Bayerische Julius-Maximilians-Universität
Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Prof. Dr. P. Tran-Gia

Traffic Modeling of Variable Bit Rate MPEG Video and its Impacts on ATM Networks

Oliver Rose

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 02/97

Würzburger Beiträge zur Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Lehrstuhl für Verteilte Systeme
Am Hubland
D-97074 Würzburg

Tel.: +49-931-888-5510
Fax.: +49-931-888-4601
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.
Gesetzt in LaTeX Computer Modern 9pt.

ISSN 1432-8801

Traffic Modeling of Variable Bit Rate MPEG Video and its Impacts on ATM Networks

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius–Maximilians–Universität Würzburg

vorgelegt von

Oliver Rose

aus

Schweinfurt

Würzburg 1997

Eingereicht am: 22.11.96
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr.-Ing. P. Tran-Gia
2. Gutachter: Prof. Dr.-Ing. Dr. h.c. P. J. Kühn
Tag der mündlichen Prüfung: 03.03.97

Danksagung

Bei aller Relevanz audiovisueller Medien, wie beispielsweise der in der vorliegenden Arbeit untersuchten, ist mir nach wie vor die unmittelbare Kommunikation mit meinen Mitmenschen die liebste.

In dieser Hinsicht schätze ich das kommunikationsfreundliche Klima an unserem Lehrstuhl. Großen Anteil trägt daran der Leiter des Lehrstuhls Prof. Phuoc Tran-Gia mit seinem stetigen Bemühen um fachliche und menschliche Impulse. Dadurch und durch sein Engagement, Kontakte mit Forschung und Industrie herzustellen, schafft er uns jungen Mitarbeitern die Möglichkeit, erfolgreich auf der Höhe der Zeit zu forschen. Ich danke ihm insbesondere dafür, daß ich während meiner Promotion zahlreiche internationale Konferenzen und Fachkollegen im In- und Ausland besuchen durfte. Nur intensiver Kontakt mit anderen Wissenschaftlern meines Gebietes ermöglichte es mir als jungem Forscher, meine eigene Arbeit richtig einschätzen zu lernen.

An dieser Stelle möchte ich sowohl meinem Doktorvater als auch dem Zweitgutachter Prof. Paul J. Kühn für die Anregungen sowie die fachlichen und hochschulpolitischen Diskussionen danken.

Einen wertvollen Beitrag zum Gelingen der Arbeit leisteten meine lieben Kollegen, sei es durch kontroverse Diskussion dissertationsrelevanter Themen, aber auch durch die zahlreichen, eher menschlichen Themen, die, durch erhebliche Anzahlen von Tassen Espresso flankiert, der eingehenden Erörterung bedurften. Ein herzliches Dankeschön an Rainer Dittmann, Dr. Thomas Fritsch, Notker Gerlich, Dr. Hermann Gold, Dr. Frank Hübner, Kenji Leibnitz, Dr. Manfred Mittler, Rainer Müller, Michael Ritter, Dr. Alexander Schömig, Dr. Thomas Stock, Kurt Tutschku und Norbert Vicari.

Einen erheblichen Anteil zum angenehmen Lehrstuhlklima trägt unsere Sekretärin Gisela Alt bei. Sie befreit uns dankenswerterweise von der Last der Verwaltungsarbeit und des Büromaterialbestellwesens.

Von großer Bedeutung für die Anfertigung der Arbeit waren auch die studentischen Hilfskräfte sowie die von mir betreuten Diplomanten Calin Barbu, Franz Busch, Egon Götz, Martin Kugler, Peter Och, Ulrike Schullerus und Horst Zölzer. Ohne ihr eifriges Mitwirken wäre die Dissertation weniger umfangreich und weniger tiefschürfend geworden.

Neben den vielen Menschen, mit denen ich am Lehrstuhl täglich Umgang pflegte und pflege, sollen auch diejenigen nicht unerwähnt bleiben, bei denen dies aufgrund der Entfernung leider nicht möglich ist. Ein großer Dank gebührt meinem australischen Kollegen Dr. Michael Frater, den ich schon zu Beginn meiner Promotion kennen- und schätzen lernte. Ihm verdanke ich viele Diskussionen über fachliche Belange, aber auch über die angelsächsische Lebensart und Sprache. Die Kooperation erwies sich als so fruchtbar, daß aus ihr einige gemeinsame Veröffentlichungen erwuchsen. Ein besonderes Erlebnis war auch die Zusammenarbeit mit europäischen Kollegen im Rahmen des COST-242-Projektes, nicht nur wissenschaftlich durch solch verkehrstheoretisches Urgestein wie Dr. James Roberts oder Prof. Jorma Virtamo, sondern auch, weil ich erkennen durfte, daß trotz der vielen Verschiedenheiten, wir doch alle echte Europäer sind.

Ein besonderer Dank gebührt meinen Eltern Renate und Gerhard Gropp dafür, daß sie mir das Studieren ermöglicht und mich immer in meinem eingeschlagenen Weg bestärkt haben.

Tiefsten Dank empfinde ich für meine Frau Bonnie. Sie hat mich stets unterstützt, auch wenn ich in den letzten Jahren immer wieder viel Zeit in meinem Büro an der Uni oder auf Reisen und eben nicht bei der Familie verbringen konnte. Dies wiegt umso schwerer, da während meiner Promotionszeit unsere beiden Kinder Jolanda und Niklas das Licht der Welt erblickten. Für die Zukunft wünsche ich mir, daß ich stets das rechte Maß zwischen Wissenschaft und Familie, die mir beide sehr am Herzen liegen, finden möge.

Würzburg, im März 1997 *Oliver Rose*

Contents

1 Introduction

Modeling is finding a representation of reality in a scale we can handle. Modern telecommunication systems are among the most complex technical contributions to our reality. Thus, tremendous amounts of modeling work was carried out and still has to be carried out for the development of the most recent telecommunications technology, the Asynchronous Transfer Mode (ATM). Most of the publications about modeling, however, are primarily concerned with the modeling of the technical system itself since the objective of research was to make it work. Currently, the viewpoint is changing from the pure technical system towards the services that are to be provided by this system and the kinds of traffic that will be carried by these services. In the early stages of ATM research and development, simple traffic models were adequate since the ideas about the traffic were rather vague. For instance, nobody thought about internet or multimedia traffic. Now, as user needs become visible, there is a demand for more accurate traffic modeling. It has to be evaluated under which conditions an ATM-based network is able to carry this traffic while meeting the user's quality requirements.

In this monograph, we devote our attention to the most complex part of multimedia traffic, the transmission of video sequences. Since uncompressed video sequences require a bandwidth that even ATM networks cannot provide for a larger number of connections, video compression standards were developed. Among these standards, the ISO Moving Pictures Expert Group

(MPEG) standard is favored for video transmission over ATM networks. There are two bit rate modes that can be used for compression, constant bit rate (CBR) or variable bit rate (VBR). We focus on VBR MPEG video since it is attractive for users and network providers. Compared to CBR video, VBR video provides a better quality for the same average bandwidth used. Assuming the same video quality, more VBR than CBR connections can be transmitted over a given ATM link. The disadvantage of VBR video is the problem of controling such traffic sources.

Based on traffic models for VBR MPEG video, we intend to identify traffic properties that may lead to problems in an ATM network and to support their solution. The traffic characteristics are determined by the MPEG coding technique and the content of the uncompressed video sequence. Thus, a detailed statistical analysis of several MPEG encoded video traces has to be the first step of the model development. Based on these results, a variety of modeling approaches, ranging from simple histograms to more esoteric self-similar processes, are tailored to match the statistical properties as good as possible. To simplify the validation of the models, logical layers of the video are introduced, such as single frames, groups of frames, scenes, etc. Finally, the models are adapted to the transmission protocols of ATM networks. The statistical properties of the models are compared to those of the measured data sets, and the models are verified by using them as performance predictors. For instance, cell loss rates or cell delays at a multiplexer buffer are determined.

Despite the complex nature of video traffic, it is our intention to keep the models as simple as possible, i.e., in a scale we can handle by both simulation and analysis. Two examples illustrate how the models can be applied in performance analyses.

1.1 MPEG

There are several standards available for the compression of digitized video sequences (see Aravind et al. (1993)). However, only the MPEG standards are currently regarded as being state-of-the-art and future-proof. MPEG is an international standard (see ISO (1993), Le Gall (1991)) for the compression of digital audio and video for storage or transmission on various digital media, including compact discs, remote video databases, movies on demand, cable television, etc. Originally, MPEG was the name of the group of people working on the standard (Moving Pictures Expert Group), but soon it became the name of the standard itself. There are several phases of the MPEG standard. We are focusing on MPEG-1 and MPEG-2 which are considered for *Broadband Integrated Services Digital Network* (B-ISDN) video services. Currently, MPEG-4 is under development, which is intended for video transmission over very low bandwidth channels.

1.1.1 MPEG-1

The MPEG-1 video compression standard, primarily aimed at coding video for digital storage media at rates of 1 to 1.5 Mbps, is well suited for a wide range of applications at a variety of bit rates. The standard mandates real-time decoding and supports features to facilitate interactivity with the stored bit stream. It only specifies the syntax for the bit stream and the decoding process. Although the intention was to design the standard for digital storage media, such as CD-ROMs, the group's goal, however, has been to develop a generic standard that can also be used in other digital video applications, such as in telecommunications. The MPEG standard has three parts:

- ◇ Part 1 describes the synchronization and multiplexing of video and audio,
- ◇ Part 2 describes video,

◇ Part 3 describes audio.

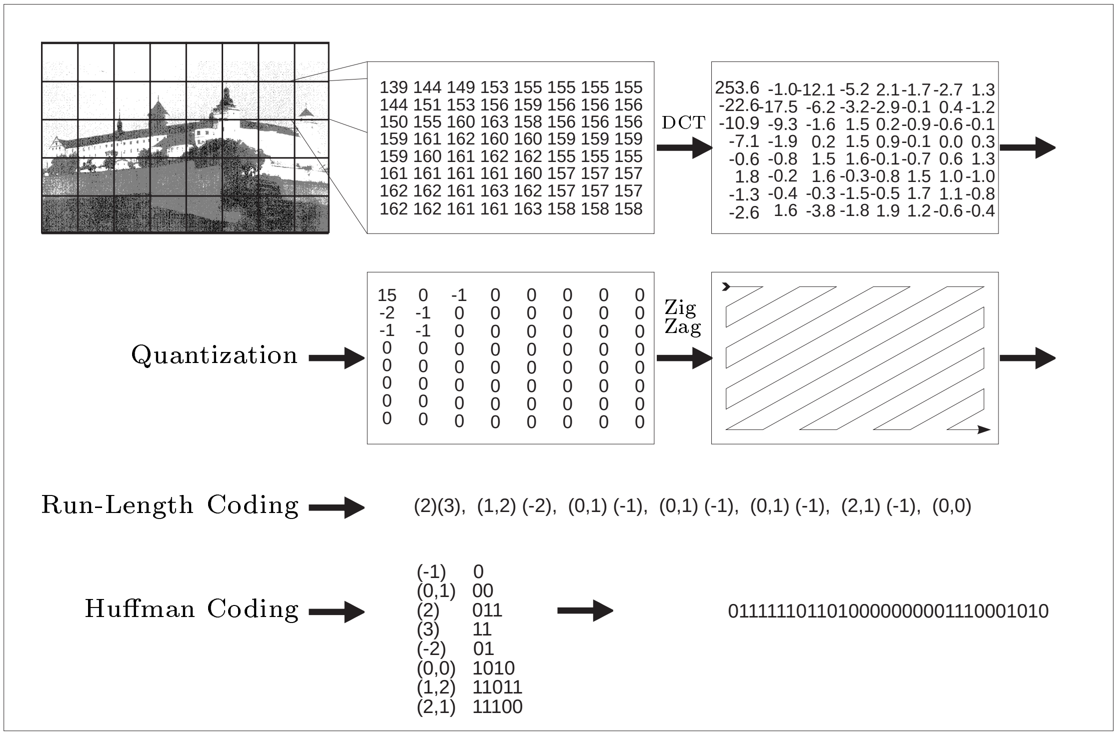
In the following, we will focus on the video part of the MPEG standard. Uncompressed digital video requires an extremely high transmission bandwidth. For instance, digitized video in North American Television Standards Committee (NTSC) resolution has a bit rate of about 100 Mbps. With digital video, compression is necessary to reduce the bit rate to suit most applications. The required degree of compression is achieved by exploiting the spatial and temporal redundancy present in a video signal. However, the compression process is inherently lossy, and the signal reconstructed from the compressed bit stream is not identical to the input video signal. Compression typically introduces artifacts into the decoded signal.

The primary requirement of the MPEG video standard is that it should achieve the highest possible quality of the decoded video at a given bit rate. In addition, most applications require some degree of resilience to bit errors. Furthermore, a variety of video formats should be supported.

Compression algorithm overview

The compression approach of MPEG video uses a combination of the ISO JPEG still image compression standard (ISO (1991)) and the CCITT H.261 video conferencing standard (CCITT (1990)). Since video is a sequence of still images, it is possible to compress or encode a video signal using techniques similar to JPEG. Such methods of compression are called *intraframe* coding techniques, where each frame of video is individually and independently compressed or encoded. Intraframe coding exploits the spatial redundancy that exists between adjacent pixels of a frame.

As in JPEG and H.261, the MPEG video-coding algorithm employs a block-based two-dimensional Discrete Cosine Transform (DCT). A frame is first divided into 8×8 blocks of pixels, and the two-dimensional DCT is then applied independently to each block. This operation results in an 8×8 block of DCT coefficients in which most of the energy in the original block is typically concentrated in a few low-frequency coefficients. A quantizer is applied to each DCT coefficient that sets many of them to zero. This quantization is responsible for the lossy nature of the compression algorithm. Compression is achieved by transmitting only the coefficients that survive the quantization operation, and by entropy coding (Huffman coding) their runs and amplitudes. Figure 1.1 shows how the bit stream for one block of a still image of Würzburg's castle is generated.



139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

253.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(2)(3), (1,2) (-2), (0,1) (-1), (0,1) (-1), (0,1) (-1), (2,1) (-1), (0,0)

(-1)	0
(0,1)	00
(2)	011
(3)	11
(-2)	01
(0,0)	1010
(1,2)	11011
(2,1)	11100

0111111011010000000001110001010

Figure 1.1: *Intraframe coding*

The quality, however, achieved by intraframe coding alone is not sufficient for typical video signals at bit rates around 1.5 Mbps. Therefore, *interframe* coding techniques are used to reduce the temporal redundancy which results from a high degree of correlation between adjacent frames. The H.261 algorithm exploits this redundancy by computing a frame-to-frame difference signal called the prediction error. In computing the prediction error, the technique of motion compensation is employed to correct for motion. A block-based approach is adopted for motion compensation, where a block of pixels, called the target block, in the frame to be encoded is matched with a set of blocks of the same size in the previous frame, called the reference frame. The block in the reference frame that best matches the target block is used as the prediction for the latter, i.e., the prediction error is computed as the difference between the target block and the best-matching block. This best-matching block is associated with a motion vector that describes the displacement between it and the target block. The motion vector information is also encoded and transmitted along with the prediction error. The prediction error itself is transmitted using the DCT-based intraframe encoding technique summarized above. In MPEG video, the block size for motion compensation is chosen to be 16×16 pixels, representing a reasonable tradeoff between the compression provided by motion compensation and the cost associated with transmitting the motion vectors.

Bidirectional temporal prediction, also called motion compensated interpolation, is a key feature of MPEG video. In bidirectional prediction, some of the video frames are encoded using two reference frames, one in the past and one in the future. A block in those frames can be predicted by another block from the past reference frame (forward prediction), or from the future reference frame (backward prediction), or by the average of two blocks, one from each reference block (interpolation). In any case, the block from the reference frame is associated with a motion vector, so that two motion vectors are used with interpolation. Frames that are bidirectionally predicted are never used as reference frames.

Bidirectional prediction provides a number of advantages. The primary one is that the compression obtained is typically higher than that obtained from forward prediction. To obtain the same picture quality, bidirectionally predicted frames can be encoded with fewer bits than frames using only forward prediction. However, bidirectional prediction introduces extra delay in the encoding process since frames must be encoded out of sequence. Further, it entails extra encoding complexity since block matching has to be performed twice for each target block.

MPEG bit stream syntax layers

The bit stream syntax should be flexible to support a variety of applications. To this end, the overall syntax is constructed in several layers, each performing a different logical function. The outermost layer is called the video *sequence layer*, which contains basic parameters such as the size of the video frames, the frame rate, the bit rate, and certain other global parameters. A wide range of values is supported for all these parameters.

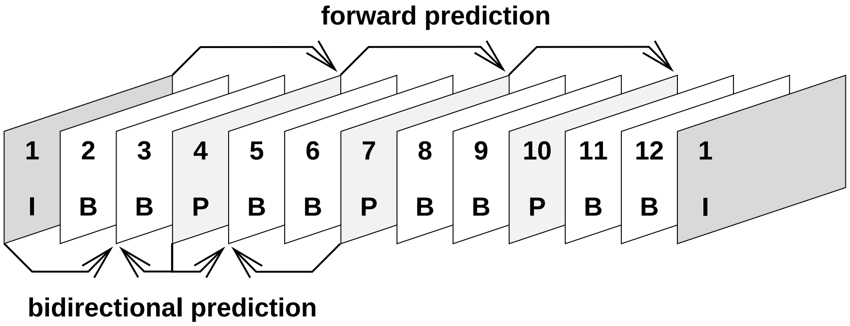


Figure 1.2: *Group of Pictures (GOP) pattern*

Inside the video sequence layer is the *Group of Pictures (GOP) layer*, which provides support for random access, fast search, and editing. A sequence is divided into a series of GOPs, where each GOP contains an intra-coded frame (I-frame) followed by an arrangement of (forward) predictive-coded frames (P-frames) and bidirectionally predicted, interpolative-coded frames (B-frames). Figure 1.2 shows the GOP pattern which we used for the encoding of our sequences and which was also used by Garrett and Willinger (1994) to encode the MPEG-1 version of the Bellcore *Star Wars* data set. The MPEG video standard allows GOPs to be of arbitrary structure and length.

The bits produced by encoding a single frame of a GOP constitute the *picture layer*. The picture layer first contains information on the type of frame that is present (I, P, or B), and the position of the frame in display order. The bits corresponding to the motion vectors and the DCT coefficients are packaged in the *slice layer*, the *macroblock layer*, and the *block layer*. Here, the block is the 8×8 DCT unit, the macroblock the 16×16 motion compensation unit, and the slice is a string of macroblocks of arbitrary length running from left to right and top to bottom across the frame. The slice layer is intended to be used for resynchronization during the decoding of a frame in the event of bit errors. Prediction registers used in the differential encoding of motion vectors are reset at the start of a slice. It is in the responsibility of the encoder to choose the appropriate length of each slice. In the macroblock layer, the motion vector bits for a macroblock are followed by the block layer, which consists of the bits for the DCT coefficients of the 8×8 blocks in the macroblock. Table 1.1 illustrates the different layers and their use.

Table 1.1: *Layers of the MPEG video bit stream syntax*

Syntax layer	Functionality
Sequence	Context
GOP	Random access; video coding
Picture	Primary coding
Slice	Resynchronization
Macroblock	Motion compensation
Block	DCT

1.1.2 MPEG-2

MPEG-1 focused on coding of single-layer (non-scalable) progressive (non-interlaced) video. The MPEG-2 standard (ISO (1994)) addresses issues of improved functionality by using scalable video coding. This means that the compressed video is separated into a base layer bit stream which contains the most important picture information and one or more enhancement layers which contain information to improve the base layer picture quality. The picture quality scaling can be achieved in the spatial or in the frequency domain. It is also possible to perform a scaling with respect to different resolutions or picture rates. In addition, the MPEG-2 standard facilitates interlaced video and offers a larger variety of motion-compensated predictions as well as improved DCT coding and quantization techniques.

1.2 ATM

In 1988, the *Asynchronous Transfer Mode (ATM)* was selected by the CCITT (now ITU-T) as the basic transmission technique for B-ISDNs. Since then, work has been continuing on the specification of the details of ATM itself, and on how ATM interfaces to different B-ISDN services. In this section, we will highlight the features and principles of ATM which are important for the modeling of VBR video traffic transmitted over ATM-based telecommunication networks.

1.2.1 Objectives

Wright (1993) lists the following main objectives that can be met using ATM transport technology:

- ◇ provide low- and high-bandwidth services,
- ◇ provide high-bandwidth transport,
- ◇ provide a single network for all services,
- ◇ provide local/wide area network integration,

- free users of bandwidth granularity,
- allow dynamically changing bandwidth,
- be future-proof.

Although these objectives cover many aspects of network services and operation, many of them can be achieved by a key feature of the ATM protocol, the use of short, fixed-length data packets called *cells*. All user and network information, whether it be voice, data, or video, is transported using the same cell format. The main advantages of short, fixed-length cells compared to variable-length packets is simplified switch design and cell processing which is necessary to achieve bandwidths in the order of hundreds or thousands of Mbits per second.

1.2.2 B-ISDN service classes

ATM provides a single mode of transportation for all telecommunication services based on cells with a 48-octet payload and a 5-octet header. However, user information comes in a variety of forms, such as voice packets, internet protocol (IP) packets, or video streams. In order to specify how this information is converted into ATM cell format, it is necessary to classify user service requirements (CCITT (1991)):

- *Class A*. Constant bit rate (CBR) service with end-to-end timing, connection oriented,
- *Class B*. Variable bit rate (VBR) service with end-to-end timing, connection oriented,
- *Class C*. Variable bit rate (VBR) service with no timing required, connection oriented,
- *Class D*. Variable bit rate (VBR) service with no timing required, connection-less.

Thus, VBR MPEG video traffic is a typical class-B service. In current ATM Forum documents, this service category is referred to as real-time variable bit rate (rt-VBR) (see ATM Forum Technical Committee (1996b)).

1.2.3 ATM-based B-ISDN protocol stack

For very detailed, i.e. cell-oriented models, it is necessary to be aware of the data-flow from the video traffic source to the transport medium which is determined by a layered set of protocols. Figure 1.3 shows a simplified version of the ATM-based B-ISDN protocol stack (see McDysan and Spohn (1994)).

<table>
<tr><td colspan="2">Higher layers</td></tr>
<tr><td rowspan="2">AAL</td><td>Convergence sublayer</td></tr>
<tr><td>Segmentation and reassembly sublayer</td></tr>
<tr><td colspan="2">ATM layer</td></tr>
<tr><td colspan="2">Physical layer</td></tr>
</table>

Figure 1.3: *Simplified ATM protocol stack*

The physical layer is responsible for the transmission of the bit stream over a given medium. The ATM layer handles multiplexing, switching, and control actions based upon information of the cell header. It passes cells to and accepts cells from the *ATM Adaption Layer (AAL)*. The AAL passes Protocol Data Units (PDU) which are generally larger than the payload of an ATM cell to the higher layers or accepts such PDUs from higher layers. In our case the PDUs will consist of compressed video images (frames) or parts of them (slices) and some protocol overhead. The AAL determines the primitives that can be used for transmission (Convergence sublayer) and the way cells are formed from the PDUs and vice versa (Segmentation and reassembly sublayer). At the moment, there are four AAL types to support the four service classes mentioned above. In the case of class-B traffic, AAL2 provides the appropriate functionality. Currently, the AAL2 standard is not yet well defined. We therefore assume that at least the following alternatives can be realized.

- *Bursty transmission.* Upon arrival of a PDU, the PDU is segmented into cells and transmitted at maximum physical layer speed as a burst of back-to-back cells.
- *Smoothed transmission.* Since PDUs arrive at the AAL at a constant rate due to the constant frame rate of a video sequence, the time when the next PDU is going to arrive is known in advance. Therefore, the AAL is able to smooth the cell stream by emitting the cells uniformly during a PDU interarrival period.

1.2.4 The traffic contract

In ATM networks, the traffic management is based on the concept of a traffic contract. The traffic contract exists for each connection and is an agreement between a user and a network across a User-Network Interface (UNI) with regard to the following aspects (see McDysan and Spohn (1994)):

- the Quality of Service (QoS) that a network is expected to provide,
- the traffic parameters that specify characteristics of the cell flow,
- the network's definition of a compliant connection.

The QoS is defined by specific parameters for cells that are conforming to the traffic contract. The following QoS parameters are negotiated according to the ATM Forum Traffic Management Specification Version 4.0 (April 1996):

- *Maximum Cell Transfer Delay* (maxCTD). It is specified as the (1-α) quantile of the Cell Transfer Delay (CTD).

- ◇ *Peak-to-peak Cell Delay Variation* (peak-to-peak CDV). It is defined as maxCTD minus the fixed CTD that could be experienced by any delivered cell on a connection during the entire connection holding time.

- ◇ *Cell Loss Ratio* (CLR). It is defined as the ratio of lost cells and total transmitted cells of a connection.

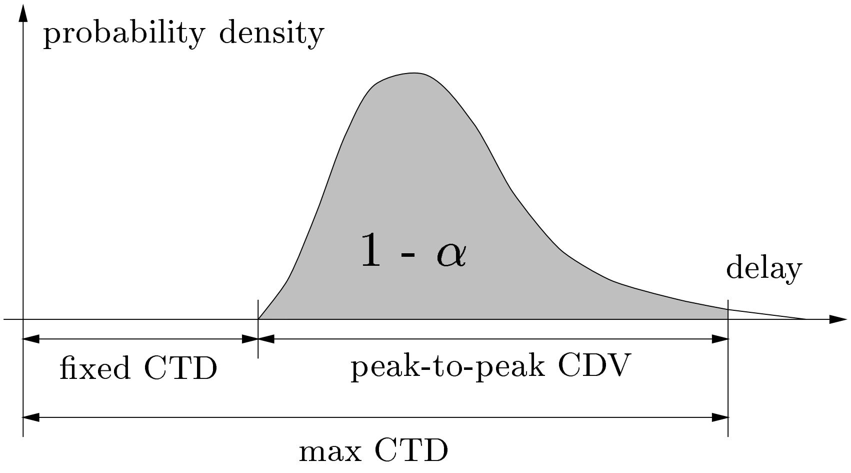


Figure 1.4: *QoS parameters*

Figure 1.4 illustrates the CTD-based parameters. The traffic parameters form a traffic descriptor which captures intrinsic source traffic characteristics. The following key traffic parameters are considered:

- ◇ *Peak Cell Rate* (PCR) = $1/T$ in units of cells per second, where T is the minimum intercell spacing in seconds (i.e. the time interval from the first bit of a cell to the first bit of the next cell).

- ◇ *Cell Delay Variation Tolerance* (CDVT) = τ in seconds. This traffic parameter normally cannot be specified by the user, but is set by the network instead. The number of cells that can be sent back-to-back at the access line rate is $\lfloor \tau/T \rfloor + 1$, where $\lfloor x \rfloor$ denotes the largest integer number smaller than x.

- ◇ *Sustainable Cell Rate* (SCR) $= 1/T_S$ is an upper bound on the average rate of a bursty traffic source.
- ◇ *Maximum Burst Size* (MBS) is the maximum number of cells that can be sent as a burst at peak rate.

The *Generic Cell Rate Algorithm (GCRA)* is used to define the conformance of cells with respect to the traffic contract. For each cell arrival, the GCRA determines whether the cell conforms to the traffic contract of the connection. There are two versions of the GCRA, namely the *Virtual Scheduling Algorithm* and the *Continuous-State Leaky Bucket Algorithm*, which are equivalent in the sense that both versions declare the same cells of a cell stream as conforming or non-conforming. In this monograph, we refer to the Virtual Scheduling Algorithm which is depicted in Figure 1.5. This algorithm was proposed first by the ITU-T Draft Recommendation I.371 (1994) to monitor the PCR.

In general, the GCRA uses a *Theoretical Arrival Time* (TAT) for the earliest time instant the next cell is expected to arrive. The TAT is initialized with the arrival time of the first cell of the connection $t_a(1)$.

For PCR enforcement, cells should be spaced by $I = T$ (the increment of the GCRA), but due to CDV a tolerance with limit $L = \tau$ is employed. If cell number k arrives later than expected, the TAT for the next cell is given by the actual arrival time plus the increment. If cell number k arrives before its TAT but not before $TAT - L$, then the TAT for the next cell is derived by incrementing the TAT for cell number k by I. Contrarily, the TAT is not changed and the cell is declared as non-conforming if it arrives earlier than $TAT - L$.

For the enforcement of the SCR, the increment parameter I is set to T_s for a SCR $1/T_s$. The limit parameter $L = \tau_s$ is called Burst Tolerance (BT)

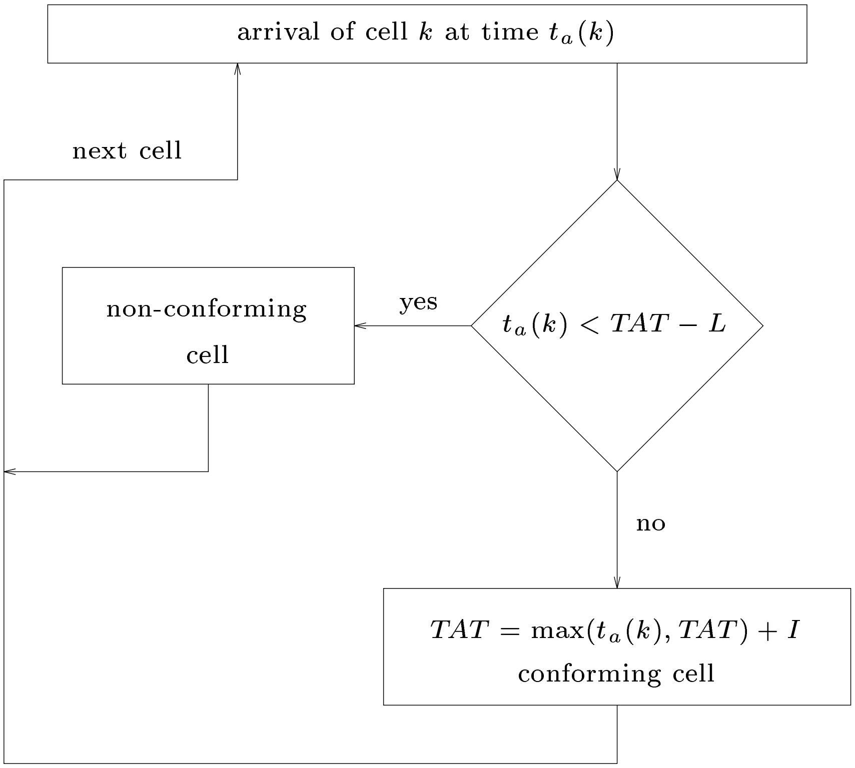


Figure 1.5: *GCRA(I,L) as Virtual Scheduling Algorithm.*

and corresponds to the MBS that can be transmitted at PCR by $MBS = \lfloor 1 + \tau_s/(T_s - T) \rfloor$.

The conformance of cells of a connection at an interface is defined in relation to the conformance algorithm, here the GCRA, and corresponding parameters specified in the connection traffic descriptor. Since we do not consider ATM cells with priorities in this monograph, we assume the CLP (Cell Loss Priority)-transparent cell flow model, i.e. the network generally disregards the CLP bit in the cell header.

1.2.5 Traffic and congestion control

In an ATM network, a balance is necessary between the efficiency achieved by allowing the traffic from different users to share the network resources and not allowing bursts in one user's traffic to cause congestion that impacts the QoS of another user. This goal can be achieved by the following means:

- ◇ *Preventive control.* Connection Admission Control (CAC) and Usage Parameter Control (UPC).
- ◇ *Reactive control.* Rerouting connections and cell discarding.

The CAC function is defined as the set of actions taken by the network at connection establishment in order to determine whether a connection can be progressed or should be rejected. This decision is based on the traffic descriptors of all existing connections, the traffic parameters of the new connection and on the QoS requirements of all connections.

The UPC is defined as the set of actions taken by the network to monitor and control traffic. Its main purpose is to protect network resources from malicious as well as unintentional misbehavior which can affect the QoS of other already established connections by detecting violations of negotiated parameters and taking appropriate actions. If the PCR $1/T$ shall be monitored at the UNI, the CDV which is introduced between the PHY SAP (Physical Layer Service Access Point) and the UNI must be tolerated using the tolerance limit τ. Thus, the PCR of an ATM connection can be monitored at the UNI using GCRA(T, τ). The SCR $1/T_s$ can be monitored at the UNI by employing a BT of $\tau_s + \tau$, i.e. with GCRA$(T_s, \tau_s + \tau)$. The choice of the BT as $\tau_s + \tau$ is motivated by the observation that a cell stream which complies with GCRA(T_s, τ_s) at the PHY SAP complies with GCRA$(T_s, \tau_s + \tau)$ at the UNI if τ is sufficient to tolerate the CDV introduced. Cells which are identified as non-conforming by the GCRA can either be discarded or optionally be tagged to be discarded in case of network congestion.

In general, reactive control is a problem not only in ATM but all high-speed networks since, if congestion is discovered somewhere in the network, large amounts of traffic are already in transit on the transmission facilities. Another problem is the limited knowledge of the network operator about the traffic emitted by a source due to the large variety of traffic characteristics.

1.3 MPEG video traffic over ATM networks

Concerning the transmission of MPEG video traffic over packet networks, ATM-based networks in our case, there are a lot of open questions with respect to both coding and telecommunication aspects (see Zhang et al. (1991)).

Since the MPEG standards suite only defines the syntax of a bit stream which a standard decoder must be able to decompress, there is a huge variety of different encoding parameter sets and modes. We will focus only on the aspects relevant to transmitting the bit stream over an ATM network.

There are two bit rate modes at the encoder output:

- ◇ *Constant Bit Rate (CBR).* The output bit rate of the encoder is held constant by means of a feedback loop control. As soon as the output buffer exceeds a given limit, the coding quality is reduced to decrease the number of bits per frame. If the number of bits per frame is too small stuffing bits are used to increase the amount of data.

- ◇ *Variable Bit Rate (VBR).* The output bit rate is variable, but the quality of the video is held approximately constant.

From the point of view of the network provider, CBR video has several advantages. Due to the known cell rate, CAC is very simple. During the holding time of the connection only this cell rate has to be controlled, i.e. only PCR monitoring takes place. Therefore UPC of such a CBR source is simple, too. For instance, the ATM Forum Technical Committee (1996a) selected constant packet rate MPEG-2 encoding in their Video on Demand Specification 1.0 .

For VBR video there are some problems. The definition of an effective bandwidth of a VBR video stream which is needed for CAC is difficult, since the statistical properties of video streams can be very different depending on the coding scheme and the content of the video sequence. Thus, it will be hard to find a small set of parameters to calculate the effective bandwidth of this type of video streams. In a close relationship to this problem is the UPC problem. The selection of the parameters of a VBR video stream to be controlled, and techniques for implementing this strategy are open questions.

Digital video has a number of properties that lead to QoS requirements that differ from other services. Among these QoS requirements are:

- ◇ *Cell loss.* The compression of digital video removes a large amount of the redundancy present in the video images. By doing this, it increases the impact of cell loss on the QoS. At present, it is not clear what cell loss probability will be tolerable. This will depend on a number of factors, including the sensitivity of the human visual system to different types of degradation. The effect of cell loss can be reduced by coders that insert particular redundant information into their output bit streams to assist the decoder to minimize the effect of cell loss.

- ◇ *Cell delay.* Due to the coding and decoding there is always some delay even when there is no media access, buffering and transmission delay. The delay requirements depend on the video service. For interactive services like video conferencing and video telephony, there should be as little delay as possible. For distribution services like video on demand and TV broadcasting, the delay usually does not constitute a problem, since the user is not able to notice it. The consequence of these requirements is that no traffic shaping can be done for interactive services, since traffic shaping always produces delay due to buffering cells. Again, the level of delay tolerable in interactive services is not precisely known. One problem will be to give a reliable statement about the maximum delay introduced by the network.

It is easier to guarantee characteristics of cell loss and delay for CBR services. However, for a given network capacity, it is possible to achieve higher quality in the decoded video using VBR compression compared to CBR. Hence, there is good reason to search for techniques for managing networks carrying VBR traffic. Possible algorithms may contain "constrained variability", either by sophisticated loop-back controls within the coder or by signaling schemes with the network, cells with different priorities (see Pancha and Zarki (1994)), or MPEG-2 multi-layer coding.

The above presentation of teletraffic engineering problems for MPEG video over ATM networks induces us to study the following scenario: a (single-layer) VBR MPEG-1 bit stream is transmitted over an ATM network without priorities. The encoding is performed based on a fixed GOP pattern. We chose MPEG-1 encoding since we intend to compare both our modeling and performance evaluations results with those of the current teletraffic literature which is mainly based on MPEG-1 data sets. Work on MPEG-2 or even MPEG-2 video data sets are not very widespread at the moment. The VBR mode was chosen due to the fact that CBR streams are not interesting for modeling and for CAC and UPC dimensioning problems. Single-layer encoding is no restriction with respect to our presented modeling approaches since the same models are applicable for each layer. Concerning the ATM transmission, we assume that the higher layer PDUs arrive in the form of single video frames at the AAL, where they are packetized into ATM cells. These cells are then transmitted equally spaced until the arrival of the next video frame. This "average per-frame bit rate" technique was proven to be the most efficient with regard to cell losses and delays (see Skelly et al. (1993), Enssle (1996)). As an alternative the PDU may consist of MPEG slices or macroblocks (see Pancha and Zarki (1994)). It should be noted, however, that differences in these approaches will only be noticeable in systems with very short buffers since they are filtered out by larger buffers.

2 Statistical analysis of MPEG video traces

The first steps in modeling a real world's stochastic process are always a thorough analysis of the technical system creating this process and of measurements of the process itself. Here, we are interested in the statistical properties of the output process of an MPEG video encoder. The structure of this output is determined by the coding parameters, such as number of slices, GOP pattern, and CBR or VBR mode. There is no known procedure to obtain the statistical parameters of the encoder output given the uncompressed video material and the coding parameters. Thus, the statistical properties of the output process can only be obtained by measurements.

From the multitude of coding parameter sets and encoder output measurements, we use the following. We focus on one-layer video data streams of MPEG-1 type. Most of the encoders will use this scheme and in case of MPEG-2 multi-layer encoding the statistical properties of the base layer are almost identical to this type of stream. We will only consider VBR encoded video sequences since CBR video is trivial from statistical analysis and modeling point of view. Concerning the encoder output, we focus on frame size measurements. Smaller items of encoded data, such as slices or macro blocks, might also be considered for statistical analysis and modeling. For instance, Chan and Leon-Garcia (1994) present an analysis of cell interarrival times for a variety of VBR video codecs. Most of the interesting performance issues, however, can be examined using frame size sequences.

