BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPARTMENT OF TELECOMMUNICATIONS AND TELEMATICS

TRAFFIC CHARACTERIZATION, MODELING AND MANAGEMENT FOR MULTIMEDIA TELESERVICES

István Cselényi

PhD Dissertation

Supervised by

Dr. Tamás Henk and Dr. Sándor Molnár
High Speed Networks Laboratory
Department of Telecommunications and Telematics
Budapest University of Technology and Economics

Budapest, Hungary
2000

István Cselényi
High Speed Networks Laboratory
Department of Telecommunications and Telematics
Budapest University of Technology and Economics[1]

[1] The reviews and the minutes of the PhD defense are available from the Dean's Office.

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
TÁVKÖZLÉSI ÉS TELEMATIKAI TANSZÉK

Tudományos vezető
Dr. Henk Tamás és Dr. Molnár Sándor
Nagysebességű Hálózatok Laboratóriuma
Távközlési és Telematikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Budapest
2000

To Györgyi, Benci & Bella

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ATM	Asynchronous Transfer Mode
CBR	Constant Bit Rate
CIT	Cell Interarrival Time
CPDF	Complementary Probability Distribution Function
DiffServ	Differentiated Services
DST	Destination host
FIFO	First In First Out
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
LAN	Local Area Network
LB	Leaky Bucket
LBA	Leaky Bucket Analysis
MAN	Metropolian Area Network
MOO	Multilevel On-Off
MTU	Maximum Transfer Unit
PMF	Probability Mass Function
QoS	Quality of Service
RSVP	Resource Reservation Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VBR	Variable Bit Rate

Abstract

Multimedia services offered by packet switched networks call for new solutions in the areas of traffic characterization, traffic modeling and traffic management. Traffic characterization should realistically describe the multimedia traffic and capture its inherent burstiness on many time scales. Traffic modeling should capture the multiple burst levels of multimedia traffic. Traffic management should find effective and scalable schemes for resource reservation utilizing the fine granularity of quality of service requirements of multimedia teleservices.

According to these problem statements, the objective of my dissertation is threefold.

I introduce a framework for characterizing the resource demand and burstiness of multimedia traffic on several time scales, in a simple way. I propose to characterize the input traffic with the Leaky Bucket curve, which gives a deterministic bound on buffer and delay. This metric directly provides the relationship between the buffer and bandwidth requirements of a rate limited, finite length traffic trace. Furthermore, I propose the Leaky Bucket Slope curve for visualizing the burst structure of traffic trace. Finally, I establish an analytic model based on the Multilevel On-Off traces, which enables the quantification of burst size and inter-burst time in a measured traffic trace.

I propose a hierarchical source model, which is capable to reproduce the burst structure of a VBR traffic source by imitating the internal traffic generation process of the end system. First I consider the multimedia traffic source as a 'black-box' and analyze only its traffic. Than I evaluate the impact of internal mechanisms in the 'white-box' traffic source on the traffic characteristics and achieve a three level, hierarchical model.

I propose a performance evaluation framework for benchmarking resource reservation schemes. I distinguish per hop and per reservation performance metrics in this framework and define new metrics for characterizing the greediness and the effectiveness of reservation schemes. I propose different new reservation protocols, which optimize one of these metrics or utilize service specific information for better reservation performance.

Kivonat

A csomagkapcsolt hálózatok multimédia szolgáltatásai új megoldásokat követelnek a forgalomjellemzés, forgalommodellezés és forgalommenedzsment területén. A multimédia forgalom valósághű leírása és sajátos csomósodási tulajdonságának több időskálán való visszatükrözése jelenti a kihívást a forgalomjellemzés számára. A forgalommodellezés új feladata a multimédia forgalom össztett csomósodási szintjeinek leírása. A forgalommenedzsment területén pedig hatékony és skálázható erőforrásfoglaló eljárásokat kell találni, amelyek kihasználják a multimédia szolgáltatások változatos minőségi követelményeit.

Disszertációm három célt tűz ki e feladatoknak megfelelően.

Bemutatok egy keretrendszert, amely egyszerű módszert kínál a multimédia forgalom erőforrásigényének és csomósodásának több időskálás jellemzésére. A bejövő forgalmat az úgynevezett ,,Lyukas Vödör" görbével írom le, amely egy determinisztikus felső korlátot határoz meg a tárolóméret és késleltetés tekintetében. Ez a mérték közvetlenül megadja a kapcsolatot egy sávkorlátozott, véges hosszúságú forgalom-minta tárolóméret- és sávszélességigénye között. Továbbá a forgalom csomósodási szerkezetének szemléltetésére az úgynevezett ,,Lyukas Vödör Meredekség" görbét ajánlom. Ezt követően egy analitikus modellt állítok fel a ,,Többszintű On-Off" minták alapán, amely lehetővé teszi a csomóméret és a csomók közt eltelt idő számszerűsítését. Végezetül számos alkalmazással érzékeltetem módszerem használhatóságát.

