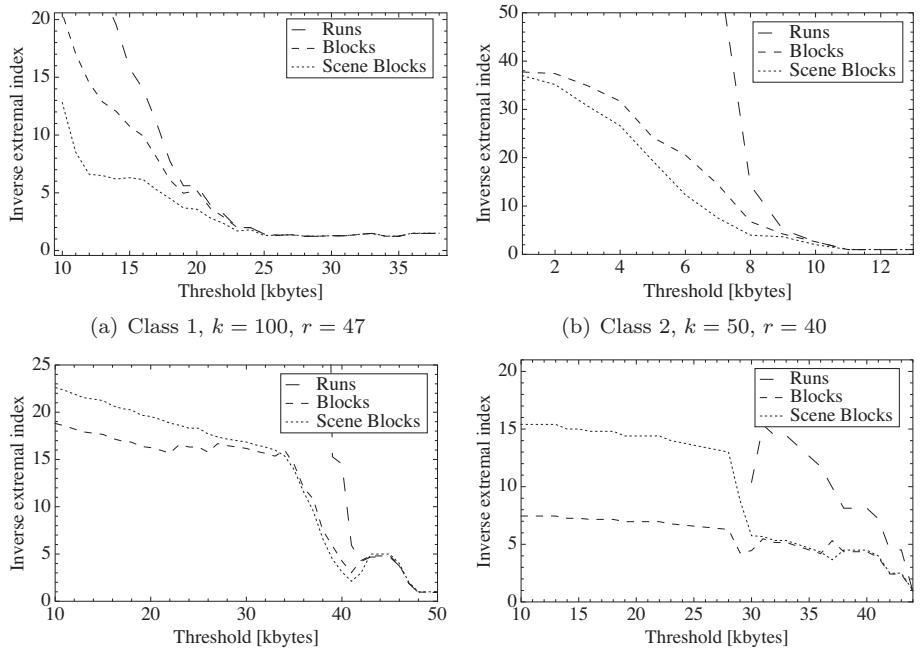
The $3D$-plot of the runs estimate in Figure 6.11 is considered in order to investigate the stability of $1/\overline{\theta}^R(u)$ with respect to both $u$ and $r$. One may conclude that the plot is stable for $u \in [20, 30]$ and for any value $r$ in the interval $[70, 150]$.

Often it is possible to choose appropriate parameters $u$ and $k$ (or $r$) with knowledge about the problem. In this work, it is proposed to use the scenes as blocks, meaning that the blocks will have unequal sizes. It implies that the parameter $r$ will be variable in Equation 6.11, i.e., a scene blocks estimator is proposed as follows:

$$\overline{\theta}^B_S(u) = \frac{\sum_{j=1}^{k} \mathbf{1}(M_{\sum_{m=0}^{j-1} r_m, \sum_{m=1}^{j} r_m} > u)}{\sum_{i=1}^{n} \mathbf{1}(X_i > u)}, \tag{6.13}$$

where $r_j$ is the number of frames in the $j$th scene, $\sum_{j=1}^{k} r_j = n$, $r_0 = 0$, and $k$ is the number of scenes. It was found in Section 6.3 that the scenes are independent, indicating that the scenes are correctly selected as blocks of frames in the scene blocks estimator. The scene blocks estimate is shown in Figure 6.10(a), together with the blocks and runs estimates.

The separation of the video trace into four classes with different average bitrates




(a) Class 1, $k = 100$, $r = 47$

(b) Class 2, $k = 50$, $r = 40$

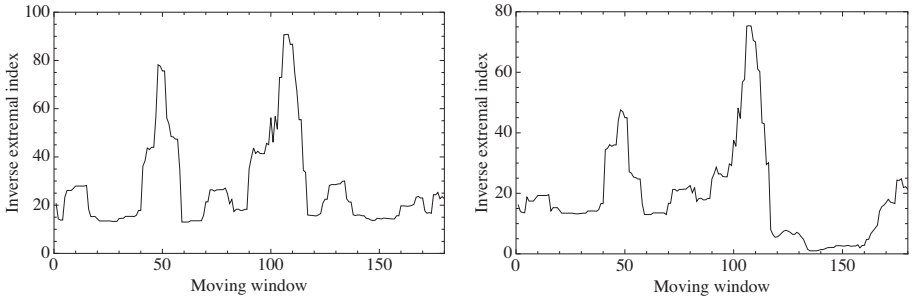
(c) Class 3, $k = 15$, $r = 19$

(d) Class 4, $k = 10$, $r = 12$

**Figure 6.12:** Blocks, scene blocks, and runs inverse estimates $1/\overline{\theta}^B$, $1/\overline{\theta}^B_S$, and $1/\overline{\theta}^R$ for the classes.

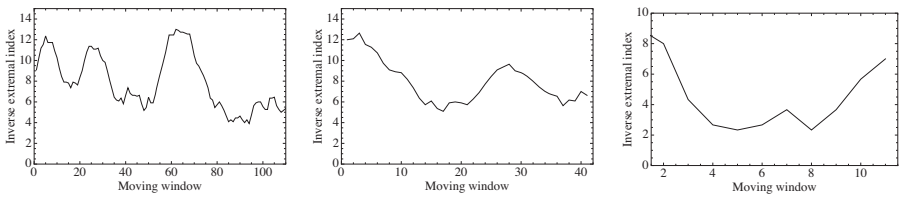
also influences the extremal index. For example, the thresholds beyond which the losses are independent for the whole trace will give no loss at all for Class 2. The extremal index is therefore investigated for each separate class, assuming stationarity of the process within each class. The blocks, scene blocks, and runs estimates are shown in Figure 6.12 as functions of the threshold $u$. The number of blocks $k$ (and $r$) is lower for Class 3 and 4 because of the lower number of observations for these classes. For Class 1 in Figure 6.12(a), the inverse extremal index is close to 1 for thresholds above 25 kbytes, which means that the losses over this threshold are weakly dependent or almost independent. For Class 2 the same is true for thresholds above 10 kbytes, while for Class 3 and 4 the same threshold as for the whole trace is needed for independence. These results can be used to calculate losses for the classes separately and is used in Chapter 7.

It can be seen that the scene blocks estimate behaves similarly to the other estimates, at least for the high thresholds. However, this method does not require the selection of an appropriate fixed $k$ (or $r$).




(a) Threshold $u = 5$ kbytes (b) Threshold $u = 8$ kbytes

**Figure 6.13:** The inverse extremal index from the scene blocks method with moving window equal to ten scenes for the entire sample.




(a) Class 1, $m$=10 (b) Class 2, $m$=10 (c) Class 3, $m$=2

**Figure 6.14:** The inverse extremal index from the scene blocks estimate for Class 1-3. The threshold $u$ is equal to the 80% quantiles of frame sizes for the individual classes.

### 6.5.1 Classification by the Extremal Index

Next, the frame sizes are classified by the value of the extremal index $\theta$. It implies that the video trace is divided according to the dependence of frames in the scenes of each class. The following procedure is proposed for the classification. The extremal index (or its inversion) is calculated sequentially for a moving window containing $m$ scenes with $m = 10$. The interpretation of the classification by $\theta$ is the variation of the dependence within the scenes. Figures 6.13 and 6.14 show the estimate $1/\widehat{\theta}$ against a number of moving windows for the scene blocks estimate for the entire trace and the classes, respectively. Scenes are selected by the 80% quantile method.

For the entire sample, periodicity in the dependence structure reflecting the high variability in the data can be observed. The periodicity exists also in Class 1 and 2 but with lower variability. For Class 3 the dependence is more stable. The closer the extremal index is to 1 (or to 0) the stronger is the asymptotic independence (or dependence) of maximal frame sizes inside scenes of the moving window. Class 4 is not shown due to the small number of scenes.

Furthermore, the extremal index can detect non-stationarity in the data as

follows. The asymptotic distribution of the inter-exceedance times was proved in [131] to be Exponential with an intensity equal to the extremal index. More exactly, under a specific mixing condition, inter-exceedance times normalized by $\overline{F}(u_n) = P\{X_i > u_n\}$ converge in distribution to a random variable $T_\theta$ which is zero with probability $1-\theta$ and strictly positive with probability $\theta$. Namely, $P\{\overline{F}(u_n)T(u_n) > t\} \to \theta \exp(-\theta t)$ for $t > 0$ as $n \to \infty$, where $T(u_n) = \min\{n \geq 1 : X_{n+1} > u_n | X_1 > u_n\}$ are inter-exceedance times in the stationary sequence $\{X_i\}$ corresponding to the threshold $u_n$. In this respect, the extremal index may detect non-stationarity in the data.

When looking at the extremal index estimates in Figure 6.14, Class 2 and 3 seem more stationary than Class 1 with respect to the more homogeneous inter-exceedance time distribution.

## 6.6 Conclusion

In this chapter there are several ideas that are interesting both regarding new statistical tools and video traffic inference. It is proposed to divide the video stream into sections and classify the sections by the average frame size. A new quantile method for scene change detection is then proposed for the classes. The resulting classes are checked regarding distribution and dependence structure.

The ACFs and Ljung-Box tests show independent scenes in the classes, but a moderate amount of LRD is found using different estimators for the Hurst parameter. The distributions of the scenes inside the classes are analyzed by the mean excess function and the tail index and it is found that the distributions of frame sizes of the selected classes can only be mixtures of classical heavy- and light-tailed distributions.

Furthermore, the extremal index which detects the changes of the stationarity and the dependence of frames within scenes is investigate. A new scene blocks estimate for the extremal index is proposed, where blocks are specified by scenes. The extremal index shows a variable dependence structure within the classes, due to the variability of the video stream.

The proposed approach is merely non-parametric. It implies that the estimation of the distribution of the underlying random variables, e.g., positive exceedances, inter-exceedance times etc., arising during the analysis is avoided.

The results of the analysis for a test flow cannot be generalized to any video flow due to high variability of video data. However, the proposed methodology for the classification of the stream can be extended to any flow including an aggregated stream.

# Chapter 7

# Estimation of Loss from Threshold Exceedance

This chapter uses the classification of the slice-based encoded StEM clip from Chapter 6, and estimates the bit loss for the video stream from the exceedance of frame sizes over a threshold. A non-parametric approach is proposed and a bufferless model of the communication system is considered. The characteristics of the losses, including the average loss and the clustering of losses, are estimated and the high quantiles of the losses are found.

## 7.1 Introduction

Characterization of the packet loss, i.e., the loss ratio and the loss distibution, is an intermediate step to estimate the perceived QoS for a video stream sent over a network as described in Section 2.2. Only bit losses due to congestion in the network are considered here. For real-time data, packets that experience extensive delay through the network will also be dropped when arriving at the receiver's playout buffer, this is not taken into account in this work.

A bufferless model of a communication system is used. In [132], the justification of the bufferless fluid model is given and the amount of bits and packets lost during congestion periods is discussed. It is argued that desirable queueing delays, especially in interactive audio and multimedia systems, require small buffers. The congestion periods are denoted loss period here, and they resemble the clusters from Chapter 6. These clusters have at least one frame exceedance over a threshold. Regarding the bufferless model, the bit losses are calculated using the exceedances of the frame sizes over a threshold during the loss periods.

The slice-based encoded StEM clip that was classified in Chapter 6 is studied, and the results from the classification are employed to estimate the bit loss for the individual classes. The loss assessment is carried out by means of the extremal index and the mean excess function, which were estimated in the previous chapter,

both for the whole trace and for the individual classes. The characteristics for the average loss volume in the loss periods as well as the number of frames between loss periods are important for estimating the burstiness of the losses.

The threshold $u$ such that the exceedances over $u$ are weak-dependent can be evaluated. In the case of independence, the high quantiles, i.e., the quantiles that are close to 100%, e.g., 99% and 99.9%, can be estimated. The amount of losses can exceed these quantiles only with a very small probability.

### 7.1.1 Chapter Outline

The rest of this chapter is organized as follows. The estimation of the average loss for the bufferless model is carried out in Section 7.2, statistics for the loss and the clustering of loss in the individual classes are also presented. In Section 7.3, the high quantiles of positive exceedances of frame sizes are calculated. The chapter is concluded in Section 7.4.

## 7.2 Estimation of Loss in the Bufferless Model

In Chapter 6, the exceedances of the video frames over the high quantiles showed that the large frame sizes are the extreme events of interest when estimating loss. It is also observed that the frame sizes can be separated into four classes where the classes have different average frame size. The loss estimation can then be carried out for the individual classes.

The estimation of the loss for the video stream under study is based on the evaluation of the cluster structure of the data as shown in Figure 7.1, where a cluster or a loss period is defined as a period in which one or more frame sizes are larger than the threshold.

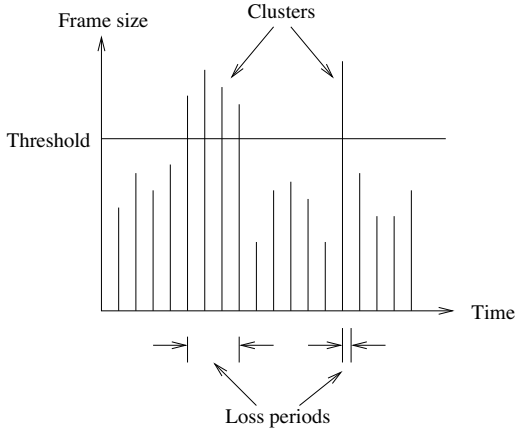
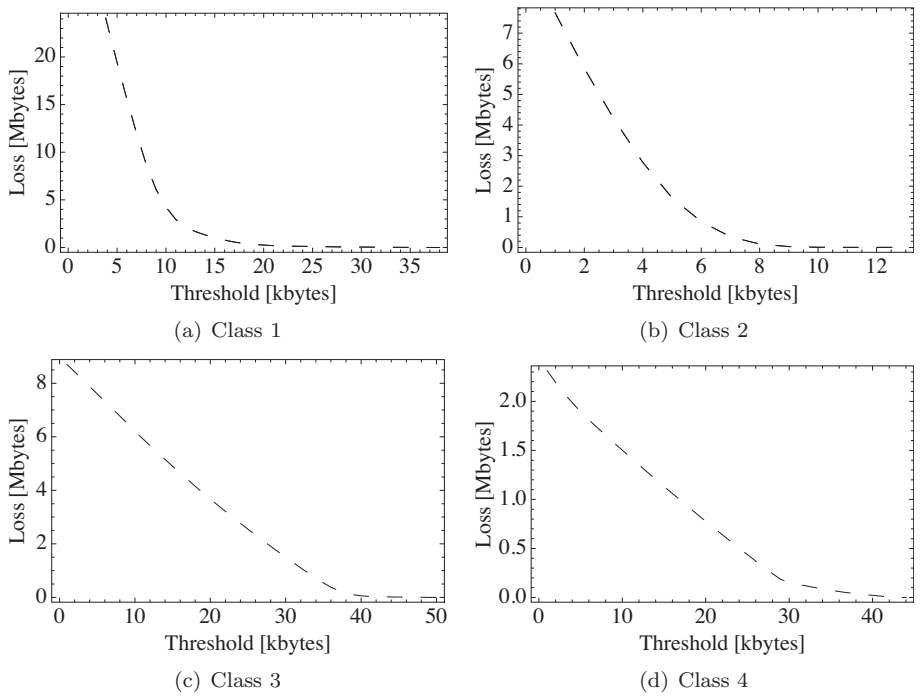



**Figure 7.1:** The cluster exceedance structure showing that the clusters correspond to the loss periods.




(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

**Figure 7.2:** The bit loss against the threshold $u$ over all clusters for each class.

In Chapter 6, the clusters are identified and the extremal index, giving the mean number of exceedances in the clusters, is evaluated in order to check the dependence and stationarity in the classes. Three different definitions are given for the clusters, giving three different estimators for the extremal index; the blocks estimator, the scene blocks estimator, and the runs estimator.

The overall bit loss $E(u)$ over all clusters for a fixed threshold $u$ coincides with the cumulative exceedance over the threhold $u$ of the entire trace, i.e.,

$$\begin{aligned} E(u) =& e(u) \cdot \theta(u)^{-1} \cdot K(u) \\ =& \frac{\sum_{i=1}^{n}(X_i - u)\mathbf{1}\{X_i > u\}}{\sum_{i=1}^{n}\mathbf{1}\{X_i > u\}} \cdot \frac{\sum_{i=1}^{n}\mathbf{1}\{X_i > u\}}{K(u)} \cdot K(u) \\ =& \sum_{i=1}^{n}(X_i - u)\mathbf{1}\{X_i > u\}, \end{aligned}$$

where $K(u)$ is the number of clusters, $e(u)$ is the mean excess function, and $\theta(u)$ is the extremal index. The estimate of the bit loss $\hat{E}(u)$ for the underlying frame size data against the threshold $u$ for all classes is shown in Figure 7.2.

Naturally, the bit loss decreases with increasing threshold. Also, the rate of decrease for the losses is different for the classes, being lowest for Class 3 and 4

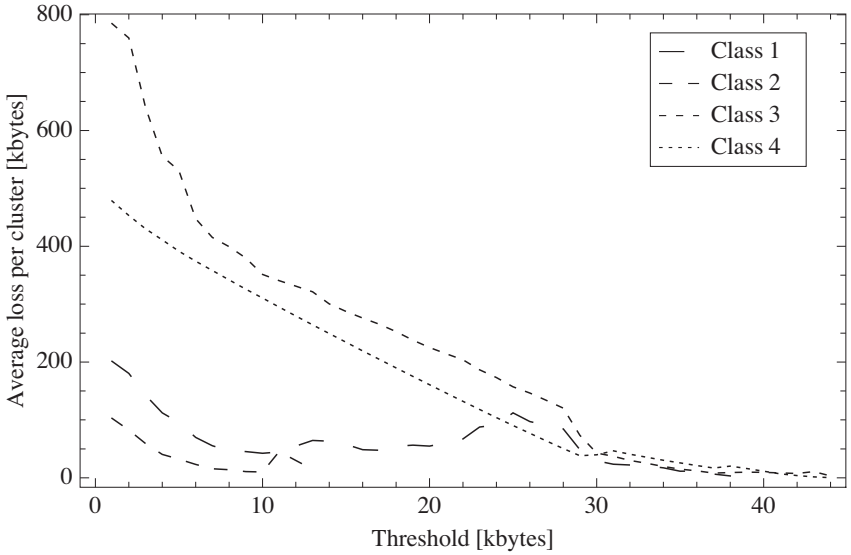



**Figure 7.3:** The average loss per cluster against the threshold $u$ for each class, using the scene blocks estimate.

because the amount of exceedances for these classes does not change much up to the thresholds $u = 40$ and $u = 30$, respectively, in contrast to Class 1 and 2.

In addition, it is interesting to study the average loss per cluster, corresponding to the average loss volume in a loss period. The average loss per cluster is determined by the sample mean excess function and the inverse extremal index estimate as $\hat{e}(u) \cdot \theta(u)^{-1}$. This is an important item for evaluating the perceived QoS, since it evaluates the burstiness of the losses. The burtiness of losses influences the perceived QoS of video traffic transmitted over a network as observed e.g., in [65]. The average loss volume in a loss period, when combined with the overall bit loss and the average length of a non-loss period, reflects the loss structure in the following sense. The occurrence of a few loss periods with a large loss volume or more frequent loss periods with a lower loss volume may influence the perceived QoS since the same overall loss ratio and different loss volumes in the loss periods give different distributions of frames between loss periods (clusters). The average loss per cluster for each class is given in Figure 7.3, calculated using the scene blocks estimate.

Statistics for the loss periods are given in Table 7.1, where also the threshold required for each class to satisfy a given loss ratio, equal to 3%, is shown. The average loss volume and average length of non-loss period are evaluated for this loss ratio. It can be seen that Class 3 needs a higher threshold than the other classes to fulfill the loss requirement, while Class 2 needs the lowest. The average loss volume in a loss period as well as the average length of a non-loss period are also shown in Table 7.1. For the required loss bound, the average loss volume

**Table 7.1:** The 3% overall loss ratio, average loss volume per loss period, average length of non-loss period, and the corresponding threshold for each class

| Class | Threshold [kbytes] | 3% loss [kbytes] | Loss volume [kbytes] | Length of non-loss period [number of frames] |
|---|---|---|---|---|
| 1 | 15 | 1033.71 | 24.23 | 110.14 |
| 2 | 8 | 109.85 | 3.39 | 63.00 |
| 3 | 37 | 258.12 | 21.44 | 24.51 |
| 4 | 35 | 74.68 | 25.72 | 42.41 |

in a loss period is approximately the same for Class 1, 3 and 4, while it is much lower for Class 2. The average length of a non-loss period is given as the average number of frames between loss periods, being largest for Class 1 and smallest for Class 3. This indicates that Class 1 has large (in terms of loss volume) and rare loss periods as far as Class 2 has more frequent and smaller loss periods and Class 3 has very frequent and large loss periods.

An overall capacity corresponding to the threshold $u$ equal to 37 kbytes is required to host all traffic classes. It is calculated as the maximum of the requirements for the individual classes. This capacity provides less than 3% overall loss ratio for Class 1-4.

As is discussed in Section 2.2, and also shown in several papers [50, 65, 67], the same loss probability and a variable number of consecutive lost packets give different perceived QoS for video traffic. Estimating the clustering of losses therefore gives valuable information that can be used for assessing the PQoS as well as being applied in the video encoding, for selecting error resilience tools.

## 7.3 High Quantile Estimation of Losses

To evaluate the high quantiles of the losses, only positive exceedances $\{Y_i = X_i - u, i = 1, 2, ..., n\}$ with unknown df $F(y)$ are considered. These are calculated from the measured frame sizes $\{X_i,\ i = 1, ..., n\}$ for a fixed threshold $u$. Since $\{Y_i\}$ form the losses in the bufferless model, their quantiles should be estimated, in particular, high quantiles close to 100%. The high quantiles of exceedances determine the bound on the amount of loss that can occur with a given probability.