First, we introduce the MPEG encoded sequences used for our statistical analysis. Simple statistics are provided such as moments and peak-to-mean ratios for both the frame sizes and the GOP sizes where the GOP size is the sum of frame sizes of one GOP. We fit model distributions to the frame and GOP size histograms and analyze the correlations of both the frames and GOPs. Finally, the long-range dependence or selfsimilar properties of the sequences are examined.

A further aspect of statistical analysis is the evaluation of the video sequences' scene length properties where a scene is defined as a sequence of consecutive frames without cuts, zooming, or panning. In this monograph, scene statistics are not considered since we do not have the technical equipment to browse through all the sequences frame by frame within a tolerable amount of time. This would be necessary to decide about the scene changes. In Section 3.3.3, a method is presented how "pseudo scenes" can be included in a model to improve its performance.

A detailed description of the techniques used for statistical inference can be found in Appendix A of this monograph.

2.1 Introduction of the video data sets

In general, long MPEG frame size traces of several thousand frames are not publically available. Therefore, traces from VHS video tapes had to be encoded at our institute. We chose several movies, TV sports events and TV shows, which were encoded at the Institute of Computer Science of the University of Würzburg using the UC Berkeley MPEG-1 software encoder (see Gong (1994)). Table 2.1 shows the sequences which we used to produce the data sets.

Table 2.1: *Overview of encoded sequences*

Movies (buy cassettes)	
dino	Jurassic Park
lambs	The Silence of the Lambs
TV sports events (recorded from cable TV)	
soccer	Soccer World Cup 1994 Final: Brazil - Italy
race	Formula 1 car race at Hockenheim/Germany 1994
atp	ATP Tennis Final 1994: Becker - Sampras
Other TV sequences (recorded from cable TV)	
terminator	Terminator 2
talk1	German talk show
talk2	Political discussion
simpsons	Cartoon
asterix	Cartoon
mr.bean	Three slapstick episodes
news	German news show
mtv	Music clips

All sequences mentioned below were encoded using the following parameter set:

- Frame rate: 25 frames per second;
- Each frame consists of one slice;
- GOP pattern: IBBPBBPBBPBB (12 frames);
- Quantizer scales: 10 (I), 14 (P), 18 (B);
- Motion vector search: logarithmic/simple; search window size: 10 half pels; reference frame: original;

◇ Encoder input: 384×288 pels in 4:2:0 color format;

◇ Number of frames per sequence: 40000 (about half an hour of video)

Due to hardware limitations, some parameters might not be optimal with respect to the quality of the MPEG video sequence. We used a Sun Sparc 20 for the image processing and encoding and captured the sequence from a VCR with a SunVideo SBus board.

2.2 Overview

Table 2.2 shows the compression rates, the mean values, Coefficients of Variation (CoV), and the peak-to-mean ratios of the frame and GOP sizes. The GOP size is defined as the sum of frame sizes of one GOP. For the sake of comparison we also present the statistical data of *starwars* as reported by Garrett and Willinger (1994). Note, that this frame size trace is from an MPEG encoded video with the same GOP pattern as our sequences. It should not be mixed up with Garrett's data set which consists of only a single frame type.

Unfortunately, even the statistical properties of the sequences of the same category, such as movies or cartoons, are not stable. For example, the measurements of *terminator* and *lambs* or of *simpsons* and *asterix* have no moments lying close together. This will lead to difficulties in finding traffic classes for MPEG video to be used for CAC.

In the remainder of this section, we mainly present the results of the statistical analysis of the *dino* and *starwars* sequence. Results for other sequences are shown if they are of particular interest.

Table 2.2: *Simple statistics of the encoded sequences*

Sequence	Compr.	**Frames**			**GOPs**		
	rate	Mean	CoV	Peak/	Mean	CoV	Peak/
	X : 1	[bits]		Mean	[bits]		Mean
asterix	119	22 348	0.90	6.6	268 282	0.47	4.0
atp	121	21 890	0.93	8.7	262 648	0.37	3.0
dino	203	13 078	1.13	9.1	156 928	0.40	4.0
lambs	363	7 312	1.53	18.4	87 634	0.60	5.3
mr.bean	150	17 647	1.17	13.0	211 368	0.50	4.1
mtv	134	19 780	1.08	12.7	237 378	0.70	6.1
news	173	15 358	1.27	12.4	184 299	0.47	6.0
race	86	30 749	0.69	6.6	369 060	0.38	3.6
simpsons	143	18 576	1.11	12.9	222 841	0.43	3.8
soccer	106	25 110	0.85	7.6	301 201	0.48	3.9
talk1	183	14 537	1.14	7.3	174 278	0.32	2.7
talk2	148	17 914	1.02	7.4	214 955	0.27	3.1
terminator	243	10 904	0.93	7.3	130 865	0.35	3.1
starwars	130	15 599	1.16	11.9	187 185	0.39	5.0

2.3 Distributions

Figures 2.1 and 2.2 show the frame size histograms of the I-, P-, and B-frames of the *dino* and the *starwars* sequence, respectively. The shapes of the curves indicate that the I-frames may be approximated by a normal probability density function, whereas the P- and B-frames have a histogram resembling a Gamma or a lognormal probability density function. Gamma or lognormal distributions are commonly suggested to model the frame and GOP sizes of MPEG video sequences (see Heyman et al. (1992), Pancha and Zarki (1992), Enssle (1994), Krunz et al. (1995)). Garrett and Willinger

(1994) report that for their data set a hybrid Gamma/Pareto probability density function provided the best approximation of the histogram.

For the probability density function fitting we use both Q-Q plots (see Appendix A.4) and visual tail comparisons. Considering the Q-Q plots, the frame and the GOP size histograms of almost all traces can be well approximated with lognormal probability density functions. Only for a small number of I-frame histograms, both normal and lognormal probability density functions lead to a reasonable approximation accuracy. For illustration, we show Q-Q plots for the *dino* I-frames (cf. Figure 2.3) and the *dino* GOPs (cf. Figure 2.4). The comparison of the right tails of the histograms and the model probability density functions shows that lognormal probability density functions approximate well the frame and GOP size histograms (cf. Figures 2.5 and 2.6).

To sum up, lognormal probability density functions are adequate model probability density functions for both frame and GOP sizes of VBR MPEG video sequences. Compared to other probability density functions the lognormal probability density function offers several advantages:

(1) The estimation of the parameters is simple compared to e.g. Gamma probability density functions, which are similar in shape.

(2) A process with a lognormal marginal distribution generates only non-negative samples. In contrast, normal (Gaussian) marginals lead to negative frame or GOP sizes of the model process. This behavior has to be corrected at the cost of increased model complexity.

(3) Processes with lognormal marginals can be generated from processes with normal marginals by using an exponential transform. This facilitates the application of models with Gaussian marginals such as autoregressive processes and several families of selfsimilar processes.

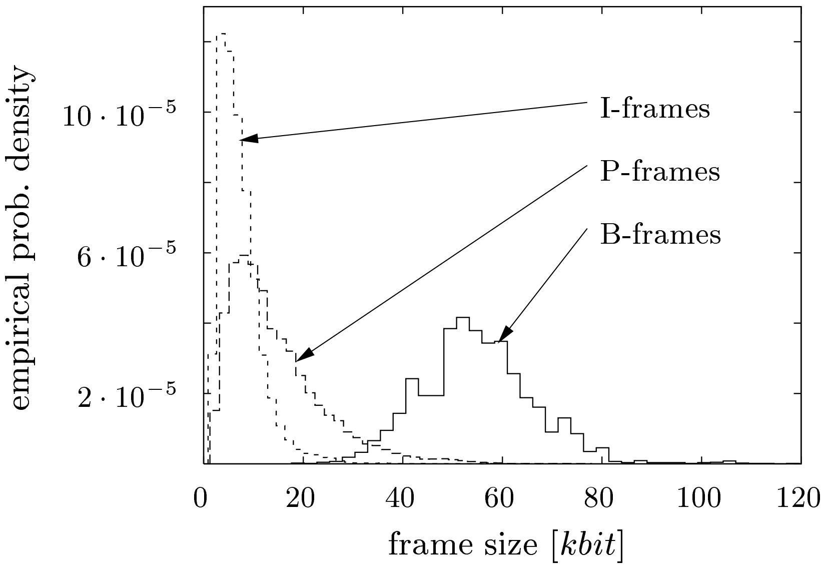


Figure 2.1: *Frame size histograms of the dino sequence*

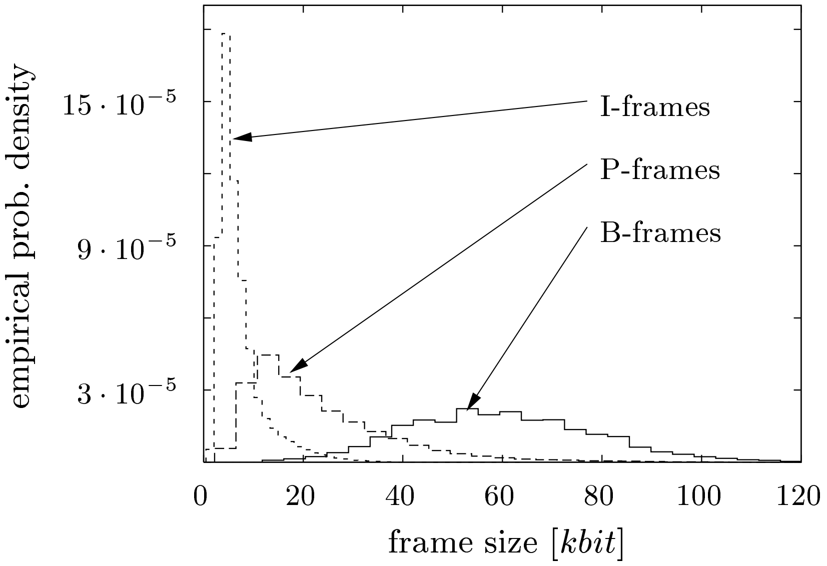


Figure 2.2: *Frame size histograms of the starwars sequence*

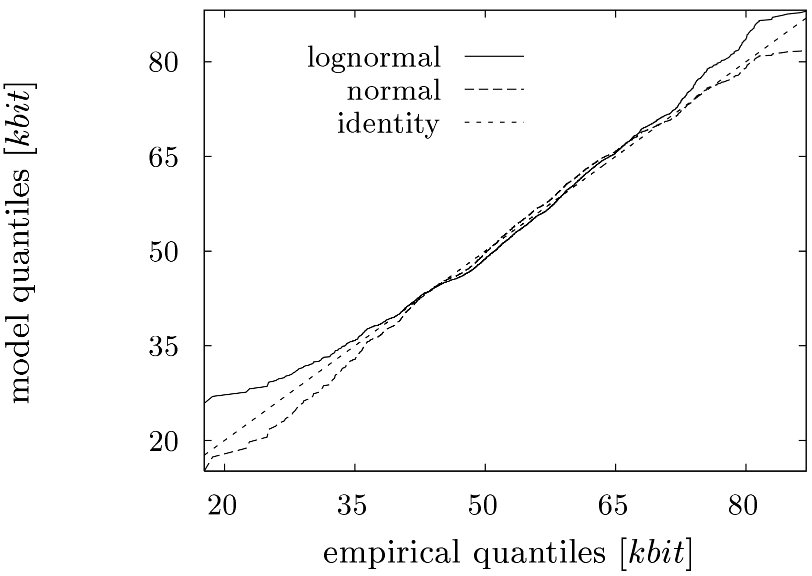


Figure 2.3: *Q-Q plot for the dino I-frames*

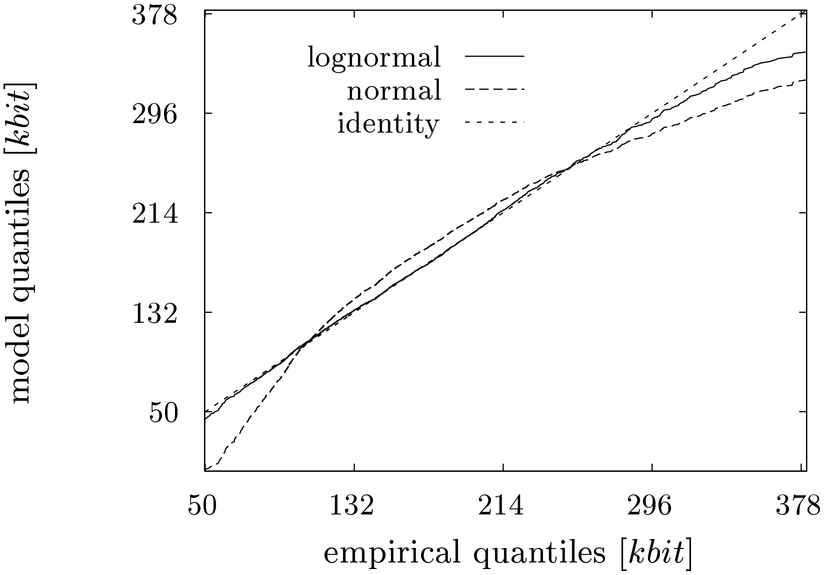


Figure 2.4: *Q-Q plot for the dino GOPs*

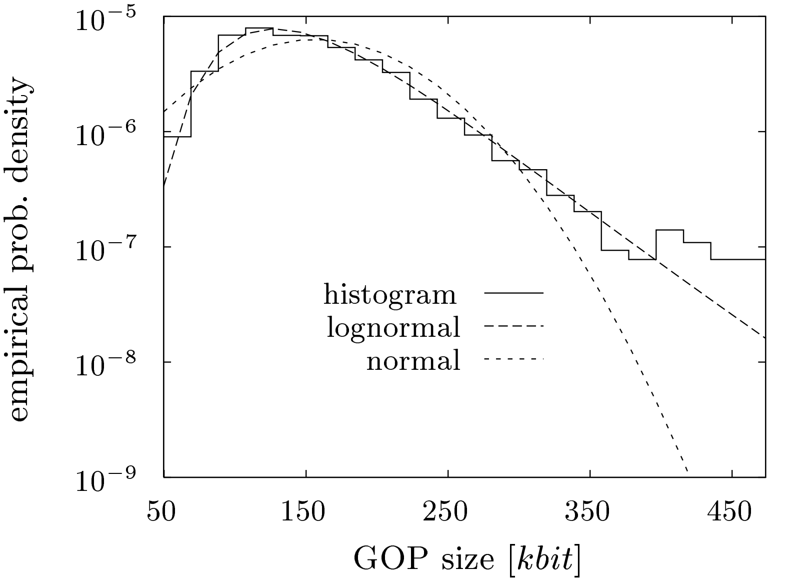


Figure 2.5: *Comparison of the tails for the dino GOPs*

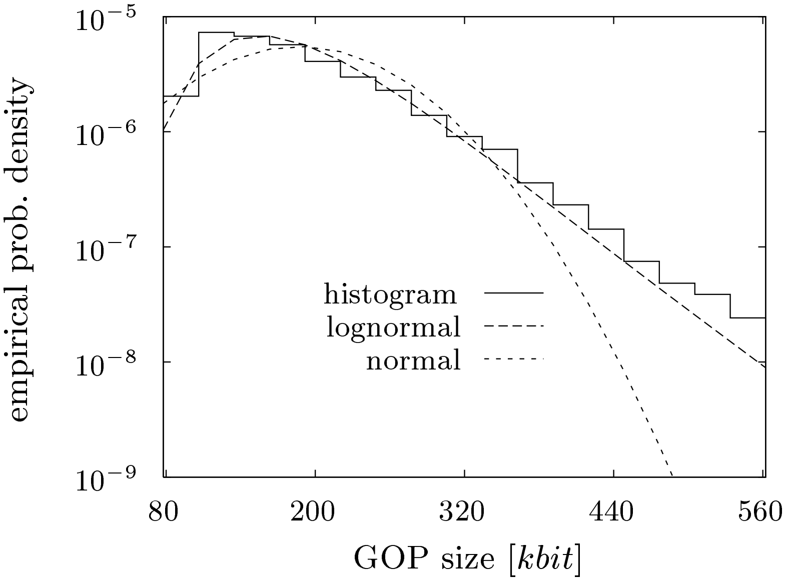


Figure 2.6: *Comparison of the tails for the starwars GOPs*

2.4 Correlations

Time-dependent statistics are important since correlations in data streams can affect the performance of the ATM network carrying this traffic. This problem is examined in Chang and Wang (1992) and Chan and Leon-Garcia (1994) in the context of video traffic and in Li and Mark (1985) and Livny et al. (1993) in a more general context. All studies report that cell losses and/or delays of the considered queuing systems are considerably higher for positvely correlated input traffic than for uncorrelated input traffic. Thus, it is important to analyze the correlation properties of video traffic.

There are a number of measures for second-order properties of empirical time series, such as autocorrelation functions (ACFs), periodograms, indexes of dispersion, and selfsimilar properties. We will focus on ACFs and the estimation of selfsimilar properties since we use this information in the sequel to determine our video traffic models. If one is more interested in burstiness measures of a given traffic sample or in indicators of non-stationarity, indexes of dispersion should also taken into consideration (see Gusella (1991)). First, autocorrelation functions of frame sizes and of GOP sizes are presented (see Appendix A.6). The frame-by-frame correlations depend on the pattern of the GOP and, in principle, always look like Figure 2.7, assuming the same GOP pattern is used for the whole sequence. The autocorrelation function clearly reflects the 12-frame GOP structure. The pattern between two I-frame peaks is therefore repeated with slowly decaying amplitude of the peaks.

If a model is needed which imitates the frame-by-frame correlations of an MPEG video traffic stream the GOP-pattern based shape of the autocorrelation function has to be considered. An approximation of the autocorrelation function is presented in Enssle (1994).

Based on the frame level correlations, it is difficult to obtain a clear picture of the mid- and long-range correlations of the video traffic stream since the curve will be dominated by the periodic GOP structure. We therefore consider the autocorrelation function of the GOP sizes.

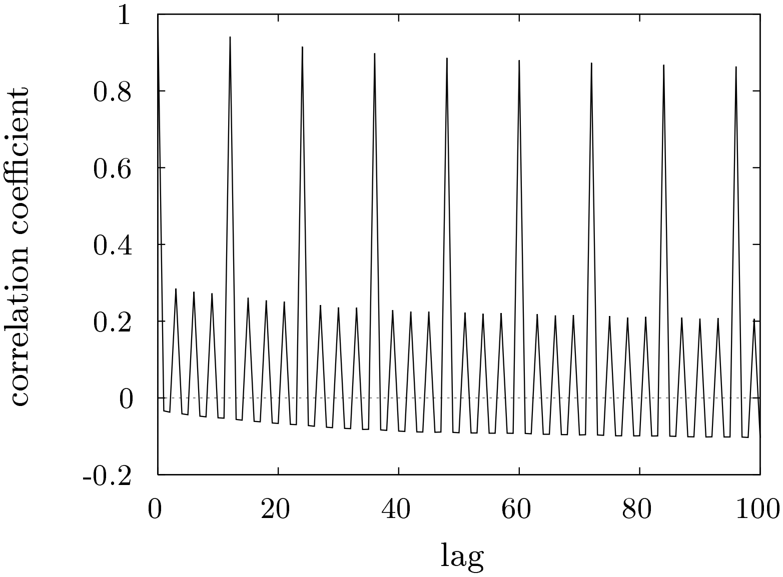


Figure 2.7: *ACF of the dino frame sizes*

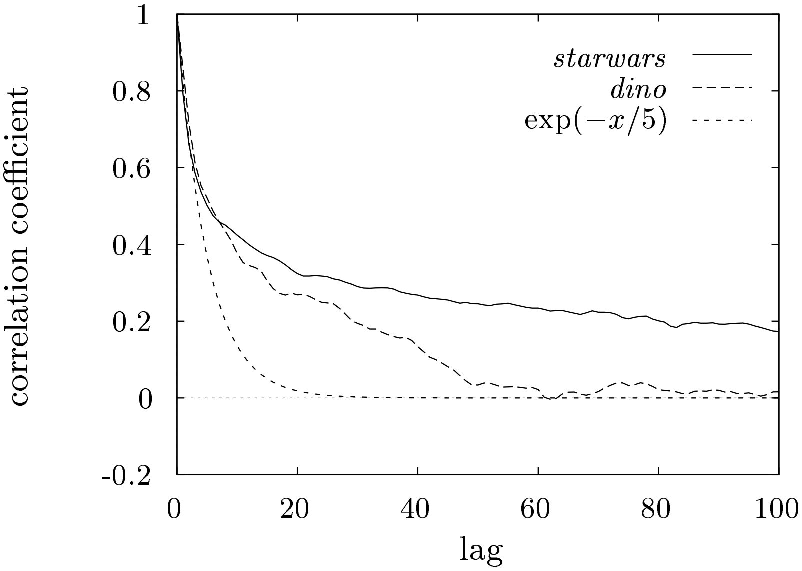


Figure 2.8: *ACF of the starwars and dino GOP sizes*

Figure 2.8 shows the autocorrelation function of the GOP sizes of the *starwars* and *dino* sequences. For comparison, we provide an exponential function matched to the empirical autocorrelation function of the first few lags. Exponential autocorrelation appears if the GOP size process is memoryless. An autocorrelation function of the statistical data decreasing less rapidly than exponentially indicates strong dependences in the GOP size process. In Figure 2.8 this is clearly the case, whereas for a few other sequences, for instance for *soccer*, the autocorrelation function and the fitted exponential function are matching well. This result makes it difficult to find a unique GOP layer modeling approach for all types of video sequence since none of the model classes will be flexible enough to model all possible correlation structures with the same accuracy.

To obtain an improved support for the decision about the appropriateness of a certain model class, we need to examine the long-range correlation behavior of the video sequences. This particular characteristic of a time series is often referred to as *selfsimilarity* or *long-range dependence.* The importance of detecting the presence of this behavior is twofold. First, queuing systems' or statistical estimators' reaction on long-range dependent input streams differs from that on uncorrelated or low-lag correlated input (see Norros (1994), Adas and Mukherjee (1995), Likhanov et al. (1995)). Second, most of the model classes, such as finite Markov chains and finite-order autoregressive processes, are not capable of modeling long-range dependence. For the comparison of model-based and trace-based performance analysis results it is therefore necessary to know whether this property is present in the video data sets or not since it can be a source of deviating results.

It is difficult, however, to evaluate long-range dependence of the video sequences by means of the autocorrelation function. A succinct way of measuring long-term dependence is through the *Hurst parameter H* (see Beran (1994), Leland et al. (1994)). We estimated the H parameter of all GOP size sequences using both *R/S analysis* and *Whittle's maximum likelihood estimator (MLE)* as described in Appendix B.4. We did not estimate the H value of the frame size sequences since we would have to aggregate the frame size sequence to avoid the effect of low-lag correlations on the estimator. Even in the case of GOP size sequences, we have to be aware of strong low-lag correlations as shown by the GOP autocorrelation functions (cf. Figure 2.8). This has several consequences for the H estimation.

Concerning the R/S analysis, we either have to cut off an adequate number of R/S samples for small ranges r, or to use aggregated sequences as input data sets. The first method may lead to errors since the wrong portion of r values is considered for the estimation. The second method leads to errors since the data set is becoming smaller.

In case of the Whittle MLE, the situation is even more complex. Here, the quality of the estimates depends on choosing an adequate underlying model. In our case, it is obvious that models without an appropriate low-lag correlation behavior such as *Fractional Gaussian Noise (FGN)* or *Fractional Autoregressive Moving-Average (FARIMA) processes* without low-lag correlations, in short FARIMA$(0, d, 0)$ with $d = H - 0.5$, lead to biased H estimates. We therefore use FARIMA$(p, d, 0)$ models with an order p larger than 0. This family of selfsimilar processes provides both low-lag and long-range correlation modeling capabilities. Then, we have to estimate the correct model order p. We solve this problem by estimating the parameters for a number of orders and choosing the parameter set where the autocorrelation curve of the corresponding FARIMA$(p, d, 0)$ provides a good approximation of the sample autocorrelations for a given number of lags (see also Appendix B.4.2).

Table 2.3 shows the H estimates for all the considered sequences. The first column consists of the results of the R/S analysis with $K = 8$, 30 ranges starting from $r_0 = 10$, and rejecting the r values with less than 4 R/S samples for the computation of the regression line. To rule out low-lag correlation influences, we determine the slope values of regression lines starting from r_0 to r_{14}. In our experience, this sequence of slope values converges to the H estimate $\widehat{H}_{RS}$ as the low-lag correlation effects fade out. The second column gives the results of the Whittle estimator $\widehat{H}_W$ assuming a FARIMA$(p, d, 0)$

process. The $\widehat{p}_W$ values are determined as described above. For comparison, the order estimate $\widehat{p}_{\text{AIC}}$ of an ordinary AR(p) process based on Akaike's Information Criterion (AIC) is presented (see Appendix B.3.2).

Table 2.3: *Hurst parameter estimates for the GOP sequences*

Sequence	R/S analysis	Whittle MLE for FARIMA($p, d, 0$)		Min AIC for AR(p)
	$\widehat{H}_{RS}$	$\widehat{H}_W$	$\widehat{p}_W$	$\widehat{p}_{\text{AIC}}$
asterix	0.86	0.85	3	11
atp	0.77	0.71	1	4
dino	0.85	0.90	1	21
lambs	0.92	0.94	7	17
mr.bean	0.95	1.00	7	22
mtv	0.86	0.93	6	18
news	0.84	0.77	1	1
race	0.71	0.56	3	6
simpsons	0.88	0.79	9	8
soccer	0.77	0.56	5	2
talk1	0.84	0.77	2	12
talk2	0.85	0.76	2	35
terminator	0.80	0.80	1	9
starwars	0.85	0.92	3	27

Note that time series without any long-range dependence have a Hurst parameter of 0.5, whereas time series of computer traffic can have H values up to 1.0 (see Garrett and Willinger (1994)). Only the *race* and *soccer* results do not clearly indicate long-range dependence. All other sequences have $\widehat{H}_{RS}$ and $\widehat{H}_W$ values larger than 0.7. For most of the sequences, the results of both estimators are in agreement. A perfect agreement can in our opinion not be forced since both estimators inherit heuristic approaches. For the R/S analysis, it is a subjective decision where the low-lag correlation effects end. For the Whittle estimator, one has to find the appropriate model process.

It has been suggested that, in case of video traffic, a larger H value reflects a larger amount of movement (see Beran (1994)). The H values, however, show that one cannot necessarily conclude a lot of movement in the video from a high H value. Even political discussion can have an H value larger than that of a soccer match. This leads to the conclusion, that long-range dependence is a property inherent in MPEG video processes independently from the content of the video sequence.

Comparing the number of parameters $\widehat{p}_W + 1$ of the FARIMA$(p, d, 0)$ processes to the $\widehat{p}_{\text{AIC}}$ values of the AR(p) processes, in almost all cases the fractal models need less parameters. In addition, an AR(p) process $(p < \infty)$ is not capable of modeling long-range dependence due to its exponentially decaying ACF.

2.5 Markovian order

Markovian models are widely used for analysis and simulation. Therefore, the Markovian properties of our video time series have to be examined. In particular, it would be of interest to determine the Markovian order of these data sets (see Appendix A.5). The previous section showed that the autocorrelation curve of the data sets does not decay exponentially and that there is a strong indication of long-range dependence. Markovian models, however, have an exponentially decaying autocorrelation function. We therefore do not attempt to estimate the Markovian order of the data sets but restrict ourselves to the discussion of the pth-order empirical entropy curves of the *dino* GOP sequence. The appearance of these curves for the other data sets is essentially the same.

Figure 2.9 shows the empirical entropies for Markov chains up to the order of 10 for a discretization of the original time series into 8, 16, and 32 intervals. The smaller the empirical entropy the better a Markov chain of this particular order is able to model the discretized time series. From the curves, we conclude that for all discretization levels the step from a histogram model (Markovian order of 0) to a first-order Markov chain provides considerable improvements in the model quality. For orders higher than 2, the curves look different. In the case of 8 intervals, there is only a small decrease in the empirical entropy while the order is increased. This indicates that a first-order Markov model is appropriate. A Markov model with 8 states, however, will lead to a poor approximation of the distribution of the GOP sizes. To obtain good models with 16 or 32 intervals, the curves indicate that the order should be larger than 5. Note, that a pth-order Markov chain ($p > 0$) based on a time series discretized into M intervals has M^p states. We would therefore need at least one million states for $M = 16$. This will not lead to

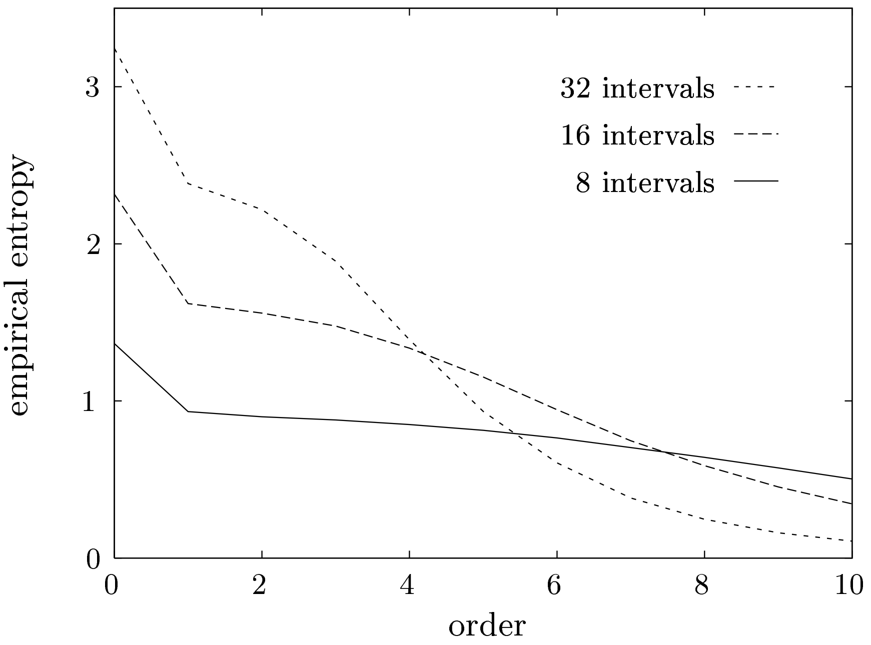


Figure 2.9: *Empirical entropies of the dino GOP sizes*

computer memory problems since most of the states will not be used. Due to basing the estimation of the probabilities on data sets, which consist only of several thousand samples, the empirical transition matrix will only contain a small number of nonzero elements. On the other hand, it is very difficult to estimate transition probabilities with any useful accuracy.

To sum up, the discussion of the Markovian properties of the GOP size sequences shows that higher-order Markov models are recommended but lead to problems of statistical significance. In Section 3.3.3, we show how this problem can partially be solved using nested Markov chains instead of higher-order Markov chains.

3 Modeling of MPEG video traffic

There are several reasons to develop models for VBR MPEG video traffic and to use them for the performance analysis of ATM networks. The first reason is to extract statistical properties of video traffic which have significant impact on the network performance. We gain a lot of insight if we are able to reduce the statistical complexity of the empirical video data sets. It is true that only the frame size trace from the output of an MPEG encoder contains all statistical information about the encoded video. However, the large number of properties makes it difficult to determine the performance and to identify how certain properties of the traffic impact on this performance. The second reason is the computational complexity of simulations of ATM networks, particularly at cell level. It often takes long simulation runs to obtain results of high accuracy. In some cases, numerical complexity can be considerably reduced using traffic models and standard analytical tools like matrix analysis or discrete-time analysis. The third reason is the need for connection traffic descriptors for video traffic. If the traffic model is simple, i.e., it has only a small number of parameters, these parameters might be used as traffic descriptors for Connection Admission Control (CAC) and Usage Parameter Control (UPC).

For model selection and development, we follow the guidelines of Buzacott and Shanthikumar (1993, p. 13). Several aspects should be considered during the modeling process:

- *Complexity versus simplicity.* More details lead to a model which is more difficult to develop, to verify, and to understand. Conversely, a simple model may not represent the data adequately and lead to inaccurate performance predictions.

- *Flexibility.* In most cases, one single model will not be appropriate to support all decisions. Therefore, it should be possible to extend the model with tolerable effort.

- *Data requirements.* The model complexity should reflect the amount of data available to estimate the model parameters. Inaccurate estimates may lead to wrong performance predictions.

- *Transparency.* The model should be designed such that not only the developer is able to understand it.

- *Efficiency.* The consumption of resources while computing the model parameters and using it for simulation or analysis studies should not exceed currently accepted limits, for instance in storage or running time.

3.1 A layered video traffic modeling scheme

For the development of video traffic models, we can exploit both knowledge about the coding technique, MPEG-1 in our case, and the statistical analysis of measured frame size sequences. The main information from the MPEG standard which we use for model development can be summarized as follows:

- There are three frame types: I-, P-, and B-frames.

- ◇ A pattern of frame types, called GOP, is repeated continuously to create the encoded frame sequence.
- ◇ The frames of one single GOP strongly depend on each other.

Moreover, if we wish to create a model at cell level, both the characteristics of the particular AAL, that is used for video transmission, and the details of any shaping applied to the cell stream before it enters the network should also be taken into account.

Based on the information presented up to this point, we are already able to develop a scheme with three layers (cf. Figure 3.1): GOP layer, frame layer and cell layer. Higher layers, such as scenes, can be added if necessary and if the statistical properties of the scene change process are available.

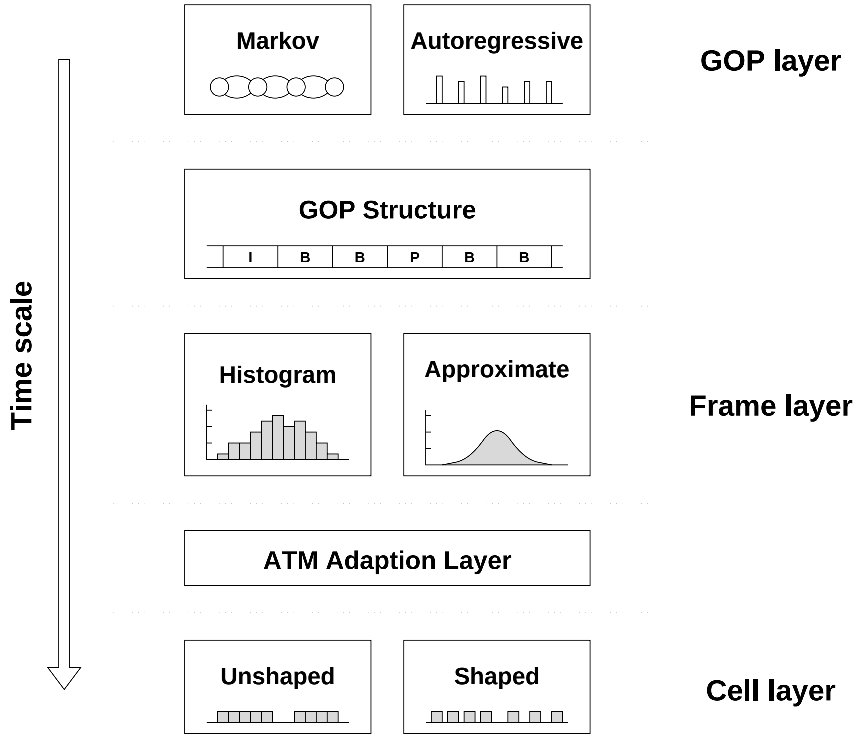


Figure 3.1: *Layered video traffic modeling scheme*

Based on the results of Chapter 2, we are able to select an appropriate stochastic process for the frame or the GOP layer. We then have to lay down the way the layers depend on each other. For example, to generate a frame size sequence based on the GOP size process, we have to consider the structure of the GOP pattern giving the order of the frame types. The simplest way to find the frame sizes based on a GOP size sample is to use a scaling factor for each frame of the GOP, where the scaling factors are the mean sizes of the frames of one GOP divided by the mean GOP size of a given data set. More complex models may use frame size histograms or approximate probability density functions to generate the frame size sequence (cf. Figure 3.1).

To obtain a cell level model, we have to decide how frames are segmented into cells. This will depend on the considered AAL and on the existence of shaping facilities between video source and ATM network. If a statistical analysis of video cell stream measurements is available it would be possible to base models directly on this material. This approach may lead to simpler models for the cell process.

The presented model development scheme is not a recipe for a perfect video traffic model. It is more the outline of a variety of stochastic modules and the description of how they interact in the case of video traffic. The model developer will have to choose which modules are appropriate for the analysis.

It should be noted that any model needs to be validated. Even quite complex models rely on simplifying assumptions and may ignore significant correlation effects. To obtain useful and reliable performance analysis results, it is important to know how these assumptions affect the results of the analysis.

3.2 Literature review

A large variety of papers about video traffic modeling can be found in the current teletraffic literature. The modeling approaches can be divided into the following classes:

- ◇ Histogram models,
- ◇ Markov models,
- ◇ Autoregressive processes,
- ◇ TES models,
- ◇ Selfsimilar models.

An overview on video models can be found in e.g. Frost and Melamed (1994) or Rose and Frater (1994). In the following sections, we give a survey of current video traffic publications. Note, that almost all papers deal with one model or one model class and only a few of them are based on the same video data sets. In addition, several data sets are produced by non-standard codecs, or codecs which are not used in ATM networks. Some of the authors do not even validate the modeling approaches by means of real data sets. We therefore do not comment on the quality of their modeling approaches. A major contribution of this monograph is to adapt the modeling approaches presented in the following to VBR MPEG video traffic and to compare the models on a common basis of data sets.

Histogram models

Histogram models are the simplest modeling approaches since they do not take into account the correlation structure of the video data sets and they need no in-depth statistical inference for the computation of their parameters.

Skelly et al. (1993) report that a histogram model provided good results for estimating the buffer occupancy distribution of an ATM multiplexer with video input sources. Shroff and Schwartz (1994) use these results to estimate end-to-end cell loss probabilities for video traffic. Both papers are based on data sets which consists of only one frame type. We mention the work of Krunz et al. (1995) in this section, since they provide a model for MPEG video traffic with all three frame types although the authors used a fitted lognormal distribution instead of a histogram. They use the simple approach of cycling through a set of three different lognormal random variables, i.e., one for each frame type, to generate a sequence which reflects the GOP pattern of MPEG coding.

Markov models

Under Markov models, we subsume all models that have a state space and a state transition probability or rate matrix that characterizes the behavior of the model, such as Markov chains, Markov Renewal Processes (MRP), Markov Modulated Poisson Processes (MMPP), etc. Numerous papers deal with Markov model approaches since the estimation of the model parameters is often straightforward and there is a large number of analysis techniques available to examine queuing systems with this type of input.