Ajánlok egy hierarchikus forrásmodellt, amely képes visszaadni egy változó bitsebességű forgalom-forrás csomósodási szerkezetét a végberendezés belső forgalomgerjesztő folyamatainak utánzása segítségével. A multimédia forrást először ,,fekete doboznak" tekintem és csak annak forgalmát elemzem. Ezután, mintegy ,,fehér dobozként", kiértékelem a forrás belső folyamatainak hatását a forgalom jellemzőire és egy háromszintű, hierarchikus modellt állítok fel.

Ismertetek egy teljesítmény kiértékelő keretrendszert erőforrás foglaló eljárások vizsgálatára. Ebben a keretrendszerben ugrásonkénti és lefoglalásonkénti teljesítmény mértékeket különböztetek meg és új mértékeket állítok fel az erőforrás foglaló eljárások mohóságának és hatékonyságának jellemzésére. Továbbá több új erőforrás foglaló protokollt ajánlok, amelyek optimalizálják ezen mértékek valamelyikét illetve a szolgáltatásra vonatkozó információt használják fel a jobb hatékonyság elérése érdekében.

Acknowledgements

Budapest, November 2000

The supervisors of this Ph.D. Dissertation have been Dr. Sándor Molnár and Dr. Tamás Henk. First of all, I wish to thank them for their guidance in my research work and for encouraging me to complete my dissertation. I would like to thank Dr. Viktória Elek, Róbert Szabó and Attila Vidács for their help and advises regarding my dissertation. I wish to thank Mr. Nils Björkman and Mr. Alexander Latour-Henner, both from Telia, for the enthusiastic working milieu and inspirational discussions which triggered me to begin research in this area. Moreover, special thanks to all the colleagues from Telia, from the university and from Ericsson Traffic Lab, who co-authored in several publications. And last but not least I am very grateful to my wife and my little children for their patience and my parents for showing me a good example, which was worth to follow.

István Cselényi

Chapter 1

General Introduction

1.1 Background

A wide spectrum of traffic modeling and management techniques is available for the single service of the telephony system [9, 11]. However, those techniques are generally not applicable for the versatile multimedia services offered by packet switched networks, which provoked intensive research in the last decades.

Traffic characterization provides parameters for describing the traffic in the traffic contract on one hand, and it supports traffic engineering with estimating the required network resources on the other hand. Multimedia services impose tight requirements on the delay, delay variation, loss and throughput performance of the network. In order to meet these constrains, preventive traffic control is proposed based on a traffic contract [35, 66]. In this traffic contract the requested connection or flow is described by a *traffic descriptor*, which consists of a set of traffic parameters [76, 66, 36]. The network can use the traffic descriptor (i) for deciding on admission of a new connection/flow (admission control), (ii) for checking whether the characteristics of an accepted connection are as declared (usage parameter control) or (iii) for enforcing the contracted characteristics in the traffic source (shaping) [34] or at the edge of a network domain (traffic conditioning) [69]. The fundamental problem is to find adequate traffic parameters for describing the traffic, since these parameters should be easy to control and significant in determining required resources. Moreover, traffic characterization has to reflect the complex burst structure of multimedia traffic.

Another challenge is to *establish models for multimedia traffic sources.* Apart from approximating models, which aim to support the analysis of multiplexing performance [5, 38, 44], a large group of traffic models targets a specific traffic source [106, 107, 108, 109, 110]. Source models can provide a simplified description of the source behaviour for setting up traffic control processes, they can precisely imitate the source in simulations or replace expensive multimedia terminals in measurements. Regarding the model building technique, the most common modeling approach is *black-box modeling* [9, 10, 11] that focuses on the

traffic and not on the traffic source. This technique tries to capture certain traffic characteristics (e.g. moments of interarrival time) by tuning the parameters of an independent traffic generation process. One drawback of these models is that they usually have a huge number of input parameters. Another problem that it is difficult to modify the behaviour of the traffic source, especially in a realistic way. These limitations do not hold for *white-box modeling*, which is fairly rare in the literature [43]. This approach utilizes the a' priori knowledge of the traffic generation process and focuses on the emulation of internal processes in the traffic source, yielding a more accurate source model. Comprehensive overviews of traffic modeling can be found in [8, 9].