High quantiles are, as a rule, located outside the range of the sample (i.e., the interval between the minimum and maximum values of the underlying sample). Hence, they cannot be evaluated by means of the empirical df or other df estimators based on the sample only. The estimators of high quantiles are based on models of the tail of the distribution [119]. The Generalized Pareto df,

$$\Psi_{\sigma,\gamma}(x) = \begin{cases} 1 - (1 + \gamma x/\sigma)^{-1/\gamma}, & \gamma \neq 0, \\ 1 - \exp(-x/\sigma), & \gamma = 0, \end{cases} \tag{7.1}$$

where $\sigma > 0$ and $x \geq 0$, as $\gamma \geq 0$; $0 \leq x \leq -\sigma/\gamma$, as $\gamma < 0$, or Pareto-type df

$$F(x) = 1 - cx^{-1/\gamma}\left(1 + dx^{-\beta} + o(x^{-\beta})\right), \tag{7.2}$$

**Table 7.2:** Extremal index using the blocks estimator for given thresholds.

| Data | $u$ | $Y_{max}$ | $N_e$ | $u_y$ | $\overline{\theta}^B(u_y)$ |
|---|---|---|---|---|---|
| Class 1 | 15 | 23.930 | 271 | 20 | 1.25 |
| Class 2 | 8 | 5.949 | 131 | 3 | 1 |
| Class 3 | 37 | 13.294 | 124 | 11 | 1 |

where $\gamma > 0$, $\beta > 0$, $c > 0$, $-\infty < d < \infty$, are often used to approximate the distribution beyond the sample location where there are no observations. To estimate the $(1-p)$th, $p \in (0,1)$, high quantile of exceedances the well known Weissman's estimator [114] is used,

$$x_p^w = Y_{(n-k_0)} \left( \frac{k_0+1}{(n+1)p} \right)^{1/\hat{\alpha}}, \qquad k_0 = 1, \ldots, n-1. \tag{7.3}$$

It was built for the Pareto model, where $Y_{(1)} \leq Y_{(2)} \leq \ldots \leq Y_{(n)}$ are the order statistics of the sample $Y_1, ..., Y_n$ and $\hat{\alpha}$ is some estimate of the tail index $\alpha$. In case of weakly dependent data, the Hill's estimate can be used for $\hat{\alpha}$ and then $k_0$ is the smoothing parameter of the Hill's estimator as described in Section 6.4.

When $p \ll n^{-1}$ the high quantiles are extrapolated outside the sample, hence losses which have not yet occurred are evaluated.

The prerequisites of the Weissman's estimator are that the underlying sample $Y_1, ..., Y_n$ is independent (or weak dependent) and stationary, and that the corresponding tail index $\alpha$ is positive. The latter indicates that the distribution is heavy-tailed. Therefore, these properties are checked. For this purpose, the extremal index $\theta(u)$ is calculated by the blocks estimator $\overline{\theta}^B(u)$ as described in Section 6.5. A value of $\theta(u)$ near 1 indicates asymptotic independence (i.e., when $n \to \infty$) of the corresponding positive exceedances.

Table 7.2 shows the threshold $u$ used to determine the positive exceedances, the number of positive exceedances $N_e$ and the maximum of these exceedances $Y_{max}$ for each class. The threshold $u$ is chosen as the threshold that provided 3% overall loss ratio for the individual classes in Table 7.1. The thresholds $u_y$ for each class are then selected to give blocks estimates of $\theta(u_y)$ close to 1. This corresponds to almost independent positive exceedances.

Figure 7.4 shows the inverse extremal index estimates using the blocks estimate against the threshold $u_y$. These plots help us to select the $u_y$ that give approximately independent positive exceedances. Class 4 is excluded from the calculation of the high quantiles and extremal index due to the small number of observations and thereby the impossibility to detect enough independent exceedances.

The high quantiles for the losses corresponding to an overall loss ratio of 3% is given in Table 7.3. The same thresholds $u$ as in Table 7.2 is used for estimating the high quantiles. These thresholds correspond to weakly dependent exceedances and are low enough to provide a sufficient number of exceedances for the calculation of the high quantiles. In [129] it is proved that a mild dependence condition is sufficient for the consistency of the high quantile estimate that is very similar to

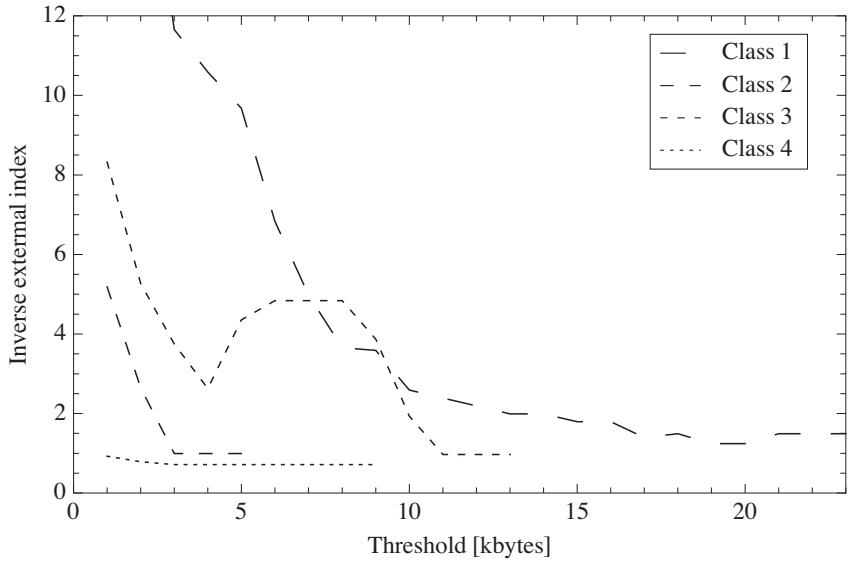



**Figure 7.4:** The extremal index estimation for positive exceedances for classes using the blocks estimate, $k = 10$.

Weissman's estimate. The thresholds $u$ normalized by time can be interpreted as the channel capacity required to get an upper bound for the losses. The latter is determined by the high quantiles.

The Hill's estimate used for the calculation of the high quantiles and its non-asymptotic confidence interval [133] are also presented in Table 7.3. This confidence interval of level $1 - \epsilon$, is calculated by formula

$$I_\varepsilon = \Big[\frac{\hat{\gamma}^H(n, k_0)}{1 + y_\epsilon N_n^{-1/2}}; \frac{\hat{\gamma}^H(n, k_0)}{1 - y_\epsilon N_n^{-1/2}}\Big], \tag{7.4}$$

where $N_n = \sum_{i=1}^{n} \mathbf{1}\{Y_i > u\}$ is the number of exceedances over the threshold $u$ for the positive exceedances $Y_i$ and $y_\epsilon$ is calculated as a quantile of the standard normal distribution $\Phi(x)$. Namely, $\Phi(-y_\epsilon) = (\epsilon/2 - C_* N_n^{-1/2})_+$, where $C_* < 0.8$ is a constant. $C_*$ is chosen such that the narrowest confidence interval is provided. Hill's estimate should be inside this interval. $\epsilon = 0.05$ is selected. The empirical quantiles are also presented.

The Hill's estimates indicate that the exceedances from all the considered classes have heavy tails. Class 2 has the lightest tail. It is reflected in the difference between the 99% and 99.9% quantiles. This difference is larger for Class 1 and 3 than for Class 2. The high quantiles indicate that the largest upper bounds of losses with probabilities 99% and 99.9% are for Class 1 and the smallest losses are arising for Class 2. This is caused by the heaviness of tail of the distribution of positive exceedances for Class 1. Nevertheless, Class 2 requires a capacity that is two times lower than Class 1. The loss for Class 3 is half of the loss for Class

**Table 7.3:** Estimation of the high quantiles for the positive exceedances data.

| Class | Tail index | | High quantile | | Empirical quantile | |
|---|---|---|---|---|---|---|
| | Hill's | 95% CI | 99% | 99.9% | 95% | 99% |
| Class 1 | 2.1234 | [1.8975, 2.4104] | 21.725 | 64.250 | 9.692 | 23.403 |
| Class 2 | 4.5174 | [3.8569, 5.4508] | 3.028 | 5.042 | 2.061 | 2.394 |
| Class 3 | 2.6528 | [2.2557, 3.2194] | 10.591 | 25.230 | 5.299 | 10.339 |

1 with a probability 99%, but the required capacity is approximately two times higher.

## 7.4 Conclusion

The bit losses arising from exceedances of frame sizes over a threshold are estimated using a bufferless model. The video stream under study is classified in Chapter 6, and the bit losses are estimated for the classes. The average loss volume in a loss period as well as the average length of a non-loss period are found for the individual classes, giving information about the clustering of losses. This gives valuable information to use for assessing the perceived QoS and can be applied in the video encoding.

Moreover, the high quantiles of bit losses, which determine the upper bounds of the bit losses for a fixed threshold are evaluated. In addition, the capacity that is required to satisfy the maximum allowed loss ratio for each class is estimated.

The approach to characterization of the losses is merely non-parametric. It implies that estimating the distribution of the underlying random variables, e.g., the positive exceedances, arising during the analysis is avoided.

# Part IV

# Characterization of Loss for Aggregated Video Using a Gaussian Model

The results in this part have been published as follows:

Astrid Undheim and Peder J. Emstad. "Distribution of Loss Periods for Aggregated Video Traffic." In *Proceedings of the ITC Specialist Seminar 18 (ITCSS'18)*, Karlskrona, Sweden, May 2008.

Astrid Undheim and Peder J. Emstad. "Distribution of Loss Volume and Estimation of Loss for Aggregated Video Traffic." Submitted for publication, 2009.

# Chapter 8

# A Gaussian Model for Aggregated Video Traffic

In this chapter, slice-based encoded video traffic is modeled as a discrete Gaussian process, taking the correlation between consecutive frames into account. This model can hence be used for studying the effects of frame correlation on the loss distribution. An aggregate of independent video streams will also be Gaussian and relations between the exceedances of frames sizes over a threshold for a basic stream and an aggregated stream are found. Results for the characteristics of exceedances for the continuous Gaussian model are introduced, and a relation between the length and volume of an excursion is found. This relation can be exploited for the discrete model as well.

## 8.1 Introduction

In general, video traffic models are needed for two purposes, to simulate such traffic in a network and to establish an analytical model. In this chapter, an analytical traffic model for video encoded using the slice-based H.264/AVC video encoding scheme as described in Chapter 3, is developed.

In the Internet, the video traffic will not appear as single streams but as aggregates of several video streams with similar means and correlation structures. Evaluation of the packet loss for the aggregate by knowledge about the single stream is of great interest. In particular, the loss probability, the distribution of the length and loss volume of a loss period, and their variation with some key parameters are interesting. To accurately grasp the distribution of the length and loss volume of a loss period it is necessary to apply a model that takes the covariance function into account. Also, it is of paramount interest to choose a type of model that can model an aggregate based on similar models of its single streams and which is parsimonious in parameters. A general Gaussian process lends itself for this purpose thanks to its simple additive properties. It only

requires knowledge of the means and the covariance functions of the aggregate members. In Chapter 4, correlation is found to exist mainly within scenes for a slice-based encoded video stream. Hence, in this work, the effect of short- and medium-range dependence on the characteristics of a loss period is studied and a general Gaussian process defined by its mean and covariance matrix is used. The exceedances of the frame sizes over a threshold constitute the loss for a video transmission over a bufferless link.

### 8.1.1 Related Work

Popular models used for simulations of video traffic in the last decades include Markov-type models and Autoregressive Moving Average (ARMA) models, and also combinations of these. These models can model short-range dependence. DPCM encoded video is modeled in [87], and both a Markov chain model and a DAR(1) model is used for modeling the dependence of the intra-scene frames. For video conferencing, the Gamma Beta AR (GBAR) model is proposed in [134], and a survey of regressive models for use in video conferencing is given in [135]. The GBAR model from [134] is extended to non-video conference video in [94], where a GOP GBAR model is proposed to explicitly model the GOP structure of MPEG video.

Long-Range Dependence (LRD) in VBR video traffic was extensively analyzed in [120], leading to the development of new types of video models. In [88], two AR(2) models are nested together to model the LRD of VBR MPEG video. One AR(2) model is used to model the mean I frame size in consecutive scenes, hence preserving the LRD. A second AR(2) process is used for modeling the deviation from the mean I frame size in each scene. This nested model is enhanced in [136], and a nested AR multinomial model is proposed for MPEG-4 video traffic modeling. Nested Markov models are proposed for video traffic modeling in [90], and it is argued that this type of model can model the LRD if the scene level is modeled as one Markov chain and the GOP sizes within the scene states is modeled as another Markov chain. A Markov renewal process is proposed for the modeling of MPEG encoded video traffic in [93], where the states correspond to different classes of GOP sizes. A scene-based Markov modulated process is also proposed for modeling of MPEG-4 video in [97] and it is shown that the model captures the LRD of the video traces under study.

Although the nested AR model from [88] and the scene-based Markov models from [90, 97] can incorporate the LRD of video traces, specific models are also developed for modeling of LRD. Self-similarity models include the Fractional Autoregressive Integrated Moving Average (FARIMA) model which is employed in [137].

A new approach to video source modeling, introduced after the discovery of LRD in video traffic, is the use of wavelet models. These models are advantageous since they can model a complicated short- and long-range dependence structure in the time domain using a short-range model in the wavelet domain. In [138], a hybrid wavelet approach is proposed, modeling the I frames in the wavelet domain

and using intra GOP correlation to model the P and B frames, using a linear model. In [98], the approach is extended to multi-layer MPEG-4 and H.264/AVC video traffic, also modeling the I frames in the base layer using wavelets.

Other proposals for video source models intended for simulation studies include the Transform Expand Sample (TES) model, combined with a Markov-Renewal model in [139] and the $M/G/\infty$ model proposed in [140].

For the analysis, a different approach to modeling is needed, which ensures analytical tractability. Renewal-type models such as Poisson models are attractive for queueing analysis models. However, these models can only model independent arrivals and are therefore unsuitable for modeling video traffic as described in [141].

Markov-type models are also attractive from a queueing analysis perspective. These can incorporate the autocorrelation using nested Markov chains as is proposed in [90]. A periodic Markov Modulated Batch Bernoulli Process (P-MMBBP) is introduced to model MPEG video traffic in [142]. This type of model can model a periodic, exponentially decaying ACF. A P-MMBBP/D/1 queueing system is analyzed, and the queue length distribution is found analytically. Finally, a matrix-analytical approach for the multiplexing of VBR sources is developed in [143], where the video traffic is modeled using a time discrete Batch Markovian Arrival Process (D-BMAP).

Fluid-type models can be used to model aggregated video traffic. In [144], a fluid-flow model is used for video traffic in order to develop a rate control scheme for video over the Internet. In [32], a fluid-flow model is proposed for the queueing analysis of a statistical multiplexer with video traffic input.

Wavelet models can be employed for queueing analysis as well. In [145], video traffic is modeled in the wavelet domain, and a queueing formula is developed for estimating the tail queue probability for an infinite buffer. A wavelet model is also proposed for the modeling of video traffic in [146], and a similar queueing analysis is performed.

For analyzing service guarantees, the token bucket traffic models such as those described in Section 2.3 are interesting. The EF class in DiffServ and the Guaranteed Service class in IntServ are both defined using network calculus server models, and these give deterministic delay guarantees to token bucket constrained input traffic. Token bucket modeling for video traffic is described in Chapter 5, and related work on video traffic modeling using token buckets is described in Section 5.1.1. In particular, in [108] and [147] a Switched Batch Bernoulli Process (SBBP) is proposed for the modeling of video traffic for the analytical evaluation of token bucket performance.

Gaussian processes with independent increments have since long been used to model queueing type systems. Fractional Brownian Motion (FBM) models, which have a corresponding increment process that is Fractional Gaussian Noise (FGN), have become popular for modeling long-range dependencies [121]. However, to use this kind of model, the trace must match the autocorrelation of FGN. In [148], this FBM model is compared to a Markov Modulated Fluid Flow (MMFF) model for the modeling of the variance function of H.264/AVC encoded video, and the variance is used as input to an effective bandwidth approach for estimating the buffer exceedance probability. The results show that it is difficult to match the

ACF of real video traces to the FBM model.

In this work, the effect of short- and medium-range dependence on the loss period is studied and a general Gaussian process defined by its mean and covariance matrix is used. This is motivated by the correlation analysis in Chapter 4, showing that correlation exists only within scenes.

Previous work on continuous Gaussian processes in [149] gives the limit distribution functions of the length, volume, and maximum height of excursions over high thresholds. However, the covariance process is continuous in this case. Based on the results in [149], a relation between the distributions of the length and volume of the excursions can be derived for the continuous process. A relationship between the length and loss volume of a loss period for a discrete process is of similar interest and is investigated further in Chapter 9.

### 8.1.2 Chapter Outline

The rest of this chapter is organized as follows. The multivariate normal distribution is described in Section 8.2, while the model for aggregated video traffic is developed in Section 8.3. The excursion characteristics for the continuous Gaussian process are described in Section 8.4 and the requirements for the correlation function for the continuous process are investigated followed by a study of two permissible functions. Finally, some conclusions are given in Section 8.5.

## 8.2 The Multivariate Normal Distribution

Let $\boldsymbol{Z} = (Z_1, ..., Z_m)$ be a vector of independent and identically distributed standard normal random variables with zero mean and covariance matrix equal to $Cov(\boldsymbol{Z}) = I_m$, where $I_m$ is the identity matrix of dimension $m$.

The random vector $\boldsymbol{Z}$ has a standard multivariate normal distribution if the density of $\boldsymbol{Z}$, $f_{\boldsymbol{Z}}(\boldsymbol{z})$ is equal to:

$$f_{\boldsymbol{Z}}(\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^m}} \exp\Big( -\frac{1}{2}\sum_{i=1}^{m} z_i^2 \Big) \tag{8.1}$$

The random vector $\boldsymbol{X} = (X_1, ..., X_m)$ is defined as $\boldsymbol{X} = \boldsymbol{\mu_X} + \Gamma \boldsymbol{Z}$, which means that $\boldsymbol{X}$ has mean $E(\boldsymbol{X}) = \boldsymbol{\mu_X} = (\mu_{X_1}, .., \mu_{X_m})$, where $\mu_{X_i} = \mu_X$ and covariance matrix $Cov(\boldsymbol{X}) = r$, where $r = \Gamma\Gamma^T$ and $r$ is assumed to be non-singular.

$\boldsymbol{X}$ has a multivariate normal distribution if the density of $\boldsymbol{X}$, $f_{\boldsymbol{X}}(\boldsymbol{x})$ is equal to:

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^m |r|}} \exp\Big( -\frac{1}{2}\sum_{i,j} (x_i - \mu_X) r_{ij}^{-1} (x_j - \mu_X) \Big) \tag{8.2}$$

where $r_{ij}^{-1}$ is element $(i, j)$ of the inverse of the covariance matrix $r$ of $\boldsymbol{X}$.

The distribution of $\boldsymbol{X}$ is determined by the mean vector, $\boldsymbol{\mu_X}$ and the covariance matrix, $r = \Gamma\Gamma^T$. For each $\boldsymbol{X}$ with these properties the corresponding $\boldsymbol{Z}$ is given by $\boldsymbol{Z} = \Gamma^{-1}(\boldsymbol{X} - \boldsymbol{\mu_X})$ which is standard multivariate normally distributed.

Throughout this work the probability of $m$ consecutive exceedances of the process $\boldsymbol{X}$ over a threshold $a$ is studied, using the Multivariate Normal Integral (MVNI).

The MVNI for the multivariate normal random vector $\boldsymbol{X}$ is then given as:

$$\begin{aligned} &Pr[a \leq X_1 \leq \infty, a \leq X_2 \leq \infty, ..., a \leq X_m \leq \infty] \\ &= \int_a^\infty \cdot\cdot \int_a^\infty f_{\boldsymbol{X}}(x_1, x_2, .., x_m) dx_1 \cdot \cdot dx_m \\ &= \frac{1}{\sqrt{(2\pi)^m |r|}} \int_a^\infty \cdot\cdot \int_a^\infty \exp\Big( -\frac{1}{2} \sum_{i,j} (x_i - \mu_X) r_{ij}^{-1} (x_j - \mu_X) \Big) dx_1 \cdot \cdot dx_m \end{aligned} \quad (8.3)$$

In the next section, the probability of exceeding the threshold is evaluated for an aggregated process and relations between this probability for the aggregate and single non-zero mean and zero mean processes are found.

## 8.3 Model for Aggregated Multimedia Traffic

In the Internet, video traffic will appear as aggregates of single video streams. For modeling purposes it is reasonable to assume independence between sources and between streams of an aggregate. Each stream is then modeled with a Gaussian process and aggregates can easily be represented based on parameters for their individual streams. The effect of correlation in a single stream and an aggregate can then be studied.

Video sources will not be synchronized in time and will start their cyclic frame transmissions randomly. The lack of synchronization is overcome by looking at the accumulated number of bits sent over an inter-frame period. The loss or even the delay can then be estimated given knowledge about the capacity of the outgoing link. This could also require knowledge of some implementation issues at the node, but is not treated in this work. To this end it is assumed that the exceedance limit for acceptable loss in an inter-frame period is given, and this exceedance limit denotes the threshold in our model.

In this section, the probability of exceeding a threshold for an aggregated video stream modeled as a Gaussian process is evaluated. The aggregate can be described as a sum of single Gaussian processes. This aggregated process can be viewed as an aggregate of $n$ statistically equal and independent processes $\boldsymbol{X}$, called *basic* processes. In other words, the focus is on the distribution of the exceedances over a threshold for the aggregated multivariate normally distributed process, $\boldsymbol{Y}$, which can be viewed as consisting of $n$ statistically equal basic processes $\boldsymbol{X}$ in which case $\boldsymbol{Y} = \sum_{k=1}^{n} \boldsymbol{X}^k$.