First, we present the models which are used to describe the video codec output at frame level. Maglaris et al. (1988) use a birth-death Markov model for the video input of their queuing systems. This model is extended by Sen et al. (1989) to a superposition of video sources and to a single video source with two activity levels. Blondia and Casals (1992) present a matrix analytic solution of a queuing system with input from a video model with multiple activity levels. Pancha and Zarki (1993a, 1993b) suggest Markov chains to model the slice and the frame size process. They report that for their data sets the transition matrices were almost tridiagonal if maximum likelihood estimates for the transition probabilities are used. A Markov chain approach is also used by Heyman et al. (1992) where the authors use the lag-1 autocorrelation of their video data and the assumption that their frame sizes are negative-binomially distributed to construct the transition matrix of the Discrete Autoregressive (DAR) model. Cohen and Heyman (1993) extend this work. In the paper of Coelho and Tohme (1993) a Markov chain is used to predict the output of a video encoder. Lucantoni et al. (1994) show that, applying their own goodness-of-fit metrics, a Markov Renewal Process outperforms the Heyman et al. approach. Frater et al. (1994) present another extension of the Heyman et al. model. They use Pareto distributed instead of exponentially distributed sojourn times for the states. In contrast to the other Markovian models, an aggregation of these model streams will lead to long-range dependent traffic. A more complex state-based model is suggested by Rodriguez-Dagnino and Leon-Garcia (1992). They use a Markov chain to model the scene changes of a video sequence. During the holding time of a state or scene frames are generated by an Autoregressive Moving Average (ARMA) process. A similar approach is used by Chandra and Reibman (1996) for modeling two-layer MPEG-2 video traffic. They discovered clusters of frames in their data sets and model the cluster changes by a Markov chain. During the holding time of a particular state the frames are generated by an autoregressive process. A different approach is followed by Heyman and Lakshman (1994). They define a statistical criterion which is used to separate the scenes of a video sequence and combine the scene length distribution with their former DAR modeling approach.

In contrast to the above Markov models, Cosmas and Odinma-Okafor (1991) model the encoder output on cell level using a Geometrically Modulated Deterministic Process (GMDP).

Autoregressive models

A class of models which also attracts a lot of researchers is the class of autoregressive (AR) processes. This class is particularly interesting since a wide range of methods known from time series analysis can be applied for parameter estimation and model characterization.

Besides the Markov model already mentioned above, Maglaris et al. (1988) present a first-order AR model for their data set. Nomura et al. (1989) also use a first-order autoregressive process to model the frame size process of their experimental encoder. In addition, they suggest to use Markov modulated AR processes for video sequences consisting of several scenes. Roberts et al. (1991) use the Maglaris et al. model for their performance considerations of a multiplexer with VBR video input traffic. A superposition of two first-order AR processes and a Markov chain is considered by Ramamurthy and Sengupta (1992). The first AR process models the short-range behavior, the second AR process the mid-range behavior, and the Markov chain models single peaks of the frame size process. Chowdhury and Sohraby (1994) compare bandwidth allocation algorithms for packet video which is modeled using the Maglaris et al. approach. One of the few models dedicated to MPEG video traffic is presented by Enssle (1994). He uses three first-order AR processes to model the MPEG frame types separately. A similar approach is used by Adas (1996) to predict the frame sizes of an MPEG trace. He used higher order AR processes. In contrast to these two approaches, Stokes (1995) uses one AR(p_G) model for the complete sequence, where p_G is the number of frames in one GOP.

Grünenfelder et al. (1991a, 1991b) suggest an ARMA model to characterize the cell interarrival process at the output of their video encoder.

TES models

Transform-Expand-Sample (TES) models were developed to model autocorrelated time series with arbitrary marginal distributions. An introduction of TES processes and the related theory is beyond the scope of this monograph. Melamed et al. (1992) give an outline of this model class. In the example section of this paper, the authors present a TES model for VBR video traffic.

Lee et al. (1994) report about a TES model which is used to model the frame size sequence at the output of an encoder with two layers. A VBR MPEG sequence is modeled by Reininger et al. (1994) using three TES processes for the different frame types. Ismail et al. (1995) extend this model for layered MPEG video traffic.

Selfsimilar models

Recently, a lot of attention has been paid to selfsimilar or long-range dependent modeling of traffic streams in communication networks. Up to now, most of the papers mainly deal with the statistical analysis of data sets, i.e., in most cases with the estimation of the Hurst parameter of an empirical sequence, and provide little information about traffic models and the analysis of queuing systems with fractal input sources. In Norros (1994) and Likhanov et al. (1995), the authors found that in G/D/1 systems with selfsimilar input the queue length distribution does not decay exponentially – as in the case of non- or short-range correlated input traffic – but hyperbolically or Weibullian.

Garrett and Willinger (1994) present a detailed statistical analysis of a two hour VBR video trace and present a FARIMA$(0, d, 0)$ model for video traffic. In Adas and Mukherjee (1995), the authors use a FARIMA$(1, d, 0)$ model for their experiments. Enssle (1995) suggests to use a FGN model for MPEG video traffic and compares its performance to a white noise process with the same marginals. In contrast to these papers, Huang et al. (1995) generate selfsimilar model traces directly from the autocorrelation function of the MPEG I-frame sizes using Hosking's method.

3.3 Video traffic models

Most of the models from the literature approximate the behavior of the frame size process at the video encoder output. This is a reasonable choice if the data sets used for the model parameter estimation consist of only one frame type. In our case, however, a frame layer model will have to reflect the periodic GOP structure of MPEG formed by three different frame types to be realistic. The modeling of this specific periodic behavior will already require a large number of parameters. Moreover, correlations over larger intervals than several frame periods are difficult to include. We therefore focus on modeling the GOP size process and use a simple method to generate the frame size process from it. Models on scene level are also possible but require complex statistical analysis. Note that larger time scales lead to smaller numbers of samples which can be used to estimate the model parameters. This often implies a decrease in statistical significance of the measurements. For instance, given a sequence of 40 000 frames and a 12 frame GOP pattern, the GOP size sequence has only about 3 000 samples and the scene length statistics will be based on a few hundred samples. In addition, scene level models work on time scales which are partly covered by GOP level models. Thus, careful consideration is necessary to justify the additional complexity of MPEG model levels higher than GOP level. One of our models, the scene-oriented model, is a GOP level model where scene level correlation information is included without extensive additional measurements. The selfsimilar model covers all time scales by definition. This does not necessarily mean, however, that selfsimilar models capture the specific characteristics of GOPs and scenes.

In the following, we discuss how frame sizes are computed from a GOP size sequence. First, we use an empirical video frame size trace to estimate the mean size of each frame type (I, P, or B) of the GOP. If we divide the mean frame sizes by the mean GOP size, we receive a scaling factor for each frame of the GOP. Note, that we do not distinguish between the individual P- and B-frames. Table 3.1 shows the scaling factors for the *starwars* and the *dino* data sets.

To generate a sequence of frame sizes, we use one of the models introduced later to generate the GOP sizes and compute the frame sizes by multiplying the GOP size with the scaling factors. As the results of the following chapters

Table 3.1: *Frame scaling factors*

	I	B	B	P	B	B	P	B	B	P	B	B
starwars	0.323	0.038	0.038	0.123	0.038	0.038	0.123	0.038	0.038	0.123	0.038	0.038
dino	0.351	0.047	0.047	0.092	0.047	0.047	0.092	0.047	0.047	0.092	0.047	0.047

show, this simple method leads to a good approximation of the frame process of the video trace. In particular, the periodic nature of the frame process is approximated with little effort. Due to the fact that both GOP and frame sizes of each type are approximately lognormally distributed, this method also leads to frame size distributions which are close to the original ones. The only frame layer information which is lost consists of the frame-by-frame correlation which is present apart from the correlation induced by the GOP pattern. As a result, a model frame trace can have a larger maximum frame size than the empirical trace while the mean and variance of both traces are the same. In several scenarios, see e.g. Chapter 4, this can lead to performance predictions which are more pessimistic than those of the empirical data sets. In those cases, we cross-check our results with models which are not based on the GOP size sequences but on the I-frame sequences of the data sets. We obtain an I-frame based vector of frame scaling factors simply by dividing the above frame scaling factors by the leading I-frame factor. The model parameters are determined from the I-frame size sequence instead of the GOP size sequence. Thus, we obtain a model which better reflects the frame size process within a GOP than the GOP-based models at the cost of a worse performance in modeling the GOP-by-GOP interdependences.

In the following sections, we introduce models of different complexity and approximation quality which we intend to compare with regard to their statistical properties and their accuracy to predict performance measures such as cell losses at ATM multiplexer buffers. The models can be divided into two classes: discrete versus continuous marginal distribution. The histogram model, the simple Markov chain model, and the scene-oriented model are based on states and have a histogram-type marginal distribution. The autoregressive model and the selfsimilar model have a lognormal marginal distribution.

3.3.1 Histogram model

The histogram model is the simplest model. It is equivalent to modeling a time series by independent and identically distributed (i.i.d.) random variables. This implies that a number of algorithms for the analysis of queuing systems with i.i.d. input traffic can be applied. The main disadvantage of this model is that any GOP-by-GOP correlation remains unmodeled. The statistical analysis of the GOP traces, however, shows considerable positive autocorrelation for at least the first 100 lags. In application scenarios where the results depend on the accuracy of the modeled autocorrelation behavior, e.g. systems with large cell buffers, the performance estimates based on the histogram model, such as losses or delays, can be several orders of magnitude too small.

Parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered GOP size trace. The only user-defined parameter of the histogram model is the number k of histogram intervals. The relative frequency h_j of the samples in the GOP size intervals and the GOP size s_j related to interval j $(j = 1, \ldots, k)$ are computed with the formulae presented in Appendix A.2. A small k value leads to a poor approximation of the marginal distribution of the empirical trace whereas a large k value may lead to problems of statistical significance due to too few observations.

Generation of a model trace

Let $\{\epsilon_i\}$ be $U(0,1)$ distributed white noise. Given the frequencies h_j and GOP sizes s_j $(j = 1, \dots, k)$ the trace $\{t_i\}$ is generated by

$$t_i \quad = \quad s_j \quad \text{with} \quad j \quad = \quad \min\left\{ l : \sum_{j=1}^{l} h_j > \epsilon_i \right\}. \tag{3.1}$$

For each of the models of this chapter we present one trace with 1 000 samples and discuss the shape and properties. The model parameters are estimated on the basis of the *dino* GOP size sequence. Figure 3.2 shows the first 1 000 samples of this data set. We point out that trace diagrams are no proof of the quality of a model. Nevertheless, traces are attractive since they illustrate the differences of the models without applying any statistical machinery.

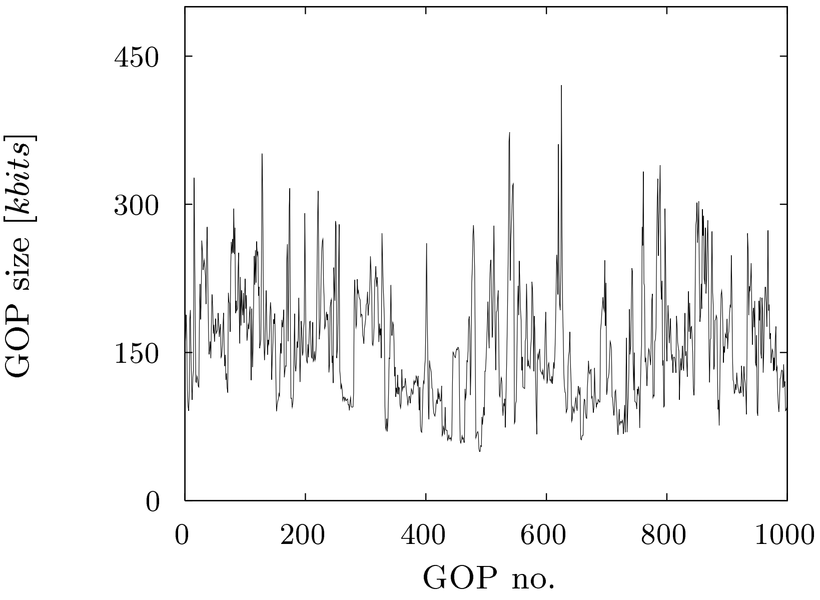


Figure 3.2: *Trace from the dino GOP size sequence*

Figure 3.3 shows a trace generated by the histogram model with a number $k = 20$ of intervals. It reflects the discrete nature of the sample sizes. Due to the fact that correlations are not modeled, no time-dependent structure is noticeable.

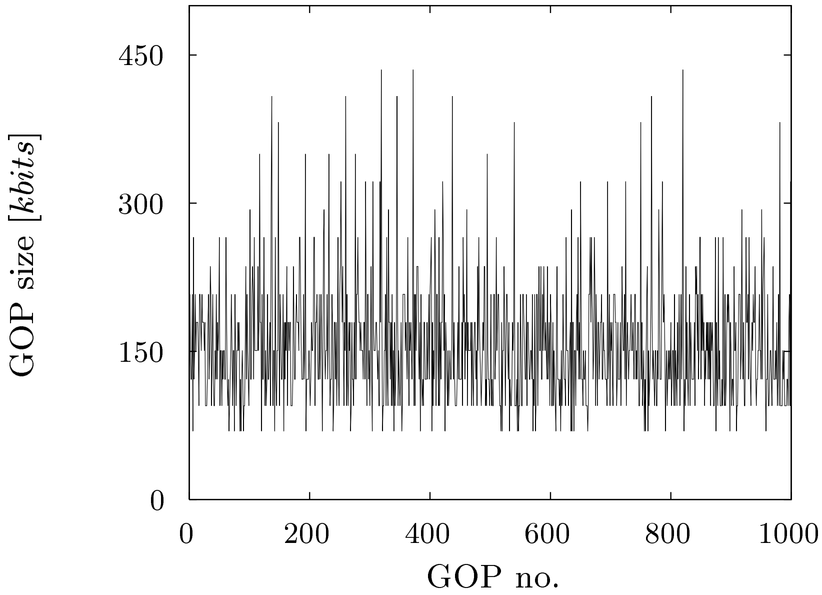


Figure 3.3: *GOP size trace generated by the histogram model*

3.3.2 Simple Markov chain model

The simple Markov chain model consists of a first-order Markov chain and is able to model the histogram and the lag-1 autocorrelation of the given data set. There are several analysis techniques for queuing systems with Markov chain type input, e.g. Neuts (1981), Neuts (1989), or Anick et al. (1982). Since the estimation of the Markovian order in Section 2.5 shows that only large orders lead to a considerable improvement of the model quality compared to the first-order case we will not consider conventional higher-order Markov chains in the sequel.

Parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered GOP size trace. The only parameter which has to be specified for the first-order Markov chain is the number M of states. The transition matrix $\mathbf{P} = [p_{ij}]$ and the GOP size vector $\mathbf{s} = [s_1, \ldots, s_M]$ can be computed with the formulae presented in Appendix B.2.2. Note, that for parameter estimation we will only consider the maximum likelihood estimates approach since the DAR(1) approach does not fit into our intended evolution from the histogram model over the simple Markov chain model to the scene-oriented model.

Generation of a model trace

Let $\{\epsilon_i\}$ be $U(0,1)$ distributed white noise. Given $\mathbf{P}$ and $\mathbf{s}$, the sequence of states $\{m_i\}$ is generated by

$$m_i \quad = \quad \min\left\{ l \, : \, \sum_{j=1}^{l} p_{m_{i-1},j} > \epsilon_i \right\}, \tag{3.2}$$

i.e., being in state m_{i-1}, the next state m_i is estimated by accumulating the transition probabilities of row m_{i-1} of $\mathbf{P}$ until a fixed but random threshold ϵ_i is exceeded. The trace $\{t_i\}$ is determined by

$$t_i \quad = \quad s_{m_i}. \tag{3.3}$$

Figure 3.4 shows a trace generated by the simple Markov chain model with $M = 20$ states. As in the histogram case, the trace clearly reflects the discrete nature of the sample sizes. Compared to the trace of the histogram model, more structure in time is present. In contrast to the empirical trace, however, the average behavior does not change in time, indicating that no long-term correlations are modeled.

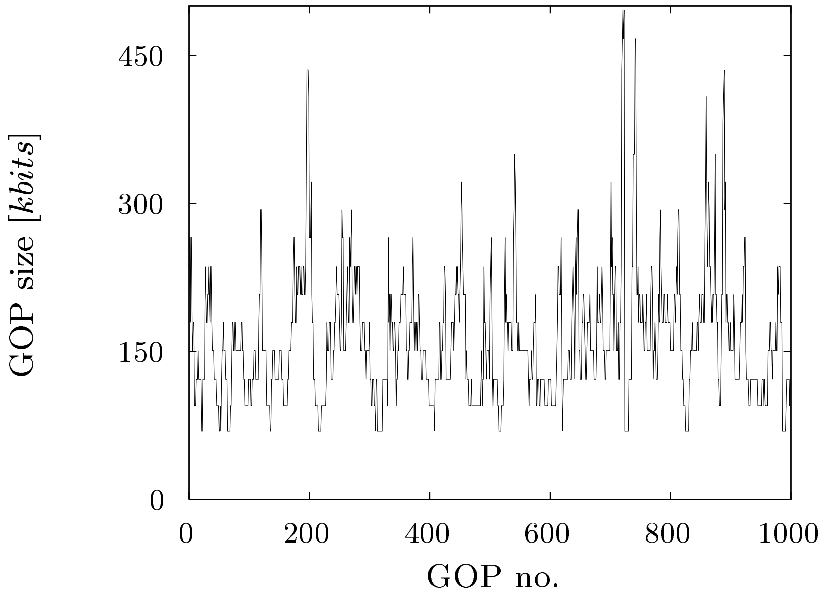


Figure 3.4: *GOP size trace generated by the simple Markov chain model*

3.3.3 Scene-oriented model

The scene-oriented model is a first-order Markov chain with a redefined set of states. The intention of this redefinition is to facilitate the modeling of scene changes and to achieve an improvement of the autocorrelation modeling properties of the Markov chain with a moderate increase in the number of states. Before we are able to classify the GOPs into different scenes we have to determine the scene boundaries of the video sequence. From a statistical point of view, it is not necessary to find out the scene boundaries of the original video sequence by watching the movie since we are not interested in a statistical analysis of the scene length. Therefore, we use a method to find these boundaries which only depends on a few statistical properties of a group of consecutive GOP sizes. These parameters should be available by simply scanning the GOP size sequence without any knowledge about the content of the video sequence. Compared to having to watch the sequence, this approach is fast and needs no additional technical equipment. In addition, the state space of the Markov chain can be automatically generated directly from the GOP size trace. Heyman and Lakshman (1994) present an algorithm to split a video trace into scenes. It is based on the fact that normally a new scene starts with a frame which is considerably larger than the preceding frames. This behavior results from the predictive encoding which is used in most video encoding schemes. The first frame of a new scene can not be predicted from former frames and therefore contains more information. In our case, however, this algorithm is not applicable since our aim is to find out scene changes among GOP sizes. Due to the summing of frames, the sudden increase of frame size at the beginning of a scene is averaged out and is not detectable anymore. Thus, we developed new algorithms to determine scene changes for GOP size sequences.

Parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered GOP size trace. In addition to the number of states M_G used to model the GOP sizes while being in a particular scene, we have to specify the number of scene classes M_S. For our traces M_G and M_S values from 10 to 20 resulted in a good approximation of both the marginal distribution and the low-lag correlations of the data sets. Note, for $M_S = 1$ the scene-oriented model degenerates to the simple Markov chain model mentioned above. To determine the scene class of each x_i we suggest the following two algorithms. Both algorithms group together GOPs into scenes which have approximately the same GOP size.

Variation-based algorithm. For the variation-based algorithm, we determine the scene boundaries based on the coefficient of variation for a sequence of consecutive GOP sizes. We add GOPs to the sequence under consideration until its weighted coefficient of variation is changing more than a preset value. The last GOP added is defined as the beginning of a new scene.

In the following, we present a more formal description of the algorithm. We first need to specify the scene change threshold ϵ_S. Let n_G denote the current GOP number and be n_S the current scene number.

(1) Set $n_G = 1$ and $n_S = 1$.
Set the current left scene boundary $b_{\text{left}}(n_S) = 1$.

(2) Increment n_G by 1. Compute the coefficient of variation $\widehat{cv}_{\text{new}}$ of $\{x_{b_{\text{left}}(n_S)}, \dots, x_{n_G}\}$.

(3) Increment n_G by 1. Set $\widehat{cv}_{\text{old}} = \widehat{cv}_{\text{new}}$. Compute the coefficient of variation $\widehat{cv}_{\text{new}}$ of $\{x_{b_{\text{left}}(n_S)}, \dots, x_{n_G}\}$.

   (i) If $|\widehat{cv}_{\text{new}} - \widehat{cv}_{\text{old}}|(n_G - b_{\text{left}} + 1) > \epsilon_S$ then set the right scene boundary $b_{\text{right}}(n_S) = n_G - 1$ and the left scene boundary of the new scene $b_{\text{left}}(n_S + 1) = n_G$. Increment n_S by 1 and go to step (2).

   (ii) If (i) does not hold go to step (3).

Iterating this algorithm over the whole GOP size sequence provides a series of N_S scene boundary pairs. Now, we compute the average GOP size $\bar{x}_S(i)$ for each scene

$$\bar{x}_S(i) = \frac{1}{b_{\text{right}}(i) - b_{\text{left}}(i) + 1} \sum_{k=b_{\text{left}}(i)}^{b_{\text{right}}(i)} x_k \quad \text{with } i = 1, \dots, N_S \tag{3.4}$$

and extend the given GOP size trace $\{x_i\}$ to a series of pairs $\{(x_i, \bar{x}_i)\}$ with

$$\bar{x}_i = \bar{x}_S(s) \quad \text{for } b_{\text{left}}(s) \le i \le b_{\text{right}}(s), \tag{3.5}$$

i.e., pairs formed by the GOP size and mean GOP size of the scene where this particular GOP is located.

Our experiments show that ϵ_S should be chosen such that the resulting average scene length is at least ten GOPs to obtain a reasonable approximation quality of the autocorrelation function. The range of lags which is well approximated has to be determined heuristically varying ϵ_S and M_S.

Average-based algorithm. The average-based algorithm consists of shifting a moving average window of size W over the trace. Compared to the variation-based algorithm, it is very simple since the scene boundaries have not to be computed explicitly. The only parameter needed for the algorithm is the window size W. The given GOP size trace $\{x_i\}$ is extended to a series of pairs $\{(x_i, \bar{x}_i)\}$ with

$$\bar{x}_i \quad = \quad \frac{1}{W} \sum_{k=i}^{i+W-1} x_k \quad \text{for} \quad i = 1, \ldots, N - W + 1. \tag{3.6}$$

Assuming the adequate number of scene classes M_S and a window of size W, the average-based algorithms leads to a model whose approximation quality of the autocorrelation curve of the data set is good for approximately the first W lags. Figure 3.5 shows this property for scene-oriented models fitted to the *dino* data. All models have $M_G = 20$ GOP classes and $M_S = 10$ scene classes, implying a 200×200 transition matrix. The window size W was varied from 1 to 100. Note, that a value of $W = 1$ results in the autocorrelation curve of the simple Markov chain model.

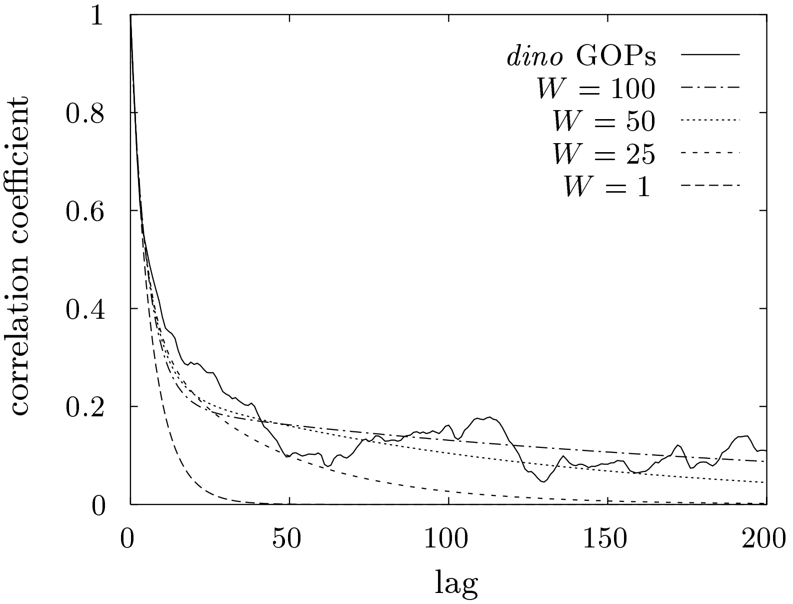


Figure 3.5: *Autocorrelation functions of the scene-oriented model*

In the sequel, we only use the average-based algorithm to determine the scene classes due to its simple way to set the expected correlation range.

Estimation of the transition matrix and the size vector. Before we start to determine the transition matrix for the scene-oriented model we have to define an appropriate state space. Each pair of the series $\{(x_i, \bar{x}_i)\}$ is related to a state (m_i^G, m_i^S), where $m_i^G \in \{1, \ldots, M_G\}$ denotes the GOP size class and $m_i^S \in \{1, \ldots, M_S\}$ the scene class of GOP i. The classes are obtained by discretizing both the x_i and $\bar{x}_i$ analogously to the first-order Markov chain case (see also Appendix B.2.2). The $M_G \cdot M_S$ states are ordered such that the states for one scene class are grouped together with ascending GOP class number. The entries p_{ij} of the transition matrix can now be estimated as shown in the appendix. The size vector $\mathbf{s}$ is determined in two steps. First, we compute the size vector of length M_G based on the x_i values as in the first-order case. Then, the vector $\mathbf{s}$ is formed by concatenating M_S copies of that vector. For instance, let $M_G = 3$ and $M_S = 2$. Then, the state space consists of $\{(1,1), (2,1), (3,1), (1,2), (2,2), (3,2)\}$ and $\mathbf{s} = [s_1, s_2, s_3, s_1, s_2, s_3]$.

Generation of a model trace

The model trace is generated analogously to the first-order Markov chain case. Figure 3.6 shows a trace generated by the scene-oriented model with $M_G = 20$ GOP classes and $M_S = 10$ scene classes. To estimate the parameters, the average-based algorithm was used with a window size $W = 100$. As for the other state based models, the sample sizes are from a discrete set. Compared to the trace of the simple Markov chain model, more variation in the average behavior is noticeable, indicating that some long-term correlation is modeled.

3.3.4 Autoregressive model

Autoregressive processes are widely used in the time series analysis literature. Due to the difficulties in analyzing systems with autoregressive in-

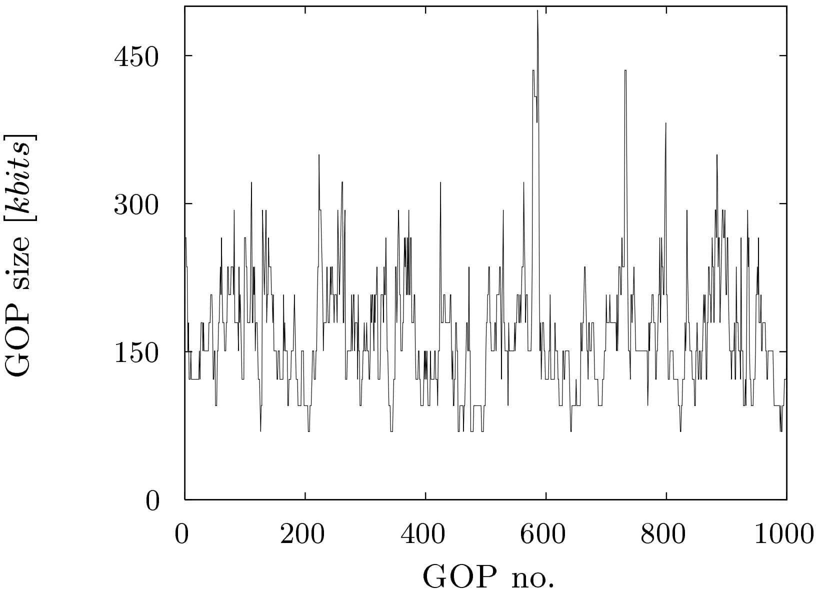


Figure 3.6: *GOP size trace generated by the scene-oriented model*

put, the queuing theory literature focuses on Markovian input streams. In our case, autoregressive models have the advantage of a smaller number of model parameters while providing a similar autocorrelation behavior and a higher accuracy in modeling the marginal distribution of the empirical traces than Markov chain models. We obtain a lognormal marginal distribution which is typical for MPEG video traffic by simply transforming the $N(\mu_n, \sigma_n)$ marginal distribution of a standard AR(p) process by an exponential function. By means of the model order p, we are able to determine the number of lags of the empirical autocorrelation curve that are modeled correctly.

Parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered GOP size trace. Assuming lognormally distributed GOP sizes, we first have to transform $\{x_i\}$ to a time series with normal marginals $\{x_i^n\}$ by $x_i^n = \log x_i$. The parameters of the normal marginal of the transformed process are given by the sample mean $\widehat{\mu}_n$ and the sample variance $\widehat{\sigma}_n^2$. Now, the model order p has to be determined. This can be done either by heuristic considerations or applying Akaike's information criterion (AIC). The values $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_p$ are given by the Yule-Walker estimates for $\{x_i^n : i = 1, \ldots, N\}$. For AIC and Yule-Walker estimates see Appendix B.3.2.

Generation of a model trace

Let $\{\epsilon_i\}$ be a $N(0,1)$ distributed white noise process. The values μ_n and σ_n are set to the given estimates $\widehat{\mu}_n$ and $\widehat{\sigma}_n^2$ or in dependence of the expected mean μ_t and variance σ_t^2 of the model trace.

$$\mu_n = \log \frac{\mu_t^2}{\sqrt{\sigma_t^2 + \mu_t^2}} \tag{3.7}$$

$$\sigma_n^2 = \log \frac{\sigma_t^2 + \mu_t^2}{\mu_t^2} \tag{3.8}$$

For a model trace of length N, we generate a trace of the Gaussian process $\{t_i^m : i = 1, \ldots, L + N\}$ applying the recurrence relation

$$t_i^m = \alpha_1 t_{i-1}^m + \ldots + \alpha_p t_{i-p}^m + \epsilon_i \tag{3.9}$$

with $t_i^m = 0$ for $i < 1$ and given the parameters $\alpha_1, \ldots, \alpha_p$. We neglect the first L samples to avoid start-up errors. The value L is determined by comparing the autocorrelation curve of the trace $\{t_i^m : i = L + 1, \ldots, L + N\}$ and the theoretical autocorrelation curve of the AR(p) process. If both curves match well the value L is large enough. The marginal distribution of trace $\{t_i^m\}$ will be Gaussian but not with the expected parameters. We therefore transform the samples t_i^m to the $N(\mu_n, \sigma_n^2)$ distributed samples t_i^n as follows.

$$t_i^n = \frac{(t_{L+i}^m - \mu_m) \cdot \sigma_n}{\sigma_m} + \mu_n \quad \text{for} \quad i = 1, \ldots, N \tag{3.10}$$

with μ_m denoting the mean and σ_m^2 the variance of $\{t_i^m\}$.

Finally, the trace $\{t_i\}$ is generated from $\{t_i^n\}$ by

$$t_i = \exp t_i^n \tag{3.11}$$

Figure 3.7 shows a trace generated by an AR(1) model with lognormal marginal distribution. In contrast to the state-based models mentioned above, the sample sizes follow a continuous distribution. This makes the trace look closer to the original data set compared to the Markov-type models. Since the average behavior does not change very much over time the trace behaves like a continuous equivalent of the simple Markov chain case.

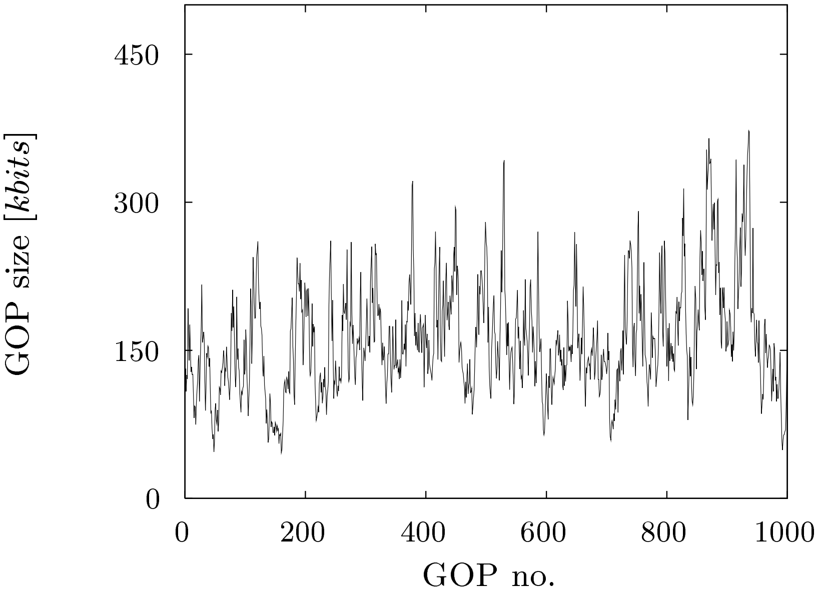


Figure 3.7: *GOP size trace generated by the auto-regressive model*

3.3.5 Selfsimilar model

Selfsimilar processes form a model class which facilitates the modeling of long-range dependence. All other models are only capable of modeling short-term correlations. The selfsimilar models allow parsimonious models but the estimation of their few parameters is not as straightforward as for the other models. Since this class of models is used in telecommunication research only for a few years, the number of papers on performance analysis in the presence of long-range dependent traffic is small. We therefore had only a few guidelines or rules of thumb for parameter estimation and performance evaluation with selfsimilar models.

Due to the results of Section 2.4, we decided to use FARIMA$(p, d, 0)$ processes with $p \geq 0$ to model the GOP size traces. Note, that $d = H - 0.5$. FGN or FARIMA$(0, d, 0)$ also facilitate the modeling of long-range dependence but offer no possibility to match the models to the low-lag correlations of the data sets if necessary. Standard FARIMA processes have a Gaussian marginal distribution. Similarly to the autoregressive models, we obtain the adequate marginal distribution by transforming the $N(\mu_n, \sigma_n)$ marginal distribution of the FARIMA process to a lognormal marginal.

Parameter estimation

Let $\{x_i : i = 1, \ldots, N\}$ denote the considered GOP size trace. Assuming lognormally distributed GOP sizes, we first have to transform $\{x_i : i = 1, \ldots, N\}$ to a time series with normal marginals $\{x_i^n : i = 1, \ldots, N\}$ by $x_i^n = \log x_i$. The parameters of the normal marginal of the transformed process are given by mean $\widehat{\mu}_n$ and variance $\widehat{\sigma}_n^2$. The model order $\widehat{p}$, the Hurst parameter $\widehat{H} = \widehat{d} + 0.5$ and the parameters $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_p$ of the AR(p) part of the FARIMA$(p, d, 0)$ process are estimated as described in Appendix B.4.2.

Generation of a model trace

In contrast to the other model traces, the selfsimilar trace has to be generated in one piece to provide the expected H value. For the generation of a Gaussian FARIMA$(p, d, 0)$ trace of length N we used the two-step algorithm suggested by Hosking (1984) . We first generate a FARIMA$(0, d, 0)$ trace of length $L + N$. Then, we add the AR(p) part by applying the appropriate recurrence relation and cut off the first L samples. The whole generation process is equivalent to that of an AR(p) trace besides the fact that $\{\epsilon_i\}$ is white noise in the AR(p) case and long-range dependent in the FARIMA($p, d, 0$) case for $d > 0$. If $d = 0$, i.e., $H = 0.5$, the statistical properties of AR(p) and FARIMA($p, d, 0$) traces are the same.

Generation of the FARIMA(0,d,0) trace. Given $d = H - 0.5$, let $\{\epsilon_i : i = 1, \ldots, L+N\}$ denote the FARIMA($0, d, 0$) trace. Set $v_0 = 1$. Choose ϵ_0 from $N(0, v_0)$. Then generate $L + N$ points by iterating the following algorithm for $k = 1, \ldots, L + N$:

$$\phi_{kk} = d/(k-d) \tag{3.12}$$

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j} \quad j = 1, \ldots, k-1 \tag{3.13}$$

$$m_k = \sum_{j=1}^{k} \phi_{kj}\epsilon_{k-j} \tag{3.14}$$

$$v_k = (1 - \phi_{kk}^2)v_{k-1} \tag{3.15}$$

Choose each ϵ_k from $N(m_k, v_k)$.

Generation of the FARIMA($p, d, 0$) trace. In the following, we partly repeat formulae from Section 3.3.4 to obtain a selfcontained model description. Given $\alpha_1, \ldots, \alpha_p$ of the AR(p) part and a Gaussian FARIMA($0, d, 0$) trace $\{\epsilon_i : i = 1, \ldots, L + N\}$. Set μ_n and σ_n to the given estimates $\widehat{\mu}_n$ and $\widehat{\sigma}_n^2$ or in dependence of the expected mean μ_t and variance σ_t^2 of the model trace.

$$\mu_n = \log \frac{\mu_t^2}{\sqrt{\sigma_t^2 + \mu_t^2}} \tag{3.16}$$

$$\sigma_n^2 = \log \frac{\sigma_t^2 + \mu_t^2}{\mu_t^2} \tag{3.17}$$

We obtain a trace of the Gaussian FARIMA($p, d, 0$) process $\{t_i^m : i = 1, \ldots, L + N\}$ applying the recurrence relation

$$t_i^m = \alpha_1 t_{i-1}^m + \ldots + \alpha_p t_{i-p}^m + \epsilon_i \tag{3.18}$$

with $t_i^m = 0$ for $i < 1$. We throw away the first L samples to avoid start-up errors. The value L is determined by comparing the autocorrelation curve of the trace $\{t_i^m : i = L+1, \ldots, L+N\}$ and the theoretical autocorrelation curve of the FARIMA$(p,d,0)$ process. If both curves match well the value L is large enough. The marginal distribution of $\{t_i^m\}$ is Gaussian but not with the expected parameters. We therefore transform the t_i^m to the $N(\mu_n, \sigma_n^2)$ distributed t_i^n by

$$t_i^n \quad = \quad \frac{(t_{L+i}^m - \mu_m) \cdot \sigma_n}{\sigma_m} + \mu_n \quad \text{for} \quad i = 1, \ldots, N \tag{3.19}$$

with μ_m denoting the mean and σ_m^2 the variance of $\{t_i^m\}$.