Traffic management aims for eliminating congestion in the network, balance the traffic load and protect against failures [76, 8]. Traffic management schemes can be classified into two classes, *reactive* or feedback control schemes [101, 102] and *proactive* or resource reservation algorithms [103, 19, 66, 15]. Reactive approaches detect and react dynamically to congestion inside the network by relying on the feedback information from the network, while proactive approaches eliminate the possibility of congestion by reserving network resources for each connection. There are numerous ways of reserving resources for multimedia services. The usage of *signaling protocols* for this purpose is widespread in ATM and IP networks (see [13, 14, 15, 16, 17, 18, T4]). Others propose to code the QoS demand in the packet header [19, 20, 21] or rely on a centralized resource manager [31, J3, C16, C18, C17, C15]. The critical issue is the scope and scalability of these alternatives, which require a performance framework for evaluation.

1.2 The Structure of the Dissertation

Chapter 2 presents the Leaky Bucket Analysis – a framework for characterizing the resource demand and burstiness of ATM traffic –, demonstrates its applicability on deterministic and measured traffic traces, and describes its application for traffic modeling, resource dimensioning, shaper design, queuing behaviour analysis and analysis of multiplexing gain.

Chapter 3 describes a hierarchical source model based on the white box modeling concept. This model can synthesize the traffic of a VBR traffic source by imitating the operation of VBR encoding, process scheduling and protocol encapsulation in a multimedia workstation. At the end of this chapter the validity of this model is investigated using the Leaky Bucket Analysis.

Chapter 4 presents a performance evaluation framework for resource reservation schemes including traditional and novel performance metrics and demonstrates the application of this framework on different reservation schemes, such as sender-oriented, receiver-oriented protocols and others using a central resource manager or service specific information.

I have primarily assumed ATM as networking technology in the first two thesis groups, while IP in the last one. However, most of the techniques I apply here can be adapted to other packet switching and multiplexing technologies.

Chapter 2

Leaky Bucket Analysis and its Applications

2.1 Introduction

This chapter introduces the Leaky Bucket Analysis (LBA) that is a traffic characterization framework providing deterministic traffic descriptors for resource dimensioning and burstiness analysis.

There are different methodologies for classifying traffic characterization approaches. Firstly, the *goal* of traffic characterization can be either to track the arrival process or to provide a bound on the traffic volume. Secondly, stochastic and deterministic *analysis methods* can be distinguished. The stochastic analysis focuses on the statistical characteristics of the traffic, while the deterministic approach is based on a deterministic algorithm or rule. Thirdly, the traffic characterization can be either static (time invariant) or dynamic (time variant). From the eight possible combinations of these categories, I give an overview on stochastic burstiness metrics and different deterministic bounds.

2.1.1 Stochastic Burstiness Metrics

Regarding the arrival process, one of the most remarkable property of multimedia traffic is the *burstiness*. Burstiness expresses the clustering phenomenon of arrivals, i.e. when arrivals tend to form clusters with relatively short inter-arrival times within the cluster separated by relatively long intervals. It has a strong relation to the correlation structure of the traffic.

A simple class of *stochastic burstiness metrics* takes only the first-order properties into account. These metrics are based on different characteristics of the marginal distribution of the inter-arrival time. The most frequently used metric of this class is the peak to mean ratio, while a popular, second-order burstiness metric is the squared coefficient of variation [11, 76]. More complex burstiness metrics use second-order properties of the traffic. The indices of dispersion [80, 81] and generalized peakedness [82] are the most well known metrics

of this class. The former include the correlation properties of the traffic, while the latter incorporates the reaction of a system to a given traffic via the complementary holding time distribution of the system [83]. Other authors [84, 85] follow the concept of self-similarity and propose the Hurst parameter, or fractal parameters as burstiness metrics. For a short overview on stochastic burstiness metrics and related traffic models refer to [2, 7].

The advantage of many stochastic burstiness metrics is that they capture the correlation of arrivals and also the time dependent variations of the arrival process. Although stochastic characterization approaches usually yield analytically tractable formulas, it is necessary to make special (sometimes unrealistic) assumptions for this purpose [79]. Moreover, the number and complexity of the parameters needed to describe the traffic does not suit to the simple parameters of actual traffic control algorithms [42].

2.1.2 Simple Deterministic Bounds

The bounding traffic characterization assumes that the entering traffic is unknown, but satisfies certain regulatory constraint (e.g. maximum queue length or loss rate). In other words this approach aims to bound the traffic rather than exactly characterize the arrival process. The exact traffic pattern for a connection is unknown, the only requirement is that the volume of the traffic should be bounded in certain ways. Although bounding characterization can either be deterministic or stochastic, I consider only the former case.