This gives the following relations for the aggregated process, $\boldsymbol{Y}$ and its basic processes. The mean vector is given by:

$$\boldsymbol{\mu_Y} = E(\boldsymbol{Y}) = n \cdot E(\boldsymbol{X}) = n \cdot \boldsymbol{\mu_X} \quad (8.4)$$

The covariance function for the aggregate process, when the basic processes are independent, is given by:

$$Cov(Y_i, Y_j) = n \cdot Cov(X_i, X_j) \tag{8.5}$$

And the elements $(i, j)$ of the covariance matrix, $q$, for the aggregated process is given by:

$$q_{ij} = n \cdot r_{ij} \tag{8.6}$$

The density of $\boldsymbol{Y}$, $f_{\boldsymbol{Y}}(\boldsymbol{y})$ is then equal to:

$$f_{\boldsymbol{Y}}(\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^m |q|}} \exp\Big( -\frac{1}{2} \sum_{i,j} (y_i - \mu_Y) q_{ij}^{-1} (y_j - \mu_Y) \Big) \tag{8.7}$$

where $\boldsymbol{\mu_Y} = n \cdot \boldsymbol{\mu_X}$, $q_{ij}^{-1}$ is element $(i, j)$ of the inverse of the covariance matrix of the aggregated process $\boldsymbol{Y}$ and $q_{ij}^{-1} = \frac{1}{n} r_{ij}^{-1}$ since:

$$q_{ij}^{-1} = \frac{\text{adj } q}{|q|} = \frac{n^{m-1} \text{adj } r}{n^m |r|} = \frac{1}{n} r_{ij}^{-1} \tag{8.8}$$

where adj $q$ (respectively $r$) is the adjoint matrix of $q$ (respectively $r$). For the determinant of the covariance matrix of the aggregated process $|q|$, the relation $|q| = n^m |r|$ is used.

The multivariate normal integral for the aggregated process, with the lower threshold $b$, then becomes:

$$\begin{aligned} &Pr[b \le Y_1 \le \infty, b \le Y_2 \le \infty, ..., b \le Y_m \le \infty] \\ &= \int_b^\infty \cdot\cdot \int_b^\infty f_{\boldsymbol{Y}}(y_1, y_2, .., y_m) dy_1 \cdot\cdot dy_m \\ &= \frac{1}{\sqrt{(2\pi)^m |q|}} \int_b^\infty \cdot\cdot \int_b^\infty \exp\Big( -\frac{1}{2} \sum_{i,j} (y_i - \mu_Y) q_{ij}^{-1} (y_j - \mu_Y) \Big) dy_1 \cdot\cdot dy_m \end{aligned} \tag{8.9}$$

In order to represent the multivariate normal integral for the aggregated process in terms of a basic process, the relations between the basic stream and the aggregated stream are used, giving:

$$\begin{aligned} &Pr[b \le Y_1 \le \infty, b \le Y_2 \le \infty, ..., b \le Y_m \le \infty] \\ &= \frac{1}{\sqrt{(2\pi)^m n^m |r|}} \int_b^\infty \cdot\cdot \int_b^\infty \exp\Big( -\frac{1}{2} \sum_{i,j} (y_i - n\mu_X) \frac{1}{n} r_{ij}^{-1} (y_j - n\mu_X) \Big) dy_1 \cdot\cdot dy_m \end{aligned} \tag{8.10}$$

The variables for the integrals are changed in order to find an expression similar to Equation 8.3, using $\boldsymbol{f} = (f_1, f_2, .., f_m)$ as the new variable of integration, where $\boldsymbol{y} = u(\boldsymbol{f})$, hence $d\boldsymbol{y} = u'(\boldsymbol{f}) \cdot d\boldsymbol{f}$. In order to get rid of the aggregating factor

$n$ ahead of $\mu_X$ and $1/n$ ahead of $r_{ij}^{-1}$ in the integral, the integration variables $f_i$ should satisfy the following relation:

$$(y_i - n\mu_X) = (f_i - \mu_X) \cdot \sqrt{n} \tag{8.11}$$

The differential is then:

$$dy_i = \sqrt{n} df_i \tag{8.12}$$

and the lower integral limits $b$ are changed accordingly to $\hat{b}$. This gives:

$$\hat{b} = \frac{b}{\sqrt{n}} - \mu_X(\sqrt{n} - 1) \tag{8.13}$$

The Jacobian of the transformation $T$ which is needed when changing variables in multiple integrals is given by:

$$\begin{aligned} J_T(f_1, f_2, .., f_m) &= \frac{\partial(y_1, y_2, ..., y_m)}{\partial(f_1, f_2, ..., f_m)} \\ &= \begin{vmatrix} \frac{\partial(y_1)}{\partial(f_1)} & \frac{\partial(y_1)}{\partial(f_2)} & \cdots & \frac{\partial(y_1)}{\partial(f_m)} \\ \frac{\partial(y_2)}{\partial(f_1)} & \frac{\partial(y_2)}{\partial(f_2)} & \cdots & \frac{\partial(y_2)}{\partial(f_m)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial(y_m)}{\partial(f_1)} & \frac{\partial(y_m)}{\partial(f_2)} & \cdots & \frac{\partial(y_m)}{\partial(f_m)} \end{vmatrix} = \sqrt{n}^m \end{aligned}$$

since $\sqrt{n}$ is the partial derivative of each of the expressions on the diagonal, while the rest of the partial derivatives are zero. Including the Jacobian when changing the variable of integration then gives: $dy_1 \cdot \cdot dy_m = \sqrt{n}^m df_1 \cdot \cdot df_m$. This results in the following integral for the aggregated process:

$$\begin{aligned} &Pr[b \leq Y_1 \leq \infty, b \leq Y_2 \leq \infty, ..., b \leq Y_m \leq \infty] \\ &= \frac{1}{\sqrt{(2\pi)^m n^m |r|}} \int_{\hat{b}}^{\infty} \cdot \cdot \int_{\hat{b}}^{\infty} \sqrt{n}^m \exp\Big(-\frac{1}{2} \sum_{i,j} \sqrt{n}(f_i - \mu_X) \frac{1}{n} r_{ij}^{-1} \sqrt{n}(f_j - \mu_X)\Big) df_1 \cdot \cdot df_m \\ &= \frac{1}{\sqrt{(2\pi)^m |r|}} \int_{\hat{b}}^{\infty} \cdot \cdot \int_{\hat{b}}^{\infty} \exp\Big(-\frac{1}{2} \sum_{i,j} (f_i - \mu_X) r_{ij}^{-1} (f_j - \mu_X)\Big) df_1 \cdot \cdot df_m \end{aligned} \tag{8.14}$$

where $|r|$ is the determinant of the covariance matrix of $\boldsymbol{X}$ and $r_{ij}^{-1}$ is element $(i, j)$ of the inverse of the covariance matrix. The integral in Equation 8.3 is recognized with another lower integration limit, hence:

$$\begin{aligned} &Pr[b \leq Y_1 \leq \infty, b \leq Y_2 \leq \infty, ..., b \leq Y_m \leq \infty] \\ &= Pr[\hat{b} \leq X_1 \leq \infty, \hat{b} \leq X_2 \leq \infty, ..., \hat{b} \leq X_m \leq \infty] \\ &= Pr\left[\frac{b}{\sqrt{n}} - \mu_X(\sqrt{n} - 1) \leq X_1 \leq \infty, .., \frac{b}{\sqrt{n}} - \mu_X(\sqrt{n} - 1) \leq X_m \leq \infty\right] \end{aligned} \tag{8.15}$$

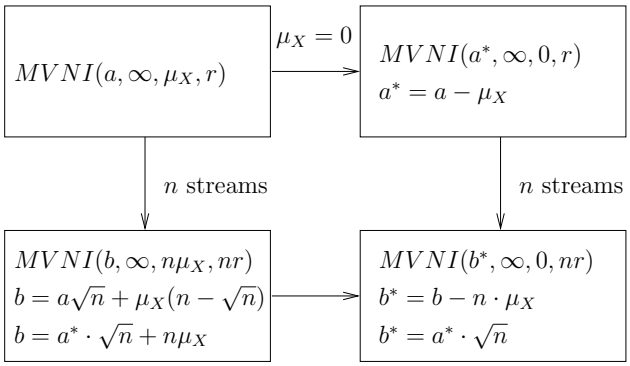



**Figure 8.1:** The relations between the single and aggregated processes.

A relation between non-zero mean and zero mean (denoted $\boldsymbol{X}^*$) single processes is straightforward, where the lower integration limit for the latter will be equal to $\hat{b} - \mu_X$. This in turns leads to the following relation between the non-zero mean aggregate and the zero-mean single process:

$$Pr[b \leq Y_1 \leq \infty, b \leq Y_2 \leq \infty, ..., b \leq Y_m \leq \infty]$$

$$= Pr\left[\frac{b}{\sqrt{n}} - \mu_X(\sqrt{n} - 1) - \mu_X \leq X_1^* \leq \infty, .., \frac{b}{\sqrt{n}} - \mu_X(\sqrt{n} - 1) - \mu_X \leq X_m^* \leq \infty\right] \tag{8.16}$$

The relations between the MVNI for the single and aggregated processes with zero mean and non-zero mean are summed up in Figure 8.1, with the limits for the MVNI (lower integration limit, upper integration limit, mean, covariance matrix).

For the expression in Equation 8.3, the threshold exceeding probabilities for different thresholds can be found numerically using the Cholesky transform as described in [150], or experimentally using simulations. In the latter case it suffices to study a single basic process.

## 8.4 Limit Distributions for Characteristics of Excursions

Expressions for the limit distributions of characteristics of excursions over a high level are found for a continuous Gaussian process in [149], as length, volume and height of an excursion. The process has zero mean and the exceedances are evaluated over the level $r$ when $r \to \infty$, where $\Delta_r$ is the length of an excursion, $m_r$ is the maximum height of excursion and $S_r$ is the area of excursion as shown in Figure 8.2.

The distribution functions for the excursion statistics over a high threshold $r$ are then given as:

$$P_1(v) = P\{r\Delta_r \geq v\} = \exp\Big\{ - \frac{\lambda_2}{8\lambda_0^2} v^2 \Big\} \tag{8.17}$$

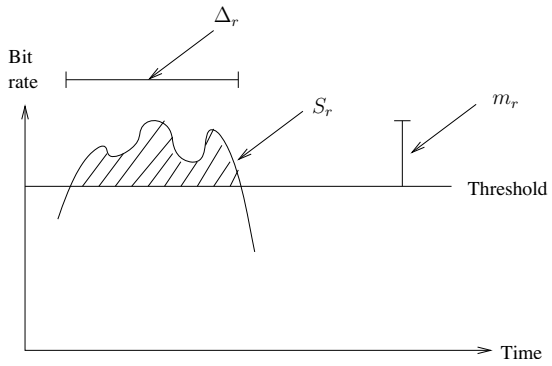



**Figure 8.2:** The characteristics of excursions for a continuous process.

$$P_2(v) = P\{rm_r \geq v\} = \exp\Big\{ - \frac{v}{\lambda_0}\Big\} \tag{8.18}$$

$$P_3(v) = P\Big\{\frac{3}{2}r^2 S_r \geq v\Big\} = \exp\Big\{ - \Big(\frac{\lambda_2}{8\lambda_0^4}\Big)^{1/3} v^{2/3}\Big\} \tag{8.19}$$

where $\lambda_k$ is given by:

$$\lambda_k = \int_0^\infty \lambda^k dF(\lambda) = \int_0^\infty \lambda^k f(\lambda) d\lambda \tag{8.20}$$

and $F(\lambda)$ and $f(\lambda)$ are the spectral distribution function and spectral density function of the autocorrelation of the process, respectively.

The spectral density for an autocorrelation function $\rho(t)$ is given as:

$$f(\lambda) = \frac{2}{\pi}\int_0^\infty \cos(\lambda t)\rho(t) dt \tag{8.21}$$

For the continuous distributions, the $i$-th order moments for the length of an excursion are defined using the complementary cdf as [13]:

$$E[r\Delta_r^i] = \int_0^\infty i \cdot v^{i-1} P_1(v) dv \tag{8.22}$$

For the distribution of the volume and maximum height of an excursion, the moments are defined in the same way:

$$E[rm_r^i] = \int_0^\infty i \cdot v^{i-1} P_2(v) dv \tag{8.23}$$

$$E\Big[\frac{3}{2}r^2 S_r^i\Big] = \int_0^\infty i \cdot v^{i-1} P_3(v) dv \tag{8.24}$$

In addition, the relationship between the distribution functions of the length and volume of an excursion is explored. It can be seen from Equation 8.20 and 8.21 that $\lambda_0$ is always equal to 1, hence:

$$P_1(v) = \exp\Big\{ - \frac{\lambda_2}{8} v^2 \Big\} \tag{8.25}$$

and

$$P_3(v) = \exp\Big\{ - \Big(\frac{\lambda_2}{8}\Big)^{1/3} v^{2/3} \Big\} \tag{8.26}$$

Taking the natural logarithm of both of them gives:

$$P_3(v) = \exp\Big\{ - \Big( - \ln P_1(v)\Big)^{1/3} \Big\} \tag{8.27}$$

and hence a very simple relation between the distribution functions of the two characteristics exists. This relation is also explored for the length and loss volume of a loss period for the discrete process in Section 9.4.

### 8.4.1 Permissible Correlation Functions

When choosing a correlation function for the analysis, two requirements must be satisfied. First, the correlation function should resemble the actual video data, and appropriate correlation functions for modeling are explored in Section 9.3.1 for the video traces described in Section 3.4. Second, conditions for the continuous process are given in [149], saying that the process should be a stationary, ergodic Gaussian random process with zero mean and a twice differentiable correlation function such that for $|t| \leq t_0$:

$$|\rho''(0) - \rho''(t)| \leq \psi(|t|), \tag{8.28}$$

where $\psi(|t|)$ is a non-decreasing and continuous function for $|t| \leq t_0$, $\psi(0) = 0$. This defines the permissible correlation functions. It is found in [149] that the function $\psi(t) = C/|\log|t||^{1+\epsilon}$ , $\epsilon > 0$ satisfies the required restriction placed on $\psi(t)$.

The inequality in Equation 8.28 is plotted in Figure 8.3 for the functions $e^{-t}$ and $e^{-t^2}$ to test if these functions are permissible. As can be seen in the figure, the condition is fulfilled for both of these functions, and they are suitable choices for the analysis based on the second requirement. However, even though this figure shows that the chosen functions are permissible, preliminary calculations showed non-converging integrals for the $\lambda$-constants for $e^{-t}$, as is shown next.

### 8.4.2 Study of Two Permissible Correlation Functions

This section starts with the calculation of the $\lambda$-constants with the correlation function $\rho(t) = e^{-t}$. This gives the following expression for the spectral density:

$$f(\lambda) = \frac{2}{\pi} \int_0^\infty \cos(\lambda t) e^{-t} dt = \frac{2}{\pi} \frac{1}{1+\lambda^2} \tag{8.29}$$

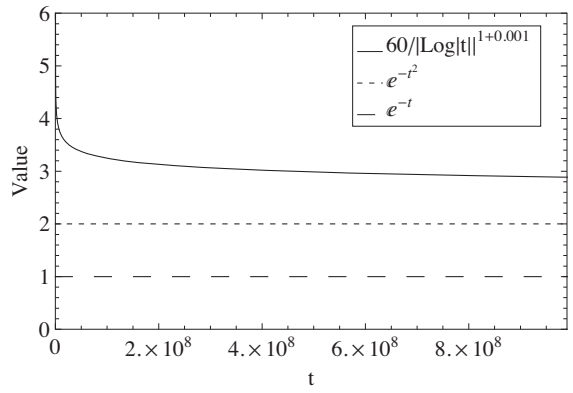



**Figure 8.3:** $|\rho''(0) - \rho''(t)|$ versus $\psi(t) = C/|\log|t||^{1+\epsilon}$ where $C = 60$ and $\epsilon = 0.001$.

and hence for the $\lambda_k$ constants:

$$\lambda_k = \frac{2}{\pi}\int_0^\infty \lambda^k \frac{2}{\pi}\frac{1}{1+\lambda^2}d\lambda \tag{8.30}$$

This integral does not converge when $k = 1, 2$. Hence, the distribution of the excursion cannot be found for this correlation function, even though it is a permissible correlation function.

Next, the calculation of the $\lambda$-constants using the correlation function $\rho(t) = e^{-t^2}$ is shown. This gives the following expression for the spectral density:

$$f(\lambda) = \frac{2}{\pi}\int_0^\infty \cos(\lambda t)e^{-t^2}dt = \frac{e^{-\frac{1}{4}\lambda^2}}{\sqrt{\pi}} \tag{8.31}$$

and hence for the $\lambda_k$ constants:

$$\lambda_k = \frac{2}{\pi}\int_0^\infty \lambda^k \frac{e^{-\frac{1}{4}\lambda^2}}{\sqrt{\pi}}d\lambda \tag{8.32}$$

The $\lambda_k$-constants will then be equal to: $\lambda_0 = 1$, $\lambda_1 = \frac{2}{\sqrt{\pi}}$ and $\lambda_2 = 2$. For these constants the moments of the excursion statistics can be evaluated. The correlation function $\rho(t) = e^{-t^2}$ is therefore employed for comparison of the characteristics of an excursion for the continuous process with the moments of the length and loss volume of a loss period for the discrete process in Section 9.4.

## 8.5 Conclusions

In this chapter, it is shown that the exceedances over a threshold for an aggregated video stream modeled as a discrete, multivariate Gaussian process can be estimated

using characteristics of exceedances for a basic stream, also modeled as a Gaussian process. This gives the required fundament to proceed with a numerical solution.

The results based on the limit distributions for the continuous process are unfortunately of limited value due to the restriction on the correlation functions and the problem of computing the $\lambda$-values even for permissible functions. Appropriate correlation functions computed from the video traces can therefore not be employed for estimation of the limit distributions. Because of these shortcomings, the numerical approach for the discrete process seems as an attractive approach to estimate the exceedances. These exceedances constitute loss periods and the numerical results for the distribution of the loss period are given in Chapter 9, together with results from simulations.

# Chapter 9

# Characteristics of Loss Periods for Aggregated Video Traffic

In this chapter, slice-based encoded video traffic is modeled using the Gaussian model as described in Chapter 8. The correlation functions are modeled using the video traces described in Chapter 3 and the exceedance probabilities of frame sizes over a threshold are found numerically. These exceedances constitute loss periods, and the distributions of the length and loss volume of a loss period are found. The loss volume gives the packet loss in the bufferless case and is also used for estimating the packet loss in a bottleneck node with small buffers.

## 9.1 Introduction

Estimation of the packet loss for a video transmission over a communication network is an important step for assessing the perceived QoS of the transmission. The loss probability is particularly important, but also the characteristics of a loss period have a profound effect, since bursty losses influence the perceived QoS in a different way than uniform distributed losses, as discussed in Section 2.2. The distribution of the loss periods, and particularly the first and second moments, is therefore of great interest and can be used to deduce the perceived QoS for a video transmission. Knowledge about the probability of subsequent packet losses in a packet stream is also important for doing video encoding, as this knowledge can be exploited to choose the best encoding parameters.

The multivariate normal integral is used for evaluating the exceedances of frame sizes over a threshold, and relations between the exceedances for the basic process and the aggregate is found in Chapter 8. The multivariate normal integral problem is solved numerically, since an analytical solution is impossible. The numerical solution is acceptable if it is accurate enough and does not take to much time.

The numerical results for the exceedance probabilities are used for estimating

the length of a loss period as well as the loss volume of a loss period for the discrete process. For a bufferless node, the total packet loss will be equal to the loss volume. In addition, an approach to packet loss estimation for a bottleneck node with a small buffer is developed, using the distribution of the loss volume for the discrete process. In addition, the correspondence between loss period characteristics for discrete processes and excursion characteristics for continuous processes is investigated. Furthermore, the relation found between the length and volume of excursions over high thresholds for continuous processes in Section 8.4 is employed for the discrete process as well.

### 9.1.1 Related Work

As described in Section 2.2, the perceived QoS for a user watching a video transmitted over a network is affected by a number of factors, including the loss ratio and the distribution of the losses. This motivates the focus on the distribution of a loss period in this work. The effect of the burst loss on the distortion is studied e.g., in [65]. A model is proposed to estimate the distortion from different types of artifacts, and the results are compared to simulations. The results show that the burst length of the loss process is important for estimating the distortion, and that loss occurring in bursts affect the distortion more than single losses of the same amount.

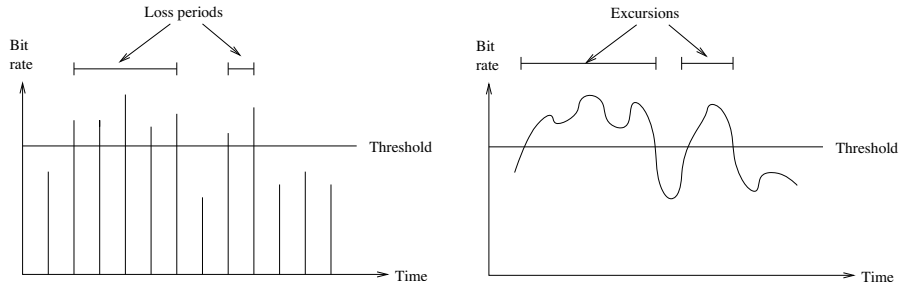
Characterization of the loss period can be viewed as a level-crossing problem. Level-crossing is a very difficult problem and there are few general results. This problem has been studied for Gaussian processes in [151] and experimental problems are investigated in [152]. This theory can be used to predict crossings over a threshold for several physical processes such as flood level. For traffic analysis purposes, the level-crossing problem in discrete time can be used for predicting packet loss over a bufferless communication link, where the level corresponds to a given threshold.

In [153], relationships between discrete and continuous Gaussian processes are investigated. In particular, results are given in the case where the discrete process is first order Markovian (AR(1)), and the continuous process has a sampled autocorrelation function which is exponential, in order to match the discrete process.

In this work, the correspondence between the excursions for the continuous Gaussian process described in Chapter 8 and the moments of the length and loss volume of a loss period for the discrete Gaussian process is investigated.