Finally, the $L(\mu_n, \sigma_n^2)$ trace $\{t_i\}$ is generated from $\{t_i^n\}$ by

$$t_i \quad = \quad \exp t_i^n \tag{3.20}$$

Figure 3.8 shows a trace generated by an FARIMA(1,d,0) model with lognormal marginal distribution. The trace has appealing similarities to the empirical trace. As for the original, its marginal distribution is continuous and the average behavior changes slowly over time.

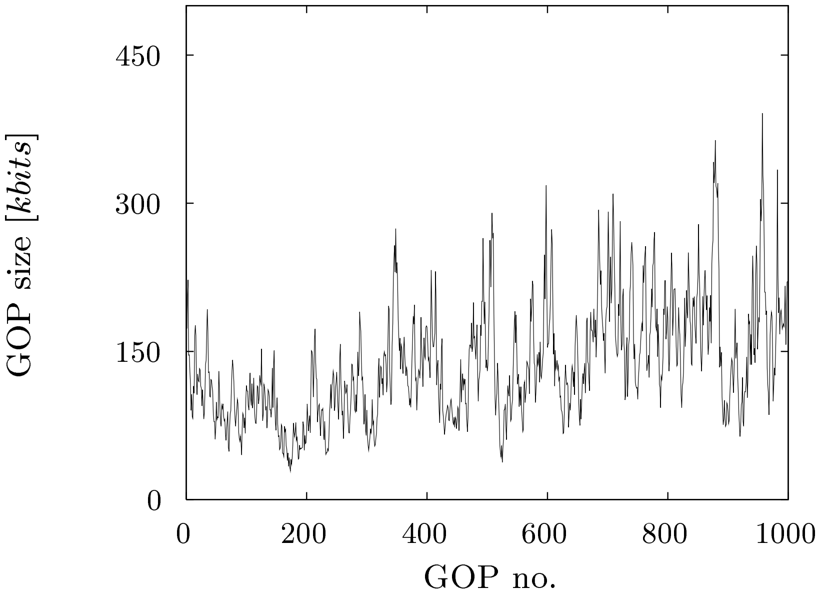


Figure 3.8: *GOP size trace generated by the selfsimilar model*

3.4 Summary

Table 3.2 classifies the models presented in this chapter by means of their marginal distribution and their autocorrelation function (ACF) properties.

Table 3.2: *Classification of the models*

	no correlations	ACF exponential	ACF sum of exponentials	ACF hyperbolic
discrete marginal distribution	histogram model	simple Markov chain model	scene-oriented model	—
continuous marginal distribution	lognormal white noise	AR(1) model	AR(p) model $p > 1$	self similar model

To complete the table, we added the lognormal distributed white noise process as the continuous equivalent of the histogram model. The state-based equivalent of the selfsimilar model requires an infinite number of states to obtain a hyperbolic ACF and is therefore not considered here. In addition, to the best knowledge of the author, there are no publications available about the parameter estimation of such Markov chains.

For illustration, Figure 3.9 shows the ACF for the first 200 lags of the *dino* GOP size sequence and the models with following parameters: simple Markov chain model with $M = 20$; scene-oriented model with $M_G = 20$, $M_S = 10$, and $W = 100$; autoregressive model with $p = 1$; selfsimilar model with $p = 1$. The ACF of the histogram model is not shown since the coefficients of correlation are 0 for lags larger than 0.

It is clearly visible that the selfsimilar model and the scene-oriented model lead to the best approximation quality of the empirical autocorrelation

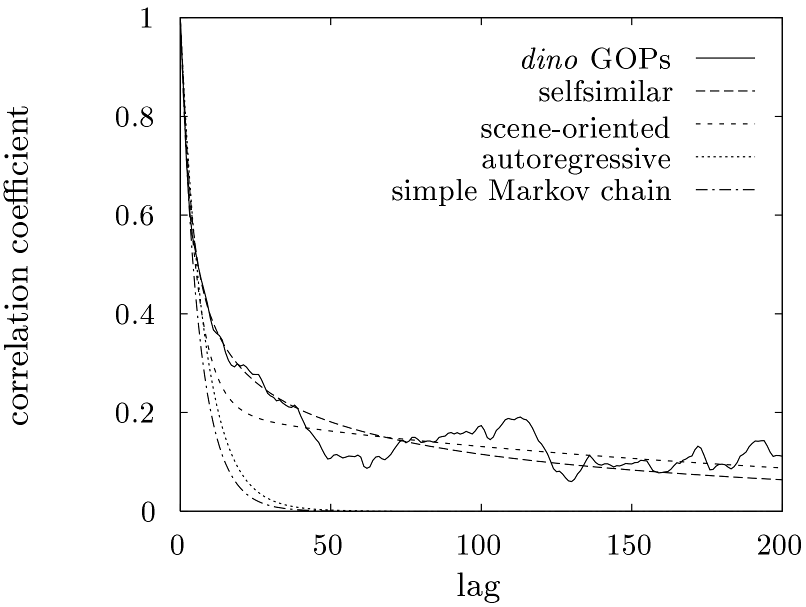


Figure 3.9: *Autocorrelation functions of the models*

up to lag 200 of the given *dino* data set. Despite the fact that the coefficient of correlation for a lag of 200 is larger for the scene-oriented model, it has to be noted that its ACF decays faster than the selfsimilar ACF for large lags. The autoregressive model and the simple Markov model show almost the same correlation behavior.

4 Model validation by simulation

In the previous chapter, we provided a comparison of the models based on their statistical properties and their sample traces. For the analysis of a queuing system, however, it is often difficult to decide in advance about the model characteristics necessary to provide useful performance estimates. In our case, the models range from the very simple histogram model to the complex scene-oriented or selfsimilar models. To illustrate how simulations support the model selection process, we present the studies of two important ATM scenarios in this chapter. The first scenario consists of a typical UPC scenario. We have a video traffic source whose Sustainable Cell Rate (SCR) is to be policed by a GCRA given a Burst Tolerance (BT). We are interested in those SCR/BT pairs where the cell loss or tag rate is below a given threshold. The second scenario is an ATM switch or concentrator multiplexing a number of video traffic input lines onto one output line. This scenario is of particular interest since we expect a number of problems for multiplexed VBR MPEG video traffic due to the periodic GOP pattern. In addition, this scenario can be used to evaluate multiplexing gains for video traffic or bandwidth estimators for CAC algorithms.

4.1 Introduction of the simulation setup

In the following, we give an introduction to the simulation we used to study the two scenarios. In both studies, we consider a queuing system with a finite buffer and a single server with constant service time. The buffer size is denoted by S and the service rate by B. The performance measures of interest are the cell loss probability P_{loss} and the mean waiting time of a cell W.

We make use of the *fluid simulation* approach to estimate the loss probabilities and delays (see Frost and Melamed (1994)). Instead of individual cell arrivals we consider frame data as a fluid, which flows into the buffer at a constant rate. There are two important benefits from the fluid approach. It is conceptually simple and it leads to enormous simulation speed-ups while the accuracy of the results is comparable to cell-oriented simulations. This method is applicable if the cell rate stays constant for a period of time which is considerably longer than the cell interarrival times. In our study, this is clearly the case since the rate stays constant for one frame duration. Due to the constant interarrival time of the frames, this bit rate is equal to *frame size* $\cdot$ *frame rate*, where for our data sets the frame rate r is $25s^{-1}$. Loss probabilities can be calculated in terms of overflow volumes. The average buffer content, which is needed to compute the mean cell delay, can be determined in terms of the time it takes to clear the current buffer.

The fluid simulation approach and the constant frame rate the simulation lead to a simple implementation of the simulation. Let us consider a frame sequence $\{f_i : i = 1, \dots, N\}$ from measurements or based on a model, a buffer size of S cells, and a system load of ρ. Since we assume that arriving frames are packetized into ATM cells with a payload of 48 octets we first have to transform each frame size f_i [bits] into $\widehat{f_i}$ [bits] by

$$\widehat{f_i} \quad = \quad \left\lceil \frac{f_i}{48 \cdot 8} \right\rceil \cdot 48 \cdot 8, \tag{4.1}$$

where $\lceil x \rceil$ is the smallest integer number larger than x. Let w_i denote the buffer content upon arrival of packetized frame i. The number of bits b served during a frame period of r^{-1} is given by

$$b = \frac{B}{r} = \frac{\widehat{\mu}}{\rho} \tag{4.2}$$

where $\widehat{\mu}$ is the sample mean of $\{\widehat{f}_i\}$. The system evolves as follows:

$$w'_{i+1} = \max\{w_i + \widehat{f}_i - b\,,\, 0\}, \tag{4.3}$$

$$w_{i+1} = \min\{w'_{i+1}\,,\, S\}. \tag{4.4}$$

The amount of loss l_i during interval i is given by

$$l_i = \max\{w'_{i+1} - S\,,\, 0\} \tag{4.5}$$

and the total loss probability P_{loss} by

$$P_{\text{loss}} = \frac{\sum\limits_{i=1}^{N} l_i}{\sum\limits_{i=1}^{N} \widehat{f}_i}. \tag{4.6}$$

Due to the constant service rate B the mean waiting time W of a cell is the average buffer occupancy divided by the service rate:

$$W = \frac{1}{N}\left[\frac{w_1}{2} + \sum_{i=2}^{N} w_i + \frac{w_{N+1}}{2}\right]\frac{1}{B}. \tag{4.7}$$

For all simulation experiments, we based our models on the *dino* data set. The results are essentially the same for our other data sets and Garrett's *starwars* MPEG data set.

4.2 Single input source

As mentioned above, a finite single server queuing system with a single input source can also be interpreted as a GCRA or a traffic shaper for a rate to be controlled of B and a CDVT or BT of S. The value P_{loss} gives the cell tag rate of the GCRA or the cell loss rate at the shaper buffer, respectively. The value W is the mean delay introduced by the shaper. Given a maximum value for P_{loss} and/or W, the user or the network provider intend to dimension the parameters B and S for a given MPEG sequence. We therefore provide diagrams which show the P_{loss} and W values for varying S and a constant B, i.e., a given system load ρ. In addition, we present diagrams where S is plotted against B for a given P_{loss}. These "iso-loss" curves are suggested in Lucantoni et al. (1994) for testing the accuracy of video models.

4.2.1 Frame and GOP regimes

Before we present a variety of curves comparing model trace and empirical trace results, we provide a diagram that highlights an interesting characteristic of a buffered VBR MPEG video traffic stream. In Figure 4.1 the solid line depicts the cell losses of a single MPEG stream for a system load of $\rho = 0.7$ in dependence of the buffer size in cells.

The striking feature of this curve is a knee at a buffer size of about 100 cells. The two other loss curves clearly show that this is the point where the system changes from frame to GOP regime. For buffer sizes of less than about 100 cells, the cell loss behavior can be reproduced with a trace that is generated simply from the histograms of each frame type based on the GOP pattern and which therefore does not inhere any GOP-by-GOP correlations. On the other hand, for buffer sizes of more than about 100 cells, the loss curve of the GOP size sequence is almost identical to that of the frame size sequence although no periodic behavior is modeled. The borderline of the two regimes is approximately three times the average frame size of about 35 cells. At the first glance, this result is surprising since one would expect the GOP length of twelve times the average frame size. Keeping in mind that due to our GOP pattern of "IBBPBBPBBPBB" a group of three consecutive frames is always formed by a large I- or P-frame and two small B-frames,

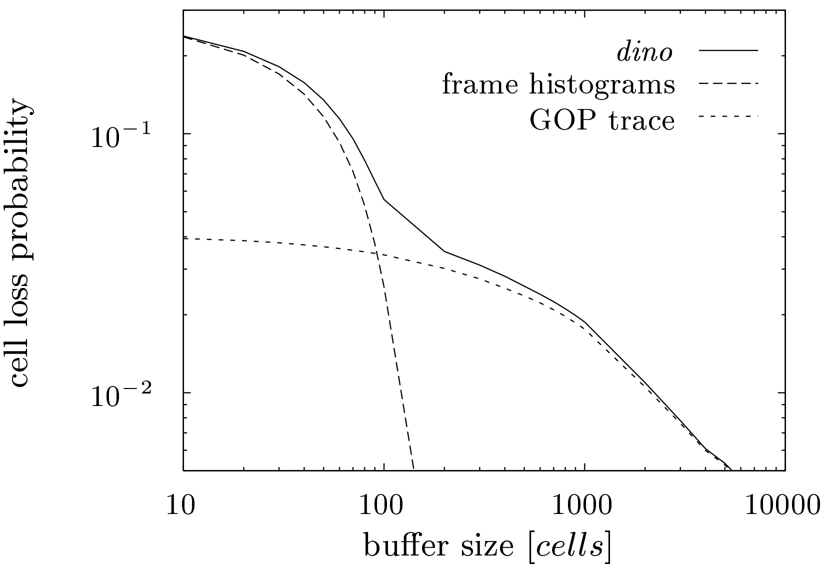


Figure 4.1: *Cell loss curve of dino frame size sequence ($\rho = 0.7$)*

it becomes clear that already a buffer size of three times the average frame size is large enough to filter out most of the periodic behavior of the video traffic stream. The cell curve has a knee for all loads larger than about 0.4. For lower loads a buffer of 100 cells already prevents from cell loss.

From modeling point of view, the following conjectures arise:

(1) For buffer sizes of less than 100 cells, simple models will already lead to good results if they reproduce the GOP pattern.

(2) For buffer sizes of more than 100 cells, models will only succeed if they mimic the GOP-by-GOP behavior in an adequate way.

4.2.2 Low load results

In the following, we compare the cell loss and average cell delays of the empirical data sets to those of the models. To avoid confusing diagrams, we separate the results according to the marginal distribution of the model traces, i.e, either discrete (histogram, simple Markov chain, and scene-oriented model) or continuous (autoregressive and selfsimilar model) and according to the data set which is used for the model parameter estimation, i.e., either GOP sizes or I-frame sizes. This results in the following four result groups: discrete GOP models, continuous GOP models, discrete I-frame models, and continuous I-frame models. Note, that the solid line always shows the curve for the *dino* frame size trace. The delays are not given in seconds but in multiples of the frame duration of 40 msec. The mean frame size is about 35 cells and the maximum frame size is 312 cells assuming a payload of 48 bytes per cell.

First, we present the results for a system load of $\rho = 0.3$. In our single source scenario low loads make sense since such a system is equivalent to a GCRA with a SCR close to the peak rate of the traffic stream. We do not provide the diagrams for the continuous GOP results since they are the same in tendency as the discrete ones.

Figure 4.2 shows that discrete GOP models clearly overestimate the cell losses for all buffer sizes. This is also the case for the continuous GOP models.

The delay predictions of the discrete and continuous GOP models tend to overestimate the empirical values for buffer sizes of less than 100 cells and to underestimate them for more than 100 cells (cf. Figure 4.3). As soon as the buffer is large enough to prevent losses, the delay stays on the same level. It is worth mentioning that for a load of 0.3, all models show almost the same behavior independently of their very different complexity. This indicates that for buffer sizes of less than 100 cells a good approximation of the GOP-by-GOP correlation behavior is not necessary.

Figures 4.4 and 4.5 show losses and delays for the discrete I-frame models. Here, we observe a very good match to the empirical curves. Thus, for low loads and buffer sizes of not more than 100 cells the model should be based only on the frame traces and that a modeling of statistical properties of the GOP traces does not improve the model quality. Since all I-frame models show the same accuracy it is sufficient to use the simple histogram model.

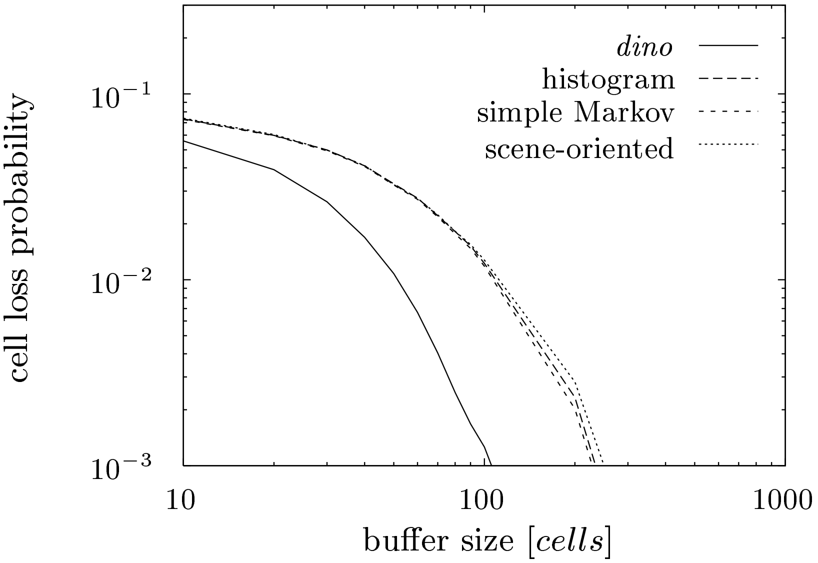


Figure 4.2: *Cell losses for discrete GOP models* ($\rho = 0.3$)

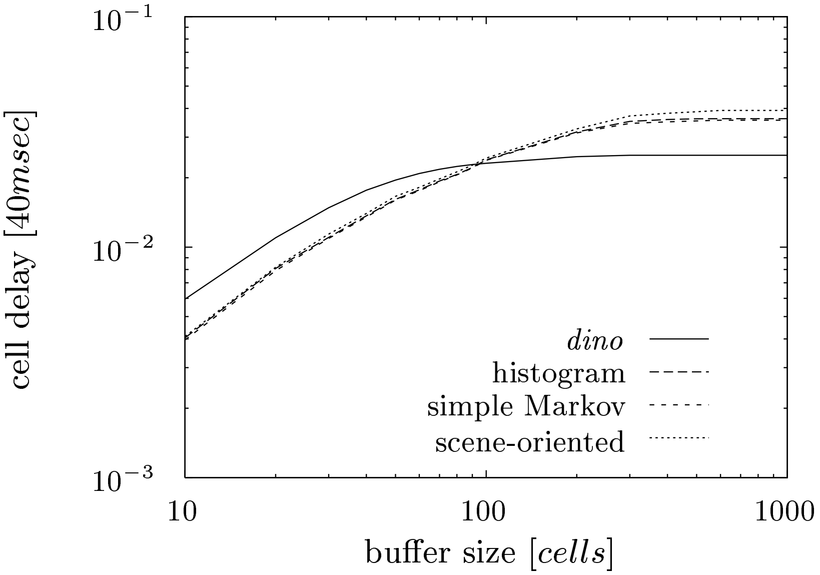


Figure 4.3: *Cell delays for discrete GOP models* ($\rho = 0.3$)

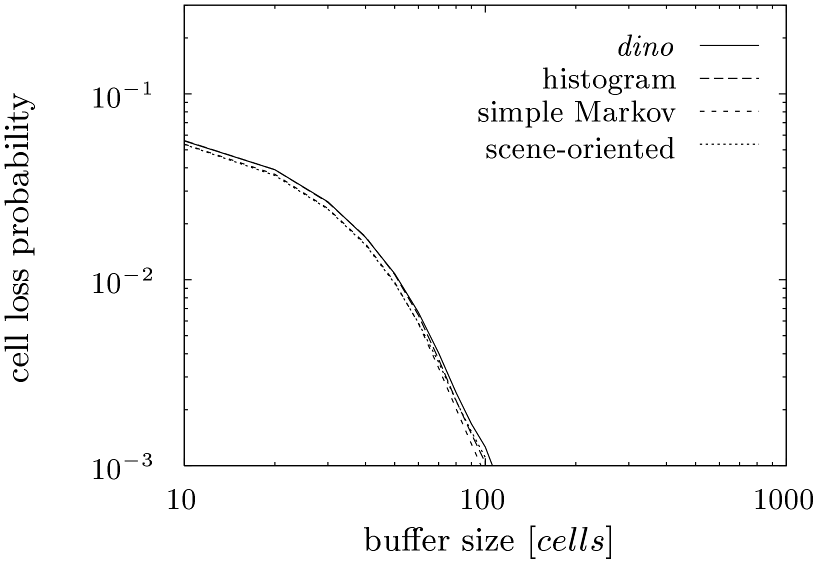


Figure 4.4: *Cell losses for discrete I-frame models ($\rho = 0.3$)*

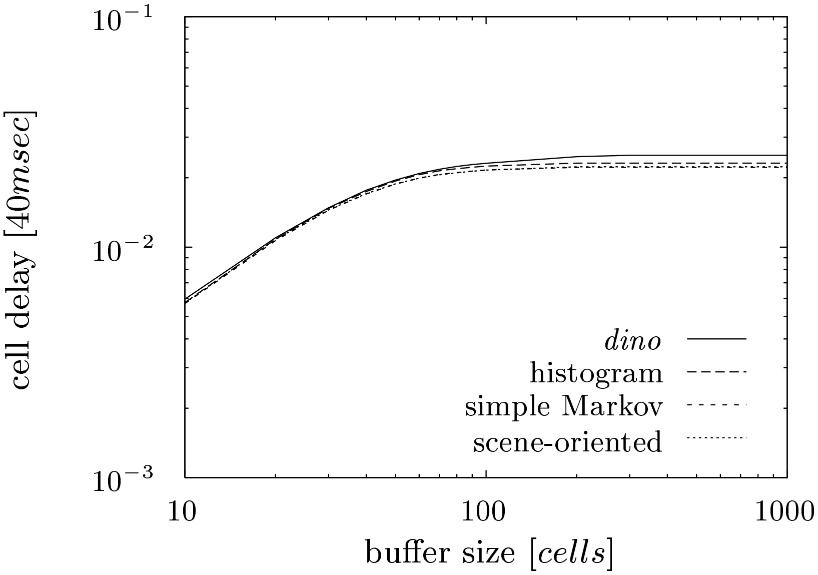


Figure 4.5: *Cell delays for discrete I-frame models ($\rho = 0.3$)*

4.2.3 High load results

Due to the graceful change in the model behavior while increasing the load, we only show a set of diagrams for a load of $\rho = 0.9$. Such a system is equivalent to a GCRA with a SCR close to the mean rate of its input traffic. In case of small buffer sizes or burst tolerances we have to expect a large amount of cell losses whereas for large buffer sizes average cell delays grow considerably.

Figures 4.6 to 4.10 show the losses and delays for all GOP models. In contrast to a load of 0.3, the model accuracy depends on the ability of the model to approximate the GOP-by-GOP behavior of the empirical trace. The histogram model only provides good approximation results for buffer sizes of less than 100 cells. Both lag-1-correlation models (autoregressive and simple Markov chain model) lead to good results up to buffer sizes of a few thousand cells. For buffer sizes of more than ten thousand cells, the scene-oriented model slightly underestimates the losses and the selfsimilar model overestimates the cell losses. As far as the delays are concerned, both models lead to very accurate estimates for the whole range of considered buffer sizes. All models besides the histogram models show the knee at 100 cells buffer size independently from discrete or continuous marginal distributions.

The I-frame models are less accurate than expected even for buffer sizes of less than 100 cells (cf. Figure 4.10). They are clearly outperformed by the GOP models for both losses and delays over the whole range of considered buffer sizes.

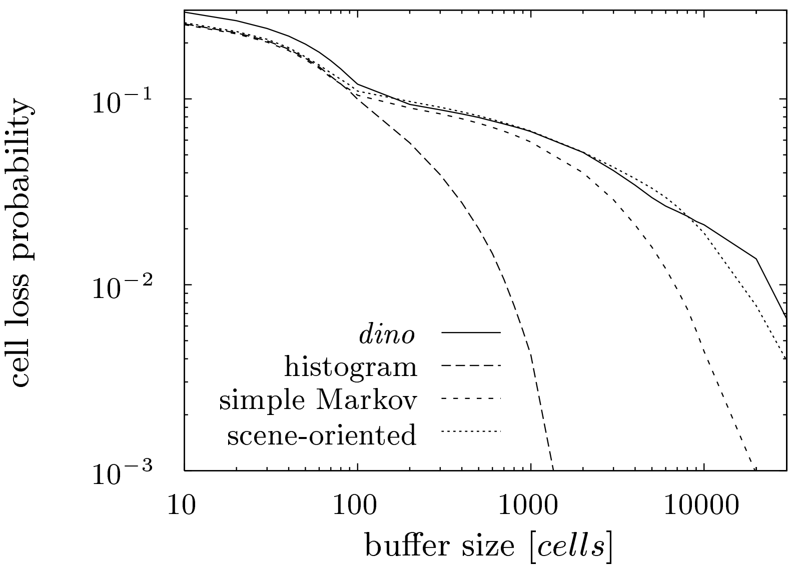


Figure 4.6: *Cell losses for discrete GOP models ($\rho = 0.9$)*

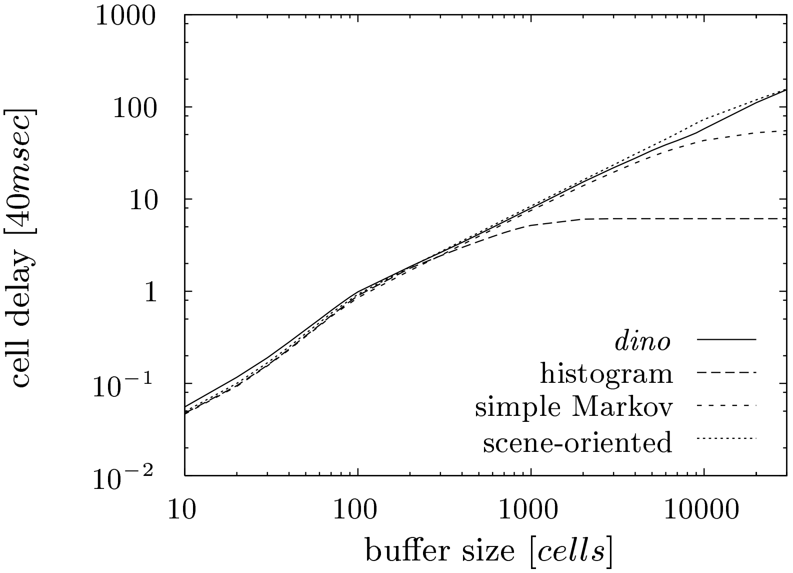


Figure 4.7: *Cell delays for discrete GOP models ($\rho = 0.9$)*

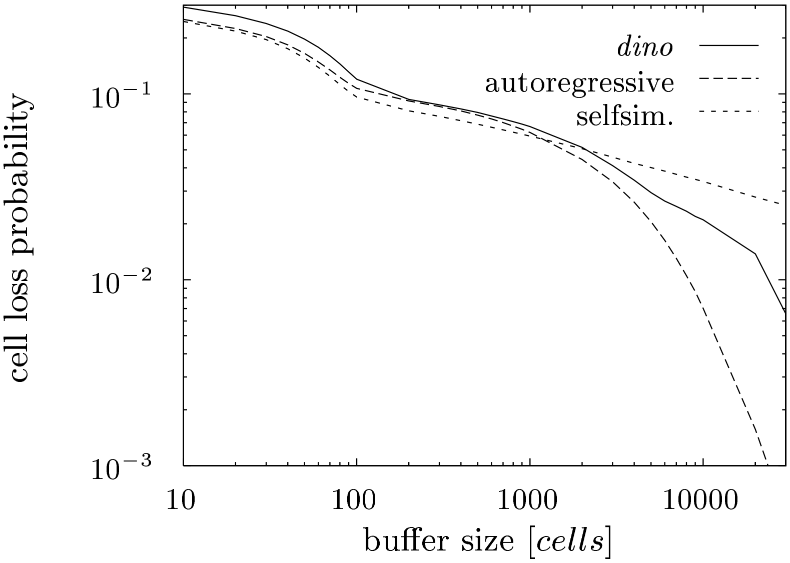


Figure 4.8: *Cell losses for continuous GOP models ($\rho = 0.9$)*

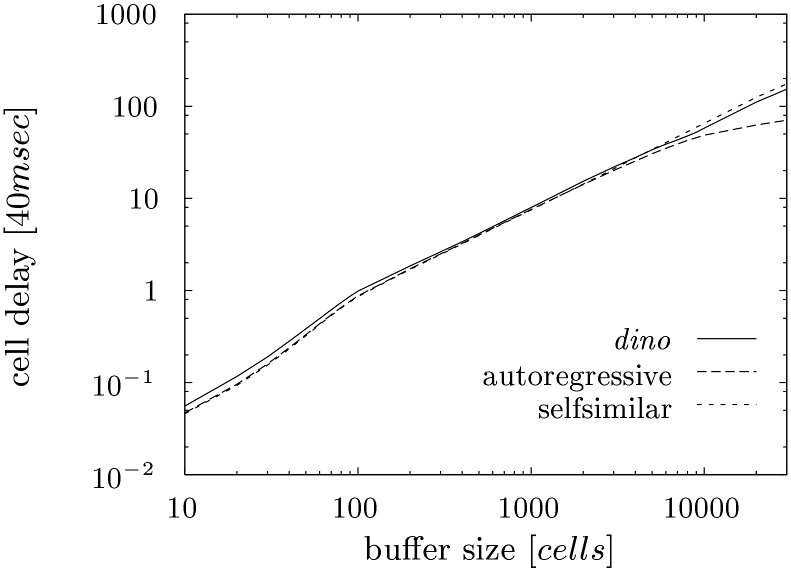


Figure 4.9: *Cell delays for continuous GOP models ($\rho = 0.9$)*

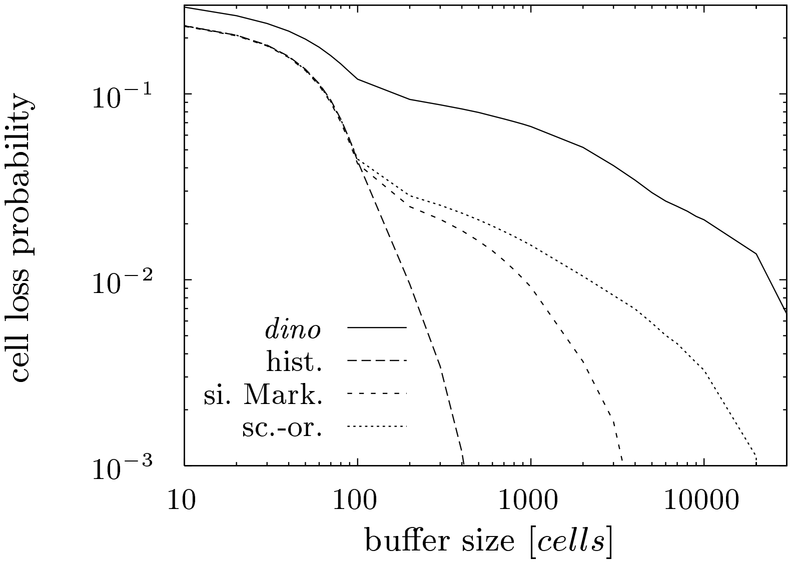


Figure 4.10: *Cell losses for discrete I-frame models ($\rho = 0.9$)*

4.2.4 Iso-loss curves

The next set of diagrams (Figures 4.11 to 4.14) shows the the empirical and model iso-loss curves, i.e., for a given target loss rate we plot the buffer size versus the SCR of the GCRA. For simplicity, we transform the SCR into the amount of data transmitted during one frame duration. This amount ranges from about the average frame size to that value where the target loss rate can be met even without a buffer. The curves were generated by means of an iterated bisection of the buffer sizes given the amount of transmitted data and the target loss rate. Considering a maximum buffer size of 100 000 cells, this approach takes 17 bisections for each data point. We provide curves for target loss rates of 0.01 and 0.001. For smaller target loss rates, the curves become problematic from statistical point of view. For higher loss rates than shown here, say 0.1, both the GOP and the I-frame histogram model lead to an accurate prediction of the iso-loss curve. From practical point of view, however, this curve is obsolete since no customer is willing to dimension a GCRA for a target loss rate of the order of 10 %.

Figures 4.11 and 4.12 show that GOP models can only be used if the expected buffer size is larger than 100 cells. For smaller buffer sizes, the results become very inaccurate. For a SCR of less than say 70 cells per frame duration in the $P_{\text{loss}} = 10^{-2}$ case and 100 cells per frame duration in the $P_{\text{loss}} = 10^{-3}$ case, the simple Markov chain approach matches well the empirical results whereas the histogram approach is not applicable. The results for the scene-oriented model are not shown since there is no substantial improvement compared to the simple Markov chain model. For a large SCR, the I-frame histogram model leads to very accurate results (cf. Figures 4.13 and 4.14). Figure 4.13 also shows that even the highly correlated scene-oriented model cannot be used for a SCR close to the mean rate of the video sequence.

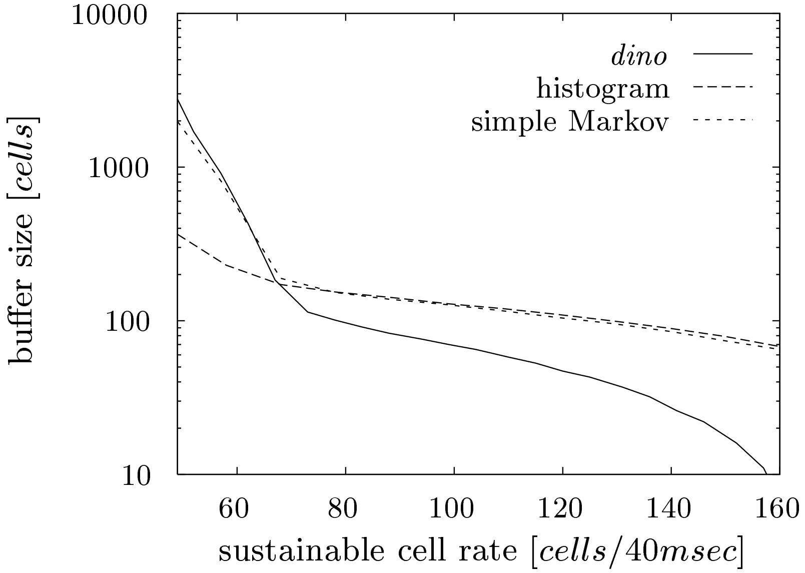


Figure 4.11: *Iso-loss curves for discrete GOP models ($P_{loss} = 10^{-2}$)*

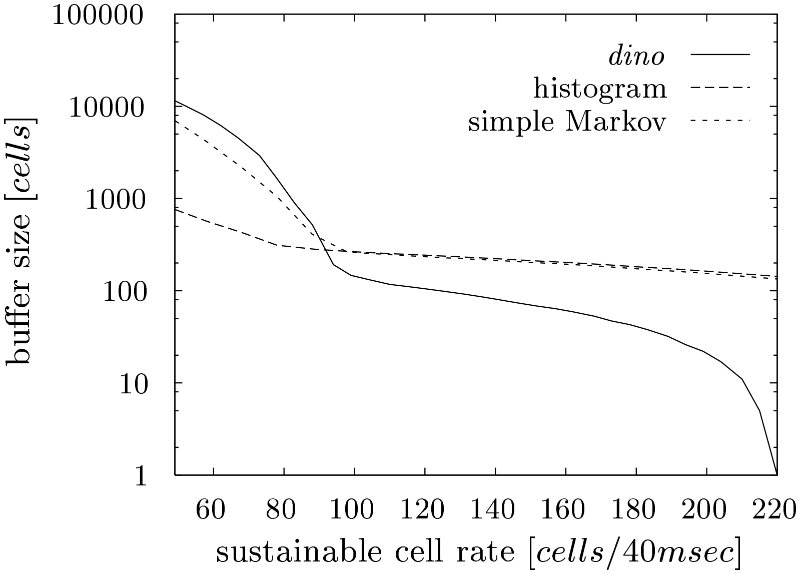


Figure 4.12: *Iso-loss curves for discrete GOP models ($P_{loss} = 10^{-3}$)*

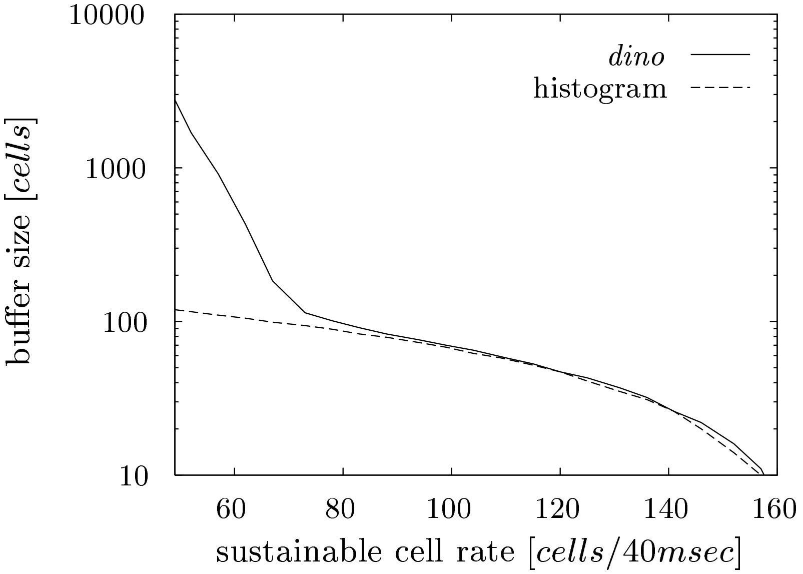


Figure 4.13: *Iso-loss curves for discrete I-frame models ($P_{loss} = 10^{-2}$)*

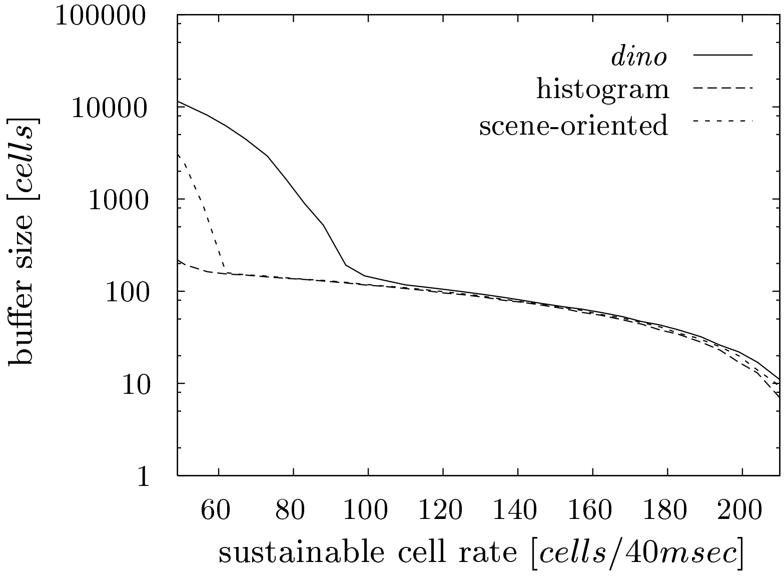


Figure 4.14: *Iso-loss curves for discrete I-frame models ($P_{loss} = 10^{-3}$)*

4.2.5 Summary

After presenting the cell loss and delay results, we reconsider our conjectures on the required model properties. The various curves show that the model accuracy does not only depend on the buffer size as assumed but also on the system load. For load of say less than 0.5, the I-frame models outperform the GOP models for buffer sizes of less than 100 cells. In all other cases, i.e., buffer sizes of more than 100 cells or loads of more than 0.5, GOP models should be used for the prediction of cell losses and delays. The decision about the model class has to be based on the buffer size. For buffers of less than 100 cells, a histogram model is already very accurate. For larger buffer sizes, lag-1-correlation models should be applied. Only for very large buffer sizes it necessary to use models with improved correlation properties.