Non-probabilistic or *deterministic bounding characterization* defines a deterministic traffic constraint function. The traffic constraint function bounds the maximum number of bits that may be generated by the source over a given time interval. The least upper bound of this function is also referred to as the minimum envelope process [35] or empirical envelope [98, 88].

A simple deterministic bound can be specified using the Leaky Bucket (LB) algorithm. This simple bound is used both in the ATM traffic descriptor [66, 76] or in the 'flow specification' of IP/RSVP messages [15, 57]. Since the LB parameters are used as traffic descriptors, the LB algorithm can also be used for traffic regulation, i.e. for Usage Parameter Control [34, 96, 97] and traffic shaping [46, 93, 95]. These are independent applications of the same algorithm. I consider the LB only as a traffic descriptor in this work. Apart from the Leaky Bucket, Rathgeb analyzes other deterministic bounding metrics, such as 'jumping window', 'triggered jumping window', 'moving window' and 'exponentially weighted moving average' (EWMA) and compares their performance [47]. His results emphasize that LB and EWMA are better traffic descriptors, because the window based mechanisms are not flexible enough to cope with the short-term statistical fluctuations of the source traffic.

The main limitation of the Leaky Bucket descriptor is that it provides a bound on burstiness only for a single service rate. The complex burst structure of multimedia traffic and the multi-rate nature of modern packet switched networks calls for a traffic characterization method which can capture the burst structure on many time scales.

2.1.3 Extended Deterministic Bounds

The single working point defined by the LB parameters should be replaced by a more sophisticated traffic constraint function in order to extend the scope of the deterministic bound.

Cruz proposed an envelope as the burstiness constraint for the traffic stream entering the network [2]. The traffic is conforming to his (σ, ρ) model, if during any interval of length u, the number of bits in that interval is less than $\sigma + \rho u$. His original model is based on the fluid flow approximation. The parameters σ and ρ can be viewed as maximum burst size and the long term bounding rate of the source, respectively. Using this calculus (σ, ρ), he also obtained delay and buffering requirements for the network elements and derived the burstiness constraint satisfied by the traffic that exits the element. Extending the original work, Cruz introduced the concept of Service Curve in [3] based on a discrete time model. This result is a corner stone of the Network Calculus framework by Le Boudec [4], which provides a set of rules and results that can be used for computing tight bounds on delays, backlogs, and arrival envelopes in a lossless setting applicable to packet networks. An alternative deterministic traffic constraint function is proposed in [89], where a traffic stream satisfies the $(X_{min}, X_{ave}, I, S_{max})$ model, if the inter-arrival time between any two packets in the stream is more than X_{min}, the average packet interarrival time during any interval of length I is more than X_{ave} and the maximum packet size is less than S_{max}. Furthermore, a traffic stream satisfies the (r, T) model [90, 91] if no more than $r \cdot T$ bits are transmitted on any interval of length T. Rather than using one bounding rate, the Deterministic Bounding Interval-Dependent (D-BIND) model [92] uses a family of rate-interval pairs where the rate is a bounding rate over the corresponding interval length. For an overview see [8].

The main benefits of deterministic bounding characterizations are that they are both general and practical. They can characterize a wide variety of bursty sources. In addition, it is sufficient for resource management algorithms to allocate resources by knowing just the bounds on the traffic volume.

2.1.4 Dynamic Deterministic Bounds

I have considered deterministic bounds with *time-invariant traffic constraint* function previously. Although this approach provides a very accurate traffic characterization for a source, its practical significance is decreased by the fact that such function can only be effectively policed by a large number of leaky buckets. Since current packet-switched networks employ simple LB mechanisms for traffic policing, the use of the minimum envelope process does not facilitate traffic policing.

Extending the static (ρ, σ) model, Cruz introduces a dynamic traffic constraint function in [1]. He uses dynamic burstiness measures for describing the relationship between the traffic entering a server and the traffic exiting the server for characterizing end-to-end network delay. Moreover, he presents new classes of service disciplines, which support delay guarantees as well as efficient statistical sharing. The dynamic envelope metric of

Cruz overlaps with the work of Parekh and Gallager, who consider *generalized processor sharing* service disciplines [86, 87]. The delay calculus of Cruz is extended based on the time varying filtering theory under the $(min, +)$ algebra in [99] and an optimal and implementable solution is provided for the general constrained traffic regulation problem with both delay and buffer constrains. Finally, Konsta proposes a dynamic envelope, that enables traffic regulation that also considers delay and buffer constrains and minimizes the number of discarded packets [100].