### 9.1.2 Chapter Outline

The rest of this chapter is organized as follows. The distribution of the length and loss volume of a loss period for the aggregated traffic is found in Section 9.2. The multivariate normal integral is calculated numerically for the single stream in Section 9.3 and the moments of the length and loss volume of a loss period for the aggregated traffic are deduced. Section 9.4 compares the first moments of a loss




(a) Loss periods, discrete process (b) Excursions, continuous process

**Figure 9.1:** Characteristics of exceedances.

periods and of an excursion for a permissible correlation function. Little's formula is employed for estimating the first moments of the length of an excursions and a loss period in Section 9.5, validating the numerical results. The calculation of the loss in a model with a small buffer, using the distribution of the loss volume in a loss period, is then given in Section 9.6. Finally, some conclusions are given in Section 9.7.

## 9.2 Characteristics of Loss

The distribution of the loss is studied, since the perceived QoS depends on the packet loss distribution in addition to the loss ratio. Discrete and continuous multivariate Gaussian processes as described in Chapter 8 are used, having the same mean and covariance. The term excursion is used for exceedances over a threshold for the continuous process, while the term loss period is used for the exceedances in the discrete case. An illustration of loss periods for a discrete process and excursions for a continuous process is given in Figure 9.1, to clarify the terminology.

As can be seen in the figure, a loss period equals a number of consecutive frames with frame sizes larger than the threshold. There is a difference between the discrete and the continuous process since the loss period of the former has a minimum length equal to 1, while there is no minimum for the continuous process. The loss period defined here is also different from the loss period defined in Chapter 7, since the latter only require one exceedance in a cluster.

The approach to estimate the moments of the length and loss volume of a loss period for the discrete process is given next.

### 9.2.1 Length of a Loss Period

An expression for the complementary cumulative distribution function (cdf) for the number of consecutive losses, $Pr[M_{\boldsymbol{X},a} > m]$ is found, where $M_{\boldsymbol{X},a}$ denotes the number of consecutive exceedances for the process $\boldsymbol{X}$ over the threshold $a$.

From the complementary cdf, known solution methods can be used to find the moments of the length of a loss period.

The complementary cdf for the number of consecutive losses, conditioning on a loss event, is then given by:

$$Pr[M_{\boldsymbol{X},a} > m] = Pr[X_{m+1} > a, X_m > a, .., X_2 > a | X_1 > a, X_0 < a] \tag{9.1}$$

Using the law of conditional probability on this last equation gives:

$$\begin{aligned} &Pr[X_{m+1} > a, X_m > a, .., X_2 > a | X_1 > a, X_0 < a] \\ &\qquad = \frac{Pr[X_{m+1} > a, X_m > a, .., X_1 > a, X_0 < a]}{Pr[X_1 > a, X_0 < a]} \end{aligned} \tag{9.2}$$

Applying the law of total probability on the numerator and denominator gives the following expression for the complementary cdf for the length of a loss period:

$$Pr[M_{\boldsymbol{X},a} > m] = \frac{Pr[X_1 > a, .., X_{m+1} > a] - Pr[X_0 > a, .., X_{m+1} > a]}{Pr[X_1 > a] - Pr[X_0 > a, X_1 > a]} \tag{9.3}$$

$M_{\boldsymbol{Y},n,b}$ is then the number of consecutive exceedances over the threshold $b$ for the process $\boldsymbol{Y}$, where the aggregate consists of $n$ basic processes. The complementary cdf for the number of consecutive exceedances for the aggregate is then given by:

$$\begin{aligned} Pr[M_{\boldsymbol{Y},n,b} > m] &= \frac{Pr[Y_1 > b, .., Y_{m+1} > b] - Pr[Y_0 > b, .., Y_{m+1} > b]}{Pr[Y_1 > b] - Pr[Y_0 > b, Y_1 > b]} \\ &= Pr[M_{\boldsymbol{X},\hat{b}} > m] = Pr[M_{\boldsymbol{X}^*,\hat{b}-\mu_X} > m] \end{aligned} \tag{9.4}$$

The above equation establishes the relations between the distribution of the length of a loss period for the aggregated process $\boldsymbol{Y}$ and the single processes $\boldsymbol{X}$ and $\boldsymbol{X}^*$. It follows that:

$$E[M^i_{\boldsymbol{Y},n,b}] = E[M^i_{\boldsymbol{X},\hat{b}}] = E[M^i_{\boldsymbol{X}^*,\hat{b}-\mu_X}] \tag{9.5}$$

where $i$-th order moments for a loss period for the aggregate process (and hence the single processes) can be calculated as:

$$E[M^i_{\boldsymbol{Y},n,b}] = \sum_m m^i \cdot Pr[M_{\boldsymbol{Y},n,b} = m] \tag{9.6}$$

The distribution of the length of a loss period for an aggregate of $n$ video streams can then be found from the basic processes, defined by their means and covariance functions. The multivariate normal integral has to be calculated numerically or by using simulation. Simulation is demanding and cumbersome, and will also inevitably suffer some inaccuracies. Simulation is hence only an alternative for cases when computation of the multivariate normal integral fails. The numerical calculation is investigated in Section 9.3 together with results from simulation.

In an aggregate, the constellation of scenes will change, governed by scene changes in each stream. It was found that scene lengths are close to a Geometric distribution in Chapter 4. This is in agreement with the negative exponential distribution (ned) in the continuous case, found to model the scene length in [96]. The length of a scene constellation in an aggregate will then also be ned. A given constellation can then be analyzed separately and a representative mix of constellations weighted together afterwards. The occurrence of the various scene constellations will determine the resulting loss.

### 9.2.2 Loss Volume in a Loss Period

The moments of the loss volume in a loss period, or loss volume in short, can be employed for estimating the packet loss in a bufferless network node, as well as the loss in a node with a small buffer. To the best of our knowledge, no exact results exist for the loss volume for a discrete, multivariate normal distribution. In this section, an approximate numerical approach for estimating the loss volume for the discrete process is developed.

To find the distribution of the loss volume, it needs to be conditioned on that the frame immediately before and after the loss period are both below the threshold. Conditioning on the process being below a value is much more cumbersome than conditioning on a particular value. Hence, several cases are investigated in the following. The notation $f_{\boldsymbol{X}}(\boldsymbol{x}) = Pr[\boldsymbol{X} = \boldsymbol{x}]$ is used as a shorthand for $f_{\boldsymbol{X}}(\boldsymbol{x}) = Pr[\boldsymbol{x} < \boldsymbol{X} \leq \boldsymbol{x} + d\boldsymbol{x}]$, and the following moments are of interest:

$$E[(X_1 - a + .. + X_m - a)^k | X_0 = x_0, X_1, ..X_m > a, X_{m+1} = x_{m+1}] \tag{9.7}$$

i.e., the moments of the loss volume.

The distribution:

$$Pr[X_1 = x_1, .., X_m = x_m | X_0 = x_0, X_{m+1} = x_{m+1}] \tag{9.8}$$

can easily be found using known results from the conditional multivariate distribution [154]. Including the final condition, $X_1, .., X_m > a$, is then a matter of normalization, namely:

$$\begin{aligned} &f_{\boldsymbol{X}}(x_1, .., x_m | X_0 = x_0, X_1, ..X_m > a, X_{m+1} = x_{m+1}) \\ &\qquad = \frac{f_{\boldsymbol{X}}(x_1, .., x_m | X_0 = x_0, X_{m+1} = x_{m+1})}{Pr[X_1, .., X_m > a | X_0 = x_0, X_{m+1} = x_{m+1})} \end{aligned} \tag{9.9}$$

The conditional distribution in Equation 9.8 is then rewritten as:

$$f_{\boldsymbol{C}}(\boldsymbol{C}_1 | \boldsymbol{C}_2 = \boldsymbol{c}) \tag{9.10}$$

where

$$\boldsymbol{C}_1 = \begin{bmatrix} X_1 \\ . \\ . \\ X_m \end{bmatrix}, \ \boldsymbol{C}_2 = \begin{bmatrix} X_0 \\ X_{m+1} \end{bmatrix} \tag{9.11}$$

and

$$\boldsymbol{c} = \begin{bmatrix} x_0 \\ x_{m+1} \end{bmatrix} \tag{9.12}$$

Results for conditional multivariate normal distributions given e.g., in [154] are then employed in the following.

For our process, the mean vectors have elements:

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \mu_{X_1} \\ \mu_{X_2} \\ \cdot \\ \cdot \\ \mu_{X_m} \end{bmatrix} \text{ and } \boldsymbol{\mu}_2 = \begin{bmatrix} \mu_{X_0} \\ \mu_{X_{m+1}} \end{bmatrix} \tag{9.13}$$

and the covariance matrix has elements:

$$\Sigma_{11} = E[\boldsymbol{C}_1 \boldsymbol{C}_1^T], \tag{9.14}$$

$$\Sigma_{12} = \begin{bmatrix} E[\boldsymbol{C}_1 X_0] & E[\boldsymbol{C}_1 X_{m+1}] \end{bmatrix}, \tag{9.15}$$

$$\Sigma_{21} = \begin{bmatrix} E[X_0 \boldsymbol{C}_1^T] \\ E[X_{m+1} \boldsymbol{C}_1^T] \end{bmatrix}, \tag{9.16}$$

and

$$\Sigma_{22} = \begin{bmatrix} E[X_0 X_0] & E[X_0 X_{m+1}] \\ E[X_0 X_{m+1}] & E[X_{m+1} X_{m+1}] \end{bmatrix} \tag{9.17}$$

Then the distribution of $\boldsymbol{C_1}$ conditioned on $\boldsymbol{C_2} = \boldsymbol{c}$ is multivariate normal $(\boldsymbol{C}_1 | \boldsymbol{C}_2 = \boldsymbol{c}) \sim N(\boldsymbol{\mu}, \Sigma)$ with mean vector

$$\boldsymbol{\mu} = \boldsymbol{\mu}_1 + \Sigma_{12} \Sigma_{22}^{-1} (\boldsymbol{c} - \boldsymbol{\mu}_2) \tag{9.18}$$

and covariance matrix

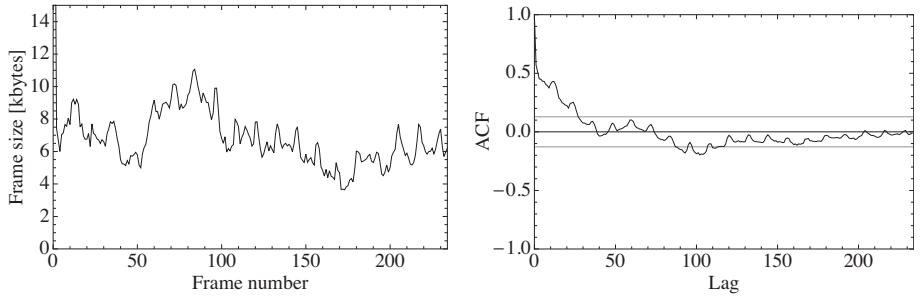
$$\Sigma = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}. \tag{9.19}$$

The final step is then to estimate this distribution and estimate the expectation function in Equation 9.7 numerically. This gives the moments of the sum of exceedances conditioned on a given length of the loss period. To find the overall moments, the expectations are unconditioned using the probability density function of the length of the loss period as found in Section 9.2.1.

The $k$-th moment for the loss volume is then found as:

$$E(S^k) = \sum_{i=1}^{m} E\left[ \Big( \sum_{j=1}^{i} X_j - a \Big)^k \right] \cdot Pr[M_{\boldsymbol{X},a} = i] \tag{9.20}$$

where $S$ is the loss volume of the loss period.




(a) Frame sizes (b) The ACF

**Figure 9.2:** One scene from the slice-based encoded StEM clip.

## 9.3 Numerical Computations and Results from Simulations

Exact calculation of the multivariate normal integral is generally not feasible [151]. However, a numerical method with acceptable accuracy is developed in [150], using the Cholesky transformation. This method for evaluating the integral is implemented in the package *mvtnorm* [155], contained in the program R [156]. Also, the generation of multivariate normal variates with a given mean vector and covariance matrix is included. The loss period characteristics over a given threshold can then easily be estimated from these variates using simulations.

The expectation function needed for evaluating the volume of a loss period can not be found using the *mvtnorm* package in R. However, a function *qsimvnef* in Matlab is available from [157] for estimating the MVN expectation for an arbitrary expectation function, also employing the algorithm from [150].

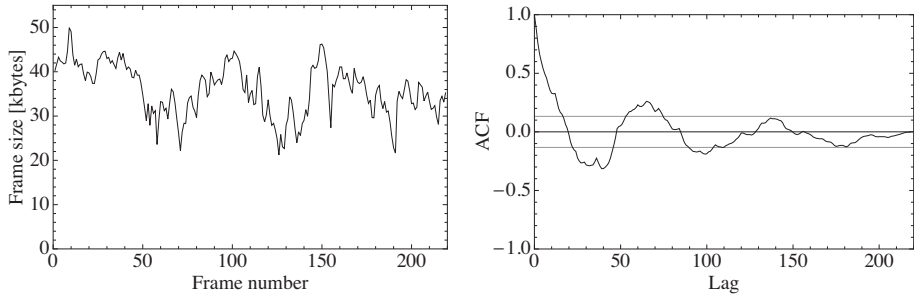
The numerical computation time for the moments of the length and loss volume of a loss period is in the order of seconds for each threshold, using a regular computer.

Next, the ACFs from real video traces are modeled, followed by the numerical and simulation results for the distributions of the loss period characteristics.

### 9.3.1 Modeling the ACF using Video Traces

Real video traces encoded using the slice-based video encoding scheme as described in Section 3.3 are used as a basis for choosing the ACFs used in the model. First, one scene of the StEM clip is employed. This clip has high variability in the average bitrate over scenes, but within each scene the bitrate is more constant. The number of bytes per frame for one scene with approximately 230 frames, as well as the ACF are shown in Figure 9.2. The ACF is high for lags $< 50$, but then decreases and stays almost within the confidence interval $1.96/\sqrt{n}$.

Second, the Mobile clip with 220 frames and a relatively stable bitrate is used.




(a) Frame sizes (b) The ACF

**Figure 9.3:** The slice-based encoded Mobile clip.

The number of bytes per frame is shown in Figure 9.3 together with the ACF. The ACF for the Mobile clip shows some oscillating behavior, and is also negligible for high lags.

The ACFs for the two video clips, when generalized to either a slowly decreasing or an oscillating function, are believed to be representative for a large number of video clips (and also audio clips). In the following they form the basis for two different types of models used for the ACF.

The covariance matrix $r$ for $\boldsymbol{X}$ needs to be a positive definite matrix. A correlation function which satisfies this requirement is [158]:

$$\rho(h) = e^{-kh^{\nu}}, \qquad 0 < \nu \leq 2 \tag{9.21}$$

For $\nu = 1$ the function becomes the ordinary negative exponential function; for $0 < \nu < 1$ it has a longer tail. A sum of two positive definite functions is also positive definite [158]. A permissible function is therefore:

$$\rho(h) = a_1 e^{-h/x_1} + a_2 e^{-h/x_2}, \qquad x_2 >> x_1 \tag{9.22}$$

where $a_1 + a_2 = 1$. This combined function can model a longer tail and is used in this study.

For the oscillating ACF the exponentially damped cosine correlation function is used:

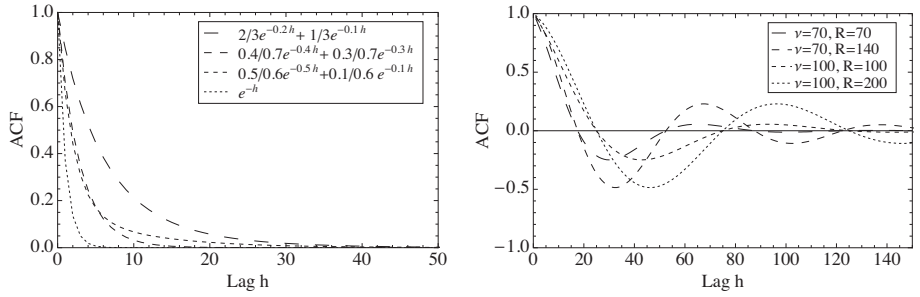
$$\rho_{cos}(h; \omega, R) = e^{-3h/R} \cos \omega h \tag{9.23}$$

where the period $\nu = 2\pi/\omega$. This function is also positive definite [158].

In Figure 9.4, the different correlation functions employed are shown. It is clear that using the exponential function with parameter 1.0 gives only short term correlation, while the combined functions and the oscillative decaying functions have higher correlation values at higher lags, resembling the ACF of the traces.

### 9.3.2 Exceedances over a Threshold

The exceedances over various thresholds are evaluated, where the thresholds are expressed as fractions of the standard deviation $\sigma_{\boldsymbol{X}}$ of $\boldsymbol{X}$ ($\boldsymbol{X}^*$). This means that




(a) Exponentially decaying (b) Oscillative decaying

**Figure 9.4:** The correlation functions employed.

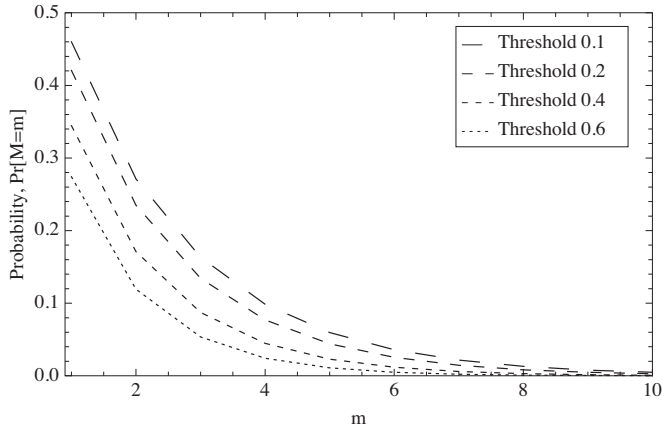



**Figure 9.5:** The probability of $m$ consecutive exceedances for variable thresholds.

only the correlation matrix is needed as input to the model.

As a first step, the probability of consecutive exceedances over the threshold for the basic, zero-mean multivariate normal process $X^*$ is evaluated. The intermediate results are shown for the exponential correlation function with parameter 1.0. Later, the first and second moments of the length of a loss period are compared for all correlation matrices.

The probabilities of consecutive losses, $Pr[M = m]$ found from the multivariate normal integral are shown in Figure 9.5. The thresholds which correspond to the lower integration limits in the MVNI are equal to 0.1, 0.2, 0.4 and 0.6 times the standard deviation. The probability of $m = 1$ consecutive exceedances gives the overall loss probability.

The thresholds $b^*$ for the zero-mean aggregate consisting of $n$ basic processes will now be found using the relation $b^* = a^* \cdot \sqrt{n}$ where $a^*$ is the threshold for the basic zero-mean process. For the aggregated stream with $n = \{2, 5, 10, 20, 50\}$, the

**Table 9.1:** The threshold $b^*$ for the zero-mean aggregate.

| $n$ | $0.1 \cdot \sigma_{\boldsymbol{X}}$ | $0.2 \cdot \sigma_{\boldsymbol{X}}$ | $0.4 \cdot \sigma_{\boldsymbol{X}}$ | $0.6 \cdot \sigma_{\boldsymbol{X}}$ |
|---|---|---|---|---|
| 2 | 0.0705 $\sigma_{\boldsymbol{Y}}$ | 0.1425 $\sigma_{\boldsymbol{Y}}$ | 0.2830 $\sigma_{\boldsymbol{Y}}$ | 0.4245 $\sigma_{\boldsymbol{Y}}$ |
| 5 | 0.0446 $\sigma_{\boldsymbol{Y}}$ | 0.0894 $\sigma_{\boldsymbol{Y}}$ | 0.1788 $\sigma_{\boldsymbol{Y}}$ | 0.2684 $\sigma_{\boldsymbol{Y}}$ |
| 10 | 0.0316 $\sigma_{\boldsymbol{Y}}$ | 0.0632 $\sigma_{\boldsymbol{Y}}$ | 0.1265 $\sigma_{\boldsymbol{Y}}$ | 0.1897 $\sigma_{\boldsymbol{Y}}$ |
| 20 | 0.0224 $\sigma_{\boldsymbol{Y}}$ | 0.0447 $\sigma_{\boldsymbol{Y}}$ | 0.0895 $\sigma_{\boldsymbol{Y}}$ | 0.1342 $\sigma_{\boldsymbol{Y}}$ |
| 50 | 0.0141 $\sigma_{\boldsymbol{Y}}$ | 0.0283 $\sigma_{\boldsymbol{Y}}$ | 0.0566 $\sigma_{\boldsymbol{Y}}$ | 0.0849 $\sigma_{\boldsymbol{Y}}$ |

**Table 9.2:** Complementary cdf for the length of a loss period.

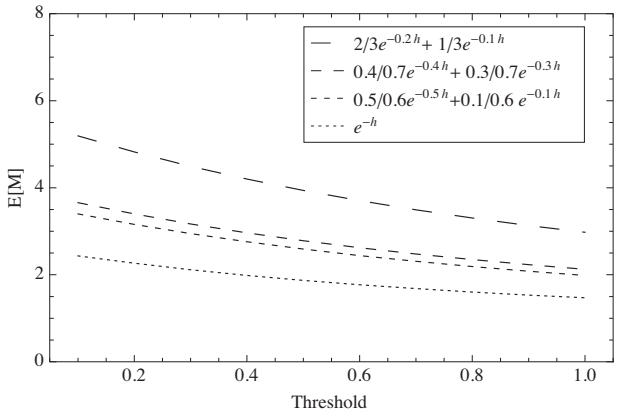
| $m$ | $0.1 \cdot \sigma_{\boldsymbol{X}}$ | $0.2 \cdot \sigma_{\boldsymbol{X}}$ | $0.4 \cdot \sigma_{\boldsymbol{X}}$ | $0.6 \cdot \sigma_{\boldsymbol{X}}$ |
|---|---|---|---|---|
| 0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 1 | 0.572 | 0.542 | 0.483 | 0.425 |
| 2 | 0.342 | 0.308 | 0.245 | 0.190 |
| 3 | 0.206 | 0.177 | 0.126 | 0.0856 |
| 4 | 0.125 | 0.101 | 0.0643 | 0.0387 |
| 5 | 0.0753 | 0.0582 | 0.0329 | 0.0175 |
| 6 | 0.0455 | 0.0333 | 0.0169 | 0.00789 |
| 7 | 0.0275 | 0.0191 | 0.00866 | 0.00356 |
| 8 | 0.0166 | 0.0110 | 0.00444 | 0.00161 |

results are shown in Table 9.1. This means that the probability of exceeding the threshold $0.1 \cdot \sigma_{\boldsymbol{X}}$ for the basic process is equal to the probability of exceeding the threshold $0.0141 \cdot \sigma_{\boldsymbol{Y}}$ (where $\sigma_{\boldsymbol{Y}} = n \cdot \sigma_{\boldsymbol{X}}$) for the aggregate consisting of $n = 50$ basic processes.