The conclusion for the approximation of iso-loss curves is twofold. For resulting buffers of less than 100 cells, i.e., a SCR close to the peak rate of the video sequence, I-frame histogram models are adequate. For buffers of more than about 100 cells, i.e., a SCR close to the mean rate, GOP-based lag-1-correlated models such as the simple Markov chain should be considered.

To sum up, in most cases GOP-based histogram or lag-1-correlated models are a good choice for the prediction of the performance measures for a single video traffic source.

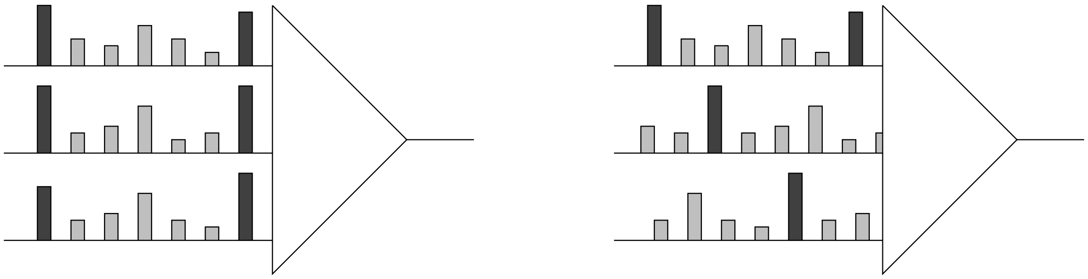
4.3 Multiplexed input sources

Besides the estimation of GCRA parameters, the estimation of possible multiplexing gains and adequate buffer sizes for a number of superposed MPEG video traffic streams is of high interest. Before we present performance results, we first have to consider the implications of multiplexing traffic streams which inhere a periodic structure.

4.3.1 Cross-correlation effects

The phase-shift with respect to the frame types of the GOP pattern introduces several performance and fairness problems. Since the average I-frame is large compared to P- and B-frames, system resources which were provided based on average bit rates will run short if too many video sources send I-frames at the same time. Moreover, the shift patterns may lead to unfairness for sources whose I-frames lie close together since they experience more losses than other sources.

For the rest of this chapter, we assume that all streams to be multiplexed have the same statistical properties and that the frames of all streams start at exactly the same instant. This is a restriction compared to real systems since there the frames may start at arbitrary instants. On the other hand, this assumption will simplify the analysis and simulation of the system considerably without introducing principle changes in its properties. Even with this simplifying assumption, we have a huge number of different phase shift opportunities for N multiplexed video streams with a GOP length of 12 frames. To our best knowledge, there is no formula published for the number of opportunities. Standard combinatorial formulae cannot be applied since both the pattern of parallel I-, P-, and B-frames and the periodic GOP pattern have to be taken into account. Andreassen (1995) provides an approximation for such a formula where P- and B-frames are not considered separately. Nevertheless, it is possible to identify worst case and optimal case scenarios. The worst case consists of the high source alignment where the I-frames of all sequences are transmitted in parallel (cf. Figure 4.15(a)). The optimal case is a low alignment scenario where the maximum number of commonly multiplexed I-frames is minimal (cf. Figure 4.15(b)). The lower bound for the cell losses in the optimal case are is given by the cell loss curves for multiplexing sources where the bit-rate is averaged over one GOP, i.e., where the periodic GOP pattern is filtered out.



(a) High alignment

(b) Low alignment

Figure 4.15: *Alignment scenarios*

In real systems, both extreme cases are possible but the average behavior is of greater interest. Since extensive simulation of all possible start scenarios is not feasible we decided to determine the start frame type of each input stream at random and to conduct a large number of simulation runs with random sets of start frames. We found that for a number of about 2000 runs the loss and delay estimates were stable. This number had to be chosen heuristically since confidence intervals are not available for this kind of simulation experiment. For the computation of the confidence intervals it would be necessary to know all equivalence classes of the shift patterns with regard to the resulting loss, i.e., patterns belong to the same class if the simulation will lead to the same cell losses. For instance, if we assume a GOP pattern of "IPP" and multiplex two sources the possible shift patterns are

$$\begin{pmatrix} I & P_1 & P_2 \\ I & P_1 & P_2 \end{pmatrix} \begin{pmatrix} I & P_1 & P_2 \\ P_2 & I & P_1 \end{pmatrix} \begin{pmatrix} I & P_1 & P_2 \\ P_1 & P_2 & I \end{pmatrix}$$

where the middle and the right pattern are equivalent with regard to cell losses. The estimation of the equivalence classes for longer GOP patterns and larger numbers of multiplexed streams is not trivial and beyond the scope of our work.

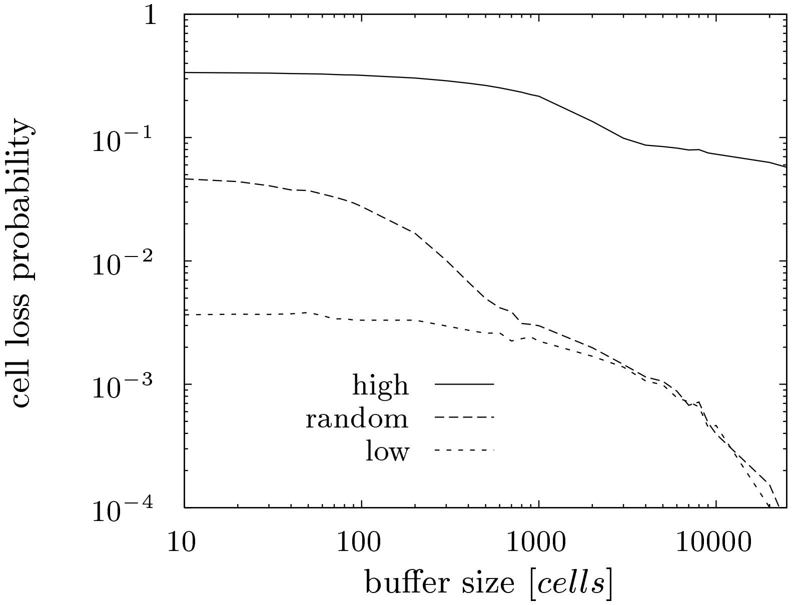


Figure 4.16: *Cell loss curve of 24 multiplexed dino frame size sequences with different alignments ($\rho = 0.9$)*

In Figure 4.16, the cell loss curves of 24 traces and different alignments are depicted. In the case of high alignment, i.e., if all traces are in phase, the losses are very high and do not decrease considerably even for large buffers. For small buffer sizes, the low alignment case leads to a smaller cell loss rates than the random case but for increasing buffer sizes the two curves converge. The reason for this striking result is that periodic effects are filtered out by larger buffers. In the following, we give an explanation of this behavior.

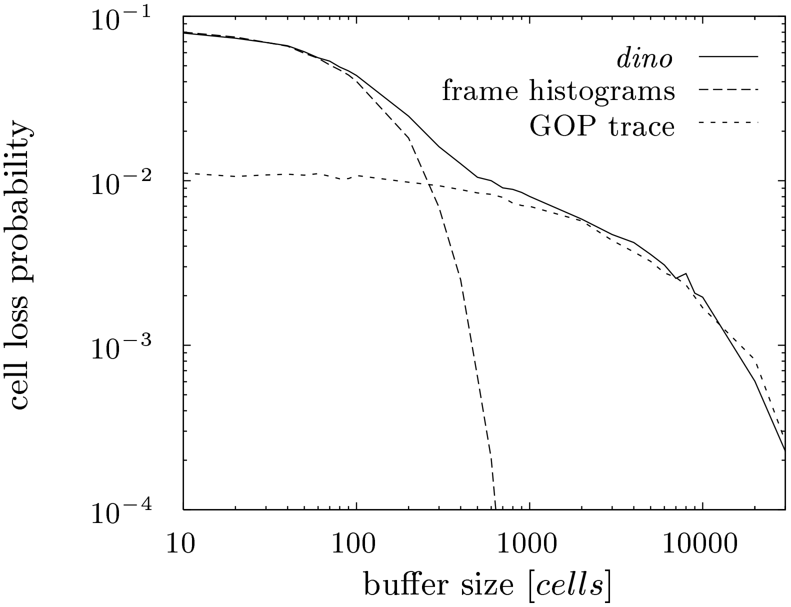


Figure 4.17: *Cell loss curve of 12 multiplexed dino frame sequences ($\rho = 0.9$)*

Figure 4.17 shows the cell losses for 12 multiplexed *dino* frame size sequences with random alignment. As in the single source case, we observe a knee in the curve. In contrast to that case, however, the knee is located at about 12 times the mean frame size (about 400 cells) and not at about 3 times the mean frame size (about 100 cells). This means that due to the random alignment multiplexing, we have to buffer the average amount of data of a GOP to filter out the periodic frame pattern. As a consequence, it is not necessary to use a model which reproduces the periodic behavior of the frame trace if the buffer size becomes large enough.

From the results presented above, there are a number of conclusions with respect to MPEG transmission over ATM networks. First, when a number of

MPEG video sources are fed into a single multiplexer, the cell loss probability is very sensitive to the alignment among the GOP structures of these sources. Consequently, it will be very difficult to guarantee the cell loss probability for a multiplexer (or alternatively to perform the CAC and UPC functions of the network) without knowing in advance how the GOP structures of the video sources are aligned.

Therefore, it would be useful for the network to be able to control this alignment. However, this is not feasible in practice. Hence, there can be no guarantee that a performance close to the best case shown in our plots will be achieved. In fact, it is certain that sometimes the performance will approach the worst case, with cell loss probabilities in the order of 10 %.

Smoothing the cell stream over intervals larger than a frame will only lead to improvements for smaller multiplexer buffer sizes. For larger buffers this will have no effect since the losses are caused by correlation effects on a GOP time scale. These effects are hard to reduce without introducing large delays. The delay requirements of the application will dictate the amount of burstiness that can be reduced by buffering or smoothing.

4.3.2 Statistical multiplexing gain

Another question which often arises in ATM networks is whether there is a gain in bandwidth by multiplexing a number of traffic streams while providing a particular QoS, say a certain maximum cell loss rate, or, in other words, which bandwidth is required for an individual traffic stream to meet a given QoS (see Kelly (1996)). In Figure 4.18, we give the bandwidth relative to the average bandwidth of the video stream which is required for a single stream to guarantee cell losses of less than 10^{-3} and 10^{-4} for a multiplexer with a buffer size of 100 cells versus the number of multiplexed sources.

Given the number of sources the required bandwidth is determined by a bisection of the bandwidth interval ranging from the average bandwidth, corresponding to the loss rate for a multiplexer load of $\rho = 1.0$, to the

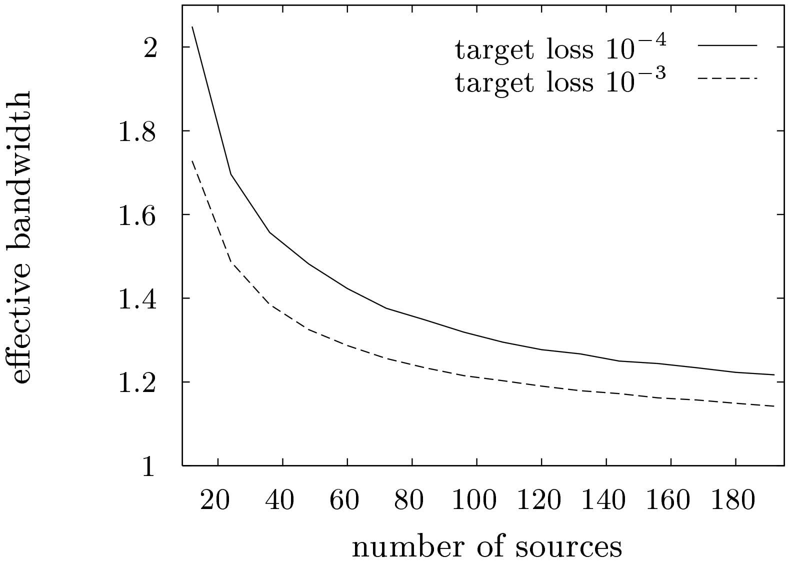


Figure 4.18: *Effective bandwidth (buffer size: 100 cells)*

peak bandwidth, corresponding to zero losses. The required bandwidth of a single video stream, also known as effective bandwidth, decreases rapidly from about two times the average bandwidth for 12 multiplexed sources and a maximum cell loss rate of 10^{-4} to about 1.5 times for 50 sources. For larger numbers of sources, the required bandwidth decreases more slowly. In principal, the same behavior can be observed for the 10^{-3} loss curve. Garrett and Willinger (1994) obtain similar results for their *starwars* data set. It is important to note that there is no linear decrease in required bandwidth. This makes it difficult to find simple approximations of this curve which are necessary for CAC functions. Chou and Chang (1996) report that they had problems to apply standard methods to find an effective bandwidth formula for video traffic due to the presence of long-range dependence in the data sets.

4.3.3 Cell losses

Now, we present the cell loss curves for multiplexed model traces under moderate and high load conditions. We restrict ourselves to a small number of diagrams since the behavior is essentially the same as in the single source case apart from the fact that the change from frame- to GOP-dominated behavior can be observed at buffer sizes of about 400 cells and not at about 100 cells.

In Figure 4.19, we present the cell loss curves for 24 multiplexed traces for an offered load of $\rho = 0.6$. The I-frame-based histogram model leads to a good approximation of the empirical *dino* trace losses whereas the GOP-based model clearly overestimates the cell losses. This observation holds if the load is small enough such that a buffer of less than about 400 cells avoids cell losses.

Figures 4.20 and 4.21 show the cell loss estimates for a variety of buffer sizes and a multiplexer load of $\rho = 0.9$ for models with discrete and continuous marginal distributions, respectively. As in the single source case, the histogram model can only be used for buffer sizes left of the knee. The first-order correlation models lead to a good approximation quality for a large range of buffer sizes. Only for very large buffer sizes of several ten thousand cells, higher-order models such as the scene-oriented or selfsimilar models should be applied.

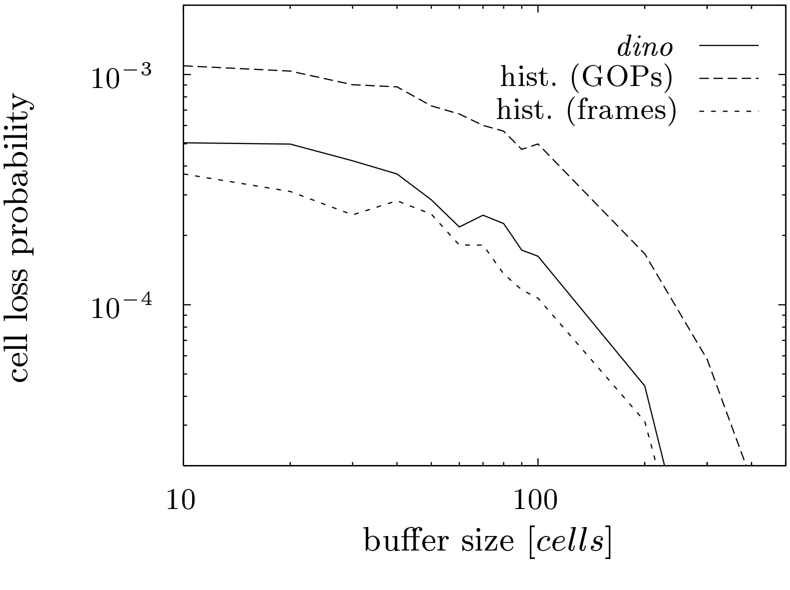


Figure 4.19: *Cell losses for 24 histogram model traces ($\rho = 0.6$)*

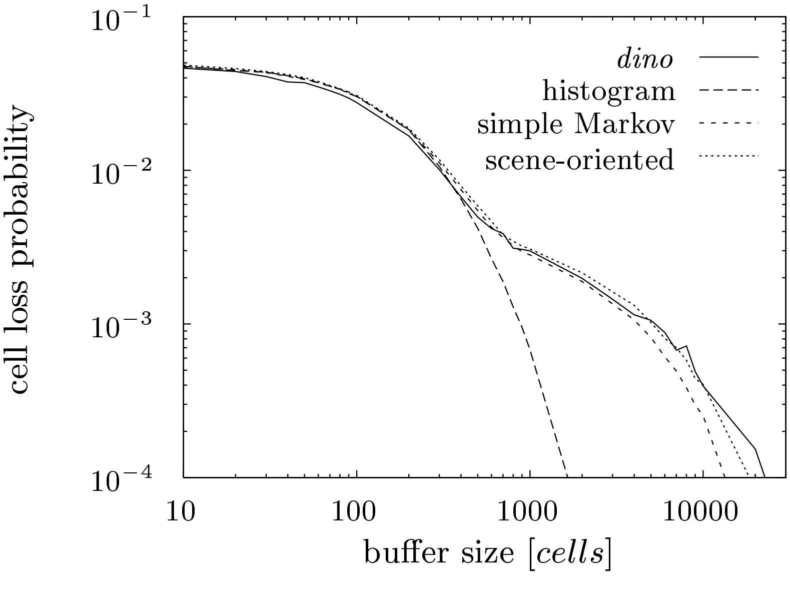


Figure 4.20: *Cell losses for 24 discrete GOP model traces ($\rho = 0.9$)*

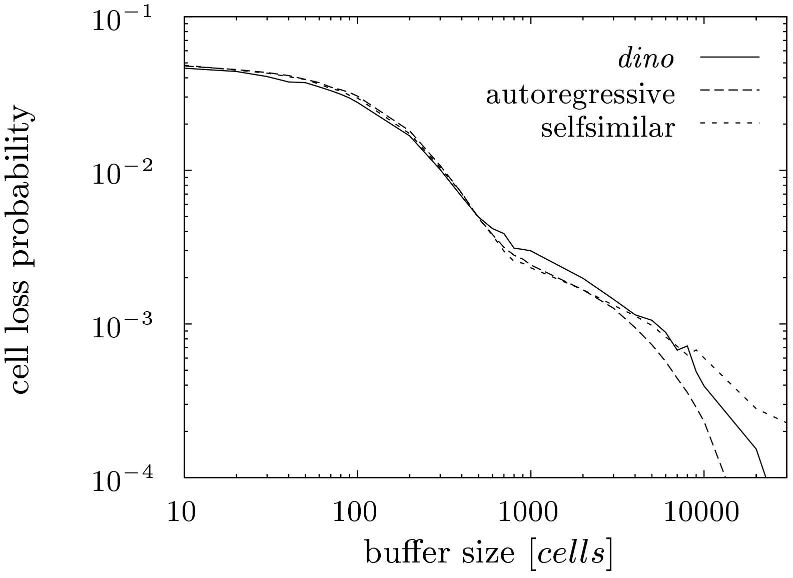


Figure 4.21: *Cell losses for 24 continuous GOP model traces ($\rho = 0.9$)*

4.3.4 Summary

The simulation results of the previous sections lead to a number of conclusions concerning the selection criteria for video traffic models. As expected, the most important system property is the buffer size. In other words, the system memory capacity determines the correlation capabilities a video model should have. In this context, we are able to identify three buffer size intervals: dominated by the periodic frame pattern, dominated by the GOP-level correlations, transition from frame to GOP regime.

If buffer sizes are smaller than the number of frames of one GOP times the average frame size the cell loss and delay results are dominated by the periodic frame pattern and considerably larger than results which are based on GOP size traces, i.e., traces without that particular periodic pattern. In the single source case, it is also possible to filter out the periodic effects for buffer sizes smaller than the average GOP length (cf. Section 4.2). This effect, however, cannot be expected in general since it depends on the particular GOP pattern used to encode the sequence. Under the frame regime, histogram models lead to good approximations of the cell loss and delay result of the empirical traces. It is mandatory to model the periodic frame pattern but not the GOP-by-GOP correlations. For moderate loads, the model parameters should be computed from the I-frame size traces, whereas for high loads, the GOP size traces should be used. The conclusions of Enssle (1996) corroborate our result that for small to moderate buffer sizes the long-range dependence effects have no influence on multiplexer cell loss results and hence have not to be modeled.

If the buffer becomes large enough to filter out the periodic frame pattern models can be based on GOP size traces and the level of complexity is considerably reduced. On the other hand, it now becomes important for the model to reflect the GOP-level correlations. For a wide range of buffer sizes, models which consider only lag-1 correlations, such as first-order Markov chains or autoregressive processes, are appropriate. Only for very large buffers, say tens of thousands of cells, models with improved correlation capabilities have to be applied.

The most complex models have to be applied for buffer sizes at about the change from frame to GOP regime. They have to include both the periodic frame pattern and GOP-level lag-1 correlations to obtain a good approximation quality.

If a model for the whole range of buffer sizes is required, either selfsimilar or scene-oriented type of models based on GOP size traces should be considered. Then, one has to be aware of the fact that these models overestimate the cell losses for low traffic intensities or multiplexer loads. The average delays predicted by these models are less affected by varying load conditions.

5 Application examples of the models

Up to now, we used the VBR MPEG video models in simulation studies only. In this chapter, we devote our attention to two examples indicating how easy-to-obtain derivates of the histogram and the simple Markov chain model can be used in combination with analytical methods. In both examples, we apply algorithms to compute cell losses which are based on the discrete-time analysis approach of Tran-Gia and Ahmadi (1988). In contrast to our simulation studies, we base our models on the Bellcore *starwars* data set, since, due to its large number of frames, the parameter estimation of the models becomes more reliable. The first example, which we called *frame-based analysis*, shows how a fluid flow simulation of a buffer with a single video source can be replaced by an analysis even if the model reflects correlations over a few lags. The second example, named *cell-based analysis*, shows how analytical methods can be used to study UPC-parameter dimensioning problems of VBR video traffic sources.

5.1 Frame-based analysis

For our analysis, we consider a one-stage queuing system with a single video traffic source. The buffer is of finite length and the service time is constant. Figure 5.1 depicts the system and provides the most important system parameters.

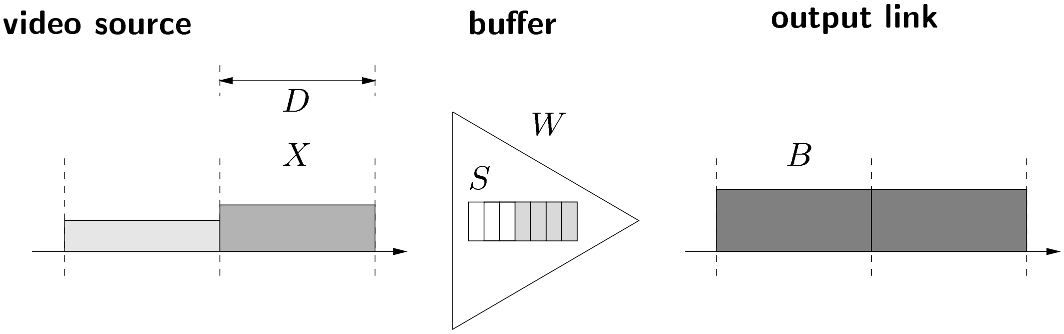


Figure 5.1: *Buffer with a single video traffic source*

For system analysis, we make the following assumptions:

- ◇ The time is discretized into frame durations D, i.e., the reciprocal of the frame rate r, and synchronized to frame starts.
- ◇ All data is discretized into ATM cells carrying 48 octets of payload, i.e., the frame sizes, the buffer content, and the amount of data transmitted during a frame duration.
- ◇ Cells of a single frame are regarded as a fluid according to the approach presented in Section 4.1.

5.1.1 Analysis without GOP correlations

First, we analyze the system using a slightly modified histogram model. Instead of using the GOP size histogram and scaling factors, we directly use the frame size distributions of the I-, P-, and B-frames derived from the video traces. Note that this model reflects the GOP pattern but does not contain any GOP-by-GOP correlations. We will use the term *uncorrelated model* to refer to this model.

The following notation is used to characterize the system:

X_n : random variable for the number of cells in the nth frame,

B : random variable for the number of cells transmitted during one frame duration D; in our case B is deterministic and we denote the constant number of cells transmitted by b,

W_n : random variable for the number of cells waiting in the buffer upon the arrival of the nth frame,

S : buffer size.

The random variables X_n, B, and W_n follow discrete distributions $x_n(k)$, $b(k)$ and $w_n(k)$, where, for instance, $x(i) = \Pr\{X_n = i\}$.

Due to our fluid-flow assumption, we are able to simplify the computation of the distribution of the buffer content $w_n(k)$ considerably as compared to standard unfinished work approaches such as, e.g., by Tran-Gia and Ahmadi (1988). The system evolution is determined by the following equation:

$$W_{n+1} \quad = \quad \min\left\{\max\{W_n + X_n - b, 0\}, S\right\}. \tag{5.1}$$

At first glance, our approach looks like a typical batch arrival process with batch size of X_n. In contrast to the original method, however, we already subtract the maximum amount of data which can be transmitted during a frame duration b upon arrival of frame n. This has to be done due to the fluid-flow assumption and the discretization of the system time into frame durations. After the arrival of one frame as a batch of cells, we have to subsume in one equation the behavior of the modeled system until the next frame arrival, i.e., equally spaced cells entering a buffer which is served at a constant rate. Integrating both arrival and service process over D, and taking into account the content of the buffer before the frame arrival as well as the limited buffer size leads to Eqn. (5.1). For illustration, Figure 5.2 shows a snapshot of the system evolution containing the three possible cases. If the buffer content plus the new video data minus the amount of data which can be transmitted exceeds the buffer size then cells are lost (left bar). If more data can be transmitted than the buffer content plus arriving data then the buffer runs empty (center bar). In all other cases the buffer contains cells and there are no losses during the frame duration (right bar).

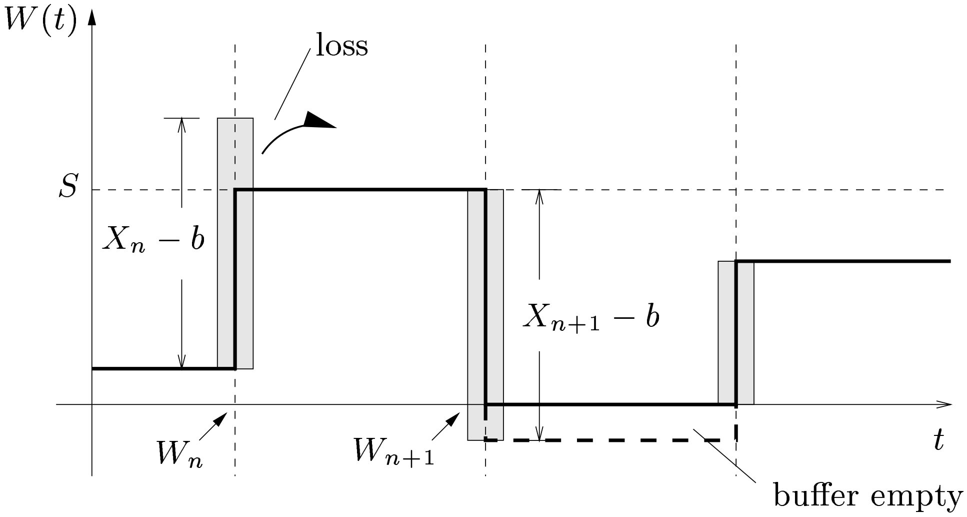


Figure 5.2: *Snapshot of the system evolution*

Eqn. (5.1) leads to the following recursion for the distribution of the buffer content:

$$w_{n+1}(k) \quad = \quad \pi^S \circ \pi_0 \left[w_n(k) \circledast x(k) \circledast \delta(k+b) \right]. \tag{5.2}$$

Here, $\circledast$ denotes the discrete convolution, π_0 and π^S are sweep operators, and $\delta(k+b)$ is the shifted Kronecker delta. They are defined as follows.

◇ *Discrete convolution.*

$$c(k) \quad = \quad a(k) \circledast b(k) \quad = \quad \sum_{j=-\infty}^{\infty} a(k-j) \cdot b(j). \tag{5.3}$$

◇ *Low sweep operator.*

$$\pi_0[a(i)] \quad = \quad \begin{cases} 0 & : \quad i < 0, \\ \sum\limits_{j=-\infty}^{0} a(j) & : \quad i = 0, \\ a(i) & : \quad i > 0. \end{cases} \tag{5.4}$$

◇ *High sweep operator.*

$$\pi^S[a(i)] \quad = \quad \begin{cases} a(i) & : \quad i < S, \\ \sum\limits_{j=S}^{\infty} a(j) & : \quad i = S, \\ 0 & : \quad i > S. \end{cases} \tag{5.5}$$

◇ *Kronecker delta.*

$$\delta(i) \quad = \quad \begin{cases} 1 & : \quad i = 0, \\ 0 & : \quad \text{otherwise.} \end{cases} \tag{5.6}$$

Figure 5.3 provides a pictorial description of Eqn. (5.2). In addition, Figure 5.4 shows a diagram of the computation of the buffer content distribution. This diagram constitutes the basic building block of the algorithms presented in the following.

Assuming that the buffer content distribution of the multiplexer converges to a steady state, $w(k) = \lim_{n\to\infty} w_n(k)$ is the solution of the fix-point equation

$$w(k) \quad = \quad \pi^S \circ \pi_0 \left[w(k) \circledast c(k) \right], \tag{5.7}$$

where

$$c(k) \quad = \quad x(k) \circledast \delta(k+b). \tag{5.8}$$

Eqn. (5.7) is the discrete-time analogue of the Lindley integral equation of $GI/GI/1-S$ queuing systems (see Kleinrock (1975)).

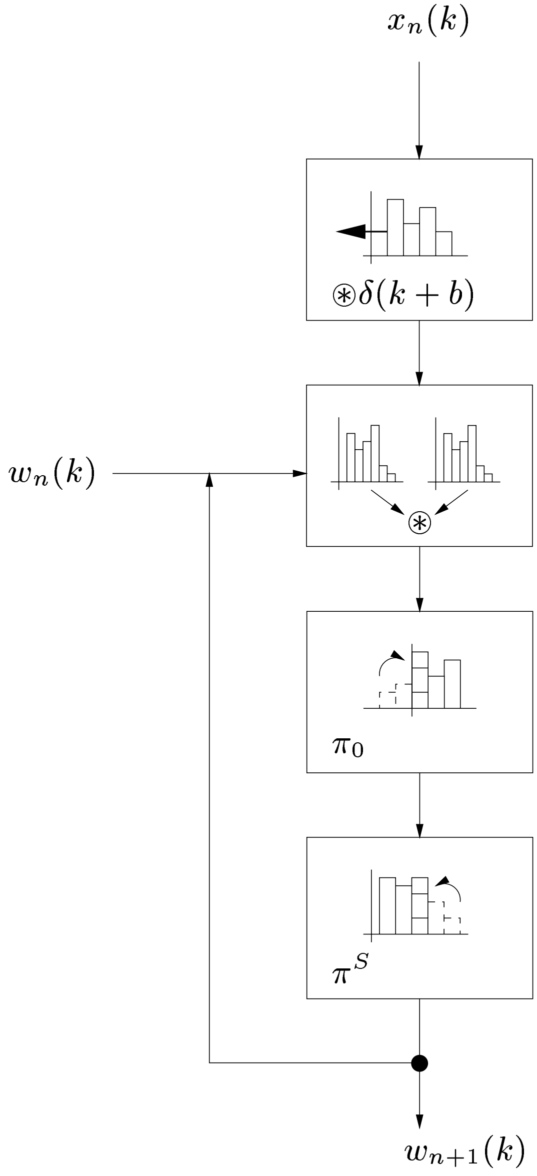


Figure 5.3: *Block scheme of the basic algorithm*

In the following, we extend this basic algorithm to cope with arrivals which do not follow a single distribution but a cyclic sequence of distributions which reflects the GOP pattern of length N of MPEG video traffic streams. A single step of the iterative system description with the general

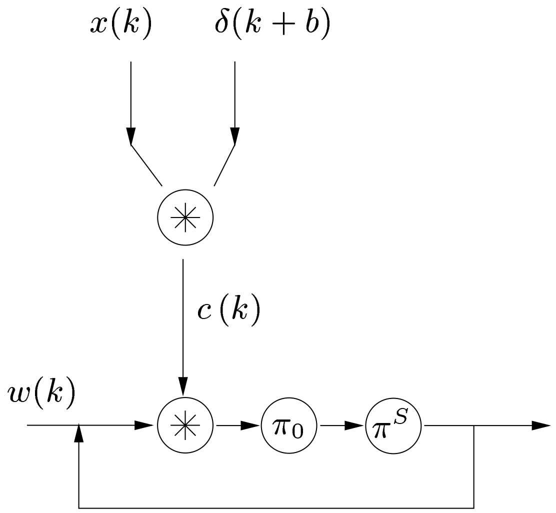


Figure 5.4: *Computation of the buffer content distribution (basic algorithm)*

distribution $x(k)$ is now replaced by a sequence of N sub-steps with the discrete distributions of the I-, P-, and B-frame sizes $x_j(k)$, $j = 1, \dots , N$, where the order is determined by the GOP pattern of the empirical data set.

In case of stationarity, we obtain the following system of equations.

$$\begin{aligned} w_{j+1}(k) &= \pi^S \circ \pi_0 \left[w_j(k) \circledast c_j(k) \right] \quad j = 1, \dots , N-1 \\ w_1(k) &= \pi^S \circ \pi_0 \left[w_{N+1}(k) \circledast c_N(k) \right], \end{aligned} \tag{5.9}$$

where $c_j(k) = x_j(k) \circledast \delta(k + b)$.

We determine the solution of this fix-point problem iteratively by applying the scheme shown in Figure 5.5. The following steps constitute the algorithm:

(1) Initialize $w_1(k)$ with the distribution of an empty system,
i.e., $w_1(0) = \delta(0)$.

(2) Compute the system functions $c_j(k)$.

(3) Apply convolution and sweep operators π_0 and π^S.

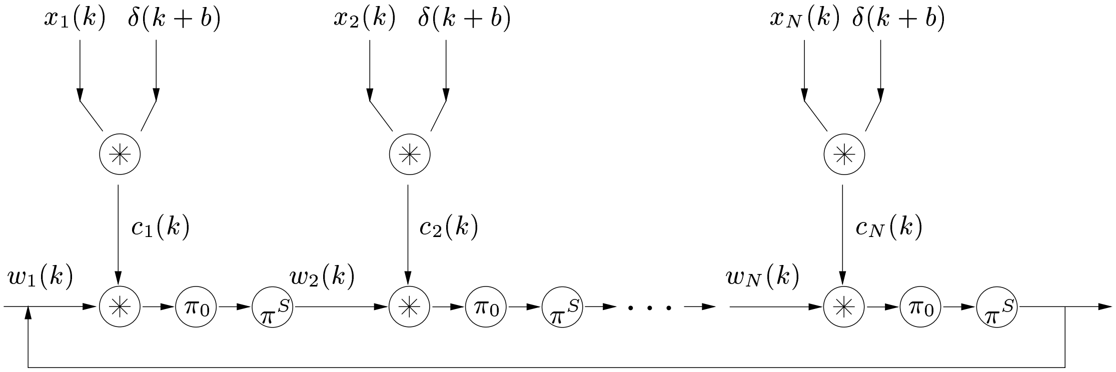


Figure 5.5: *Computation of the buffer content distribution (uncorrelated model)*

(4) Repeat this operation for each system function (in total N times).

(5) If the convergence criterion is not met, go to Step (3).

The buffer content distributions $w_j(k)$ are now used to compute the cell loss probability P_{loss}. Since we have to consider the whole GOP the loss probability is the average loss during a GOP duration divided by the average GOP size

$$P_{\mathrm{loss}} = \frac{\mathrm{E}[L]}{\mathrm{E}[Y]} = \frac{\sum_{j=1}^{N} \mathrm{E}[L_j]}{\sum_{j=1}^{N} \mathrm{E}[X_j]}, \tag{5.10}$$

where L (L_j) denotes the amount of cells lost during one GOP (frame) duration, and Y (X_j) is the GOP (frame) size.

To determine the average amount $\mathrm{E}[L_j]$ of losses during a frame duration, we have to consider all cases where

$$k > (S-i)+b \qquad i = 0, \ldots, S, \tag{5.11}$$

with k cells arriving at a buffer that contains i cells. In other words, cells are lost if

$$k = (S-i)+b+1, \ \ldots, x_{\max}, \tag{5.12}$$

where $x_{\max}$ denotes the maximum frame size of all frames.

The average amount of losses is then computed by

$$\mathrm{E}[L_j] \quad = \quad \sum_{i=0}^{S} w_j(i) \cdot \left[\sum_{l=1}^{x_{\max}-(S-i)-b} x_j\left((S-i)+b+l\right) \cdot l \right]. \qquad (5.13)$$

5.1.2 Analysis with GOP-correlations

From the results of Chapter 4, we concluded that the histogram model leads only to good cell loss predictions if the multiplexer buffers are small. For larger buffers, models including GOP-by-GOP correlations, such as the simple Markov chain model or the autoregressive model, clearly outperform models without GOP correlations. Therefore we extend the model of the previous section to include GOP-by-GOP correlations for a small number of lags. We will use the term *correlated model* to refer to the model.

In addition to the notations introduced for the uncorrelated model we use the following terms:

X_{n_j} : random variable for the number of cells in the jth frame of the nth GOP following the distribution $x_j(k)$ with $j = 1, \ldots, N$.

Y_{n-h} : random variable for the class of GOP $n-h$, where GOP n is currently being processed by the algorithm.

The notion of GOP classes is identical to that of the scene-oriented model, i.e., the GOP class of a given GOP is the sequence number of the GOP size histogram interval in which this particular GOP falls. To obtain M GOP classes a histogram with M intervals is used. GOP classes have to be introduced instead of actual GOP sizes to keep the model numerically tractable.

For the correlated model we replace the frame sizes X_n by $(h+1)$-dimensional vectors $(X_n, Y_{n-1}, \ldots, Y_{n-h})$, where

$$x(k, t_1, \ldots, t_h) \quad = \quad \Pr\{X = k, Y_{n-1} = t_1, \ldots, Y_{n-h} = t_h\}. \qquad (5.14)$$

This means that the current frame size depends on the GOP sizes of one or more previous GOPs. For illustration see Figure 5.6. Note, that for simplification the conditional probabilities are not computed for each frame of the GOP but for each frame type.

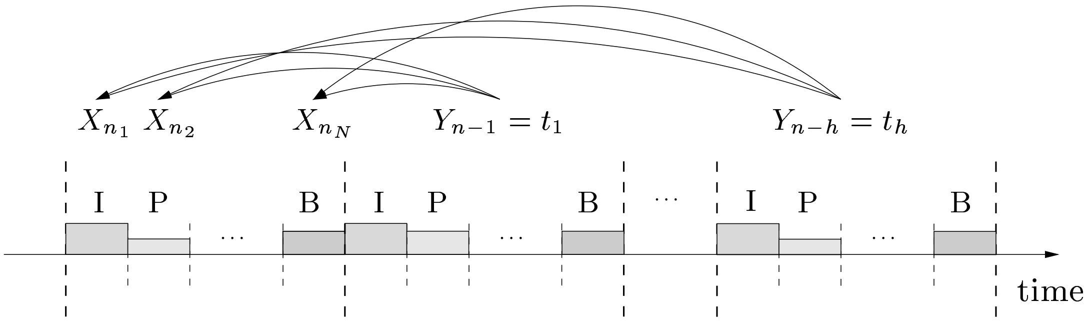


Figure 5.6: *Conditional frame size probabilities*

To simplify the notation we define $\mathbf{t} = (t_1, \ldots, t_h)$ to represent the currently considered conditions, for instance, $x(k, t_1, \ldots, t_h) = x(k, \mathbf{t})$.

The conditional frame size distributions $x_j^{(\mathbf{t})}(k)$ are given by

$$x_j^{(\mathbf{t})}(k) \quad = \quad \frac{x_j(k, \mathbf{t})}{\sum_k x_j(k, \mathbf{t})}. \tag{5.15}$$

Hence, we obtain a new set of equations which determine the evolution of the buffer content.

$$\begin{aligned} w_{j+1}^{(\mathbf{t})}(k) &= \pi^S \circ \pi_0 \left[w_j^{(\mathbf{t})}(k) \circledast x_j^{(\mathbf{t})}(k) \circledast \delta(k+b) \right], \\ w_1^{(\mathbf{t})}(k) &= \sum_t q \cdot w_{N+1}^{(\mathbf{t})}(k) \end{aligned} \tag{5.16}$$

for $j = 1, \ldots, N$ and $\mathbf{t} \in \{1, \ldots, M\}^h$. Let N denote the GOP length, M the number of GOP classes, and q the probability of the vector of conditions $\mathbf{t}$.

The system of equations can be solved by an iteration as depicted in Figure 5.7. The operator in the hexagon is introduced to improve the readability.

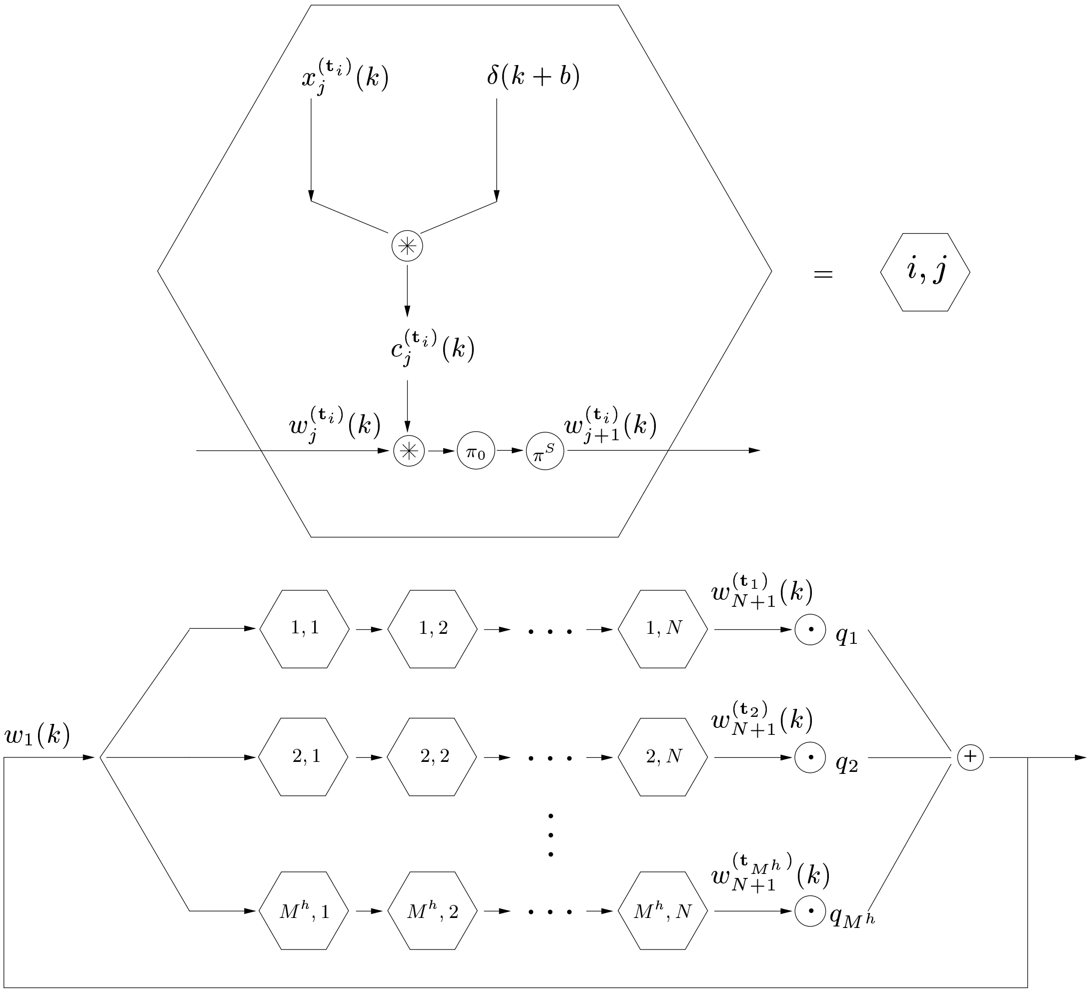

$x_j^{(\mathbf{t}_i)}(k)$
$\delta(k+b)$
$c_j^{(\mathbf{t}_i)}(k)$
$w_j^{(\mathbf{t}_i)}(k)$
π_0
π^S
$w_{j+1}^{(\mathbf{t}_i)}(k)$
=
i, j
$w_1(k)$
$1,1$
$1,2$
$1,N$
$2,1$
$2,2$
$2,N$
$M^h,1$
$M^h,2$
M^h,N
$w_{N+1}^{(\mathbf{t}_1)}(k)$
$w_{N+1}^{(\mathbf{t}_2)}(k)$
$w_{N+1}^{(\mathbf{t}_{M^h})}(k)$
q_1
q_2
q_{M^h}

Figure 5.7: *Computation of the buffer content distribution (correlated model)*

Starting with a common buffer content distribution for all cases at the beginning of a GOP, we have to compute the iterations for each set of conditions $\mathbf{t}_i$ separately. Each of these sequences of iterations forms a row of the computing scheme. At the end of the GOP, all these distributions have to be aggregated to a single distribution by means of the weights q_i. For $h = 0$, Figure 5.7 reduces to Figure 5.5.

The cell loss probability has to be computed as a weighted sum of the loss probabilities on each path of Figure 5.7:

$$P_{\text{loss}} = \sum_{i=1}^{M^h} q_i \cdot p_{\text{loss}}^{(\mathbf{t}_i)} \tag{5.17}$$

with (see Eqn. (5.10))

$$\begin{aligned} p_{\text{loss}}^{(\mathbf{t}_i)} &= \\ &= \frac{\sum_{j=1}^{N} \left\{ \sum_{k=0}^{S} w_j^{(\mathbf{t}_i)}(k) \cdot \left[\sum_{l=1}^{x_{\max}-(S-k)-b} x_j\left((S-k)+b+l\right) \cdot l \right] \right\}}{\sum_{j=1}^{N} \mathrm{E}[X_j]} . \end{aligned} \tag{5.18}$$

5.1.3 Numerical results

In this section, we present cell loss results for both the uncorrelated model and the correlated model with $h = 1, 2, 3$. The frame size distributions were derived from the Bellcore *starwars* data set, i.e. $N = 12$. The number of GOP classes M is 5. Table 5.1 shows some statistical data of the video stream.

Table 5.1: *Statistical data of the starwars sequence.*

Frame type	Number	Average [*cells*]	Min [*cells*]	Max [*cells*]	CoV
all	174126	41.12	2	483	1.15
I	14511	157.74	31	483	0.33
P	43531	60.58	6	454	0.63
B	116084	19.25	2	169	0.65

Figure 5.8 shows the cell loss results for a buffer size of 100 cells. The offered load ρ is given by $\rho = \mathrm{E}[X]/b$, where $\mathrm{E}[X]$ denotes the average frame size of the whole sequence. For buffers of this size and smaller, all model variants lead to approximately the same good prediction of the cell losses of the trace-driven simulation. Therefore, as already concluded from the results of Chapter 4, it is sufficient to use simple histogram-based models in the presence of buffers with a size in the order of the average frame size.

The situation is different for larger buffers. Here, as depicted in Figure 5.9 for a buffer size of 1000 cells, the modeling of the GOP-by-GOP correlation becomes important. This is reflected by the ranking of the model approximation quality, where the correlated model with $h = 3$ is several orders of magnitude better than the uncorrelated model.

In Figure 5.10, we fix the offered load at $\rho = 0.9$ and compute the cell loss probabilities for a wide range of buffer sizes. Only the model with $h = 3$ leads to acceptable estimates of the trace-based cell losses up to a buffer size of about 400 cells. The predicted losses of the other models are accurate up to a buffer size of 100 cells and underestimate the losses for larger buffer sizes.

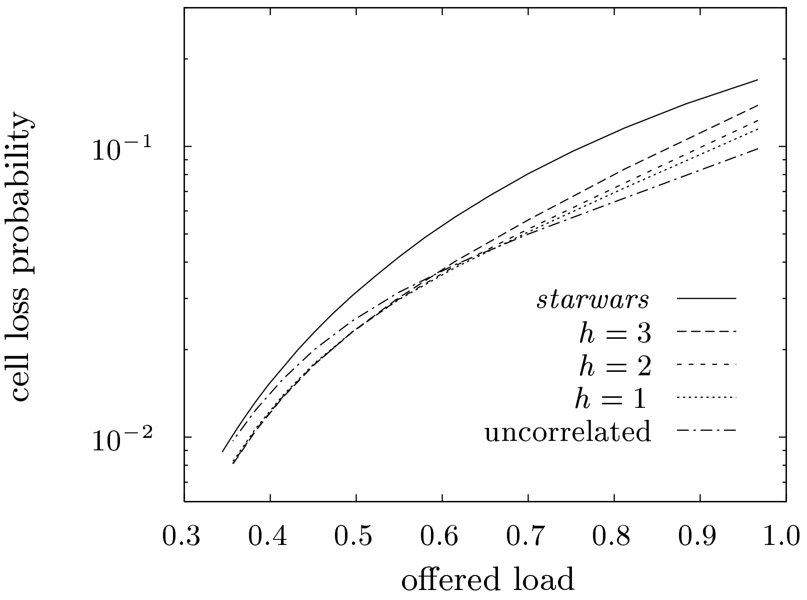


Figure 5.8: *Cell loss curves for a buffer size of 100 cells*

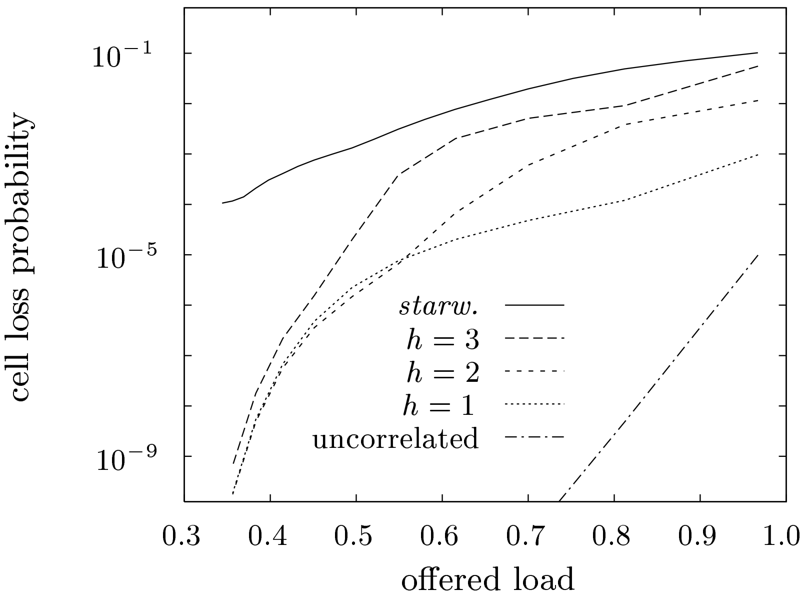


Figure 5.9: *Cell loss curves for a buffer size of 1000 cells*

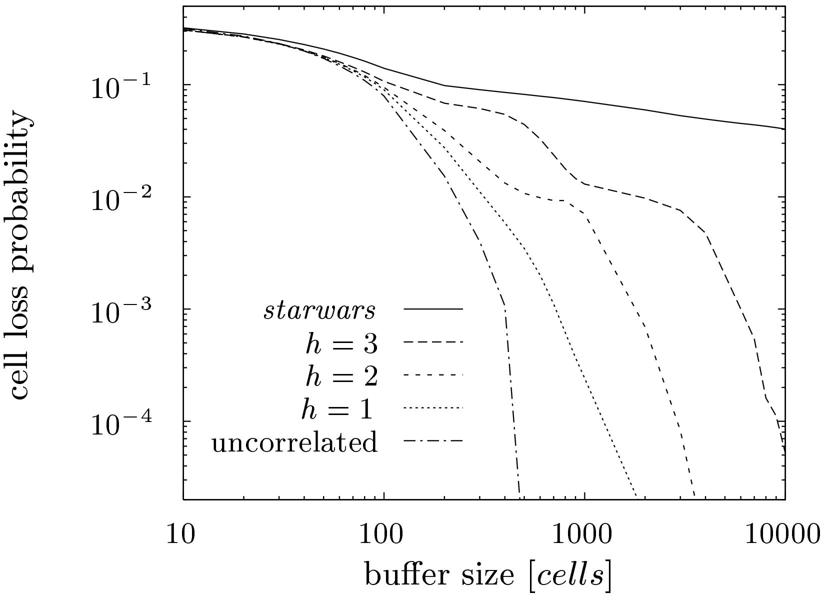


Figure 5.10: *Cell loss curves for $\rho = 0.9$*

5.1.4 Summary

In the above section, we presented an analysis technique which is capable to make use of some models developed in Chapter 3. It is possible to consider the different frame size distributions, the cyclic GOP pattern, and the GOP-by-GOP correlation for a small number of lags. Taking into account the results of Chapter 4, the cell loss prediction quality is as expected.

The implementation of the algorithm is simple but there are a number of problems with respect to memory consumption and running time. Generally, the iteration takes longer to converge if the buffer size is increased or the offered load decreased. In addition, for the correlated model the time to converge is considerably longer than for the uncorrelated model. Another difficulty that arises for the correlated model is the memory requirements to store the conditional frame sizes. Even if we apply sophisticated storage techniques the correlation order which can be used for experiments is clearly limited since the number of probabilities grows exponentially with the value of h. Another subject which should be considered in greater detail in another study is the convergence behavior of the algorithms. The problems listed above are not specific to our model and to our traffic but can generally be found while applying this discrete-time analysis technique.

5.2 Cell-based analysis

In this section, we present an analytical approach for the dimensioning of the GCRA for VBR MPEG video cell stream monitoring in ATM networks (cf. Section 1.2.5). Again, the analysis technique is based on the discrete-time analysis approach of Tran-Gia and Ahmadi (1988). However, in contrast to the previous section, we do not work on frame scale but on cell scale with respect to the time discretization. This section generalizes the results of Rose and Ritter (1995). With the analysis provided, it is also possible to consider a spaced video cell stream and not only a video stream where the frames are transmitted as a back-to-back burst of cells.

5.2.1 UPC of video traffic

As with other services, a video connection that has been accepted by the CAC mechanism has to be monitored by the UPC to check whether it fulfills the traffic contract negotiated with the network.

Unlike with other services, for video traffic it is hard to determine the key traffic parameters. The only parameter which is available without difficulties is the PCR of the coder adaptor to the ATM network, i.e. the transmission capacity of the ATM adaptor. If a more detailed description of the cell stream of the video connection is needed one has to know the video sequence in advance, but this will only be the case for movie broadcast services or video data base retrievals. Video connections which consist of live video transmission, like broadcasts of sports events, will suffer from a lack of information about the cell stream.

One possibility to overcome this problem is the definition of video categories with different safety bandwidth requirements, for example, categories with respect to the frequency of scene changes or the set of tolerated camera actions. But even if we would be able to compute a variety of parameters of the video cell stream in advance, we have to decide which of these parameters will be used for CAC and UPC.

The PCR is already standardized as source traffic descriptor by the ITU-T (1994). To improve the estimates of CAC functions, it would be desirable to know the Mean Cell Rate (MCR) of the traffic stream. This traffic descriptor is useless in the case of video, however, since it cannot be policed efficiently by any UPC function due to the burstiness of this kind of traffic. Studies concerning MCR policing can be found in e.g. Ritter and Tran-Gia (1995) or Roberts (1992). Therefore, the introduction of the Sustainable Cell Rate (SCR) as source traffic descriptor was discussed in the standardization bodies. We investigate whether it is advantageous to use the SCR as control parameter of video cell streams and how to dimension the required parameters.

Several simulation studies were carried out considering the dimensioning of UPC parameters for VBR video traffic. Rathgeb (1993) examines the behavior of leaky buckets and other policers for several video traces. He concludes that leak rates which are close to the mean rate of the stream are not practical since they require large buffers in the network and consequently lead to large transmission delays. Smoothing at the source would only shift the source of delay from inside the network to its edges. Reibman and Berger (1995) come to similar conclusions in their study. In addition, they note that due to the large SCR which has to be chosen the multiplexing gain for video sources will be much lower than expected. To overcome the UPC dimensioning problems, Reininger et al. (1995) suggest a scheme for the renegotiation of the traffic parameters during connection time.

5.2.2 Traffic model

The basic idea of our model is to describe the coder output process by an array of frame size distributions of the specific GOP pattern of a video sequence as in Section 5.1. From the *starwars* sequence we will therefore obtain a sequence of 12 different distributions. The only frame-by-frame correlation information which is used in our model is the order of the frame size distributions given by the GOP pattern. The long-term dependences among frames of consecutive GOPs, e.g. the correlations introduced by similar pictures of one movie scene, seem to be less important in our case.

To sum up, the GOP pattern is the only correlation information which is used for our model. To describe the cell stream produced by the ATM adaptor of the MPEG coder, the following is assumed:

- a single-layer coder is used,
- the ATM adaptor and the transmission link have the same capacity,
- the ATM adaptor transmits the cells with a given intercell distance,
- the first cell of a frame is transmitted at the beginning of the frame.

This means that one frame at a time will arrive at the ATM layer, the packetization takes place, and the ATM cells are transmitted with the maximum rate of the adaptor taking into account the given spacing distance. In our opinion, this way of transmission might not be optimal with respect to cell loss due to cell discarding at the UPC, but it seems to be a realistic assumption.

The modeled cell stream can be described by the following parameters:

- *frame duration* D, which is measured in cells and can be calculated by $D = B/r$, where B denotes the maximum output rate of the ATM adaptor in $cells/sec$ and r denotes the frame rate of the video sequence in $frames/sec$. Of course, the maximum frame size of the encoded video sequence always has to be smaller than D cells.

- ◇ *intercell distance d_{cell}*, i.e., each used slot is followed by $d_{\text{cell}} - 1$ empty slots. If $d_{\text{cell}} = 1$, the cells are transmitted back-to-back. The maximum value of d_{cell} is $\lfloor D/x_{\max} \rfloor$, where $x_{\max}$ is the number of cells of the largest frame.

- ◇ *frame size distributions* $x_1(i), \dots, x_N(i)$ for a sequence with a GOP of N frames, which are sampled from real MPEG-coded video data. Here, $x_j(i)$ denotes the probability that frame number j of a GOP has a length of i cells.

In Figure 5.11, the cell streams of a simple "IP"-GOP sequence are shown as an example, i.e. $N = 2$, with frame size distributions $x_1(i)$ and $x_2(i)$. Every $D = 13$ cells a new frame is starting, where its size in cells is computed from its distribution. Figure 5.11 (a) shows a cell stream with $d_{\text{cell}} = 1$, and (b) a cell stream with $d_{\text{cell}} = 2$.

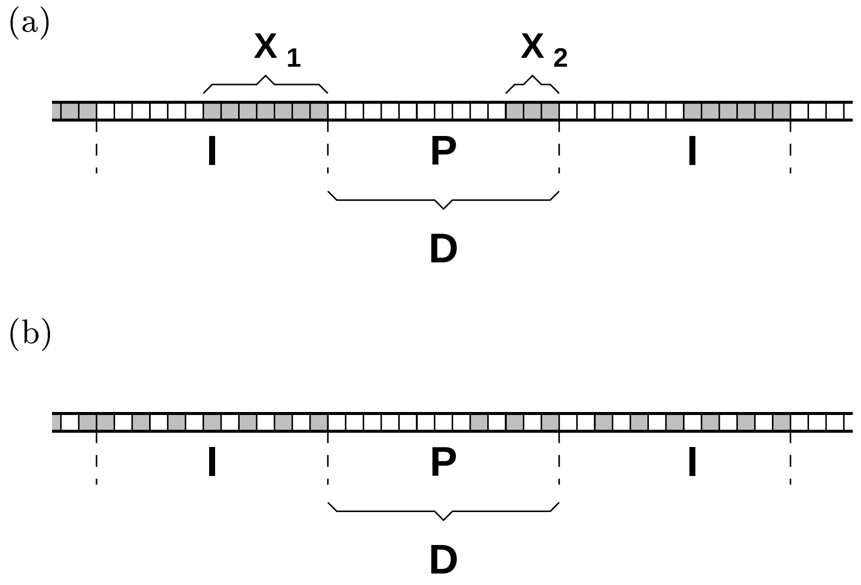


Figure 5.11: *Examples of the model cell stream.*

5.2.3 Cell loss analysis

As in Section 5.1, the cell loss estimation algorithm based on the discrete-time analysis of the $GI^{[X]}/D/1-S$ queuing model presented in Tran-Gia and Ahmadi (1988). This analysis technique was applied in Hübner (1994) to analyze the GCRA if the input traffic is assumed to follow a renewal process. An extension which deals with ON/OFF sources was presented in Ritter and Tran-Gia (1995). Based on this approach, we develop an algorithm to cope with the cyclic occurrence of frames of different types in MPEG coded video sequences.

The current state of the GCRA(T_s, τ_s) is described by a discrete-time random variable $Z(t)$, which represents the remaining time until the next cell is expected to arrive (see Hübner (1994)). A cell arriving at time t_0 seeing the GCRA in state $Z(t_0) = i$ is considered to be conforming for $i \leq \tau_s$, otherwise non-conforming.

For the presentation of the algorithm, the following notation is used:

X_j : discrete random variable for the size of frame number j in the GOP measured in cells,

$Z^-_{j,k}$: $Z(t)$ just *before* the beginning of the k-th slot in frame number j in the GOP,

$Z^+_{j,k}$: $Z(t)$ just *after* the beginning of the k-th slot in frame number j in the GOP.

The distributions of $Z^-_{j,k}$ and $Z^+_{j,k}$ are $z^-_{j,k}(i)$ and $z^+_{j,k}(i)$, respectively. The frame sizes X_j are assumed to follow renewal processes with distributions $x_j(i)$.

Figure 5.12 shows a snapshot of the system evolution during the arrival of frame j. Each arriving cell increments the remaining time $Z(t)$ by T_s. If $Z(t)$ is less than τ_s arriving cells are conforming. If it is larger arriving cells are rejected or tagged. In our example the fifth cell is rejected due to violation of the limit τ_s.

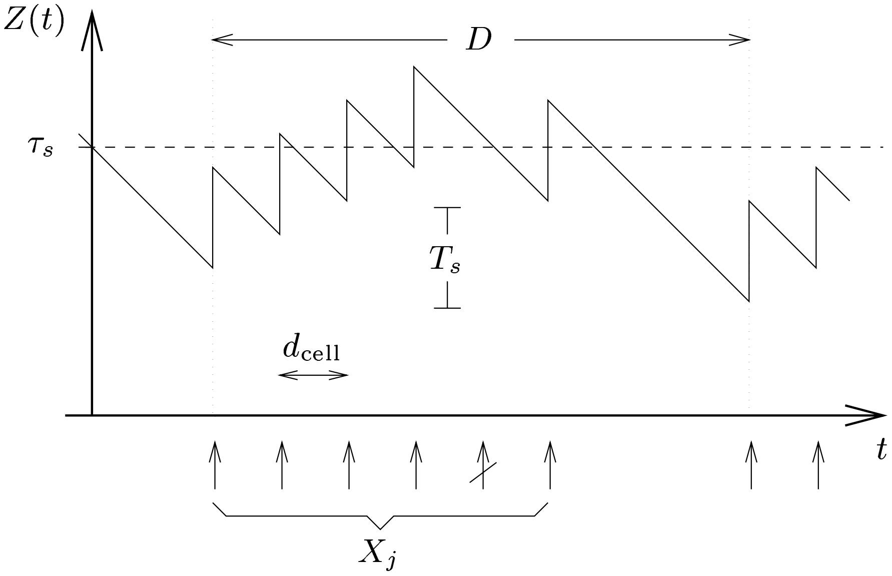


Figure 5.12: *Snapshot of the system evolution*

Let us consider a particular frame in the GOP, say frame number j and assume a frame size of $\lceil D/d_{\text{cell}} \rceil$ cells. This corresponds to a frame which has the maximum possible size to be transmitted with an intercell spacing of d_{cell} cells.

If k modulo d_{cell} equals 0, i.e., a cell is arriving at slot k then $Z^+_{j,k}$ is determined from $Z^-_{j,k}$ by

$$Z^+_{j,k} = \begin{cases} Z^-_{j,k} & : \quad Z^-_{j,k} > \tau_s, \\ Z^-_{j,k} + T_s & : \quad Z^-_{j,k} \leq \tau_s. \end{cases} \tag{5.19}$$

Otherwise, $Z^+_{j,k}$ is computed by

$$Z^+_{j,k} = Z^-_{j,k}. \tag{5.20}$$

According to Eqn. (5.19), the corresponding distributions can be obtained by

$$z^+_{j,k}(i) = \begin{cases} 0 & : \quad 0 \le i \le \tau_s, \\ z^-_{j,k}(i) & : \quad \tau_s < i < T_s, \\ z^-_{j,k}(i) + z^-_{j,k}(i - T_s) & : \quad T_s \le i \le T_s + \tau_s \end{cases} \tag{5.21}$$

for $\tau_s < T_s$. For $\tau_s \ge T_s$ the following holds:

$$z^+_{j,k}(i) = \begin{cases} 0 & : \quad 0 \le i < T_s, \\ z^-_{j,k}(i - T_s) & : \quad T_s \le i \le \tau_s, \\ z^-_{j,k}(i) + z^-_{j,k}(i - T_s) & : \quad \tau_s < i \le T_s + \tau_s. \end{cases} \tag{5.22}$$

In case of Eqn. (5.20), i.e. no cell arrival in slot k (k modulo $d_{\text{cell}} \ne 0$), we obtain the distributions $z^+_{j,k}(i)$ by

$$z^+_{j,k}(i) = z^-_{j,k}(i). \tag{5.23}$$

The computation of $Z^-_{j,k+1}$ is driven by the decrease of $Z(t)$ until it reaches zero, i.e.,

$$Z^-_{j,k+1} = \max\{0, Z^+_{j,k} - 1\}. \tag{5.24}$$

Therefore, the distributions are determined by

$$z^-_{j,k+1}(i) = \begin{cases} z^+_{j,k}(0) + z^+_{j,k}(1) & : \quad i = 0, \\ z^+_{j,k}(i + 1) & : \quad 0 < i < T_s + \tau_s, \\ 0 & : \quad i = T_s + \tau_s. \end{cases} \tag{5.25}$$

The next step is the computation of the state of the GCRA(T_s, τ_s) at the beginning of the next frame boundary. Since we computed $Z^-_{j,k}$ assuming a frame size equal to $\lceil D/d_{\text{cell}} \rceil$ cells, we now have to take into account the different possible frame sizes. The state of the GCRA(T_s, τ_s) just before the frame boundary in dependence of the size m of the current frame is given by

$$Z^*_{j,D,j} = \max\{0, Z^-_{j,m^*} - (D - m^*)\}, \tag{5.26}$$

if we define $m^* = (m-1) \cdot d_{\text{cell}} + 1$. The reason for this is that there are no cell arrivals in the last $(D - m^*)$ slots of the frame period. We obtain the corresponding distributions by

$$z^*_{j,D,m}(i) \;=\; \begin{cases} \sum_{l=0}^{D-m^*} z^-_{j,m^*}(l) & : \; i = 0 \\ z^-_{j,m^*}(i + (D - m^*)) & : \; 0 < i \leq T_s + \tau_s - (D - m^*) \\ 0 & : \; T_s + \tau_s - (D - m^*) \\ & \quad < i \leq T_s + \tau_s \end{cases} \qquad (5.27)$$

Using $Z^*_{j,D,m}$, the system state just before the next frame boundary is given by

$$Z^-_{j+1,0} \;=\; Z^*_{j,D,X_j}, \qquad (5.28)$$

where $(j+1)$ is computed modulo N. To obtain $z^-_{j+1,0}(i)$, we have to multiply the system state distributions just before the frame boundary $z^*_{j,D,m}(i)$ by the probabilities of observing a frame of size m for the frame type j. This leads to the following equation:

$$z^-_{j+1,0}(i) \;=\; \sum_{m=1}^{\lceil D/d_{\text{cell}} \rceil} x_j(m) \cdot z^*_{j,D,m}(i). \qquad (5.29)$$

Now, the distributions in equilibrium can be derived by applying iteratively the equations presented above with respect to the GOP used and the current slot in each frame.

Given the equilibrium system state distributions just before the cell arrivals, the probability $p_j(k)$ to observe a non-conforming cell at slot k $(k = 0, \ldots, D-1)$ in a frame of type j is

$$p_j(k) \;=\; \sum_{i=\tau_s+1}^{T_s+\tau_s} z^-_{j,k}(i) \qquad \text{for} \qquad j = 1, \ldots, N. \qquad (5.30)$$

To derive the probability P_j to observe a non-conforming cell in a given frame in the GOP, the $p_j(k)$ values have to be multiplied with the complementary cumulative probability distribution $x_j^c(i)$ of the frame size distribution $x_j(i)$. This has only to be done for values of k, where cell arrivals are possible, i.e. k modulo $d_{\text{cell}} = 0$. Furthermore, a normalization is required:

$$P_j \quad = \quad \frac{\sum\limits_{k=0}^{\lceil D-1/d_{\text{cell}} \rceil} p_j(k \cdot d_{\text{cell}}) \cdot x_j^c(k)}{\sum\limits_{k=0}^{\lceil D-1/d_{\text{cell}} \rceil} x_j^c(k)} \qquad \text{for} \qquad j = 1, \ldots, N. \tag{5.31}$$

The overall cell loss probability can now be obtained by

$$P_{\text{loss}} \quad = \quad \frac{\sum\limits_{j=1}^{N} P_j \cdot \mathrm{E}[X_j]}{\sum\limits_{j=1}^{N} \mathrm{E}[X_j]}. \tag{5.32}$$

5.2.4 Numerical results

Parameters and configuration

In this section we present numerical results based on simulation and analysis to show the effectiveness of SCR monitoring of video cell streams and point out some interesting properties for dimensioning the UPC function. We focus on four ATM adaptor capacities: 150 *Mbps*, 75 *Mbps*, 37.5 *Mbps*, and 34 *Mbps*. For the considered *starwars* sequence, this leads to the frame durations D and minimum (T_{peak}) and maximum values (T_{mean}) for the SCR parameter T_s shown in Table 5.2.

It is important to note that all simulation results were produced using directly the sequence of the frame sizes of the *starwars* movie and not by means of the traffic model.

Table 5.2: *ATM adaptor and SCR parameters*

Capacity	D	T_{peak}	T_{mean}
150 *Mbps*	14740	30	360
75 *Mbps*	7370	15	180
37.5 *Mbps*	3685	7	90
34 *Mbps*	3340	6	81

Simulation study

We first give simulation results for a capacity of $B = 150$ *Mbps* to show how the parameters T_s and τ_s affect the cell loss probability. Figure 5.13 depicts the loss curves for T_s ranging from 30 to 110, where τ_s is measured in multiples of the frame duration D.

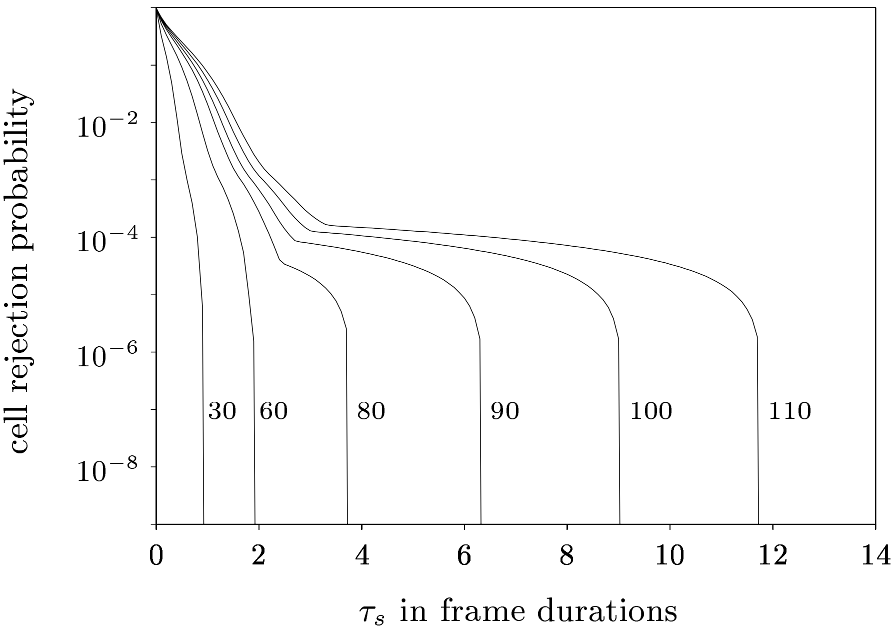


Figure 5.13: *Dependence of cell rejection probability on T_s ($B = 150$ Mbps)*

For small values of T_s, the losses decrease very fast, whereas for values of T_s larger than 60 a knee in the curve can be observed. The knee is always located at a value τ_s for which the UPC function tolerates bursts which have the maximum frame length of the video sequence. If a certain value of τ_s is reached, e.g. $\tau_s \approx 3.8D$ for $T_s = 80$, the loss probability drops to zero, i.e., there are no losses as soon as the parameter τ_s is large enough to force the UPC function to accept consecutive bursts of several frames. We prove this assumption by using only the frame data of GOPs with a high mean frame length, i.e. worst case GOPs, for the simulation and obtain the same drop-down locations of the curves. For all values of T_s in Figure 5.13, small loss probabilities can be achieved. It should be noted, however, that already for $T_s = 90$ a buffer capacity in the network elements of about 1000 cells is required to store the burst of this single connection that is tolerated for a cell loss probability of less than 10^{-6}. Generally, buffer sizes in ATM networks are in the order of 10^3 cells. Therefore, the value T_s should be chosen to be close to T_{peak} to obtain realistic values for the required buffer size, e.g. for $T_s = 60$ a buffer size of about 500 cells is needed.

Figure 5.14 shows that the SCR owns a certain scalability property. The cell loss curves remain almost identical if the parameters T_s and τ_s are scaled by the same factor as the adaptor capacity. In Figure 5.14, two groups of curves for $T_s = 60$ and $T_s = 100$ are depicted. We start with a capacity of 150 *Mbps* and use the scaling factors 1.0, 0.5, and 0.25, i.e., $T_s = 60$ for 150 *Mbps*, $T_s = 30$ for 75 *Mbps*, and $T_s = 15$ for 37.5 *Mbps*. To allow for a comparison of these curves, the horizontal axis has to be scaled accordingly. Note that the video sequence which was used to create the input for the UPC function was not scaled.

Figure 5.14 shows that the curves of the two groups are matching well. In general, the curves are matching better for large values of T_s.

By means of this scalability property, the SCR parameters for a large variety of adaptor capacities can easily be calculated by multiplying a constant if the curve for a single capacity is known.

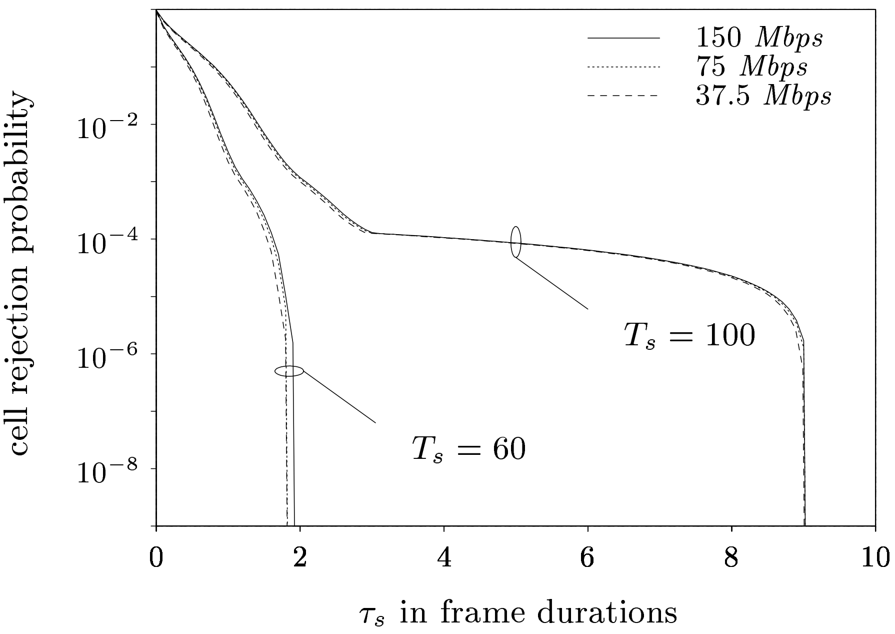


Figure 5.14: *Scalability properties of policing parameters*

Analytical results

Now, we investigate the accuracy of the analysis presented in Section 5.2.3. Generally, the results are of exact nature if the frame sizes in the GOP follow renewal processes. In reality, however, correlations can be observed.