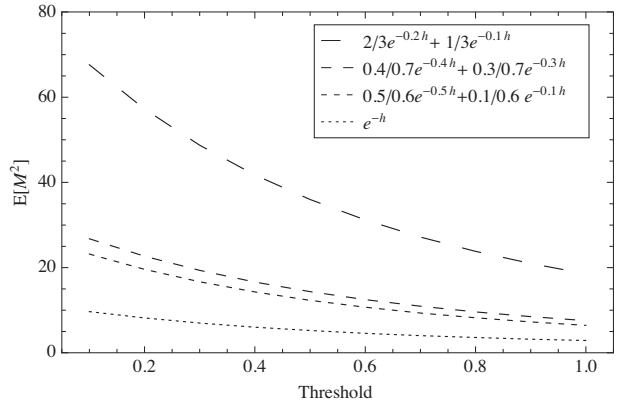
### 9.3.3 Length of a Loss Period

For estimating the length of a loss period, the next step is to calculate the complementary cdf $Pr[M_{\boldsymbol{X}^*,a^*} > m]$ and $Pr[M_{\boldsymbol{Y}^*,n,b^*} > m]$ for the basic and aggregated processes respectively. Equation 9.3 is used, and the same thresholds as before are employed for the basic and aggregated processes. It is conditioned on the occurrence of a loss, therefore the probability of more than zero losses is equal to 1 for all thresholds. The results from these calculations are shown in Table 9.2. The complementary cdf for the length of a loss period is then the same for the aggregate, where the corresponding threshold is found in Table 9.1.

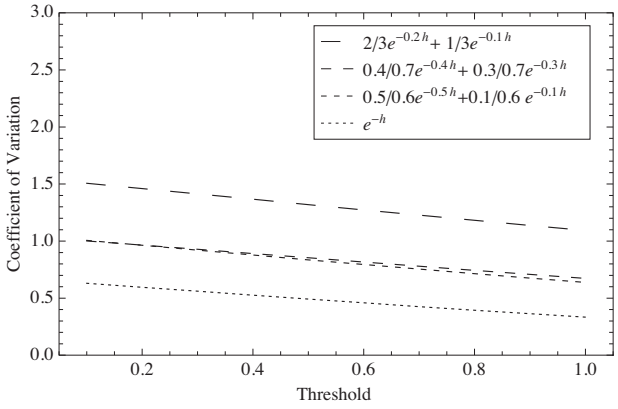
The moments of the length of a loss period is calculated using Equation 9.6. First, the results for the exponential correlation functions are shown. The first moments are shown in Figure 9.6(a), with different covariance matrices and for variable thresholds. The moments for the basic process with the threshold given in the figure correspond to the moments for the aggregated process of $n$ basic processes where the threshold for the aggregated process for each $n$ is given in Table 9.1. In Figure 9.6(b), the second moment for the number of consecutive losses is shown for the same thresholds and covariance matrices as above, while in Figure




(a) First moment

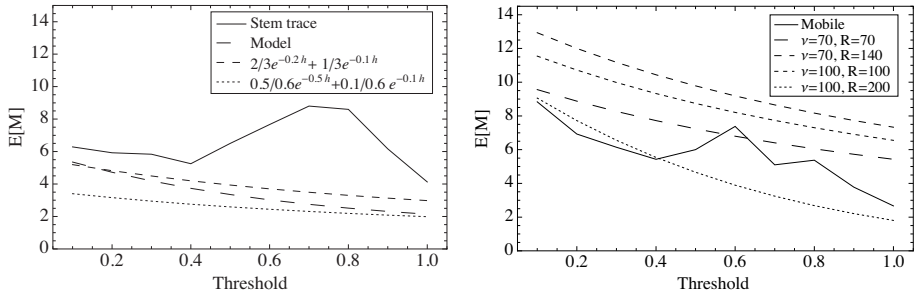



(b) Second moment




(c) The coefficient of variation

**Figure 9.6:** Characteristics of the length of a loss period.




(a) StEM clip (b) Mobile clip

**Figure 9.7:** First moment of the length of a loss period found from the numerical calculation together with the average length of a loss period from the traces.

9.6(c), the coefficient of variation is shown. A value of the coefficient of variation around 1 indicates a Geometric distribution, values lower than 1 and higher than 1 indicate Hypogeometric and Hypergeometric distributions respectively.

As expected, there are large differences in the moments of the length of a loss period for the different covariance functions. This indicates that while the loss periods are relatively short and frequent when the parameter of the exponential function is equal to 1.0, the loss periods become longer and less frequent when the correlation is given by a combination of exponential functions with parameters 0.1 and 0.2.

In Figure 9.7(a), the first moment of the length of a loss period for the StEM clip, using the real correlation values of the clip in the model, is compared to the sample average and sample second moment from direct inspection of the trace. Also, the first moment for two different ACFs are shown. The difference between the first moment from the the trace and from the model is partly due to a single long loss period for the trace. For the oscillative decreasing correlation functions, the first moments are shown in Figure 9.7(b). In addition to the moments calculated from the model, also the sample moments of the length of a loss period from the Mobile trace are shown, found from direct inspection of the trace. As can be seen, the sample average and sample second moment are less smooth than the moments from the model. This is due to the low number of frames in the clip and hence only a few loss periods contribute to the calculation of the sample moments for high thresholds. More and larger clips are hence needed to validate the model.

Finally, the length of a loss period is found using simulation. 20 different samples are generated, with 3000 variates in each. The first moment including Student's $t$-confidence intervals for the correlation function $e^{-h^2}$, is shown in Figure 9.8 together with the results from the numerical computation. As can be seen in the figure, the results overlap completely for the numerical computation and simulation.

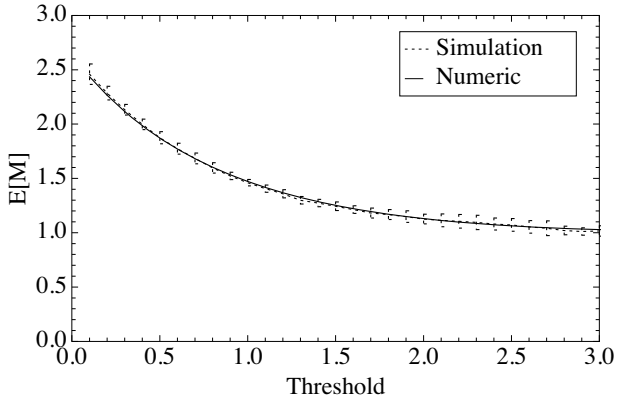



**Figure 9.8:** First moment of the length of a loss period from numerical calculation and simulation with correlation function $e^{-h^2}$.

### 9.3.4 Loss Volume of a Loss Period

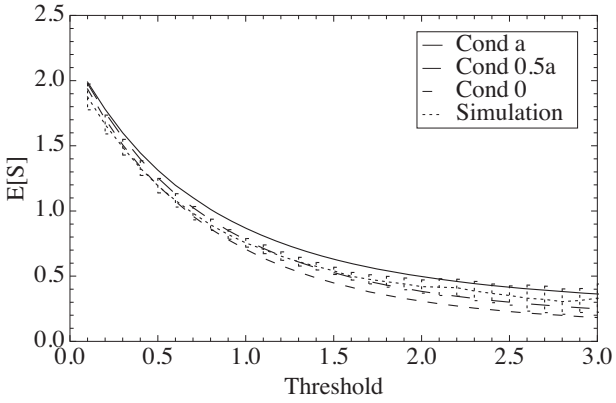
The first moment of the loss volume from numerical computation is shown in Figure 9.9(a), together with the results from simulation. The correlation function $\rho(h) = e^{-h^2}$ is used, in order to compare with the results from the continuous process. Three different sets of results are shown, varying the condition on the frames before and after the loss period, with $x_0, x_{m+1} = a$ as the worst case and $x_0, x_{m+1} = 0$ as the best case in terms of a smaller first moment. $x_0, x_{m+1} = 0.5a$ gives the best correspondence to the simulation results. It is clear that decreasing the condition values influences the expectation. Conditioning on $x_0, x_{m+1} = a$ gives too high moments for the loss volume as compared to simulations, while the lowest condition value gives a smaller loss volume than the loss volume from simulations for high thresholds.

The second moment of the loss volume from the numerical computation is shown in Figure 9.9(b), together with the results from simulations, including confidence intervals. Similar discrepancies as for the first moment are observed.

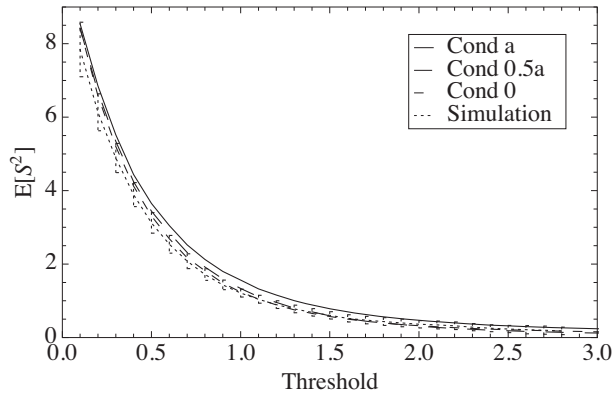
Finally, to overcome the discrepancies due to conditioning on explicit values of $X_0$ and $X_{m+1}$, conditioning on the expected values of $X_0$ and $X_{m+1}$ given that they are below the threshold is done.

$$\begin{aligned} x_0 &= E[X_0 | X_0 < a, X_1 > a] \\ x_{m+1} &= E[X_{m+1} | X_m > a, X_{m+1} < a] \end{aligned} \tag{9.24}$$

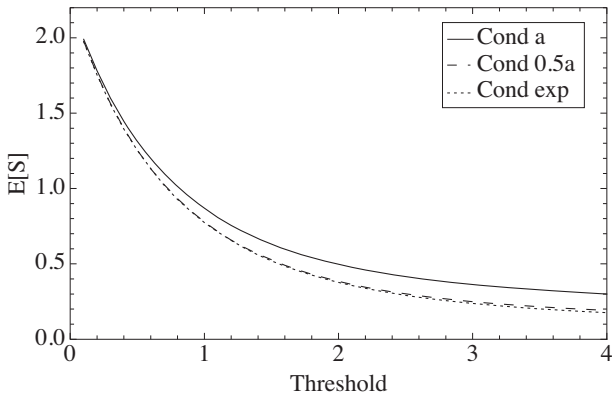
The results are given in Figure 9.9(c), and show that conditioning on the expected value gives results very close to conditioning on $x_0, x_{m+1} = 0.5a$, where the latter is a simpler approach.




(a) First moment of loss volume

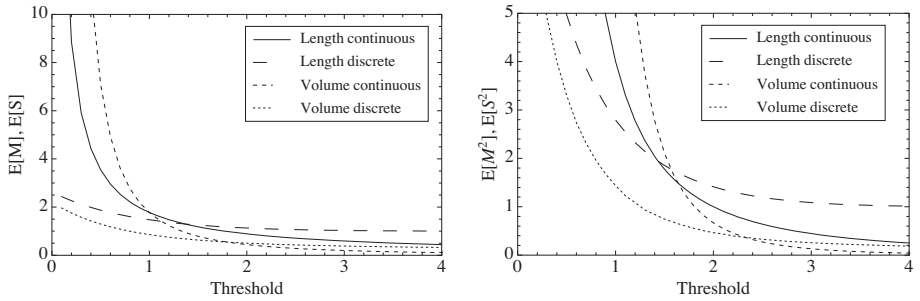



(b) Second moment of loss volume




(c) First moment of loss volume

**Figure 9.9:** Moments of loss volume in a loss period found from numerical computation and simulation.




(a) First moment (b) Second moment

**Figure 9.10:** Moments of the length and loss volume of a loss period, continuous process and discrete process.

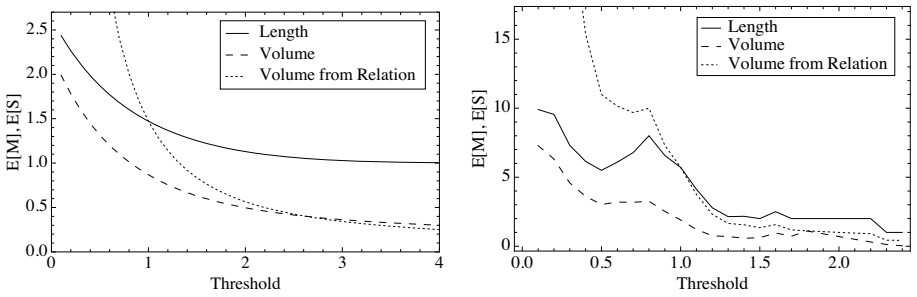
## 9.4 Comparison of Loss Periods and Excursions

In order to check the agreement between characteristics of excursions for the continuous process and the discrete process, the quantiles for loss within a certain percent are investigated. The focus is on 5, 3 and 1% loss, and these quantiles are found for the standard normal distribution $N(0,1)$ as 1.645 for 5% loss, 1.881 for 3% loss, and 2.326 for 1% loss.

Acceptable loss is typically below 3%, or even lower for real-time video as described in Section 2.2 meaning that thresholds above 1.8 are of highest interest. Whether the correspondence between the moments for the continuous process and the discrete process over these thresholds is satisfactory or not should then be investigated.

The limit distributions, $P_1(v)$, $P_2(v)$ and $P_3(v)$ for the correlation function $\rho(t) = e^{-t^2}$ can now be estimated for different thresholds. A comparison between the first moments of the length and volume of the excursions from the continuous model and likewise for the discrete process using numerical evaluation of the multivariate normal integral is shown in Figure 9.10(a), while the second moments are shown in Figure 9.10(b).

The results from [149] are valid only when the threshold $r \to \infty$, which explain the discrepancies for low thresholds in Figure 9.10(a) and 9.10(b). For thresholds higher than approximately 1.5, the results for the discrete and continuous processes are comparable, covering the interesting range of the thresholds ($> 1.8$) for which the loss probability is acceptable. Next, the first moment of the continuous process goes towards zero for high thresholds, while for the discrete process it is conditioned on a loss event and the first moment of the length is always higher than 1.




(a) Model (b) Mobile trace

**Figure 9.11:** First moment of length and loss volume of a loss period, together with the loss volume from the relation.

### 9.4.1 Using a Relation Between Length and Volume

For the continuous Gaussian process, a relation between the length and volume of an excursion over high thresholds exists, as shown in Equation 8.27. In this section, it is investigated if this relation is valid at high thresholds for a discrete process as well.

As a first step, the moments of the length of a loss period are found for the discrete model using the MVNI integral. The first moment of the loss volume is then calculated from the length using Equation 8.27, and the results are shown in Figure 9.11(a), together with the first moment of the loss volume found from the numerical method, conditioning on $x_0, x_{m+1} = a$.

As can be seen in the figure, the relation between the length and volume of the excursion for the continuous process is a good approximation also in the discrete case. Estimating the loss volume from the relation gives almost identical first moment as for the numerical estimation at high thresholds, using the condition $x_0, x_{m+1} = a$. This means that the first moment of the loss volume estimated from the relation is a bit higher than the loss volume from simulations.

Next, the first moments of the length and loss volume of a loss period from the Mobile trace are shown in Figure 9.11(b), together with the loss volume calculated using the relation between the length and the volume. For high thresholds, the relation gives a close approximation, as was also seen for the discrete process.

To investigate the usefulness of the relation further, the differences between the true loss volume and the loss volume from the relation for each threshold are evaluated for the Mobile clip. If the differences are independent, it is an indication that the relation can be employed. In Figure 9.12, the ACF of the differences between the first moment of the loss volume from the relation and the first moment of the loss volume from the trace is shown. The ACF is negligible for lags $> 1$, which is a first indication that the differences are independent. In addition, the Ljung-Box test is calculated for the differences samples. This test, as described in [101], evaluates the sum of the ACF up to a given lag. If this sum is too large, the samples are probably not independent. In our case, the Ljung-Box test gives

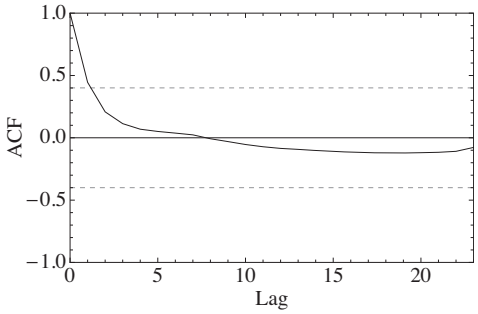



**Figure 9.12:** ACF of the differences from comparing the first moment of the loss volume from the Mobile trace with the loss volume from the relation.

a clear indication of independent differences. Together, these two tests indicate that the relation for the continuous process is valid for estimating the loss volume from the length of a loss period for the discrete process.

## 9.5 Expected Length of a Loss Period and an Excursion Using Little

As an alternative to finding the first moment of the length of a loss period and an excursion, Little's formula [13] can be employed to estimate the expected length, when the intensity into the loss area is known.

Little's queueing formula gives a relation between the average number in a system state, the arrival intensity into the state and the average sojourn time in the state.

For the discrete process, the intensity into the loss area, $\Lambda_D$, is known as the probability of having a frame larger than the threshold when the previous frame was below the threshold. This is given as:

$$\Lambda_D = Pr[X_1 > a, X_0 \leq a] = Pr[X_1 > a] - Pr[X_0 > a, X_1 > a] \tag{9.25}$$

The expected length of a loss period ($\overline{W}_D$) is then found using the intensity into the loss area ($\Lambda_D$) and the expected number in the loss state ($\overline{L}_D$), where the latter is merely the probability of being in the loss state, given as $Pr[X_0 > a]$.

$$\overline{W}_D = \frac{\overline{L}_D}{\Lambda_D} = \frac{Pr[X_0 > a]}{Pr[X_1 > a] - Pr[X_0 > a, X_1 > a]} \tag{9.26}$$

For a continuous process, the intensity into the loss area, $\Lambda_C$, is given in [159] and [153]:

$$\Lambda_C = \frac{1}{2\pi} \cdot \sqrt{\frac{-\rho''(0)}{\rho(0)}} \cdot e^{(-\frac{a^2}{2\rho(0)})} = \frac{1}{2\pi} \cdot \sqrt{-\rho''(0)} \cdot e^{(-\frac{a^2}{2})} \tag{9.27}$$

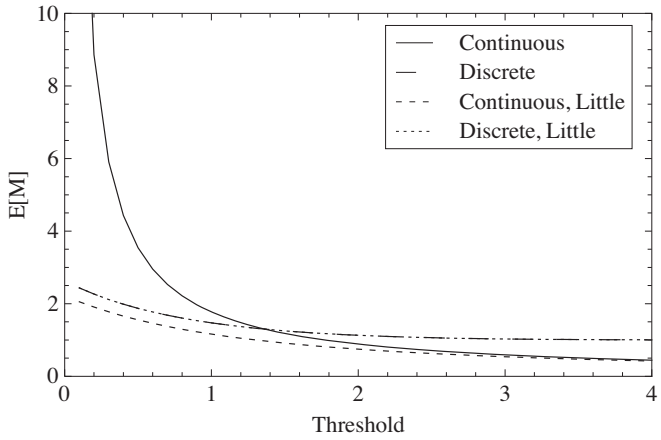



**Figure 9.13:** First moment of the length of a loss period and length of an excursion, together with the results found using Little's formula.

for the threshold $a$. Using Little again then gives:

$$\overline{W}_C = \frac{\overline{L}_C}{\Lambda_C} \tag{9.28}$$

where the probability of being in the loss state is taken from the standard normal distribution for the given threshold $a$.

The results from using Little on the discrete and continuous processes with correlation function $\rho(h) = e^{-h^2}$ and $\rho(t) = e^{-t^2}$, respectively, are shown in Figure 9.13 together with the previous results for the discrete and continuous processes. As can be seen, the results from the numerical evaluation for the discrete process are identical to those using Little, validating the numerical approach. For the continuous process, the difference between the moment found from Little and found using the limit distribution from [149] is large for low thresholds, but for thresholds higher than 2.0 they coincide. This is as expected since the limit distribution is valid only for high thresholds.

## 9.6 The Approximate Loss with a Small Buffer

In a bufferless system, the bit loss is estimated by the distribution of the loss volume in a loss period. Next, when the buffer size in a node is finite and relatively small, the loss in a loss period is determined by the buffer size, the buffer content at the beginning of the loss period, and the loss volume of a loss period in a bufferless system. Loss with a small buffer can then be approximated from the characteristics of the bufferless system.