In Figure 5.15, the overall cell loss probability for different choices of d_{cell} is plotted over τ_s to verify the accuracy of our analysis. We use an adaptor capacity of 34 *Mbps* and a SCR with $T_s = 15$. The relative difference between the analytical and simulation values is always smaller than 1%.

Since the accuracy does not depend on the choice of the intercell distance (cf. Figure 5.15) we use $d_{\text{cell}} = 1$ for the following numerical examples. Furthermore, loss curves for I-, P-, B-frames only, as well as for all frames are shown.

In Figure 5.16, results for the same adaptor capacity of 34 *Mbps* and a higher SCR with $T_s = 10$ are presented. A relative difference of 1% can also

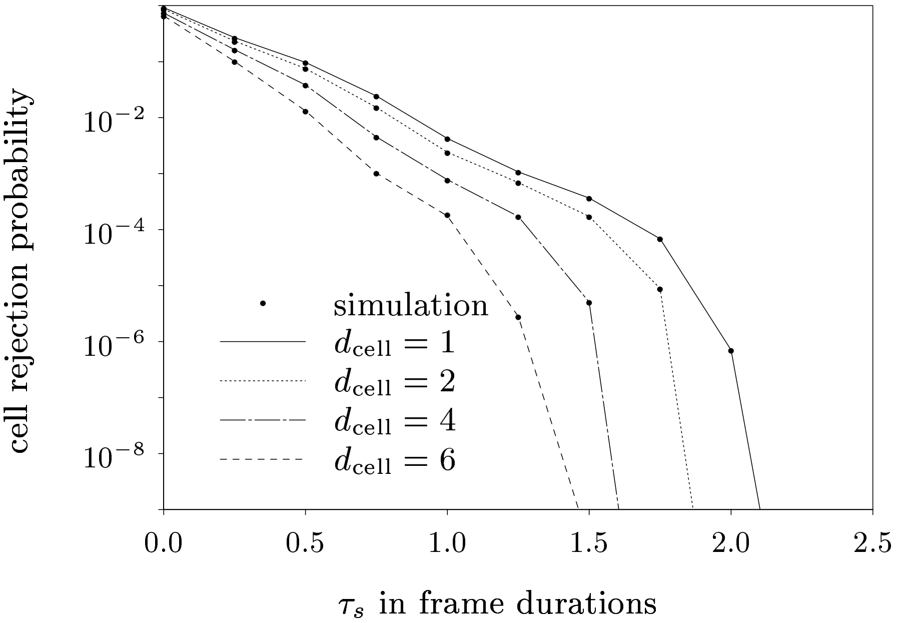


Figure 5.15: *Approximation accuracy ($T_s = 15$, $B = 34$ Mbps).*

be observed for the loss curves of the single frame types I, P, and B. This leads to the conclusion that our simple video coder output model is appropriate for the estimation of the cell losses for this type of UPC function.

For the parameter set of Figure 5.16, the B-frames always experience less losses than the P-frames, and the P-frames less losses than the I-frames. This seems to be obvious since the mean frame size of the B-frames is smaller than the one of the P-frames, and the mean frame size of the P-frames smaller than the one of the I-frames.

However, as presented in Figure 5.17, crossing of the loss curves of different frame types is possible. This behavior depends on the long-term correlations of the video sequence used. Moreover, there is no crossing of the curves, if the value of T_s is chosen to be close to T_{peak} and τ_s can be chosen small, too. If the value of τ_s is larger than about two frame durations the analytical results underestimate the cell losses, since the MPEG model does not take into account GOP-by-GOP correlations of the video sequence. This

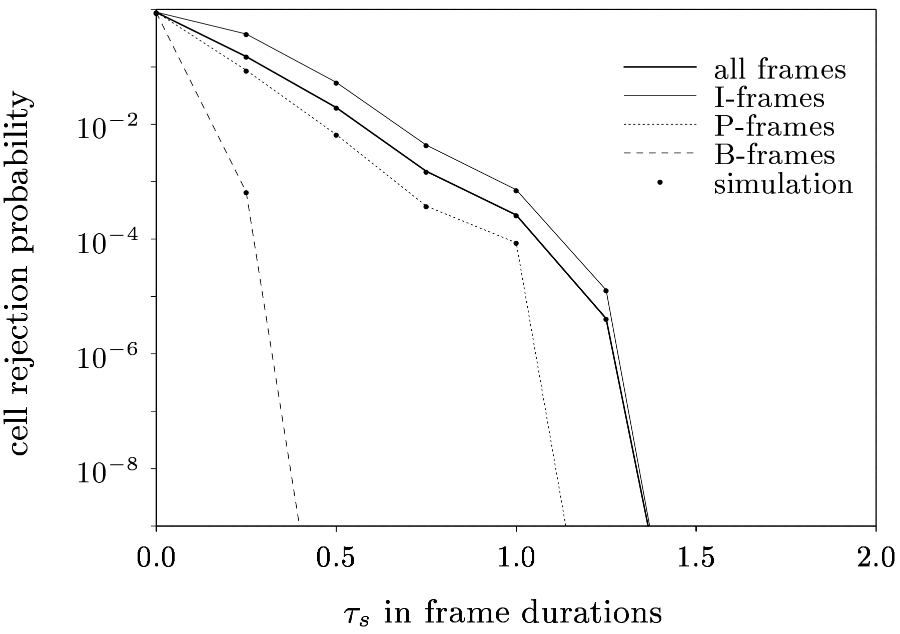


Figure 5.16: *Approximation accuracy ($T_s = 10$, $B = 34$ Mbps, $d_{cell} = 1$)*

effect is not problematic, however, since large values of τ_s would lead to large buffers within the network. For useful values of T_s and τ_s the analysis is very accurate (cf. Figures 5.16 and 5.18).

Figure 5.18 shows the curves for a 150 *Mbps* ATM adaptor and a value of $T_s = 30$. The behavior of the curves is similar to that of Figure 5.16.

5.2.5 Summary

The results show that the analysis using this simple model is very accurate compared to the simulation results based on real MPEG video data. A minor drawback of the analysis technique is that the computation time depends on the frame duration D, i.e. the ATM adaptor capacity. Large capacities lead to time consuming computations. For all parameter sets considered in this section, no numerical problems have occurred. In a number of cases, the analysis took considerably longer than the simulation. However, for large

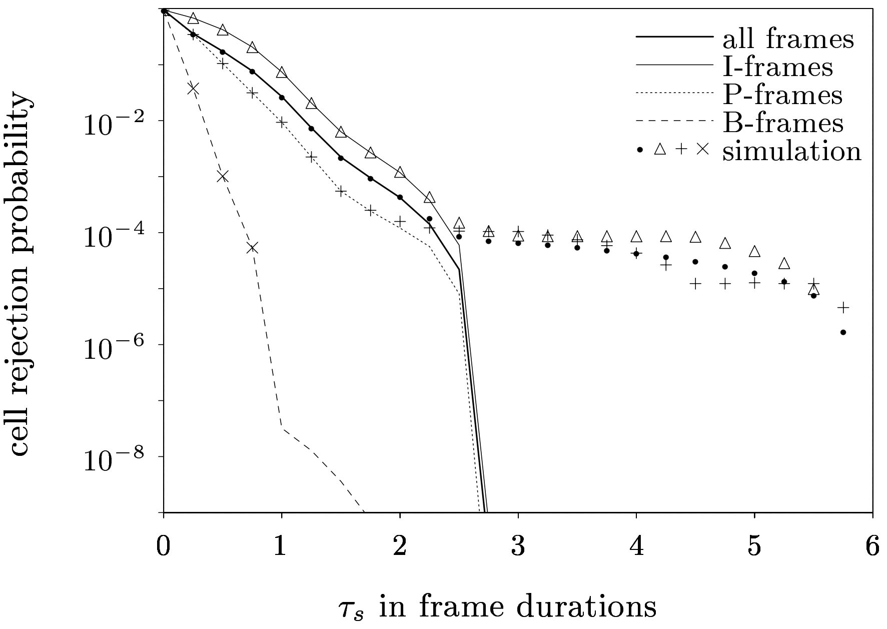


Figure 5.17: *Approx. accuracy divergence ($T_s = 20$, $B = 34$ Mbps, $d_{cell} = 1$)*

link capacities this can be avoided if we make use of the scalability property of the SCR parameters.

As far as the dimensioning of the GCRA parameters T_s and τ_s is concerned, the analytical and simulation results lead to several conclusions. To deal with reasonable buffer sizes of the network elements, it is necessary to keep the parameter T_s close to the PCR of the video sequence considered. In this case, both the required buffer can be kept small and small values of τ_s can be achieved. The parameter τ_s should always be chosen at least as large as the maximum frame size of the video sequence times T_s to obtain small loss probabilities.

Unfortunately, the loss curves show that the I-frames which contain the most important information of the MPEG frames experience higher losses than the other frame types. Discarding of cells on a frame-type basis could therefore lead to an improvement of the video quality (see Ramanathan et al. (1993)).

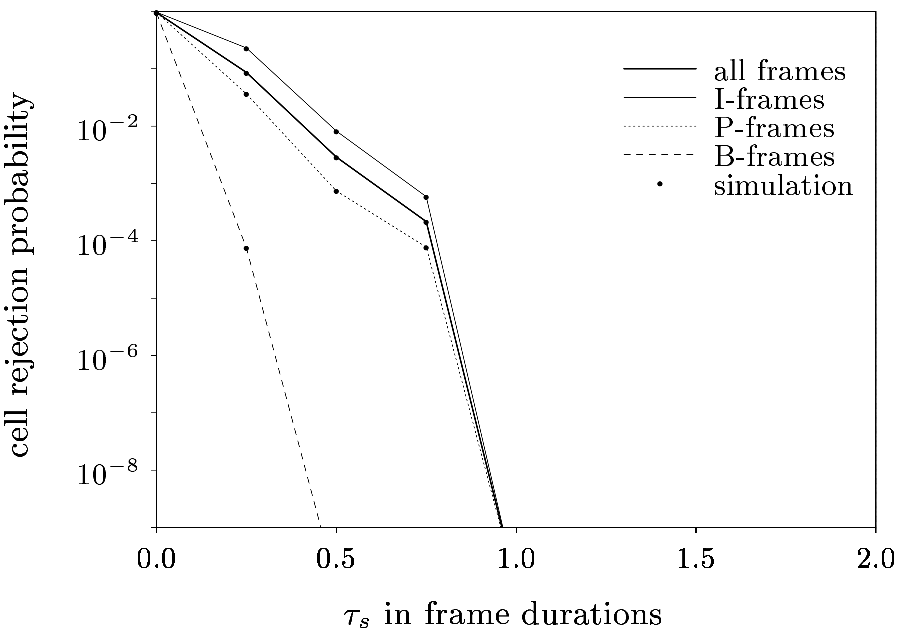


Figure 5.18: *Cell rejection analysis ($T_s = 30$, $B = 150$ Mbps, $d_{cell} = 1$)*

For video sequences with rapidly changing scene contents like action movies or sports events, the SCR generally will lie close to the PCR if τ_s is chosen reasonably. This implies a poor multiplexing gain. For sequences like video conferencing or video telephony, however, the SCR can be dimensioned remarkably lower than the PCR due to minor changes in the scene content.

The results presented in this section are based on the a priori knowledge of the traffic descriptor of the video source. In real systems, it will be difficult to determine the exact parameters for video traffic which have to be negotiated at connection set up. In most cases, it will even be impossible for the user to influence the coding parameters in order to obtain a certain descriptor.

6 Conclusion

This monograph was concerned with the modeling of VBR MPEG video and the impacts of this kind of traffic on ATM-based communication networks.

After a review of the MPEG coding standard and the basic principles of the ATM technology, we presented a thorough statistical inference of the frame and GOP (Group of Pictures) size traces of more than ten half-hour MPEG-1-encoded video sequences, where GOP size is defined as sum of sizes of a number of consecutive frames. For comparison, the Bellcore *Star Wars* MPEG data set was also analyzed since it is often used as a benchmark sequence in video modeling literature. A great deal of work was spent to examine the correlation structure of MPEG frame and GOP size sequences. The main results are as follows:

- ◇ The frame size traces inhere a periodic pattern due to the GOP-based encoding.
- ◇ The GOP size traces show considerable positive correlation over the first few hundred lags.
- ◇ There is a strong indication of long-range dependence for almost all sequences.

In other words, the correlation properties change fundamentally with the time scale. As a consequence, the memory capacity of the system to be examined, in our case the size of the ATM switch buffers, will be an important model selection criterion.

The presence of time-scale-dependent correlation characteristics induced the development of a layered modeling approach. Only then, the correlation effects can be decoupled in order to create simple and transparent but yet accurate models. We therefore modeled the GOP size process and derived the frame sizes from the GOP sizes based on the GOP pattern. As an alternative, the I-frame size process was modeled and the remaining frame sizes of a GOP were generated from the size of the leading I-frame. Since we intended to provide and compare MPEG video traffic models for a wide range of system parameters and analysis techniques, we discussed the properties and parameter estimation of the following models:

- histogram model,
- simple Markov chain model,
- scene-oriented model,
- autoregressive model,
- selfsimilar model.

These models represent a large part of the currently applied traffic modeling approaches. The models are of different correlation complexity and the marginal distributions of the generated samples are either discrete or continuous. All models were validated from the statistical view point and with respect to their accuracy in predicting performance measures, such as cell loss and cell delay at switch buffers. In addition, possible problems of the model parameter estimation were outlined.

In the course of the simulation study, we discovered that the system behavior is governed by two regimes which can be clearly separated. For small buffers, i.e., in the order of the average frame size of the trace, the system behavior is dominated by the periodic GOP pattern. In this case, it is sufficient to use the histogram model which contains no GOP-by-GOP correlations. If the buffer size is large enough to filter out the periodic pattern the GOP correlations play the most important role. Then, the simple Markov chain model or the autoregressive model should be applied. Only if the buffer sizes are huge, i.e., tens of thousands of cells, the models with the most complex GOP correlation structure, such as the scene-oriented model or the selfsimilar model, should to be used. In addition, we found that, under most load conditions, the GOP-based models outperform the I-frame-based models in terms of accuracy.

Another important aspect, which was considered by the simulation study, is the multiplexing of several MPEG video traffic streams. A number of problems were identified which occur due to the periodic GOP pattern and which might lead to performance degradation and unfairness.

To support the use of the presented models in performance analysis work, two detailed examples were provided. The first example consists of a discrete-time analysis on frame level. Algorithms were developed for the estimation of cell losses for modeling approaches with and without GOP level correlations. The second example shows that a discrete-time analysis on cell level can be used to determine the parameters for the usage parameter control of a VBR MPEG video cell stream.

In the introduction, we expressed our intention to provide traffic engineers with models for VBR MPEG video traffic in a scale that can be handled. In general, it is possible to find such models at the cost of a limited range of applications. In our opinion, however, a single video model for all purposes is neither necessary nor useful.

We close this monograph with a bon mot of George Box emphasizing that modeling should never be an end in itself but an aid to understand our complex world.

"All models are wrong but some of them are useful."

Appendix

The following chapters provide the definition of statistical terms and an outline of statistical methods used in the main part of this monograph. Some paragraphs repeat fundamental definitions to keep the monograph as selfcontained as possible. Most of the definitions are from Law and Kelton (1991). However, the subsections about selfsimilarity and parameter estimation of selfsimilar processes offer information which is not widespread at the moment. One of the few textbooks on selfsimilar processes and their parameter estimation is Beran (1994).

For the remainder of the appendix, we use the following notational conventions. Finite time series of length N, e.g. obtained from measurements, are denoted by $\{x_t : t = 1, \ldots, N\}$. Stochastic processes are denoted by $\{X_t : t \in \mathbb{Z}\}$. In both cases, we omit the index set if possible. A letter with a hat indicates an estimator. For instance, $\widehat{\mu}$ denotes the estimator of the mean μ.

A. Definitions and methods for statistical inference

A.1. Summary statistics

There is a large number of sample estimates which can be used to characterize the given data set $\{x_t\}$. In our work, we make use of the following:

- Sample mean (measure of central tendency):

$$\widehat{\mu} \quad = \quad \frac{1}{N}\sum_{t=1}^{N} x_t. \tag{A.1}$$

- Sample variance (measure of variability):

$$\widehat{\sigma}^2 \quad = \quad \frac{1}{N-1}\sum_{t=1}^{N}(x_t - \widehat{\mu})^2. \tag{A.2}$$

- Sample coefficient of variation (alternative measure of variability):

$$\widehat{cv} \quad = \quad \frac{\widehat{\sigma}}{\widehat{\mu}}. \tag{A.3}$$

- ◇ Sample peak-to-mean ratio (measure of burstiness):

$$\widehat{b} \quad = \quad \frac{\max_t x_t}{\widehat{\mu}}. \tag{A.4}$$

A.2. Histograms

For a continuous data set, a histogram is a graphical estimate of the density function corresponding to the distribution of the samples x_t. To compute a histogram, we break up the range of values covered by the data set into k disjoint adjacent intervals $[b_0, b_1), [b_1, b_2), \ldots, [b_{k-1}, b_k)$ where the interval width $\Delta b = b_i - b_{i-1}$ is constant. The histogram is defined as the function

$$h(x) \quad = \quad \begin{cases} 0 & \text{if} \quad x < b_0, \\ h_j/\Delta b & \text{if} \quad b_{j-1} \le x < b_j \quad \text{for} \quad j = 1, 2, \ldots, k, \\ 0 & \text{if} \quad b_k \le x, \end{cases} \tag{A.5}$$

where h_j is the proportion of the samples falling into the interval $[b_{j-1}, b_j)$. The value Δb should be varied until a "smooth" histogram is obtained.

For the statistical analysis and the estimation of the model parameters, we set $b_0 = \min_t x_t$, $b_k = \max_t x_t$, and $\Delta b = (b_k - b_0)/k$. If necessary, each interval is related to a frame or GOP size s_i, which is defined as the average size of the samples that lie in the interval $[b_{i-1}, b_i)$.

If the histogram is used for matching a model distribution to the data set with emphasis on the correct matching of the tail, then both the histogram and the model probability density function should be plotted in log-scale (Figure 2.6). Otherwise, differences in the tails may not be identifiable. In addition, histogram intervals, which contain less than say 5 samples, are not considered for the matching due to the reduced statistical significance of these intervals.

A.3. Model distributions

A.3.1. Uniform distributions

The probability density function of the uniform distribution $U(a,b)$ is given by

$$f(x) \quad = \quad \begin{cases} \dfrac{1}{b-a} & \text{if} \quad a \leq x < b, \\ 0 & \text{otherwise.} \end{cases} \tag{A.6}$$

Its mean is $(b-a)/2$.

A.3.2. Normal distributions

The probability density function of the normal distribution $N(\mu_N, \sigma_N^2)$ is given by

$$f(x) \quad = \quad \frac{1}{\sqrt{2\pi\sigma_N^2}} \exp\left[\frac{-(x-\mu_N)^2}{2\sigma_N^2}\right] \tag{A.7}$$

with mean μ_N and variance σ_N^2. There is no closed form of the normal distribution function.

For a given data set, the MLE (maximum likelihood estimator) for the mean is $\widehat{\mu}_N = \widehat{\mu}$ and the MLE for the variance is $\widehat{\sigma}_N^2 = (N-1)/N \cdot \widehat{\sigma}^2$.

Figure A.1 depicts the normal and lognormal probability density functions fitted to the *dino* GOP size trace. A logscale plot of the same functions can be found in Figure 2.5.

We generate a $N(\mu_N, \sigma_N^2)$ distributed random variable X by means of two $U(0,1)$ distributed random variables U and V applying the method of Hoover and Perry(1989, p. 266).

$$x \quad = \quad \left(\sqrt{-2\log u}\sin 2\pi v\right) \cdot \sigma_N + \mu_N. \tag{A.8}$$

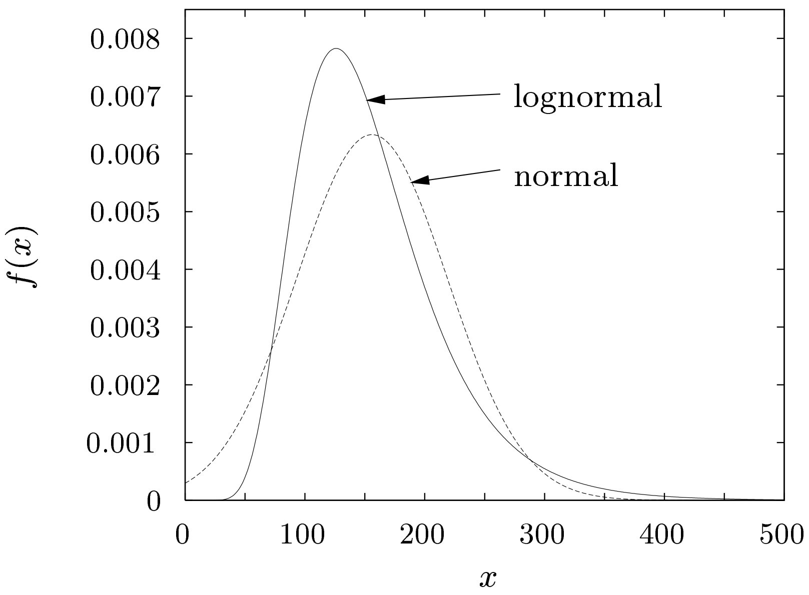


Figure A.1.: *Normal and lognormal density functions*

A.3.3. Lognormal distributions

The probability density function of the lognormal distribution $LN(\mu_L, \sigma_L^2)$ is given by

$$f(x) \quad = \quad \begin{cases} \dfrac{1}{x\sqrt{2\pi\sigma_L^2}} \exp\left[\dfrac{-(\log x - \mu_L)^2}{2\sigma_L^2}\right] & \text{if} \quad x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{A.9}$$

with mean $\exp(\mu_L) + \sigma_L^2/2$ and variance $\exp(\mu_L) + \sigma_L^2\left[\exp(\sigma_L^2) - 1\right]$. There is no closed form of the lognormal distribution function.

For a given data set, the MLE for μ_L is $\widehat{\mu}_L = (\sum_{t=1}^{N} \log x_t)/N$ and the MLE for σ_L^2 is $\widehat{\sigma}_L^2 = [\sum_{t=1}^{N}(\log x_t - \widehat{\mu}_L)^2]/N$.

A.4. Q-Q plots

Let $F(x)$ denote the distribution function of a continuous random variable. For $0 < q < 1$, the *q-quantile* of $F(x)$ is defined as the value

x_q solving $F(x_q) = q$. Let $q_i = (i - 0.5)/N$ for $i = 1, 2, \ldots, N$, i.e., $0 < q_i < 1$. For any continuous data set, a *quantile-quantile (Q-Q) plot* is a graph of the q_i-quantiles of a model distributions function $\widehat{F}(x)$, i.e., $x_{q_i}^{\text{model}} = \widehat{F}^{-1}(q_i)$, versus the q_i-quantile of the sample distribution function $\tilde{F}_N(x)$, i.e., $x_{q_i}^{\text{sample}} = \tilde{F}_N^{-1}(q_i)$.

If $\widehat{F}(x)$ is equal to the true underlying distribution $F(x)$ and if the sample size N is large, then $|\widehat{F}(x) - \tilde{F}_N(x)|$ will be small and the Q-Q plot will be approximately linear with an intercept of 0 and a slope of 1 (cf. Figures 2.3 and 2.4).

For normal and lognormal distributions, the inversion of the distribution is mathematically intractable. In both cases we use Riemann sums of the probability density functions $\widehat{f}(x)$ to obtain approximations of the integrals necessary to solve $q_i = \int_{-\infty}^{x_{q_i}} \widehat{f}(x)dx$.

The estimation of the sample q_i-quantiles is simple. First, we sort $\{x_t\}$ in ascending order and obtain $\{x_t^{\text{sorted}}\}$. Now, the q_i-quantile of $\tilde{F}_N(x)$ is given by x_i^{sorted}.

A.5. Markovian order

A process $\{X_t\}$ has a *Markovian order* p $(p = 0, 1, 2, \ldots)$ if

$$\begin{aligned} \Pr\{X_t = j | X_{t-1} = i_{t-1}, \ldots, X_0 = i_0\} &= \\ = \quad \Pr\{X_t = j | X_{t-1} = i_{t-1}, \ldots, X_{t-p} = i_{t-p}\}. \end{aligned} \tag{A.10}$$

In the coding theory literature, we find several methods to estimate the Markovian order of a given sequence or time series. We focus on the method of Merhav et al. (1989) which is based on estimating the pth-order empirical entropy $H(q_x^p)$ of a time series $\{x_t\}$. Before we are able to apply their method, we have to convert our frame or GOP size time series $\{y_t\}$ into a series of discrete states as follows. Given a number M of states of the Markov chain, the converted samples are determined by

$$x_t = \begin{cases} 1 & \text{if} \quad y_t = \min_t y_t, \\ \left\lceil \dfrac{y_t - \min\limits_t y_t}{\max\limits_t y_t - \min\limits_t y_t} \cdot M \right\rceil & \text{otherwise.} \end{cases} \tag{A.11}$$

Let $A = \{1, \ldots, M\}$ and $s_t = (x_{t-1}, x_{t-2}, \ldots, x_{t-p}) \in A^p$ with $1 \leq t \leq N$. We denote by $\delta(x_t, u, s_t, s)$ the indicator function for $x_t = u$ and $s_t = s$ with $u \in A$ and $s \in A^p$). Now let

$$q_x^p(u, s) = \frac{1}{N} \sum_{t=1} N\delta(x_t, u, s_t, s), \tag{A.12}$$

$$q_x^p(s) = \sum_{u \in A} q_x^p(u, s), \tag{A.13}$$

$$q_x^p(u|s) = \begin{cases} q_x^p(u, s)/q_x^p(s) & \text{if} \quad q_x^p(s) > 0, \\ 0 & \text{if} \quad q_x^p(s) = 0. \end{cases} \tag{A.14}$$

We define the pth-order *empirical entropy* as follows:

$$H(q_x^p) = -\sum_{s \in A^p} q_x^p(s) \sum_{u \in A} q_x^p(u|s) \log_2 q_x^p(u|s). \tag{A.15}$$

Merhav et al. (1989) use $H(q_x^p)$ to construct Markovian order estimators which are based on the assumption that an upper bound on the order exists. Since we cannot guarantee this behavior for our empirical video sequences, we give an interpretation of the $H(q_x^l)$ curve $(0 \leq l \leq l_{\max})$ for a number of values M (cf. Figure 2.9) instead of estimating the Markovian order. A large decrease of the empirical entropy from l to $l+1$ indicates that the model accuracy will be considerably improved if its order is increased from l to $l+1$. On the other hand, a small decrease indicates that increasing the model order will only lead to minor improvements.

A.6. Correlations, spectra, and periodograms

A.6.1. Definitions for stochastic processes

The autocorrelation coefficient ρ_k of $\{X_t\}$ for lag k ($k \in \mathbb{Z}$) is defined by

$$\rho_k \quad = \quad \frac{\mathrm{E}[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2} \tag{A.16}$$

with mean $\mu = \mathrm{E}[X_t]$ and variance $\sigma^2 = \mathrm{Var}[X_t]$. The curve obtained for several lags is known as *autocorrelation function (ACF).*

The spectral density $f(\lambda)$ of $\{X_t\}$, also referred to as spectrum, is defined by

$$f(\lambda) \quad = \quad \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \rho_k \exp(i\lambda k) \tag{A.17}$$

with $\lambda \in [-\pi, \pi]$.

A.6.2. Definitions for empirical data sets

The sample autocorrelation coefficient $\widehat{\rho}_k$ of $\{x_t\}$ for lag k ($k = 0, 1, \ldots, N$) is defined by

$$\widehat{\rho}_k \quad = \quad \frac{1}{N} \sum_{t=1}^{N-k} \frac{(x_t - \widehat{\mu}) \cdot (x_{t+k} - \widehat{\mu})}{\widehat{\sigma}^2}. \tag{A.18}$$

The sample periodogram $I(\lambda)$ is given by

$$I(\lambda) \quad = \quad \frac{1}{2\pi N} \left\{ \left[\sum_{t=1}^{N} (x_t - \widehat{\mu}) \cos \lambda t \right]^2 + \left[\sum_{t=1}^{N} (x_t - \widehat{\mu}) \sin \lambda t \right]^2 \right\} \tag{A.19}$$

with $\lambda \in [-\pi, \pi]$.

For a long time series $\{x_t\}$ the computation of $\widehat{\rho}_k$ with Eqn. (A.18) or $I(\lambda)$ with Eqn. (A.19) can be very time-consuming. We therefore exploit the fundamental relationship of $\{x_t\}$, $\{\widehat{\rho}_k\}$, and $I(\lambda)$ depicted in the following diagram:

$$\begin{tikzcd}
\text{time series } \{x_t\} \arrow[r, shift left, "\mathcal{F}"] \arrow[d, "\text{Eqn. (A.18)}"'] & F(\lambda) \arrow[l, shift left, "\mathcal{F}^{-1}"] \arrow[d, "\text{Eqn. (A.22)}"] \\
\text{autocorrelation fct. } \{\widehat{\rho}_k\} \arrow[r, shift left, "\mathcal{F}"] & \text{periodogram } I(\lambda) \arrow[l, shift left, "\mathcal{F}^{-1}"]
\end{tikzcd}$$

$\mathcal{F}$ denotes the discrete Fourier transform of $\{x_t : t = 1, \dots, N\}$ defined by

$$\{y_k\} \quad = \quad \mathcal{F}_{\{x_t\}}(\lambda_k) \quad = \quad \sum_{t=1}^{N} x_t \exp(i\lambda_k t) \tag{A.20}$$

and $\mathcal{F}^{-1}$ the inverse of the Fourier transform defined by

$$\{x_t\} \quad = \quad \mathcal{F}^{-1}_{\{y_k\}}(\lambda_t) \quad = \quad \frac{1}{N} \sum_{k=1}^{N} y_k \exp(-i\lambda_t k) \tag{A.21}$$

with $\lambda_k = [2\pi \cdot (k-1)]/N$, $k = 1, \dots, N$. $I(\lambda)$ is derived from $F(\lambda)$ by

$$I(\lambda) \quad = \quad |F(\lambda)|^2. \tag{A.22}$$

This leads to further alternatives estimating $I(\lambda)$, i.e.,

$$\begin{aligned} I(\lambda) \quad &= \quad \frac{1}{2\pi} \sum_{k=-(N-1)}^{N-1} \widehat{\rho}_k \exp(i\lambda k) && \text{(A.23)} \\ &= \quad \frac{1}{2\pi N} \left| \sum_{t=1}^{N} x_t \exp(i\lambda) \right|^2 . && \text{(A.24)} \end{aligned}$$

All $\mathcal{F}$ and $\mathcal{F}^{-1}$ calculations can be carried out using *FFT (Fast Fourier Transform)* techniques which speed up the computation considerably. For instance, we did not compute $\{\widehat{\rho}_k\}$ by Eqn. (A.18) but via $I(\lambda)$.

A.7. Selfsimilarity

A.7.1. Definition

The most common way to define selfsimilarity of a process $\{X_t : t \in \mathbb{Z}\}$ is by means of its distribution: if $\{X_{at}\}$ and $a^H \{X_t\}$ have identical finite-dimensional distributions for all $a > 0$, then $\{X_t\}$ is selfsimilar with parameter H (see Taqqu (1988)). In our case, however, we need a definition which is more related to the properties of time series and which is more appropriate for the development of estimators for the selfsimilarity parameter H (see Willinger et al. (1995)).

Let $\{X_t : t = 0, 1, 2, \dots\}$ be a covariance stationary stochastic process with mean μ, variance σ^2, and autocorrelation function $\rho_k, k \geq 0$. In particular, we assume that $\{X_t\}$ has an autocorrelation function of the form

$$\rho_k \sim k^{-\beta} L(k) \quad \text{as} \quad k \to \infty, \tag{A.25}$$

where $0 < \beta < 1$ and L is slowly varying at infinity. For simplicity, we assume that L is asymptotically constant. For each $m = 1, 2, \dots,$ let $\{X_t^{(m)}, t = 1, 2, \dots\}$ denote the process obtained by averaging the original process $\{X_t\}$ over non-overlapping blocks of size m, i.e. $\{X_t^{(m)}\}$ is given by $X_t^{(m)} = (X_{(t-1)m} + ... + X_{tm-1})/m$.

A process $\{X_t\}$ is called *(exactly) second-order selfsimilar* with selfsimilarity parameter $H = 1 - \beta/2$ if, for all $m = 1, 2, \dots,$ $\sigma^2_{X^{(m)}} = \sigma^2 m^{-\beta}$, and

$$\rho_k^{(m)} \quad = \quad \rho_k \quad = \quad \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + |k-1|^{2H}\right], \quad k \geq 0, \tag{A.26}$$

where $\{\rho_k^{(m)}\}$ denotes the autocorrelation function of $\{X_t^{(m)}\}$.

A process $\{X_t\}$ is called *(asymptotically) second-order selfsimilar* with selfsimilarity parameter $H = 1 - \beta/2$ if, for all k large enough,

$$\rho_k^{(m)} \to \rho_k \quad \text{as} \quad m \to \infty. \tag{A.27}$$

In other words, $\{X_t\}$ is second-order selfsimilar if the corresponding aggregated processes $\{X_t^{(m)}\}$ are the same as $\{X_t\}$ or become indistinguishable from $\{X_t\}$ at least with respect to their autocorrelation functions.

A.7.2. Properties of selfsimilar processes

The following properties of selfsimilar processes are equivalent:

- ◇ *Hurst effect.* The rescaled adjusted range statistic (see section B.4.2) is characterized by a power law: $\mathrm{E}[R(m)/S(m)] \sim a_1 m^H$ as $m \to \infty$ with $0.5 < H < 1$.
- ◇ *Slowly decaying variances.* The variances of the sample mean are decaying more slowly than the reciprocal of the sample size, i.e. $\sigma^2_{X(m)} \sim a_2 m^{2H-2}$ as $m \to \infty$, with $0.5 < H < 1$. As a consequence, classical statistical tests and confidence intervals lead to erroneous results.
- ◇ *Long-range dependence.* The autocorrelations decay hyperbolically rather than exponentially. This implies a non-summable autocorrelation function $\sum_k \rho_k = \infty$. This implies that even though the ρ_k's are individually small for large lags, their cumulative effect is important.
- ◇ *1/f-noise.* The spectral density $f(\cdot)$ obeys a power law near the origin, i.e. $f(\lambda) \sim a_3 \lambda^{1-2H}$, as $\lambda \to 0$, with $0.5 < H < 1$.

The constants a_i are finite, positive, and independent of m or λ, respectively.

In contrast to the above properties, short-range dependent processes, i.e. $H = 0.5$, show the following characteristics:

- ◇ $\sigma^2_{X(m)} \sim a_1 m^{-1}$.
- ◇ $0 < \sum_k \rho_k < \infty$.
- ◇ $f(\lambda)$ at $\lambda = 0$ is positive and finite.

B. Stochastic processes

B.1. White noise processes

A stochastic process $\{X_t\}$ is called *white noise process*, in short white noise, if the samples X_t form an independent and identically distributed (i.i.d.) sequence of random variables. It is completely characterized by the arbitrary but fixed distribution of the X_t.

The spectrum of the white noise process is given by

$$f(\lambda) = 1 \tag{B.1}$$

with $\lambda \in [-\pi, \pi]$.

B.2. Markov chains

B.2.1. Definition and characteristics

We define a *Markov chain* as a discrete-time Markov process with a discrete state space. We assume an irreducible homogeneous Markov chain with a finite state space $\{1, \dots, M\}$ and positive recurrent states. These assumptions are achieved for our Markov models by construction. This Markov chain $\{X_t\}$ is characterized by its one-step transition matrix $\mathbf{P} = [p_{ij}]$ containing the conditional probabilities

$$p_{ij} \quad = \quad \Pr\{X_t = j | X_{t-1} = i\}, \quad \text{with} \quad i, j \in \{1, \dots, M\}. \tag{B.2}$$

If the Markov chain is also aperiodic there exists a steady-state distribution $\boldsymbol{\pi} = [\pi_1, \dots, \pi_M]$ where π_i denotes the probability that the Markov chain is in state i in equilibrium. The vector $\boldsymbol{\pi}$ is given by the solution of

$$\boldsymbol{\pi} \quad = \quad \boldsymbol{\pi}\mathbf{P}, \qquad \sum \boldsymbol{\pi} = 1. \tag{B.3}$$

The above definition can be generalized to pth-order Markov chains by appropriately redefining the state space. For instance, we can describe a 2nd-order Markov chain by the transition probabilities of state pairs of the process $\{X_t\}$ in the following way:

$$p_{(i,j)(j,k)} \quad = \quad \Pr\{X_t = k | X_{t-1} = j, X_{t-2} = i\}. \tag{B.4}$$

In the remainder of this section we focus on first-order Markov chains but the results can easily be applied to higher-order Markov chains. For the modeling of the frame or GOP processes it is necessary to relate every state of the Markov model to a particular frame or GOP size denoted by s_i with $i \in \{1, \dots, M\}$. In other words, every time the Markov state enters state i a frame or GOP of size s_i is generated. The Markov model is fully characterized by the transition matrix $\mathbf{P}$ and the size vector $\mathbf{s} = [s_1, \dots, s_M]$. The mean and variance of this process are given by

$$\mu \quad = \quad \boldsymbol{\pi} \cdot \mathbf{s}^T, \qquad \sigma^2 \quad = \quad \sum \boldsymbol{\pi} \cdot diag(\mathbf{s})^2 - \mu^2 \tag{B.5}$$

where $diag(\mathbf{s})$ denotes a square diagonal matrix with diagonal $\mathbf{s}$. The correlation coefficient for lag k is obtained by

$$\rho_k \quad = \quad \frac{1}{\sigma^2}\left[\sum \boldsymbol{\pi} \cdot diag(\mathbf{s}) \cdot \mathbf{P}^k \cdot diag(\mathbf{s}) - \mu^2\right]. \tag{B.6}$$

B.2.2. Parameter estimation

There are several methods to determine the transition probability matrix. We focus on the most popular approach: to determine maximum likelihood estimates of the transition probabilities as suggested by e.g. Billingsley (1961). The main disadvantage of this method is the number parameters involved to describe the model. Therefore Heyman et al. (1992) suggest to use the approach of Jacobs and Lewis (1983). Assuming a negative binomial distribution of the samples, this approach needs only two parameters to describe the marginal distribution and one correlation parameter for the generation of the transition probability matrix. We do not consider this approach since an improvement in approximating correlations at larger lags cannot be achieved as natural as for the maximum likelihood estimates method. Hwang and Li (1995) report about a tool which generates the transition matrix based on the histogram and power spectrum of the empirical data set. As an example, they provide results for a video trace which indicate that their approach leads to reasonable results. We do not consider their complex approach since the approach presented in the following already lead to a good model quality.

Before we estimate the entries of the transitions matrix $\mathbf{P}$ we have to convert our frame or GOP size time series $\{y_t\}$ into discrete states as follows. Given a number M of states of the Markov chain, the discretized samples are determined by

$$\begin{aligned} x_t &= 1 && \text{if} \quad y_t = \min\nolimits_t y_t \\ x_t &= \left\lceil \frac{y_t - \min\limits_t y_t}{\max\limits_t y_t - \min\limits_t y_t} \cdot M \right\rceil && \text{otherwise} \end{aligned} \tag{B.7}$$

Let $\delta_p(x_t, i, x_{t+1}, j)$ denote the indicator function of $x_t = i$ and $x_{t+1} = j$ with $t = 1, \ldots, N-1$ and $i, j \in \{1, \ldots, M\}$ and let $\delta_s(x_t, i)$ denote the indicator function of $x_t = i$ with $t = 1, \ldots, N$ and $i \in \{1, \ldots, M\}$. The MLE of p_{ij} is given by

$$\widehat{p}_{ij} = \frac{\displaystyle\sum_{t=1}^{N-1} \delta_p(x_t, i, x_{t+1}, j)}{\displaystyle\sum_{k=1}^{M} \sum_{t=1}^{N-1} \delta_p(x_t, i, x_{t+1}, k)} \tag{B.8}$$

and the sizes s_i by

$$s_i = \frac{\displaystyle\sum_{t=1}^{N} \delta_s(x_t, i) \cdot y_t}{\displaystyle\sum_{t=1}^{N} \delta_s(x_t, i)}. \tag{B.9}$$

B.3. Autoregressive processes

B.3.1. Definition and characteristics

A stochastic process $\{X_t\}$ is called *autoregressive process of order* p, in short AR(p) process, if it is defined by the recurrence relation

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + \epsilon_t, \quad t \in \mathbb{Z},\ \alpha_i \in \mathbb{R}, \tag{B.10}$$

where $\{\epsilon_t\}$ is a white noise process.

If we define the characteristic function of the AR(p) process as

$$\alpha(x) = 1 - \sum_{i=1}^{p} \alpha_i x^i \tag{B.11}$$

we obtain a more concise form of Eqn. (B.10):

$$\alpha(B) X_t = \epsilon_t, \tag{B.12}$$

where B denotes the backshift operator, i.e., $BX_t = X_{t-1}$.

The spectrum of the AR(p) process is given by

$$f(\lambda) \quad = \quad \frac{\sigma_\epsilon^2}{\alpha(\exp i\lambda)} \tag{B.13}$$

with $\lambda \in [-\pi, \pi]$.

B.3.2. Parameter estimation

Given an order p, we use the *Yule-Walker estimates* $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_p$ as parameters for the model AR(p) process of a given time series $\{x_t\}$. They are obtained by solving the following system of linear equations:

$$\begin{bmatrix} \widehat{\rho}_0 & \widehat{\rho}_1 & \cdots & \widehat{\rho}_{p-2} & \widehat{\rho}_{p-1} \\ \widehat{\rho}_1 & \widehat{\rho}_0 & \cdots & \widehat{\rho}_{p-3} & \widehat{\rho}_{p-2} \\ \vdots & \vdots & & \vdots & \vdots \\ \widehat{\rho}_{p-1} & \widehat{\rho}_{p-2} & \cdots & \widehat{\rho}_1 & \widehat{\rho}_0 \end{bmatrix} \begin{bmatrix} \widehat{\alpha}_1 \\ \widehat{\alpha}_2 \\ \vdots \\ \widehat{\alpha}_p \end{bmatrix} = \begin{bmatrix} \widehat{\rho}_1 \\ \widehat{\rho}_2 \\ \vdots \\ \widehat{\rho}_p \end{bmatrix} \tag{B.14}$$

For Gaussian sequences this estimator is asymptotically normal and efficient.

We estimate the process order $\widehat{p}$ by minimizing *Akaike's Information Criterion (AIC)* which is defined by

$$AIC(p) \quad = \quad \log \tilde{\sigma}_p^2 + \frac{2 \cdot p}{N} \tag{B.15}$$

where $\tilde{\sigma}_p^2$ is the variance of the residuals $\{\widehat{\epsilon}_k\}$ given by

$$\widehat{\epsilon}_t \quad = \quad x_t - (\widehat{\alpha}_1 x_{t-1} + \ldots + \widehat{\alpha}_p x_{t-p}) \tag{B.16}$$

B.4. Selfsimilar processes

B.4.1. Definitions and characteristics

For the analysis and simulation of systems where selfsimilar processes are involved, there is a need of processes which exhibit this particular property.

In this section, we give the definitions of two processes of this type, namely the *fractional Gaussian noise (FGN)*(see Mandelbrot and Ness (1968)) and the *fractional autoregressive integrated moving-average processes (FARIMA)* (see Hosking (1984)).

These processes were introduced to facilitate parsimonious modeling of long-range dependent time series. Traditional models, for instance *autoregressive (AR)* and *autoregressive integrated moving-average (ARIMA) processes*, are only capable to model the short-range portion of the correlations of empirical data sets, and even for a large number of coefficients the long-range portion will remain unmodeled.

The FGN process with parameter $H \in (0,1)$ is a stationary Gaussian process with mean μ, variance σ^2, and autocorrelation function ρ_k, $k > 0$ as in Eqn. (A.26). It is exactly selfsimilar if $0.5 < H < 1$. Its spectrum is defined by

$$f(\lambda) \quad = \quad \frac{\sigma^2}{\pi} \sin(\pi H)\, \Gamma(2H+1)\,(1-\cos\lambda) \sum_{j=-\infty}^{\infty} |\lambda + 2\pi j|^{-2H-1}. \tag{B.17}$$

FGN(H) can also be defined as the process which has the same correlation function as the process of unit increments $\Delta B_H(t) = B_H(t) - B_H(t-1)$ of fractional Brownian motion with exponent H.

Instead of using the increment process of fractional Brownian motion, we can also start from Brownian motion and its discrete time analogue, the random walk. This will lead to the class of FARIMA processes, since the random walk can also be defined as an FARIMA(0,1,0) process. For a FARIMA(p,d,q) process the orders p and q are the classical ARMA parameters and $d = H - 0.5$ is the fractional difference parameter.

With $(1-B)X_t = \epsilon_t$ being the representation of the FARIMA(0,1,0) process, where B denotes the back-shift operator and $\{\epsilon_t\}$ a white noise process with variance σ_ϵ^2, we generalize this definition for the FARIMA$(0,d,0)$ processes as follows:

$$(1-B)^d X_t \quad = \quad a_t \quad \text{for} \quad -0.5 < d < 0.5. \tag{B.18}$$

The fractional difference operator $(1-B)^d$ is defined by the binomial series $(1-B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-B)^k$. The spectrum is

$$f(\lambda) \quad = \quad \frac{\sigma_\epsilon^2}{2\pi}(2\sin\frac{\lambda}{2})^{-2d} \quad = \quad \frac{\sigma_\epsilon^2}{2\pi}|1-e^{i\lambda}|^{-2d} \text{ for } 0 < \lambda \leq \pi. \quad \text{(B.19)}$$

Thus, $f(\lambda) \sim \lambda^{-2d} = \lambda^{1-2H}$ as $\lambda \to 0$.

If a good approximation of both the short-range and the long-range correlations is mandatory the FARIMA(p,d,q) processes with non-zero p and/or q should be used. A FGN process is not applicable in such cases since its low-lag correlation behavior cannot be fitted to that one of empirical data sets. In our studies, we focus on FARIMA$(p,d,0)$ processes which have a spectrum given by

$$f(\lambda) \quad = \quad \frac{\sigma_\epsilon^2}{2\pi} \cdot \frac{(2\sin\frac{\lambda}{2})^{-2d}}{\alpha(\exp i\lambda)} \quad \text{for} \quad 0 < \lambda \leq \pi \qquad \text{(B.20)}$$

where $\alpha(\cdot)$ is defined as in Section B.3.1.

B.4.2. Parameter estimation

The properties of selfsimilar processes (see section Section A.7.2) lead to the following methods to estimate H (see Willinger et al. (1995) and references therein):

(1) time-domain analysis: *R/S statistics*,

(2) analysis of the variances of the aggregated processes $X^{(m)}$: *variance-time plots*, and

(3) periodogram-based analysis in the frequency-domain: *Whittle's maximum likelihood estimator*.

In the following, we focus on the alternatives (1) and (3).

R/S analysis

The feature that makes *R/S analysis* particularly attractive is its robustness against changes in the marginal distribution, even for long-tailed or skew distributions. On the other hand, for marginal distributions which are close to normality a dramatic loss in efficiency is reported, and, to our best knowledge, no detailed analysis of robustness of R/S statistics was carried out yet. Given an empirical time series $\{x_t : t = 1, \dots , N\}$, the whole series is subdivided into K non-overlapping blocks. Now, we compute the rescaled adjusted range $R(t_i, d)/S(t_i, d)$ for a number of ranges r, where $t_i = \lfloor N/K \rfloor (i-1)+1$ are the starting points of the blocks which satisfy $(t_i - 1) + r \leq N$.

$$\begin{aligned} R(t_i, r) \quad &= \quad \max\{0, W(t_i, 1), \dots , W(t_i, r)\} - \\ &\quad - \min\{0, W(t_i, 1), \dots , W(t_i, r)\}, \end{aligned} \qquad \text{(B.21)}$$

where

$$W(t_i, k) \quad = \quad \sum_{j=1}^{k} x_{t_i+j-1} - k \cdot \left(\frac{1}{r} \sum_{j=1}^{r} x_{t_i+j-1} \right), \quad k = 1, \dots , r. \quad \text{(B.22)}$$

Let $S^2(t_i, r)$ denote the sample variance of $x_{t_i}, \dots , x_{t_i+r-1}$. For each value r we obtain a number of R/S samples. For small values r there are K samples. The number decreases for larger ranges r due to the limiting condition on the t_i values mentioned above. These samples are computed for logarithmically spaced values r, i.e. $r_{l+1} = m \cdot r_l$ with $m > 1$, starting with a value r_0 of about 10. Plotting $\log [R(t_i, r)/S(t_i, d)]$ versus $\log r$ results in the R/S plot.

Next, a least squares line is fitted to the points of the R/S plot, where the R/S samples of the extremal ranges are not considered. The R/S samples of the smallest ranges are dominated by short-range correlations and samples of large ranges are statistically insignificant if the number of samples per range is less than say 5. The slope of the regression line for these R/S samples is an estimate for the Hurst parameter H. Both the number of blocks K

and the number of values r should not be chosen too small. In addition, some care has to be taken when deciding about the end of the transient, i.e. which of the small values of r should not be taken into consideration for the regression line. In practice, it has to be checked whether different parameter settings lead to consistent H estimates for $\{x_t^{(m)}\}$ with different aggregation levels m.

Figure B.1 shows the R/S plot of the *dino* GOP sequence with $K = 8$ and 30 columns of R/S samples. The regression line has a slope of 0.8 indicating a Hurst parameter estimate of $\widehat{H} = 0.8$.

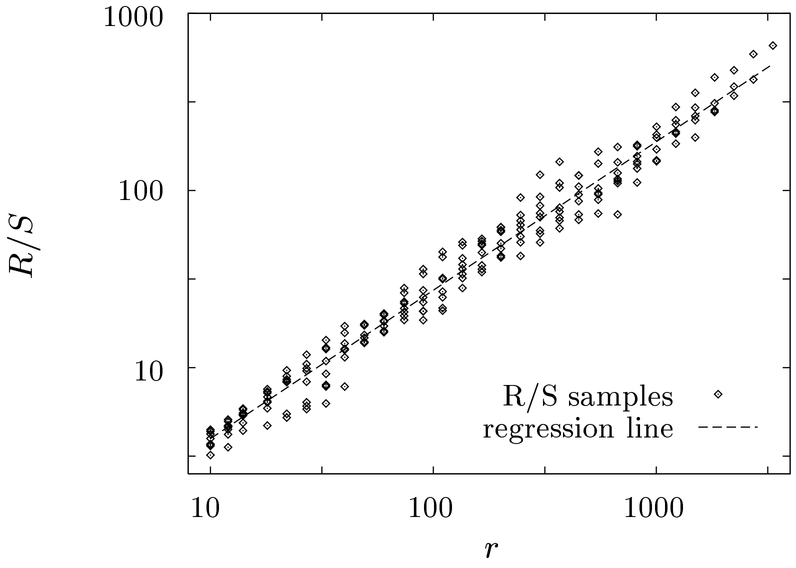


Figure B.1.: *R/S plot of the dino GOP sequence*

Periodogram-based analysis

If more information is needed on the H-estimate, such as confidence intervals, or information on the estimator itself, such as efficiency and robustness, periodogram-based estimators are used. In addition, these estimators facilitate the estimation of short-range correlation parameters. The main idea of this method is to assume a certain selfsimilar process type, say a FARIMA(p, d, q) process, and to fit the parameters of this process to the given empirical sample. The fitting should be optimal in the sense that the periodogram of the sample and the spectral density of the process are minimizing a given goodness-of-fit function.

As mentioned above, the spectral density of selfsimilar processes obeys a power law near the origin. Thus, the first idea to determine the Hurst parameter H is simply to plot the periodogram in a log-log grid, and to compute the slope of a regression line which is fitted to a number of low frequencies. This should be an estimate of $1 - 2H$. In most of the cases this will lead to a wrong estimate of H since the periodogram is not appropriate to estimate the spectral density (see Schlittgen and Streitberg (1995)). More sophisticated methods have to be applied to obtain useful estimates of H.

Several periodogram-based estimators can be found in the literature. In this paper we will focus on an MLE as presented in Beran (1992) and Willinger et al. (1995) which is based on Whittle's approximate MLE for Gaussian processes (1953). For Gaussian sequences this estimator is asymptotically normal and efficient (see Fox and Taqqu (1986), Dahlhaus (1989)).

The spectral density of the selfsimilar process is denoted by $f(\lambda; \theta)$, where the parameter vector of the process $\theta = (\theta_1, \dots, \theta_M)$ is structured as follows. $\theta_1 = \sigma_\epsilon^2$ is a scale parameter, where σ_ϵ^2 is the variance of the innovation ϵ of the infinite AR-representation of the process, i.e., $X_t = \sum_{i=1}^{\infty} \alpha_i X_{t-i} + \epsilon_t$. This implies $\int_{-\pi}^{\pi} \log\{f[\lambda; (1, \theta_2, \dots, \theta_M)]\}\, d\lambda = 0$. θ_2 denotes the Hurst parameter H. If necessary, the parameters θ_3 to θ_M describe the short-range behavior of the process. For FGN and FARIMA$(0, d, 0)$, only σ_ϵ^2 and H have to be considered. With $\eta = (\theta_2, \dots, \theta_M)$, the Whittle estimator $\widehat{\eta}$ of η minimizes the quality-of-fit function

$$Q(\eta) \quad = \quad \int_{-\pi}^{\pi} \frac{I(\lambda)}{f(\lambda; (1, \eta))} d\lambda \tag{B.23}$$

where $I(\cdot)$ denotes the periodogram of the given time series of length N

defined by Eqn. (A.24). $\widehat{H}$ is given by $\widehat{\theta}_2$ and the estimate of σ_ϵ^2 by

$$\widehat{\sigma}_\epsilon^2 \quad = \quad \int_{-\pi}^{\pi} \frac{I(\lambda)}{f(\lambda;(1,\widehat{\eta}))} d\lambda. \tag{B.24}$$

The approximate 95%-confidence interval of the $\widehat{\eta}_i$ is given by

$$\widehat{\eta}_i \quad \pm \quad 1.96\sqrt{\frac{V_{ii}}{N}} \tag{B.25}$$

where $V = 2D^{-1}$ and the matrix D is defined by

$$D_{ij} \quad = \quad \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\partial}{\partial\eta_i}\log f(\lambda)\frac{\partial}{\partial\eta_j}\log f(\lambda)d\lambda \quad i,j \in \{1,\ldots,M\}. \tag{B.26}$$

For implementation details, we suggest to consider Chapter 12.1 of Beran (1994), where an $S+$ listing of the Whittle estimator is provided. Given some knowledge in numerical analysis, no special library functions are necessary to implement the above formulae. However, FFT, vector, and matrix functions would make the programming more convenient.

In practice, there are two problems which may have an effect on the robustness of the estimator:

◇ *Deviations from the model spectrum assumed.* Deviations at higher frequencies lead to a bias in the estimate of H. One possible solution is to estimate H only from periodogram ordinates at low frequencies. For large data sets, one can also aggregate the data over non-overlapping blocks of length m and compute several $\widehat{H}^{(m)}$ for the $X^{(m)}$. We, however, prefer using a model process which is also able to model short range correlations such as FARIMA(p,d,q) processes. Then, we are able to use the full length time series what leads to a higher estimation accuracy. On the other hand, we have the problem to determine the appropriate process orders p and q.

- ◇ *Deviations from Gaussianity.* Gaussianity can often be achieved by transforming the data, but then it has to be proven that the estimates of H for the original and the transformed data sets are identical (see Huang et al. (1995)). For instance, this is the case for the log-transformation $\{y_k\} = \log\{x_k\}$. We apply the log-transformation to our approximately lognormally distributed video data sets to obtain a Gaussian marginal distribution.

As mentioned above, we have to determine the model orders p and q if we assume a FARIMA(p, d, q) model for our data set. Since we focus on FARIMA$(p, d, 0)$ processes we have to find an appropriate value of p for our model. In contrast to order estimators for AR (see Eqn. (B.15)) or ARMA models, only little is known about such estimators for FARIMA models. Hosking (1984) reports about an extension of Akaike's Information Criterion (AIC) for FARIMA(p, d, q) models. Beran (1995) and Beran and Bhansali (1996) report about model selection and parameter estimation of FARIMA(p, d, q) processes. Nevertheless, we developed our own graphical method to determine the value p for the following reasons. Our primary interest is not the statistical analysis as such, but the computation of model parameters. We intend to determine the smallest number of model parameters possible that leads to good predictions of the behavior of the modeled system. This has not necessarily to be the same number which is provided by a statistical order estimator designed for objectives such as normality or consistence.

Our data sets are processed as follows. We first log-transform our data set to obtain approximately Gaussian marginals. Then, we compute the FARIMA parameters for a number of p values starting with $p = 0$. For each of these values we plot the corresponding FARIMA spectrum and the periodogram of the data set in a log-log grid.

Figure B.2 shows the spectrum of the fitted FARIMA$(1, 0.4, 0)$ process and the periodogram of the *dino* GOP size trace. This type of plot shows whether the slope of the spectrum is correct for the low frequency part, i.e.,

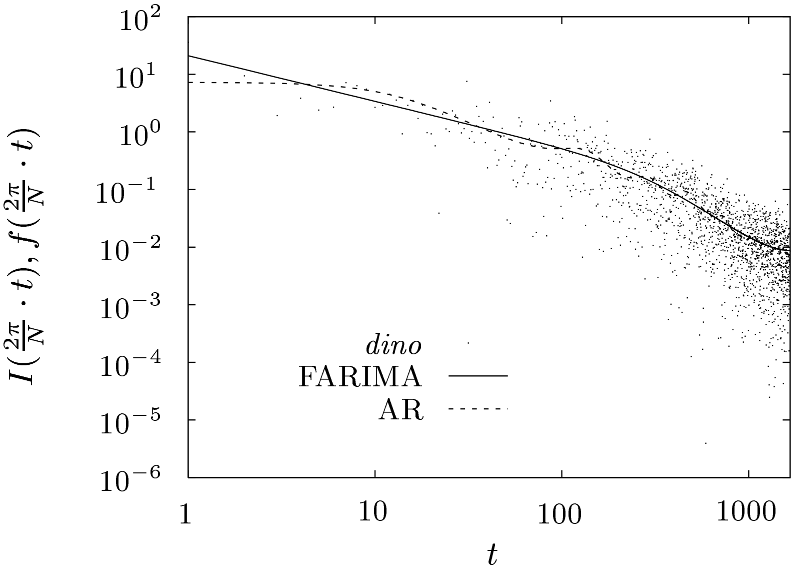


Figure B.2.: *Periodogram and fitted spectrum of dino GOP sizes*

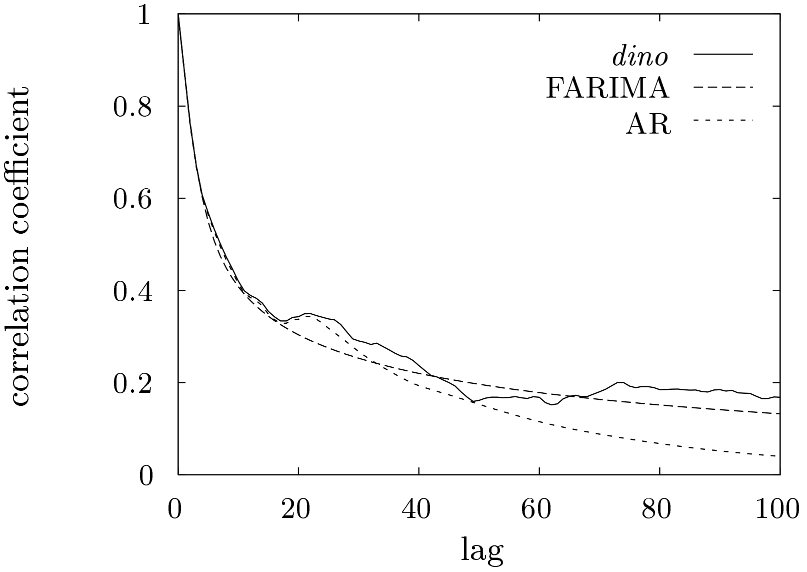


Figure B.3.: *ACF of dino GOP sizes and model processes*

the long-range correlations. However, it is hard to decide whether the model is also appropriate for the high frequency part, i.e., the short-range correlations. We therefore transform both the spectrum and the periodogram using an inverse Fourier transform (see Section A.6) to obtain the autocorrelation functions.

Figure B.3 shows the sample ACF of the *dino* GOP sizes and the ACF of the FARIMA process mentioned above. The FARIMA$(1, 0.4, 0)$ process provides a good approximation of both short and long-range correlations of the *dino* GOP size trace up to a lag of 100. Note, that due to our graphical approach to determine the order p, no error bounds on this estimate can be given.

Bibliography

Adas, A. (1996). Supporting real time VBR video using dynamic reservation based on linear prediction. In *Proceedings of the Infocom '96*, pp. 1476–1483.

Adas, A. and A. Mukherjee (1995). On resource management and QoS guarantees for long range dependent traffic. In *Proceedings of the Infocom '95*, pp. 779–787.

Andreassen, R. (1995). Cell losses of multiplexed VBR MPEG sources in an ATM-multiplexer. In *Proceedings of the 12th Nordic Teletraffic Seminar.*

Anick, D., D. Mitra, and M. M. Sondhi (1982). Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal 61*(8), 1870–1894.

Aravind, R., G. L. Cash, D. L. Duttweiler, H.-M. Hang, B. G. Haskell, and A. Puri (1993). Image and video coding standards. *AT&T Technical Journal* (January), 67–89.

ATM Forum Technical Committee (1996a). *Audiovisual Multimedia Services: Video on Demand Specification 1.0.*

ATM Forum Technical Committee (1996b). *Traffic Management Specification. Version 4.0.*

Beran, J. (1992). Statistical methods for data with long-range dependence. *Statistical Science 7*(4), 404–427.

Beran, J. (1994). *Statistics for long-memory processes.* New York: Chapman & Hall.

Beran, J. (1995). Maximum likelihood estimation of the differencing parameter for invertible short- and long-memory ARIMA models. *Journal of the Royal Statistical Society B 57*(4), 695–672.

Beran, J. and R. J. Bhansali (1996). On unified model selection for stationary and nonstationary short- and long-memory autoregressive processes. Department of Economics and Statistics, University of Konstanz, Germany. Preprint.

Billingsley, P. (1961). *Statistical Inference for Markov Processes.* Chicago: University of Chicago Press.

Blondia, C. and O. Casals (1992). Statistical multiplexing of VBR sources: A matrix-analytic approach. *Performance Evaluation 16*, 5–20.

Buzacott, J. A. and J. G. Shanthikumar (1993). *Stochastic Models of Manufacturing Systems.* Englewood Cliffs: Prentice-Hall.

CCITT (1990). *Video Codec for Audiovisual Services at px64 kbit/s. Recommendation H.261.*

CCITT (1991). *BISDN ATM Adaption Layer (AAL) Functional Description. Recommendation I.321.*

Chan, S. K. and A. Leon-Garcia (1994). Analysis of cell inter-arrival from VBR video codecs. In *Proceedings of the Infocom '94*, pp. 350–357.

Chandra, K. and A. R. Reibman (1996). Modeling two-layer SNR-scalable MPEG-2 video traffic. In *Proceedings of the 7th International Workshop on Packet Video.*

Chang, C.-J. and S.-Y. Wang (1992). Performance analysis of a statistical multiplexer for integrated service in the customer-premise equipment.

Computer Networks and ISDN Systems 25, 191–201.

Chou, L.-S. and C.-S. Chang (1996). Experiments of the theory of effective bandwidth for Markov sources and video traces. In *Proceedings of the Infocom '96*, pp. 497–504.

Chowdhury, S. and K. Sohraby (1994). Bandwidth allocation algorithms for packet video in ATM networks. *Computer Networks and ISDN Systems 26*, 1215–1223.

Coelho, R. and S. Tohme (1993). Video coding mechanism to predict video traffic in ATM networks. In *Proceedings of the Globecom '93*, pp. 447–451.

Cohen, D. M. and D. P. Heyman (1993). Performance modeling of video teleconferencing in ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology 3*(6), 408–420.

Cosmas, J. and A. Odinma-Okafor (1991). Characterisation of variable rate video codecs in ATM to a geometrically modulated deterministic process model. In *Proceedings of the ITC-13*, pp. 773–780.

Dahlhaus, R. (1989). Efficient parameter estimation for self-similar processes. *The Annals of Statistics 17*(4), 1749–1766.

Enssle, J. (1994). Modelling and statistical multiplexing of VBR MPEG compressed video in ATM networks. In *Proceedings of the 4th Open Workshop on High Speed Networks, Brest, France*, pp. 59–67.

Enssle, J. (1995). Modelling of short and long term properties of VBR MPEG compressed video in ATM networks. In *Proceedings of the Silicon Valley Networking Conference and Exposition*, pp. 95–107.

Enssle, J. (1996). The impact of VBR MPEG video traffic on ATM multiplexer performance and its evaluation. In *Proceedings of the 4th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks.*

Fox, R. and M. S. Taqqu (1986). Large-sample properties of parameter estimates for strongly dependent stationary Gaussian time series. *The Annals of Statistics 14*(2), 517–532.

Frater, M. R., J. F. Arnold, and P. Tan (1994). A new statistical model for traffic generated by VBR CODECs for television on the broadband ISDN. *IEEE Transactions on Circuits & Systems for Video Technology 4*(6), 521–526.

Frost, V. S. and B. Melamed (1994). Traffic modeling for telecommunication networks. *IEEE Communications Magazine 32*(3), 70–81.

Garrett, M. W. and W. Willinger (1994). Analysis, modeling and generation of self-similar VBR video traffic. In *Proceedings of the ACM SIGCOMM '94 Conference*, pp. 269–280.

Gong, K. L. (1994). *Berkeley MPEG-1 Video Encoder, User's Guide.* University of California, Berkeley, Computer Science Division-EECS.

Grünenfelder, R., J. Cosmas, S. Manthorpe, and A. Odinma-Okafor (1991a). Characterization of video codecs as autoregressive moving average processes and related queueing system performance. *IEEE Journal on Selected Areas in Communications 9*(3), 284–193.

Grünenfelder, R., J. Cosmas, S. Manthorpe, and A. Odinma-Okafor (1991b). Measurement and ARMA model of video codecs in an ATM environment. In *Proceedings of the ITC-13*, pp. 981–985.

Gusella, R. (1991). Characterizing the variability of arrival processes with indexes of dispersion. *IEEE Journal on Selected Areas in Communications 9*(2), 203–211.

Heyman, D. P. and T. V. Lakshman (1994). Source models for VBR broadcast-video traffic. In *Proceedings of the Infocom '94*, pp. 664–671.

Heyman, D. P., A. Tabatabai, and T. V. Lakshman (1992). Statistical analysis and simulation study of video teleconference traffic in ATM networks. *IEEE Transactions on Circuits and Systems for Video Technology 2*(1), 49–59.

Hoover, S. V. and R. F. Perry (1989). *Simulation: A Problem-Solving Approach.* Reading: Addison-Wesley.

Hosking, J. R. M. (1984). Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research 20*(12), 1898–1908.

Huang, C., M. Devetsikiotis, I. Lambadaris, and A. R. Kaye (1995). Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. In *Proceedings of the ACM SIGCOMM '95 Conference.*

Hübner, F. (1994). Dimensioning of a peak cell rate monitor algorithm using discrete-time analysis. In *Proceedings of the ITC-14*, pp. 1415–1424.

Hwang, C.-L. and S.-Q. Li (1995). On the convergence of traffic measurement and queueing analysis: A Statistical-MAtch Queueing (SMAQ) Tool. In *Proceedings of the Infocom '95*, pp. 602–612.

Ismail, M. R., I. E. Lambadaris, M. Devetsikiotis, and A. R. Kaye (1995). Modelling prioritized MPEG video using TES and a frame spreading strategy for transmission in ATM networks. In *Proceedings of the Infocom '95*, pp. 762–770.

ISO (1991). *Digital Compression and Coding of Continuous-tone Still Images – Part 1: Requirements and Guidelines. Draft International Standard: ISO/IEC DIS 10918-1.*

ISO (1993). *Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 Mbit/s – Part 2: Video. International Standard: ISO/IEC IS 11172-2.*

ISO (1994). *Generic Coding of Moving Pictures and Associated Audio – Part 2: Video. International Standard: ISO/IEC IS 13818-2.*

ITU-T (1994). *Recommendation I.371: Traffic control and congestion control in B-ISDN (frozen issue).*

Jacobs, P. A. and P. A. W. Lewis (1983). Time series generated by mixtures. *Journal on Time Series Analysis 4*(1), 19–36.

Kelly, F. P. (1996). Notes on effective bandwidths. In F. P. Kelly, S. Zachary, and I. Ziedins (Eds.), *Stochastic Networks: Theory and Applications*, pp. 141–168. Oxford: Oxford University Press.

Kleinrock, L. (1975). *Queueing Systems – Volume 1: Theory*. New York: Wiley.

Krunz, M., R. Sass, and H. Hughes (1995). Statistical characteristics and multiplexing of MPEG streams. In *Proceedings of the Infocom '95*, pp. 455–462.

Law, A. M. and W. D. Kelton (1991). *Simulation modeling and analysis* (2nd ed.). New York: Mc Graw-Hill.

Le Gall, D. (1991). MPEG: A video compression standard for multimedia applications. *Communications of the ACM 34*(4), 46–58.

Lee, D.-S., B. Melamed, A. R. Reibman, and B. Sengupta (1994). TES modeling for analysis of a video multiplexer. *Performance Evaluation 16*, 21–34.

Leland, W. E., M. S. Taqqu, W. Willinger, and D. V. Wilson (1994). On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking 2*(1), 1–15.

Li, S.-Q. and J. W. Mark (1985). Performance of voice/data integration on a TDM system. *IEEE Transactions on Communications COM-33*(12), 1265–1273.

Likhanov, N., B. Tsybakov, and N. D. Georganas (1995). Analysis of an ATM buffer with self-similar ("fractal") input traffic. In *Proceedings of the Infocom '95*, pp. 985–992.

Livny, M., B. Melamed, and A. K. Tsiolis (1993). The impact of autocorrelation on queuing systems. *Management Science 39*(3), 322–339.

Lucantoni, D. M., M. F. Neuts, and A. R. Reibman (1994). Methods for performance evaluation of VBR video traffic models. *IEEE/ACM Transactions on Networking 2*(2), 176–180.

Maglaris, B., D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins (1988). Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications 36*(7), 834–844.

Mandelbrot, B. B. and J. W. V. Ness (1968). Fractional brownian motions, fractional noises and applications. *SIAM Review 10*(4), 422–437.

McDysan, D. E. and D. L. Spohn (1994). *ATM: theory and application.* New York: McGraw-Hill.

Melamed, B., J. R. Hill, and D. Goldsman (1992). The TES methodology: Modelling empirical stationary time series. In *Proceeding of the '92 Winter Simulation Conference*, pp. 135–144.

Merhav, N., M. Gutman, and J. Ziv (1989). On the estimation of the order of a Markov chain and universal data compression. *IEEE Transactions on Information Theory 35*(5), 1014–1019.

Neuts, M. F. (1981). *Matrix-geometric solutions in stochastic models.* Baltimore: Johns Hopkins University Press.

Neuts, M. F. (1989). *Structured stochastic matrices of M/G/1 type and their applications.* New York: Dekker.

Nomura, M., T. Fujii, and N. Ohta (1989). Basic characteristics of variable rate video coding in ATM environment. *IEEE Journal on Selected Areas in Communications 7*(5), 752–760.

Norros, I. (1994). A storage model with self-similar input. *Queueing Systems 16*, 387–396.

Pancha, P. and M. E. Zarki (1992). A look at the MPEG video coding standard for variable bit rate video transmission. In *Proceedings of the Infocom '92*, pp. 85–94.

Pancha, P. and M. E. Zarki (1993a). Bandwidth requirements of variable bit rate MPEG sources in ATM networks. In *Proceedings of the Conference on Modelling and Performance Evaluation of ATM Technology, Martinique*, pp. 5.2.1–25.

Pancha, P. and M. E. Zarki (1993b). Bandwidth requirements of variable bit rate MPEG sources in ATM networks. In *Proceedings of the Infocom '93*, pp. 902–909.

Pancha, P. and M. E. Zarki (1994). MPEG coding for variable bit rate video transmission. *IEEE Communications Magazine 32*(5), 54–66.

Ramamurthy, G. and B. Sengupta (1992). Modelling and analysis of a variable bit rate video multiplexer. In *Proceedings of the Infocom '92*, pp. 6C.1.1–11.

Ramanathan, S., P. V. Rangan, and H. M. Vin (1993). Frame-induced packet discarding: An efficient strategy for video networking. In *Proceedings of the Fourth International Workshop on Network and Operating Systems Support for Digital Audio and Video*.

Rathgeb, E. P. (1993). Policing of realistic VBR video traffic in an ATM network. *International Journal of Digital and Analog Communication Systems 6*, 213–226.

Reibman, A. R. and A. W. Berger (1995). Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking 3*, 329–339.

Reininger, D., B. Melamed, and D. Raychaudhuri (1994). Variable bit rate MPEG video: Characteristics, modeling and multiplexing. In *Proceedings of the ITC-14*, pp. 295–306.

Reininger, D., G. Ramamurthy, and D. Raychaudhuri (1995). VBR MPEG video coding with dynamic bandwidth renegotiation. In *Proceedings of the ICC '95.*

Ritter, M. and P. Tran-Gia (1995). Performance analysis of cell rate monitoring mechanisms in atm systems. In T. Hasegawa, G. Pujolle, H. Takagi, and Y. Takahashi (Eds.), *Local and Metropolitan Communication Systems. Volume 3*, pp. 129–150. London: Chapman & Hall.

Roberts, J. W. (1992). *Performance evaluation and design of multiservice networks.* Luxembourg: Commission of the European Communities.

Roberts, J. W., J. Guibert, and A. Simonian (1991). Network performance considerations in the design of a VBR codec. In *Proceedings of the ITC-13 Workshop on Queueing, Performance and Control in ATM*, pp. 77–82.

Rodriguez-Dagnino, R. M. and A. Leon-Garcia (1992). Broadband traffic characterization. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering.*

Rose, O. and M. R. Frater (1994). A comparison of models for VBR video traffic sources in B-ISDN. In *IFIP Transactions C-24: Broadband Communications, II*, pp. 275 – 287. Amsterdam: North-Holland.

Rose, O. and M. Ritter (1995). MPEG-video sources in ATM-systems – a new approach for the dimensioning of policing functions. In T. Hasegawa, G. Pujolle, H. Takagi, and Y. Takahashi (Eds.), *Local and Metropolitan Communication Systems. Volume 3*, pp. 108–126. London: Chapman & Hall.

Schlittgen, R. and B. H. J. Streitberg (1995). *Zeitreihenanalyse* (6th ed.). München: Oldenbourg. (In German).

Sen, P., B. Maglaris, N.-E. Rikli, and D. Anastassiou (1989). Models for packet switching of variable-bit-rate video sources. *IEEE Journal on Selected Areas in Communications 7*(5), 865–869.

Shroff, N. and M. Schwartz (1994). Video modeling within networks using deterministic smoothing at the source. In *Proceedings of the Infocom '94*, pp. 342–349.

Skelly, P., M. Schwarz, and S. Dixit (1993). A histogram-based model for video traffic behavior in an ATM multiplexer. *IEEE/ACM Transactions on Networking 1*(4), 446–459.

Stokes, O. L. (1995). *Transmission of MPEG Compressed Video Through B-ISDN ATM Networks.* Ph. D. thesis, North Carolina State University.

Taqqu, M. S. (1988). Self-similar processes. In *Encyclopedia of Statistical Sciences 8*, pp. 352–357. New York: Wiley.

Tran-Gia, P. and H. Ahmadi (1988). Analysis of a discrete-time $G^{[X]}/D/1-S$ queueing system with applications in packet-switching systems. In *Proceedings of the Infocom '88*, pp. 861–870.

Whittle, P. (1953). Estimation and information in stationary time series. *Arkiv för Matematik 2*(23), 423–434.

Willinger, W., M. S. Taqqu, W. E. Leland, and D. V. Wilson (1995). Self-similarity in high-speed packet traffic: Analysis and modeling of Ethernet traffic measurements. *Statistical Science 10*(1), 67–85.

Wright, D. (1993). *Broadband: Business Services, Technologies, and Strategic Impact.* Boston: Artech House.

Zhang, Y.-Q., W. W. Wu, K. S. Kim, R. L. Pickholtz, and J. Ramasastry (1991). Variable bit-rate video transmission in the broadband ISDN environment. *Proceedings of the IEEE 79*(2), 214–222.

Index

ISSN 1432-8801