The excess capacity in a non-loss period is the unused capacity at a time instant. The excess volume ($V$) is then defined as the accumulated excess capacity

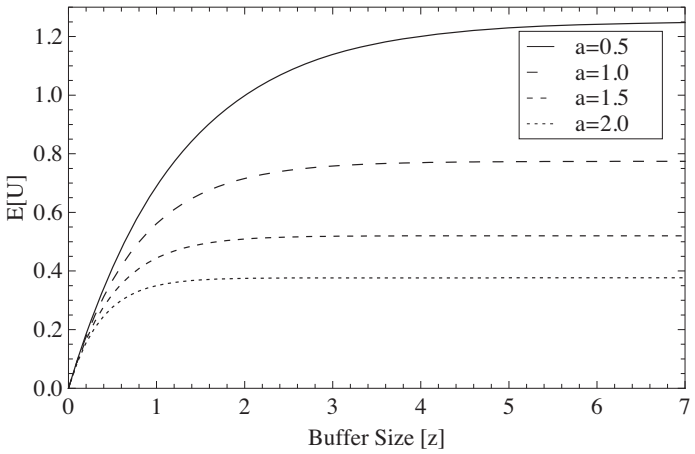



**Figure 9.14:** Expected buffer content at the end of a loss period, for different thresholds $a$.

in a non-loss period and gives the available capacity that can be used for emptying the buffer before the next loss period. When a non-loss period has a larger excess volume than the remaining buffer content at the end of a loss period, the loss can be found from the loss volume in a loss period in the bufferless system, since the buffer is then always empty at the beginning of a new loss period.

The probability of an empty buffer at the start of a loss period for given thresholds and buffer sizes is therefore investigated. The buffer size is denoted $z$ and the buffer content is denoted $U$. The buffer content at the end of a loss period is then given as $U = \min(z, S)$, where $S$ is the loss volume of a loss period in the bufferless model.

The expected buffer content at the end of a loss period is then:

$$E[U] = P(S \geq z) \cdot z + P(0 < S < z) \cdot E[S|0 < S < z] \tag{9.29}$$

The expected buffer content is shown in Figure 9.14 for different buffer sizes and thresholds.

The excess volume in a non-loss period can be found numerically using the same procedure as for the loss volume of a loss period. The probability of the excess volume in a non-loss period being larger than the remaining buffer content at the end of a loss period is then given as:

$$\begin{aligned} Pr[V > U] &= \int_{u=0}^{z} Pr[V > U|U = u] \cdot Pr[U = u]du \\ &= \int_{u=0}^{z} \int_{v=u}^{\infty} f_U(u) f_V(v) du dv \end{aligned} \tag{9.30}$$

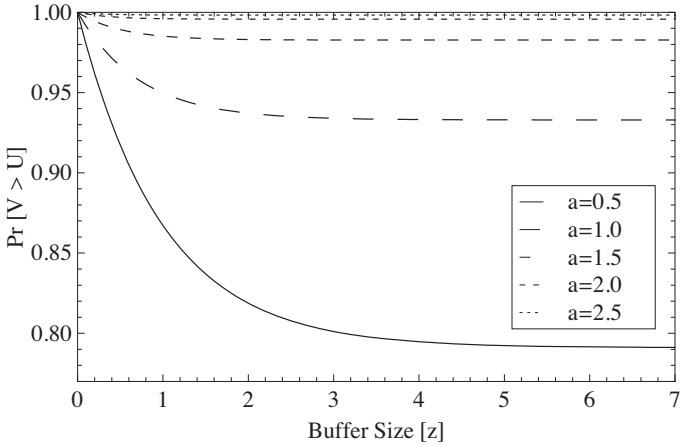



**Figure 9.15:** The probability of the excess volume in a non-loss period being larger than the remaining buffer content at the end of a loss period, for different thresholds $a$.

where the density function is $f_U(u)$ for the remaining content and $f_V(v)$ for the excess volume, respectively. It is assumed that the remaining buffer content and the excess volume are independent. For the cases studied, the mean and variance of $S$ and $V$ are known. Both of them have a coefficient of variation around 1, and are therefore modeled with an Exponential distribution. The remaining buffer content will have the same distribution as the loss volume for $U < z$, and be equal to $z$ for $U = z$.

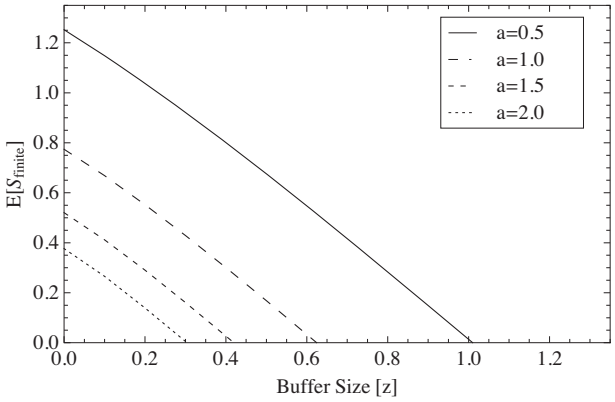
The probability of an excess volume in a non-loss period being larger than the remaining buffer content at the end of a loss period for thresholds equal to 0.5-2.5 is shown in Figure 9.15. As can be seen, the probability that the buffer is empty at the beginning of a loss period is high for high thresholds.

When it is justified that the probability of an empty buffer at the beginning of a loss period is high, the amount of loss ($S_{finite}$) is given by $S_{finite} = \max(S - z, 0)$. The expected loss is then given as:

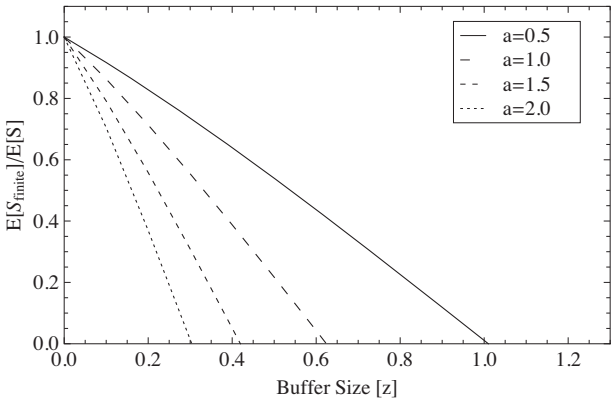
$$E[S_{finite}] = P(S > z) \cdot (E[S|S > z] - z) + P(S \leq z) \cdot 0 \tag{9.31}$$

The expected loss is shown in Figure 9.16(a), for thresholds from 0.5 to 2.0. Clearly, even a very small buffer significantly reduces the expected loss.

Finally, the reduction in the expected loss in a loss period by introduction of a small buffer is given in Figure 9.16(b) as $E[S_{finite}]/E[S]$. This can be interpreted as the gain in terms of reduced loss by including a buffer of the given size.




(a) Expected loss as function of buffer size




(b) The reduction in loss by introduction of a buffer.

**Figure 9.16:** Expected loss with a small buffer, and the corresponding reduction in loss compared to the bufferless case.

## 9.7 Conclusion

The QoS perceived by a user watching a video transmitted over the Internet depends on the packet loss both through its value and its distribution. The moments of the length of a loss period are evaluated with different thresholds and different correlation structures and show that the first and second moments vary with the correlation structure for the same loss probabilities.

The distribution of the loss volume of a loss period is important for estimating the expected loss in a node with a small buffer, in addition to giving the loss directly in a bufferless node. The results for the discrete process are compared to the results for a continuous process, based on results for limit distributions. Comparison of the first and second moments of the length and the volume of a loss period for the continuous and discrete process gives satisfactory correspondence for

high thresholds. This is as expected, since the results for the continuous process are only valid for high thresholds. The relation between the length and volume of an excursion over high thresholds for the continuous process is found to be useful for the discrete process as well for the examples studied.

A first validation of the Gaussian model is done using real video traces, however, more work is required. Also, more effort should be put into the specification of correlation matrices for real video traces. It also remains to find the distribution of losses for a targeted single stream in the aggregate.

# Part V

# Router Models for Quality of Service Assessment

The results in this part have been published as follows:

Astrid Undheim, Yuming Jiang, and Peder J. Emstad. "Network Calculus Approach to Router Modeling Using External Measurements." In *Proceedings of the International Conference on Communications and Networking in China (ChinaCom)*, Shanghai, China, August 2007.

# Chapter 10

# Router Modeling with External Measurements

In this chapter, a measurement approach for estimating parameters for the Guaranteed Rate (GR) server model and the Packet Scale Rate Guarantee (PSRG) server model is proposed. The idea of the proposed approach is to conduct measurements externally on the router and to estimate the desired parameters using burst and backlog period statistics. This is of particular importance since these parameters cannot be obtained or verified through theoretical analysis for a real network router. The characterization of the router is valid for all kinds of input traffic, and a bound on the router delay can be provided when the characteristics of the input traffic, such as the token bucket parameters, are known. This is explored for the slice-based encoded video streams that were characterized using token buckets in Chapter 5.

## 10.1 Introduction

When analyzing service in a communication network, models specifying the service of network nodes are a necessity. Moreover, the deployment of real-time applications such as streaming video and voice over IP on top of the current best-effort Internet has governed the need for QoS guarantees. These guarantees may be specified by server models from the network calculus domain [10], showing the theoretical service guaranteed to traffic flows by a network node. Utilizing these server models represents a new approach to router modeling, using only few parameters that may easily be estimated for a network router.

Two proposals for Internet QoS guarantees have been given by the IETF, namely the IntServ architecture [7] and the DiffServ architecture [8]. The GR server model [75] and the PSRG server model [25] specify the Guaranteed Service [18] by an IntServ router and the Expedited Forwarding (EF) PHB [23] by a DiffServ router respectively, as described in Chapter 2, giving the rationale for choosing

them for router modeling in this work. These models use a rate and an error term parameter to characterize each output interface of the router. The stationarity assumption that is often needed by and hence restricts the use of traditional queueing models is avoided in these models, and they are independent of the router architecture and the input traffic. The theoretical values for the rate and error term parameters under GR and PSRG are known for many types of schedulers such as First In First Out (FIFO), Weighted Fair Queueing (WFQ) and Deficit Round Robin (DRR) [72, 73], and delay guarantees are given when the input traffic is constrained by a token bucket model as seen in Chapter 5. However, the complex architecture of a router is not taken into account in calculating these theoretical values, and hence the rate parameter may be smaller and the error term larger for a real router than for the ideal scheduler.

The purpose of this work is to investigate the use of GR and PSRG server models to characterize a router, with particular focus on estimating the parameters for these models through external measurements. As mentioned above, the parameters for the GR and PSRG server models include a rate parameter and an error parameter for each output interface. These parameters may be obtained in several ways: (1) through theoretical analysis, (2) through simulating the router, (3) through actively probing the router and (4) through passive measurements. The first two methods require detailed knowledge of the router which includes the internal switching/routing architecture and the scheduling and buffer properties [160]. However, such information is often not available. For the third method, active probing introduces extra load in the network, which may be undesirable in an operational setting. In this work, the focus is on using passive measurements to estimate the required parameters, providing an elegant solution to the parameter estimation when theoretical values cannot be obtained or these values need to be verified.

### 10.1.1 Related Work

For modeling network routers, traditional queueing models [161] and models taking the detailed router architecture into account [162] have been employed. However, the limitations imposed by these models, such as input traffic constraints and dependence on the router architecture, restrict their usability.

Using external, passive measurements has come up as an attractive approach to estimate parameters for a network router model. In [163], external measurements are used for estimating the router performance and a model of the router is developed. The model incorporates a minimum processing time as well as a fluid output queue. Measurements on network routers have also been used to evaluate service guarantee mechanisms. In [164], measurements on router backlog periods are used for identifying the scheduling algorithms employed in the routers as well as estimating the rate limiter parameters. Passive measurements are used together with maximum likelihood estimation to determine the parameters of interest, based on the estimated arrival and service rates.

Simulation is another approach to parameter estimation for a network router,

requiring a model of the router. In [160], a simulation approach is proposed to estimate parameters for a non-FIFO node using input and output statistics for throughput analysis. Backlog periods are identified and an approach to estimation of the parameters for a rate-latency service curve is described.

### 10.1.2 Chapter Outline

The rest of this chapter is organized as follows. In Section 10.2, the GR and PSRG server models considered for router modeling are introduced. Also, approaches appropriate for parameter estimation are presented as well as the relationship to the GR and PSRG server models. In Section 10.3, techniques for estimation of the different parameters are given. The measurement setup, results from the measurements, and the mapping to the chosen router models are presented in Section 10.4. Finally, conclusions are given in Section 10.5.

## 10.2 Network Calculus Approach to Router Modeling

The structure of a generic router mainly consists of three parts: (i) the router line cards that hold the input and output ports, (ii) the routing processor which runs the operating system and computes the router forwarding tables, and finally, (iii) the switching fabric, most often consisting of buses, shared memory or a cross-bar switch. The server models from the network calculus domain usually describe a router as simply consisting of buffered output links, not taking the processing and switching time into account. This is a simplification and in [165], the router delay is separated into queueing time, processing/switching time, and transmission time, showing that the processing/switching time is a significant part of the total router delay. This implies that the allocated rate is lower and the error term is higher for a real router than they are in the theoretical model.

Several different models have been proposed to describe the behaviour of a server and analyze service guarantees under network calculus [25, 71–73]. The GR and PSRG server models are defined to specify the guaranteed service of an IntServ router and the EF PHB class for a DiffServ router respectively, and are therefore the most interesting server models for router modeling. Modeling routers using GR and PSRG requires that the allocated rate as well as the error term are determined or estimated. These models were defined in Section 2.3.1, but are repeated here for the convenience of the reader.

For a GR server with rate $r$ and error term $E$, it is guaranteed that the $j$th arriving packet is transmitted by time [73]:

$$d^j \leq VFT^j + E \,, \tag{10.1}$$

where the VFT is defined as:

$$VFT^j = \max\{a^j, VFT^{j-1}\} + \frac{l^j}{r} \tag{10.2}$$

where $a^j$ is the arrival time of packet $j$ and $l^j$ is the packet size.

For a PSRG server with rate $r$ and error term $E$, it is guaranteed that the $j$th arriving packet is transmitted by time [25]:

$$d^j \leq PFT^j + E \ , \tag{10.3}$$

where the PFT is defined as:

$$PFT^j = \max\{a^j, \min\{PFT^{j-1}, d^{j-1}\}\} + \frac{l^j}{r} \ , \tag{10.4}$$

With an estimate for the rate parameter, a direct matching approach based on Equations 10.1-10.4 can be used to find the error parameters. However, when these parameters need to be obtained through measurements, this approach is cumbersome, because the VFT or PFT for each packet must be calculated, and the error term for each packet must be found in order to find the smallest term such that the delay guarantee is fulfilled.

### 10.2.1 Approach to Parameter Estimation

In this work, a novel approach to parameter estimation for GR and PSRG is advocated. It has been proved in the literature that the GR model is closely related to the service curve model [10, 74] and the PSRG model is equivalent to the adaptive service curve model, both defined under network calculus [10, 25]. Based on these relationships and available results for getting the service curve or the adaptive service curve of a server, it is proposed to conduct measurements on the burst periods and/or the backlog periods and from the measurements to further deduce the parameters for the GR and PSRG models.

A router backlog period is defined as a maximum time interval in which one or more packets are received but not yet served by the router. The router is then continuously backlogged for $k$ packet transmissions in the interval $[a^j, d^{j+k-1}]$ if [164]:

$$d^{j+m} > a^{j+m+1} \ , \tag{10.5}$$

for all $0 \leq m < k-1$ for $k \geq 2$.

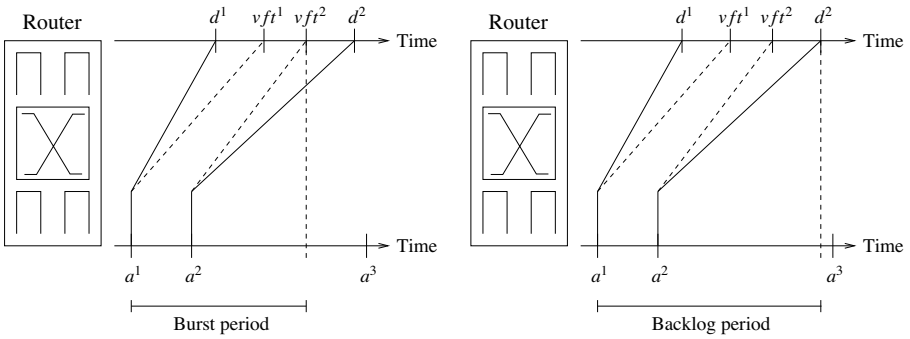
A router burst period is defined as a time period $[t^0, t^*]$ where the average arrival rate at time $t$ in $[t^0, t^*]$ is always at or above the reserved rate, $r$ [71]. That is:

$$A(t^0, t) \geq r(t - t^0) \ , \tag{10.6}$$

where $A(s, t)$ denotes the amount of traffic arrived in the time interval $[s, t]$.

The difference between burst period and backlog period is shown in Figure 10.1, where a burst period is defined between $a^1$ and $vft^2$ and the router is backlogged from $a^1$ to $d^2$.

The following results provide the theoretical basis for the proposed approach to estimating parameters for GR and PSRG. Their proofs follow directly from Theorem 2 in [74] and Theorem 2.1 in [72] respectively and are hence omitted.




(a) Burst Period (b) Backlog Period

**Figure 10.1:** The router burst and backlog periods ended by the VFT and the departure time respectively.

**Lemma 1.** *Consider a network node. If for any time $t$ in a burst period $[t^0, t^*]$, the amount of service provided by the node in time interval $[t^0, t]$ satisfies*

$$W(t^0, t) \geq r(t - t^0 - \Theta)^+ \tag{10.7}$$

*then the node is a GR server with rate $r$ and error term $E$:*

$$E = \Theta - \frac{L^{min}}{r} , \tag{10.8}$$

*where $L^{min}$ is the minimum packet length, and $(x)^+ = max\{x, 0\}$.*

**Lemma 2.** *Consider a network node. If for any time $t$ in a backlog period $[t^0, t^*]$, the amount of service provided by the node in time interval $[t^0, t]$ satisfies*

$$W(t^0, t) \geq r(t - t^0 - \Theta)^+ \tag{10.9}$$

*then the node is a PSRG server with rate $r$ and error term $E$:*

$$E = \Theta . \tag{10.10}$$

With Lemma 1 and Lemma 2, it is clear that to obtain the rate and error term for the GR and PSRG server models, it is sufficient to measure the rate and $\Theta$, called the latency term in the rest of the chapter, respectively. Next, the approach to determe these parameters through measurements is presented.

## 10.3 External Measurements for Router Model Parameterization

This section details the approach to parameter estimation for Lemma 1 and Lemma 2 through measurements. The measurement results are then used to estimate the parameters for the GR and PSRG server models.

### 10.3.1 Estimation of the Rate Parameter

As seen from the definitions, the backlog periods are defined using the arrival and departure times, and can be used to find an estimate for the allocated rate. In this work, the maximum throughput measured over backlog periods is used as the rate parameter. This corresponds to the theoretical rate parameter for the server models with FIFO scheduling that ideally should be equal to the capacity on the outgoing link.

The throughput or the amount of service received in the time interval $[s, s+t]$ of length $t$ when the router is continuously backlogged is defined as [164]:

$$R(t) = \frac{U[s, s+t]}{t} \ , \tag{10.11}$$

where $U[s, s+t]$ is the number of bits serviced in the time interval in which the router is backlogged. The estimation of the throughput is for the full length of each backlog period, hence $s$ is the arrival time of the first packet and $s+t$ is the departure time of the last packet in the backlog period.

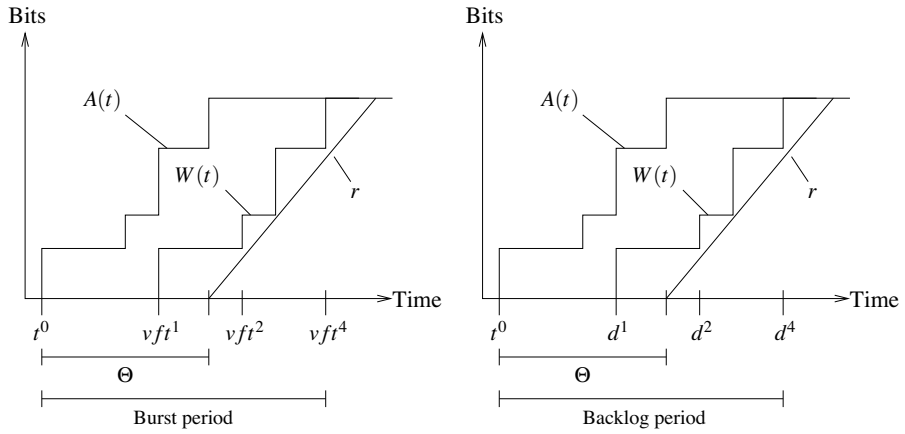
The maximum $R(t)$ is then taken as the estimate for the allocated rate, from which, the VFTs and PFTs can be estimated and the burst periods identified.

### 10.3.2 Estimation of the Error Parameters

To estimate the error parameters, several methods are available. In this work, it is first investigated how to directly use the definitions of GR and PSRG to estimate the error terms. Furthermore, backlog period and burst period measurements are used for parameter estimation for Lemma 1 and Lemma 2, based on which error terms for the GR and PSRG models are estimated.

Once the rate is estimated, the VFTs and PFTs are found using the arrival and departure time of each packet and the definitions in Equations 10.2 and 10.4. With this at hand, the error term for the GR and PSRG server models can be estimated by the maximum error samples found using the departure time, the VFT/PFT for each packet and the definitions from Equations 10.1 and 10.3.

The error terms found from these definitions give the lowest possible error terms for the measurement samples. The method will however require several computations for each packet for finding the VFT/PFT and the corresponding error term, and the complexity is proportional to the load. Simpler methods for the error parameter estimation are therefore required.




(a) From burst period (b) From backlog period

**Figure 10.2:** Latency term estimation from burst/backlog period.

As described earlier, Lemma 1 and Lemma 2 define service during burst and backlog periods respectively. A procedure for latency term estimation for Lemma 1 and Lemma 2 followed by error parameter estimation for the GR and PSRG server models is proposed in the following.

The VFT of each packet is found using the allocated rate, $r$, and the burst periods are identified. The maximum latency in each burst period is calculated by finding the curve with rate equal to $r$ bounding the amount of service received. The latency in each burst period is the distance from $t_0$ to the intersection of $r$ with the time axis as shown in Figure 10.2(a). The estimate for the latency term $\Theta$ for Lemma 1 is found as the maximum of the sample latencies. The error parameter, $E$, for the GR server model is then found using the latency term estimate from the burst period analysis and the relationship from Equation 10.8.

The maximum latency in each backlog period is calculated by finding the curve with rate equal to $r$ bounding the amount of service received as shown in Figure 10.2(b). The latency is the distance between $t^0$ and the intersection of $r$ with the time axis and the latency term $\Theta$ for Lemma 2 is found as the maximum of the sample latencies. The error parameter, $E$, for the PSRG server model is found using the latency term estimate from the backlog period analysis and the relationship from Equation 10.10.

### 10.3.3 Estimating the Processing Time

The GR and PSRG server models account for the queueing delay due to packets of the same (possibly aggregated) flow and the transmission time on the outgoing link, given a certain allocated rate. With no queueing, a maximum rate lower than the theoretical capacity on the outgoing link will be due to internal processing.

A lower bound on the processing time may then be found by investigating the processing time for packets transmitted during backlog periods with only one packet in backlog. In these periods, packets will experience an empty router and hence no queueing delay.

The sample rates for the packets transmitted in backlog periods with only one packet in backlog are found by:

$$r^j = \frac{l^j}{d^j - a^j} \ , \tag{10.12}$$

where $r^j$ is the rate experienced by packet $j$. The total delay, $d^j - a^j$ may also be specified using the theoretical capacity $C$ and the processing time $\delta^j$ as follows:

$$d^j - a^j = \delta^j + \frac{l^j}{C} \ , \tag{10.13}$$

where $\delta^j$ is the processing delay for packet $j$. When inserting Equation 10.13 into Equation 10.12 an expression for $\delta^j$ is found:

$$\delta^j = \frac{l^j \cdot (1 - \frac{r^j}{C})}{r^j} \ . \tag{10.14}$$

The minimum of these samples is accounted for in an allocated rate lower than the capacity. The error term must then account for the internal router delay due to processing times higher than the minimum. The result from the evaluation of the processing time is shown in Section 10.4.3.

## 10.4 Results from Measurements

The idea of the proposed approach is to obtain packet information through passive measurement on input and output links of a router. This information is then used to calculate the rate and error term parameters in its GR and PSRG models.

### 10.4.1 Measurement Setup

DAG-cards [166] are used for the external measurements. These cards are capable of capturing a chosen amount of information from the IP-header such as arrival time, packet length, and type of service. With such information, delay, loss, throughput as well as burst period and backlog period statistics can be gathered.

The measurement setup is shown in Figure 10.3 with the studied router in the center and DAG cards for traffic capturing. The DAG cards are synchronized using a GPS receiver, giving satisfactory timestamp accuracy. All links are Ethernet links. An ordinary PC is used to produce measurement traffic using tcpreplay [167], capable of replaying any file in the .pcap format. The trace files are pre-recorded multimedia traces in the MPEG4 encoding format.

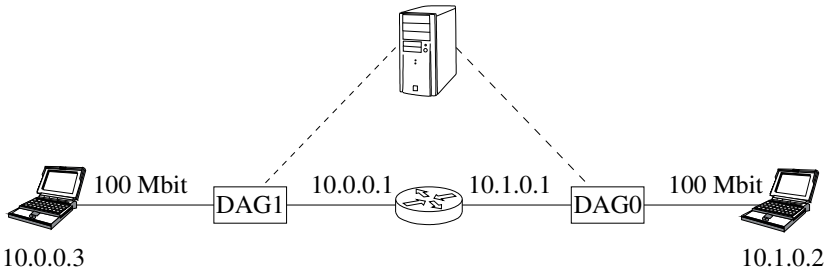



**Figure 10.3:** The measurement setup with the DAG-cards indicated.

### 10.4.2 Parameter Estimation for a FIFO Scheduler

Without loss of generality and for ease of explanation, the considered router uses FIFO for packet scheduling as do (most) routers in the current Internet.

The maximum estimated rate is $9.6597 \cdot 10^7$ bps while the capacity on the link is 100 Mbps. The rate parameter should ideally be equal to the capacity, as for the theoretical models. However, because of the processing time, a lower maximum rate is experienced. The estimated latency terms for Lemma 1 and Lemma 2 found from the burst/backlog periods are shown in Table 10.1, together with the theoretical values. The theoretical values for a FIFO scheduler are given in [72, 74], and are equal to $L^{max}/C$, where $L^{max}$ is the maximum packet size.

**Table 10.1:** The estimated latency parameters for Lemma 1 and Lemma 2.

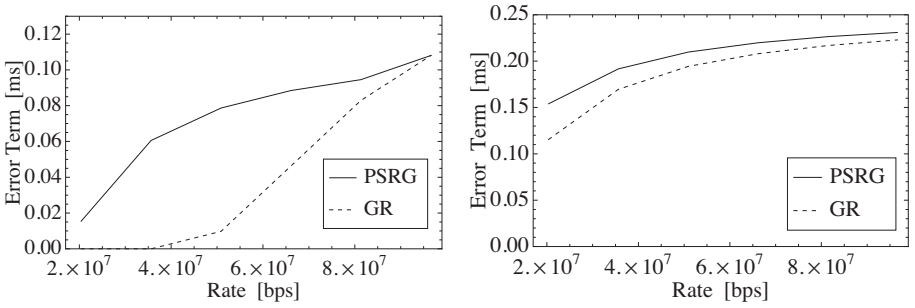
| Model | Theoretical Latency | Burst/Backlog Latency |
|---|---|---|
| Lemma 1 | 0.12112 ms | 0.2309 ms |
| Lemma 2 | 0.12112 ms | 0.2309 ms |

The estimated error terms for the GR and PSRG server models are shown in Table 10.2, where the parameters are found using direct evaluation and burst/backlog period estimation together with the relationships from Section 10.2.

**Table 10.2:** The estimated parameters for the GR and PSRG server models.

| Model | Theoretical Error | Direct Error | Burst/Backlog Error |
|---|---|---|---|
| GR | 0 ms | 0.10808 ms | 0.2230 ms |
| PSRG | 0.12112 ms | 0.10808 ms | 0.2309 ms |

The error terms found from direct evaluation is the same for the GR and PSRG server models. This is obvious since the maximum measured rate in a backlog period is used as the rate parameter. This means that the server never runs early compared to the ideal constant rate server and the departure time of a packet is then never before the PFT. Also, the error term for PSRG found from direct evaluation is lower than the theoretical value, since the latter, i.e., the theoretical value from [72], is based on a more stringent server definition [72].




(a) Direct Evaluation (b) Burst/Backlog Period Evaluation

**Figure 10.4:** Parameter curves with different rate parameter estimates.

The error terms from the backlog/burst period estimations are higher than the error terms from the direct evaluation, as expected since the latency is calculated by assuming that all packets are transmitted at the maximum rate. According to results in [74], the error term for a GR server model with FIFO scheduling should be 0. The error terms found from the measurements are however higher, since the packets experience delays due to processing time higher than the minimum processing time.

The parameters estimated from the direct evaluation and the burst/backlog periods are also shown in Figure 10.4, where the rate parameter is varied between the minimum estimated rate from the backlog periods to the maximum estimated rate in the backlog periods.

For the direct evaluation, the curve for the PSRG server model is significantly higher than for the GR server model for low rates because the error term for PSRG also includes jitter and the server runs faster than the allocated rate. For an allocated rate equal to the minimum estimated throughput, the error term for the GR server model is zero, in agreement with the theoretical model. Estimating the error terms from burst and backlog periods and the server model relationships gives approximately the same error terms as can be seen in Figure 10.4(b), but the curves are higher than those from the direct evaluation.

For token bucket constrained traffic flows, such as the StEM and Mobile video traces investigated in Chapter 5, the results from the router parameterization give the delay bound of the streams through the specific router. The delay when the router is specified as a GR server with FIFO scheduling will be bounded according to Equation 5.2, as:

$$\begin{aligned} D &\leq \frac{\sigma}{r} + E \\ &\leq \frac{80 \cdot 8 \cdot 1000 \text{ bit}}{9.6597 \cdot 10^7 \text{ bps}} + 0.00010808 \text{ s} = 6.733 \text{ ms} \end{aligned} \tag{10.15}$$

for a token bucket size of 80 kbytes, and a token generation rate lower than the allocated rate. If the allocated rate at the router is equal to the token generation

rate of the input flow, the delay bound will be given as:

$$D \leq \frac{80 \cdot 8 \cdot 1000 \text{ bit}}{5.0 \cdot 10^6 \text{ bps}} + 0.00010808 \text{ s} = 128.108 \text{ ms} \tag{10.16}$$

where the token generation rate is 5000 kbps, corresponding to the knee point of the token bucket curve for the StEM clip in Figure 5.1(a).

### 10.4.3 Processing Time

When looking at the throughput in backlog periods with only one packet, the minimum processing time through the router is found, as discussed earlier, and as shown in Table 10.3.

**Table 10.3:** The processing time evaluated in backlog periods.

| Max rate | Min proc, $\delta^j_{min}$ | Max proc $\delta^j_{max}$ |
|---|---|---|
| $9.3457 \cdot 10^7$ bps | 0.00524 ms | 0.11296 ms |

The maximum rate is lower here than for the rate parameter estimation using all backlog periods because the former is estimated from backlog periods with only one packet in backlog and no packets are processed while others are transmitted. The minimum processing time is accounted for in an allocated rate lower than the capacity on the outgoing link as explained earlier. It is also interesting to look at the maximum processing time, which is approximately equal to the difference between the theoretical latency parameter and the estimated latency parameters, explaining the higher value for the latter.

The processing time is highly variable although all samples are from backlog periods with one packet in backlog. A packet size dependent component as found in [165] can explain some of this variability, which also means that the error term for the server models may be divided into a packet size dependent and independent part. Also, router internal operations such at router table updates and garbage collection may cause additional delay. These are irregular and can be fairly large. All these issues need further investigation.

## 10.5 Conclusion

The GR and PSRG server models are used for router modeling by IntServ and DiffServ. In this chapter, it is shown that the parameters of these models for a network router can be found by using external measurements on the router. Specifically, the proposed approach uses relationships between the GR and PSRG models and the server models in the network calculus domain that have close relation to the burst and backlog period. Then based on the relationships, burst period and backlog period measurements are performed and parameter estimation for GR and PSRG is further conducted with satisfactory results. In addition, the measurement results are used to analyze the processing delay in the router. The

processing time is identified as the cause of a lower value of the rate parameter and a higher value of the latency parameter from the measurements, compared to the theoretical values.

The proposed approach provides another means for deciding GR/PSRG parameters when the router architecture is so complex that theoretically determining these parameters is impossible. In addition, the approach is a simple method for the user of a router to verify the router's GR or PSRG specification provided by the vendor.

Future work includes more extensive measurements with different input traffic as well as measurements with alternative scheduling disciplines on the output link.

# Part VI

# Concluding Remarks

# Chapter 11

# Conclusions

This thesis focuses on quality of service issues for video transmissions over the Internet. When real-time traffic is transmitted over resource constrained networks, stringent requirements to the network performance parameters, throughput, delay, delay jitter, and packet loss are imposed in order to guarantee a satisfactory user experience. In this respect, service guarantees need to be provided by the network in order to accommodate these requirements.

The work presented in this thesis focuses on characterization and modeling of video traffic, in particular the new slice-based H.264/AVC encoded video traffic. Estimation of the performance parameters and in particular the packet loss is of high interest. The loss period distribution will influence the perceived QoS and is important in addition to the knowledge about the total amount of loss.

Traffic characterization is necessary for developing traffic models and Part II of the thesis focuses on video traffic characterization for slice-based encoded video. This characterization is important since the slice-based encoded video presents a new type of traffic which has not been studied before. The slice-based encoded video has no GOP structure and the frames within the scenes are statistically equal. Important findings include the negligible autocorrelation for frames in different scenes, different from regular frame-based encoded video where the I frames cause high, periodic correlation over scene boundaries. However, there is non-negligible correlation between the size of the scene changes frames and the average frame size in the same scene as well as non-negligible ACF for frames inside the scenes, as is expected. The distributions of the scene lengths and the average frame sizes in the scenes are both found to be close to a Gamma distribution for the stream studied. Simple network simulations show that the slice-based stream performs better than the frame-based stream in terms of lower loss and delay when the buffer size is small.

Resource reservation and admission control are important functions for providing service guarantees in a network. For these functions, traffic characteristics described using token bucket traffic models are usually needed. The token bucket parameters of two slice-based encoded video clips are estimated using simulations.

The StEM clip has frequent scene changes and the resulting token bucket curves are therefore almost identical for the slice-based and frame-based streams. However, for the Mobile clip without scene changes, the token bucket parameters are significantly lower for the slice-based stream than for the frame-based stream and less resources are needed for the former to ensure equal delay guarantees.

In Part III of the thesis, non-parametric analysis of slice-based encoded video traffic is pursued. The sequence of frame sizes is divided into sections, which are classified according to the average frame size. A new quantile method for scene change detection is then employed for the classes. The resulting classes are checked regarding dependence structure, type of distribution, heaviness of tails, and stationarity. The ACFs and Ljung-Box tests show independent scenes in the classes, but a moderate amount of LRD is found using different estimators for the Hurst parameter. The mean excess function shows that the distributions of frame sizes of the selected classes can only be mixtures of classical heavy- and light-tailed distributions and the Hill's estimates give the number of finite moments for the frame distributions inside the classes. The video data are also classified by the value of the extremal index. The extremal index detects changes in the stationarity and dependence of frames within scenes. The dependence structure within the classes is variable due to the variability of the video stream. Non-parametric methods are also employed for estimation of loss using a bufferless model. The average bit loss in a loss period, the average length between loss periods and the overall bit losses in the bufferless model are found. These statistics give information about the distribution of the losses. Moreover, the high quantiles of bit losses are evaluated. These quantiles give statistical guarantees for the amount of loss which can occur.

The correlation analysis in Part II showed that frame correlation exists only within scenes. Based on this, in Part IV of the thesis a discrete Gaussian model is proposed for the slice-based encoded video traffic, taking the autocorrelation for the frames inside a scene into account. The Gaussian model is a simple model and is found to give accurate results for the cases studied, and the same approach can be applied to audio and video traffic. The exceedances of the video frames over a threshold then constitute loss periods of variable lengths and variable loss volumes. Utilizing the additive properties of the Gaussian process, the distribution of a loss period for an aggregated video stream is estimated using characteristics of a basic stream. The loss volume gives the loss in a bufferless node. In addition, the distribution of the loss volume is used for estimating the expected loss in a node with a small buffer. The moments of the length and loss volume of a loss period for the discrete process are shown to be comparable to the results for a continuous process, over high thresholds. The simple relation between the length and loss volume of a loss period for the continuous process is found to hold also for the discrete process.

GR and PSRG server models have been proposed for analyzing service guarantees in IntServ and DiffServ. In Part V of the thesis, it is shown that the parameters of these models for a network router can be found from external measurements on the router. Specifically, the proposed approach uses relationships between the GR and PSRG models and the network calculus server models that have close relations

to the burst and backlog period. Then based on the relationships, measurement results are analyzed for burst and backlog periods and parameter estimation for GR and PSRG are conducted with satisfactory results. These results are then used for calculating the delay bound for the token bucket constrained slice-based encoded video streams from Part II. In addition, the measurement results are used to analyze the router processing delay, which is identified as the cause of a lower value of the rate parameter and a higher value of the error parameter from measurements, compared to the theoretical values.

## 11.1 Future Work

This thesis addresses the estimation of some important network performance parameters. This should be followed by a mapping to the perceived QoS. Some work has already been done on this mapping, as discussed in the thesis, but it is a complicated task because of the wide diversity in the video applications and in the coding techniques applied. Subjective tests addresses this problem, and work on such subjective tests is currently in progress at the Q2S centre.

Both for the characterization and modeling of the video traffic, more and longer video traces should be studied for the slice-based video encoding scheme. The slice-based video encoding scheme is only implemented in software. Together with the lack of raw video available, this causes difficulty in producing long video clips of required quality.

Both the regular characterization and the token bucket characterization would benefit from being applied to the classes, since the frames inside the classes are more stationary and homogeneous. In addition, video source models such as ARMA models can be defined for the different classes.

For the Gaussian modeling, several issues remain to be studied. A first validation of the Gaussian model is done using real video traces. However, more work is required for the validation. Validating that an aggregate of slice-based encoded video can be modeled as a multivariate Gaussian process is needed, using real measurements. This is not done due to the same reasons as above, the lack of suitable traces. Also, more efforts should be put into the modeling of correlation matrices from real video traces. The accuracy of this modeling depends on the length of the video, and should be studied further for more and longer video clips.

It also remains to find the distribution of losses for a single stream in the aggregate. In this respect, it is interesting to study the length and loss volume of a loss period for a single stream with known characteristics for the aggregate.

For the network calculus router modeling, only the simple FIFO scheduler was studied. Using alternative scheduling disciplines on the output link as well as applying different input traffic, possible with different service classes, are of high interest. Measurements to address these issues are continued in another project.

# Bibliography


[1] V. G. Cerf and R. E. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, no. 5, pp. 637–648, May 1974.

[2] W. B. Norton, "Video Internet: The Next Wave of Massive Disruption to the U.S. Peering Ecosystem," in *Asian Pacific Regional Internet Conference on Operational Technologies (APRICOT)*, Bali, Indonesia, February-March 2007.

[3] "Youtube," [Online]. Available: http://www.youtube.com.

[4] ITU-T, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264, November 2007.

[5] ——, "Terms and Definitions Related to Quality of Service and Network Performance Including Dependability," ITU-T Recommendation E.800, August 1994.

[6] F. Pereira and I. Burnett, "Universal Multimedia Experiences for Tomorrow," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 63–73, March 2003.

[7] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," IETF RFC 1633, July 1994.

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

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

[10] J. Y. Le Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*. Springer-Verlag, 2001.

[11] M. Fidler, Y. Lin, P. J. Emstad, and A. Perkis, "Efficient Smoothing of Robust VBR Video Traffic by Explicit Slice-Based Mode Type Selection," in *Proceedings of the 4th IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, USA, January 2007, pp. 880–884.

[12] S. Keshav, *An Engineering Approach to Computer Networking*. Addison-Wesley, 1997.

[13] V. B. Iversen, *Data- og teletrafikteori*. Den Private Ingeniørfond, 1999.

[14] ITU-T, "Communications Quality of Service: A Framework and Definitions," ITU-T Recommendation G.1000, November 2001.

[15] S. Shenker and J. Wroclawski, "Network Element Service Specification Template," IETF RFC 2216, September 1997.

[16] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)," IETF RFC 2205, September 1997.

[17] S. Shenker and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements," IETF RFC 2215, September 1997.

[18] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service," IETF RFC 2212, September 1997.

[19] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," IETF RFC 2211, September 1997.

[20] M. S. Blumenthal and D. D. Clark, "Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World," *ACM Transactions on Internet Technology*, vol. 1, no. 1, pp. 70–109, August 2001.

[21] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A Framework for Integrated Services Operation over DiffServ Networks," IETF RFC 2998, November 2000.

[22] F. Baker, C. Iturralde, F. L. Faucheur, and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations," IETF RFC 3175, September 2001.

[23] B. Davie, A. Charny, J. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," IETF RFC 3246, March 2002.

[24] J. Heinanen, F. Baker, W. Weiss, and J. Wroclowski, "Assured Forwarding PHB Group," IETF RFC 2597, June 1999.

[25] J. C. R. Bennett, K. Benson, A. Charny, W. F. Courtney, and J.-Y. Le Boudec, "Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 529–540, August 2002.

[26] S. Floyd and V. Jacobsen, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

[27] B. Teitelbaum, S. Hares, L. Dunn, R. Neilson, V. Narayan, and F. Reichmeyer, "Internet2 QBone: Building a Testbed for Differentiated Services," *IEEE Network*, vol. 13, no. 5, pp. 8–16, September/October 1999.

[28] J. Kimura, F. A. Tobagi, J.-M. Pulido, and P. J. Emstad, "Perceived Quality and Bandwidth Characterization of Layered MPEG-2 Video Coding," in *Proceedings of the SPIE International Symposium on Voice, Video and Data Communications*, Boston, USA, September 1999.

[29] B. Teitelbaum and S. Shalunov, "Why Premium IP Service Has Not Deloyed (and Probably Never Will)," Internet2 QoS Working Group Informal Document, May 2002.

[30] B. Davie, "Deployment Experience with Differentiated Services," in *Proceedings of the ACM SIGCOMM 2003 Workshops*, August 2003, pp. 131–136.

[31] ITU-T, "End-user Multimedia QoS Categories," ITU-T Recommendation G.1010, November 2001.

[32] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance Models of Statistical Multiplexing in Packet Video Communications," *IEEE Transactions on Networking*, vol. 36, no. 7, pp. 834–844, July 1988.

[33] M. N. Garcia, A. Raake, and P. List, "Towards Content-related Features for Parametric Video Quality Prediction of IPTV Services," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '08)*, Las Vegas, USA, March-April 2008, pp. 757–760.

[34] D. Wu, Y. T. Hou, W. Zhu, Y.-Q. Zhang, and J. M. Peha, "Streaming Video over the Internet: Approaches and Directions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 282–300, March 2001.

[35] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, July 2003.

[36] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, " RTP Payload Format for H.264 Video," IETF RFC 3984, February 2005.

[37] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in Wireless Environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, July 2003.

[38] J. Postel, "Internet Protocol," IETF RFC 791, September 1981.

[39] V. G. Cerf and Y. D. C. Sunshine, "Specification of Internet Transmission Control Program," IETF RFC 675, December 1974.

[40] J. Postel, "Transmission Control Protocol," IETF RFC 793, September 1981.

[41] ——, "User Datagram Protocol," IETF RFC 768, August 1980.

[42] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobsen, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 3550, July 2003.

[43] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," IETF RFC 2326, April 1998.

[44] P. Chimento and J. Ishac, "Defining Network Capacity," IETF RFC 5136, February 2008.

[45] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM," IETF RFC 2679, September 1999.

[46] ——, "A Round-trip Delay Metric for IPPM," IETF RFC 2681, September 1999.

[47] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)," IETF RFC 3393, November 2002.

[48] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Packet Loss Metric for IPPM," IETF RFC 2680, September 1999.

[49] R. Koodli and R. Ravikanth, "One-way Loss Pattern Sample Metrics," IETF RFC 3357, August 2002.

[50] S. Tao, J. Apostolopoulos, and R. Guerin, "Real-Time Monitoring of Video Quality in IP Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1052–1065, October 2008.

[51] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, September 1995.

[52] ITU-T, "Vocabulary for Performance and Quality of Service," ITU-T Recommendation P.10, July 2006.

[53] ——, "Methods for Subjective Determination of Transmission Quality," ITU-T Recommendation P.800, August 1996.

[54] S. Winkler and P. Mohandas, "The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics," *IEEE Transaction on Broadcasting*, vol. 54, no. 3, pp. 660–668, September 2008.

[55] Z. Wang, L. Lu, and A. C. Bovik, "Video Quality Assessment Based on Structural Distortion Measurement," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, February 2004.

[56] M. Masry, S. Hemami, and Y. Sermadevi, "A Scalable Wavelet-based Video Distortion Metric and Applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 260–273, February 2006.

[57] O. I. Hillestad, "Evaluating and Enhancing the Performance of IP-based Streaming Media Services and Applications," Ph.D. dissertation, NTNU, May 2007.

[58] Video Quality Experts Group (VQEG), "Multimedia Phase I, Final Report," www.vqeg.org, 2008.

[59] ITU-T, "Objective Perceptual Multimedia Video Quality Measurement in the Presence of a Full Reference," ITU-T Recommendation J.247, August 2008.

[60] ——, "Perceptual Audiovisual Quality Measurement Techniques for Multimedia Services over Digital Cable Television Networks in the Presence of a Reduced Bandwidth Reference," ITU-T Recommendation J.246, August 2008.

[61] ——, "The E-model: A Computational Model for Use in Transmission Planning," ITU-T Recommendation G.107, August 2008.

[62] A. Raake, "Predicting Speech Quality under Random Packet Loss: Individual Impairment and Additivity with other Network Impairments," *Acta Acustica united with Acustica*, vol. 90, no. 6, pp. 1061–1083, 2004.

[63] ——, "Short- and Long-Term Packet Loss Behavior: Towards Speech Quality Prediction for Arbitrary Loss Distributions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1957–1968, November 2006.

[64] J. E. Voldhaug, E. Hellerud, A. Undheim, E. Austreim, U. P. Svensson, and P. J. Emstad, "Influence of Sender Parameters and Network Architecture on Perceived Audio Quality," *Acta Acustica United with Acustica*, vol. 94, pp. 1–11, February 2008.

[65] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of Packet Loss for Compressed Video: Does Burst-Length Matter?" in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 5, April 2003.

[66] A. R. Reibman, V. A. Vaishampayan, and Y. Seradevi, "Quality Monitoring of Video Over a Packet Network," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 327–334, April 2004.

[67] S. Mohamed and G. Rubino, "A Study of Real-Time Packet Video Quality Using Random Neural Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 12, pp. 1071–1083, December 2002.

[68] A. Raake, M. N. Garcia, S. Möller, J. Berger, F. Kling, P. List, J. Johann, and C. Heidemann, "T-V-Model: Parameter-based Prediction of IPTV Quality," in *Proceedings IEEE International Conference on Acoustics, Speech, and*

*Signal Processing (ICASSP '08)*, Las Vegas, USA, March-April 2008, pp. 1149–1152.

[69] C. S. Chang, *Performance Guarantees in Communications Networks*. Springer-Verlag, 2000.

[70] Y. Jiang and Y. Liu, *Stochastic Network Calculus*. Springer, 2008.

[71] D. Stiliadis and A. Varma, "Latency Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 611–624, January 1998.

[72] Y. Jiang, "Per-Domain Packet Scale Rate Guarantee for Expedited Forwarding," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 630–643, June 2006.

[73] P. Goyal and H. M. Vin, "Generalized Guaranteed Rate Scheduling: A Framework," *IEEE/ACM Transactions on Networking*, vol. 5, no. 4, pp. 561–571, August 1997.

[74] Y. Jiang, "Relationship Between Guaranteed Rate Server and Latency Rate Server," *Computer Networks*, vol. 43, no. 3, pp. 307–315, October 2003.

[75] P. Goyal, S. S. Lam, and H. M. Vin, "Determining End-to-End Delay Bounds in Heterogeneous Networks," in *Proceedings of the Workshop on Network and Operating System Support for Digital Audio and Video*, April 1995, pp. 287–298.

[76] A. W. Berger and W. Whitt, "The Impact of a Job Buffer in a Token-Bank Rate-Control Throttle," *Stochastic Models*, vol. 8, no. 4, pp. 685–717, 1992.

[77] S. S. M. Wittevrongel and H. L. Bruneel, "Analytic Study of the Queueing Performance and the Departure Process of a Leaky Bucket with Bursty Input Traffic," *AEÜ International Journal of Electronics and Communications*, vol. 50, no. 1, pp. 1–10, 1996.

[78] A. Undheim, Y. Lin, and P. J. Emstad, "Characterization of Slice-based H.264/AVC Encoded Video Traffic," in *Proceedings of the 4th European Conference on Universal Multiservice Networks (ECUMN 2007)*, Toulouse, France, February 2007, pp. 263–272.

[79] Y. Lin, A. N. Kim, E. Gurses, and A. Perkis, "On the Error Resilience of Rate Smoothing using Explicit Slice-Based Mode Selection," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, July 2007, pp. 2134–2137.

[80] ——, "Rate-Distortion Optimized I-Slice Selection for Low Delay Video Transmission," in *Proceedings of the IEEE 9th Workshop on Multimedia Signal Processing (MMSP)*, Crete, Greece, October 2007, pp. 115–118.

[81] A. Undheim and P. J. Emstad, "Characterization of Slice-based H.264/AVC Encoded Video Traffic Using Token Buckets," *Telecommunication Systems*, vol. 39, no. 2, pp. 63–76, October 2008.

[82] N. Markovich, A. Undheim, and P. J. Emstad, "Slice-based VBR Video Traffic-Estimation of Link Loss by Exceedance," in *Proceedings of the 4th International Telecommunication Networking Workshop on QoS in Multi-service IP Networks (QoS-IP)*, Venize, Italy, February 2008, pp. 112–117.

[83] ——, "Classification of Slice-based VBR Video Traffic and Estimation of Link Loss by Exceedance," *To appear in Computer Networks*, 2009.

[84] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[85] H.264/AVC Reference Software JM 10.1, Last modified August 2007, http://iphome.hhi.de-/suehring/tml/download.

[86] StEM, "DCI & ASC Digital Cinema Initiatives and the American Society of Cinematographers-StEM Mini-movie Access Procedure," Available at http://www.dcimovies.com, 2004.

[87] D. P. Heyman and T. V. Lakshman, "Source Models for VBR Broadcast-Video Traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 1, pp. 40–48, February 1996.

[88] D. Liu, E. I. Sara, and W. Suun, "Nested Auto-Regressive Processes for MPEG-Encoded Video Traffic Modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 2, pp. 169–183, February 2001.

[89] M. Krunz and H. Hughes, "A Traffic Model for MPEG-coded VBR Streams," in *Proceedings of the ACM SIGMETRICS'95 Conference*, Ottawa, Canada, May 1995, pp. 47–55.

[90] O. Rose, "Simple and Efficient Models for Variable Bit Rate MPEG Video Traffic," *Performance Evaluation*, vol. 30, no. 1-2, pp. 69–85, July 1997.

[91] M. Krunz and S. K. Tripathi, "On the Characterization of VBR MPEG Streams," in *Proceedings of the ACM SIGMETRICS '97 Conference*, Seattle, USA, June 1997, pp. 192–202.

[92] O. Rose, "Statistical Properties of MPEG Video Traffic and Their Impact on Traffic Modeling in ATM Systems," in *Proceedings of the 20th Conference on Local Computer Networks*, Minneapolis, USA, October 1995, pp. 397–406.

[93] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Model Full-Length Video Using Markov-Modulated Gamma-Based Framework," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 638–649, August 2003.

[94] M. Frey and S. Nguyen-Quang, "A Gamma-Based Framework for Modeling Variable-Rate MPEG Video Sources: The GOP GBAR Model," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 710–719, December 2000.

[95] H. Koumaras, C. Skianis, G. Gardikis, and A. Kourtis, "Analysis of H.264 Video Encoded Traffic," in *Proceedings of INC 2005 Fifth International Network Conference*, Greece, July 2005.

[96] M. Krunz and A. M. Ramasamy, "The Correlation Structure for a Class of Scene-Based Video Models and its Impact on the Dimensioning of Video Buffers," *IEEE Transactions on Multimedia*, vol. 2, no. 1, pp. 27–36, March 2000.

[97] Y. Sun and J. N. Daigle, "A Source Model of Video Traffic Based on Full-Length VBR MPEG4 Video Traces," in *Proceedings of IEEE Globecom*, vol. 2, St. Louis, USA, November-December 2005, pp. 766–770.

[98] M. Dai and D. Loguinov, "Analysis and Modeling of MPEG-4 and H.264 Multi-Layer Video Traffic," in *Proceedings of the IEEE Infocom*, vol. 4, no. 24, March 2005, pp. 2257–2267.

[99] T. Y. Mazraani and G. M. Parulkar, "Performance Analysis of the Ethernet under Conditions of Bursty Traffic," in *Proceedings of IEEE GLOBECOM'92*, Orlando, USA, December 1992, pp. 592–596.

[100] A. A. Lazar, G. Pacifici, and D. E. Pendarakis, "Modeling Video Sources for Real-Time Scheduling," in *Proceedings of IEEE GLOBECOM '93*, vol. 2, Houston, USA, November/December 1993, pp. 835–839.

[101] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, 2nd ed. Springer Texts in Statistics, 2002.

[102] S. Xu and Z. Huang, "A Gamma Autoregressive Video Model on ATM Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 2, pp. 138–142, April 1998.

[103] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 49–59, March 1992.

[104] F. P. Kelly, "Effective Bandwidths at Multi-class Queues," *Queueing Systems*, vol. 9, no. 1-2, pp. 5–16, October 1991.

[105] G. Birtwistle, "DEMOS: Discrete Event Modelling on Simula," MacMillan, 1978.

[106] R. Bruno, R. Garroppo, and S. Giordano, "Estimation of Token Bucket Parameters for VoIP Traffic," in *Proceedings of the IEEE Conference on High Performance, Switching and Routing*, Germany, June 2000, pp. 353–356.

[107] S.-H. Park and S.-J. Ko, "Evaluation of Token Bucket Parameters for VBR MPEG Video Transmission over the Internet," *IEICE Transactions on Communication*, vol. E85-B, no. 1, pp. 43–51, January 2002.

[108] A. Lombardo, G. Morabito, and G. Schembra, "Traffic Specification for MPEG Video Transmission over the Internet," in *Proceedings of the IEEE International Conference on Communications (ICC 2000)*, New Orleans, USA, June 2000, pp. 853–857.

[109] M. F. Alam, M. Atiquzzaman, and M. A. Karim, "Traffic Shaping for MPEG Video Transmission over the Next Generation Internet," *Computer Communications*, vol. 23, no. 14-15, pp. 1336–1348, August 2000.

[110] G. Procissi, A. Garg, M. Gerla, and M. Y. Sanadidi, "Token Bucket Characterization of Long-range Dependent Traffic," *Computer Communications*, vol. 25, no. 11-12, pp. 1009–1017, July 2002.

[111] R. G. Garroppo, S. Giordano, S. Niccolini, and F. Russo, "A Simulation Analysis of Aggregation Strategies in a $WF^2Q+$ Schedulers Network," in *Proceedings of the Workshop on IP Telephony*, New York, USA, April 2001.

[112] J. Beran, *Statistics for Long-Memory Processes*. Chapman & Hall, 1994.

[113] H. Kettani and J. A. Gubner, "A Novel Approach to the Estimation of the Hurst Parameter in Self-Similar Traffic," in *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks*, Tampa, USA, November 2002, pp. 160–165.

[114] I. Weissman, "Estimation of Parameters and Large Quantiles Based on the $k$ Largest Observations," *Journal of American Statistical Association*, vol. 73, no. 364, pp. 812–815, December 1978.

[115] J. Beirlant, Y. Goegebeur, J. Segers, J. Teugels, D. D. Waal, and C. Ferro, *Statistics of Extremes-Theory and Applications*. J. Wiley & Sons, 2004.

[116] P. Embrechts, C. Klüppelberg, and T. Mikosch, *Modelling Extremal Events for Finance and Insurance*. Springer, Berlin, 1997.

[117] A. Ledford and J. Tawn, "Diagnostic for Dependence within Time Series Extremes," *Journal of the Royal Statistical Society. Series B*, vol. 65, no. 2, pp. 521–543, April 2003.

[118] R. Smith and I. Weissman, "Estimating the Extremal Index," *Journal of the Royal Statistical Society. Series B*, vol. 56, no. 3, pp. 515–528, 1994.

[119] N. Markovich, *Non-Parametric Estimation of Univariate Heavy-Tailed Data*. J. Wiley & Sons, 2007.

[120] J. Beran, R. Sherman, M. S. 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, February/March/April 1995.

[121] I. Norros, "A Storage Model with Self-Similar Input," *Queueing Systems*, vol. 16, no. 3-4, pp. 387–396, September 1994.

[122] B. K. Ryu and A. Elwalid, "The Importance of Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities," in *Proceeding of the ACM Sigcomm*, Stanford, USA, August 1996.

[123] G. M. Ljung and G. E. P. Box, "On a Measure of Lack of Fit in Time Series Models," *Biometrika*, vol. 65, no. 2, pp. 297–303, August 1978.

[124] M. Taqqu, V. Teverovsky, and W. Willinger, "Estimators for Long-Range Dependence: an Empirical Study," *Fractals*, vol. 3, no. 4, pp. 785–798, 1995.

[125] H. Hurst, "Long-Term Storage Capacity of Reservoirs," *Transactions of American Society of Civil Engineers*, vol. 16, pp. 770–808, 1951.

[126] P. Abry and D. Veitch, "Wavelet Analysis of Long-Range Dependent Traffic," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 2–15, January 1998.

[127] O. Rose, "Estimation of the Hurst Parameter of Long-Range Dependent Time Series," University of Würzburg Institute of Computer Science, Tech. Rep. 137, February 1996.

[128] T. Higuchi, "Approach to an Irregular Time Series on the Basis of the Fractal Theory," *Physica D*, vol. 31, no. 2, pp. 277–283, June 1988.

[129] S. Y. Novak, "Inference on Heavy Tails from Dependent Data," *Siberian Advances in Mathematics*, vol. 12, no. 2, pp. 73–96, 2002.

[130] P. Hall, "Using the Bootstrap to Estimate Mean Squared Error and Select Smoothing Parameter in Nonparametric Problems," *Journal of Multivariate Analysis*, vol. 32, no. 2, pp. 177–203, April 1990.

[131] C. Ferro and J. Segers, "Inference for Clusters of Extreme Values," *Journal of the Royal Statistical Society, Series B*, vol. 65, no. 2, pp. 545–556, 2003.

[132] V. Naumov and P. Emstad, "Analysis of Losses in a Bufferless Transmission Link," in *Proceedings of the International Teletraffic Congress (ITC'20)*, Ottawa, Canada, June 2007, pp. 913–924.

[133] S. Y. Novak, "On Self-Normalized Sums," *Mathematical Methods of Statistics*, vol. 9, no. 4, pp. 415–436, 2000.

[134] D. P. Heyman, "The GBAR Source Model for VBR Video Conferences," *IEEE/ACM Transactions on Networking*, vol. 5, no. 4, pp. 554–560, August 1997.

[135] A. Alheraish, "Autoregressive Video Conference Models," *International Journal of Network Management*, vol. 14, no. 5, pp. 329–337, September 2004.

[136] C. H. Liew, C. K. Kodikara, and A. M. Kondoz, "MPEG-encoded Variable Bit-rate Video Traffic Modeling," *IEE Proceedings Communications*, vol. 152, no. 5, pp. 749–756, October 2005.

[137] M. W. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," in *Proceedings of the ACM Sigcomm*, London, UK, September 1994.

[138] M. Dai, D. Loguinov, and H. Radha, "A Hybrid Wavelet Framework for Modeling VBR Video Traffic," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, vol. 5, Singapore, October 2004, pp. 3125–3128.

[139] B. Melamed and D. E. Pendarakis, "Modeling Full-Length VBR Video Using Markov-Renewal-Modulated TES Models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 600–611, June 1998.

[140] M. M. Krunz and A. M. Makowski, "Modeling Video Traffic Using $M/G/\infty$ Input Processes: A Compromise Between Markovian and LRD Models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 5, pp. 733–748, June 1998.

[141] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, June 1995.

[142] B. D. Choi, G. U. Hwang, Y. W. Jung, and H. Chung, "The Periodic Markov Modulated Bernoulli Process and its Application to MPEG Video Traffic," *Performance Evaluation*, vol. 32, no. 4, pp. 301–317, May 1998.

[143] C. Blondia and O. Casals, "Statistical Multiplexing of VBR Sources: A Matrix-analytic Approach," *Performance Evaluation*, vol. 16, no. 1-3, pp. 5–20, November 1992.

[144] Z. G. Li, L. Xiao, C. Zhu, and P. Feng, "A Novel Rate Control Scheme for Video over the Internet," in *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '02)*, Orlando, USA, May 2002, pp. 2065–2068.

[145] V. J. Ribeiro, R. H. Riedi, M. S. Crouse, and R. G. Baraniuk, "Multiscale Queuing Analysis of Long-Range Dependent Network Traffic," in *Proceedings of the IEEE Infocom*, March 2000.

[146] S. Ma and C. Ji, "Modeling Heterogeneous Network Traffic in Wavelet Domain," *IEEE/ACM Transactions on Networking*, vol. 9, no. 5, pp. 634–649, October 2001.

[147] A. Lombardo, G. Schembra, and G. Morabito, "Traffic Specifications for the Transmission of Stored MPEG Video on the Internet," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 5–17, March 2001.

[148] Z. Avramova, D. De Vleeschauwer, K. Laevens, S. Wittevrongel, and H. Bruneel, "Modelling H.264/AVC VBR Video Traffic: Comparison of a Markov and a Self-Similar Source Model," *Telecommunication Systems*, vol. 39, no. 2, pp. 91–102, October 2008.

[149] Y. K. Belyaev and V. P. Nosko, "Characteristics of Excursions Above a High Level for a Gaussian Process and Its Envelope," *Theory of Probability and Its Applications*, vol. 14, no. 2, pp. 296–309, January 1969.

[150] A. Genz, "Numerical Computation of Multivariate Normal Probabilites," *Journal of Computational and Graphical Statistics*, vol. 1, no. 2, pp. 141–149, June 1992.

[151] I. F. Blake and W. C. Lindsey, "Level-Crossing Problems for Random Processes," *IEEE Transactions on Information Theory*, vol. 19, no. 3, pp. 295–315, May 1973.

[152] T. Mimaki and T. Munakata, "Experimental Results on the Level-Crossing Intervals of Gaussian Processes," *IEEE Transactions on Information Theory*, vol. 24, no. 4, pp. 515–519, July 1978.

[153] D. R. Morgan, "On Level-Crossing Excursions of Gaussian Low-Pass Random Processes," *IEEE Transactions on Signal Processing*, vol. 55, no. 7, pp. 3623–3632, July 2007.

[154] S. F. Arnold, *The Theory of Linear Models and Multivariate Analysis*. Wiley, 1981.

[155] A. Genz and F. Bretz, *The mvtnorm Package*, July 2007. [Online]. Available: http://cran.r-project.org/doc/packages/mvtnorm.pdf

[156] R Development Core Team, "R project." [Online]. Available: www.r-project.org

[157] [Online]. Available: http://www.math.wsu.edu/faculty/genz/homepage

[158] P. Abrahamsen, "A Review of Gaussian Random Fields and Correlation Functions," Norwegian Computation Center, Technical report 917, 1997.

[159] C. E. Rasmussen and C. K. I. Williamson, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[160] L. Alcuri, G. Barbera, and G. D'Acquisto, "Service Curve Estimation by Measurement: An Input Output Analysis of a Softswitch Model," in *Proceedings of the 3rd International Workshop on QoS in Multiservice Networks*, Catania, Italy, February 2005, pp. 49–60.

[161] C.-J. Chang and A. A. Nilsson, "Queuing Networks Modelling for a Packet Router Architecture using the DTM Technology," in *Proceedings of the IEEE International Conference on Communications*, vol. 1, New Orleans, USA, June 2000, pp. 186–191.

[162] L.-S. Peh and W. J. Dally, "A Delay Model for Router Microarchitectures," *IEEE Micro*, vol. 21, no. 1, pp. 26–34, January-February 2001.

[163] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, "Bridging Router Performance and Queuing Theory," in *Proceedings of the ACM Sigmetrics - Performance*, New York, USA, June 2004.

[164] A. Kuzmanovic and E. W. Knightly, "Measuring Service in Multi-Class Networks," in *Proceedings of the IEEE Infocom*, vol. 3, Anchorage, USA, April 2001, pp. 1281–1289.

[165] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of Measured Single-Hop Delay from an Operational Backbone Network," in *Proceedings of the IEEE Infocom*, vol. 2, New York, USA, June 2002, pp. 535–544.

[166] "Endace," http://www.endace.com.

[167] "Tcpreplay," http://tcpreplay.synfin.net/trac/.