Following the deterministic bounding approach, I have established a traffic characterization framework, called *Leaky Bucket Analysis*, that provides a deterministic bound on a traffic trace on different time scales in a discrete time framework. It directly provides the relationship between the buffering requirement of a trace and the service rate. Moreover, it visualizes the burst structure of the trace and – with the help of a special deterministic traffic model, the so called Multilevel On-Off trace – quantifies the burst size and the burst interarrival times in the trace.

The rest of this chapter is organized as follows. Section 2.2.1 presents the Leaky Bucket Analysis (LBA) framework by defining the deterministic traffic descriptors and related metrics. Section 2.2.2 illustrates this traffic characterization method on artificial, deterministic traffic traces, while Section 2.2.3 demonstrates the applicability of LBA on real ATM traffic traces captured by measurements. The robustness of LBA as a tool for characterizing a specific traffic type is investigated in Section 2.3. The deterministic metrics of LBA are fitted to two analytic models. The first model produces a recursive traffic pattern, called Multilevel On-Off Trace, while the second is a two-level fluid flow model. These models are described in Section 2.4.1 and Section 2.4.2, respectively. Finally, several applications of LBA are presented in Section 2.5 such as resource dimensioning, shaper design, queuing behaviour analysis and analysis of multiplexing gain.

2.2 The Leaky Bucket Analysis

In this section, I describe and illustrate a deterministic traffic analysis method for characterizing bursty traffic sources, called the Leaky Bucket Analysis (LBA) [B1, C1, C3].

Let us consider discrete cell arrivals and departures in a discrete-time queuing model with FIFO service discipline and no priority scheduling according to the following definitions.

Definitions

Arrival Function Let the series $a_1, a_2, \ldots, a_N$ $(N \in \mathbb{Z}^+)$ denote the absolute arrival time of cells in a traffic trace of length T. The arrival function of this trace can be defined as:

$$\mu(k) = \sum_{i=1}^{N} \delta(k - a_i), \qquad k = 0, 1, 2, \ldots, T, \tag{2.1}$$

where k denotes the discrete time and $\delta(k)$ is the Kronecker delta function which equals to one, if $k = 0$, and otherwise equals to zero. As a short form, I will refer to a traffic trace of length T, containing N arrivals as $\mu(T, N)$.

Cumulative Arrival Function In order to express the amount of cells, which have arrived in the interval $[1, k]$ the cumulative arrival function can be used:

$$m(k) = \sum_{i=1}^{N} \epsilon(k - a_i) = \sum_{j=0}^{k} \mu(j), \qquad k = 0, 1, \ldots, T, \tag{2.2}$$

where $\epsilon(k)$ is the 'step function' which equals to one, if $k \geq 0$, and otherwise equals to zero.

Service Function The service function can be defined similarly to Equation (2.1), using the series of absolute service times t in the queue:

$$\eta(k, t) = \sum_{j=1}^{\lfloor \frac{T}{t} \rfloor} \delta(k - jt), \qquad k = 0, 1, \ldots, T, \quad t \in \{1, 2, 3, \ldots, T\}. \tag{2.3}$$

The selected set of service time corresponds to serving one incoming cell in every $1, 2, 3, \ldots$ time slot[1]. Because of this choice, the service rate (i.e. the reciprocal of service time) is $r \in \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{T}\}$ in this framework.

2.2.1 The Leaky Bucket Curve

The basic metric of the Leaky Bucket Analysis is the *Leaky Bucket Curve*[2] that specifies the upper bound for the queue length ξ when a finite trace of length T is served at rate r in a $G/D/1$ queuing system:

$$\begin{aligned} \xi(0, r) &= 0, \\ \xi(k+1, r) &= \max\left\{\xi(k, r) + \mu(k) - \eta\left(k, \frac{1}{r}\right), 0\right\}, \qquad k = 0, 1, \ldots, T, \end{aligned} \tag{2.4}$$

$$q(r) = \max_{k \in \{0,1,\ldots,T\}} \{\xi(k, r)\}, \qquad r \in \left\{1, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \ldots, \tfrac{1}{T}\right\}, \tag{2.5}$$

where $q(r)$ denotes the discrete Leaky Bucket (LB) Curve.

The dual pair of the LB curve, i.e. the maximum queue length as a function of service time t can be given as:

$$q^t(t) = q\left(\tfrac{1}{t}\right), \qquad t \in \{1, 2, 3, \ldots, T\}. \tag{2.6}$$

[1] This kind of deterministic service process is widely used in actual ATM switches.

[2] Burstiness Curve or Backlog Curve might be a more intuitive name for this curve. I have chosen the notion, Leaky Bucket Curve, since the most practical application of this curve is to determine proper parameters for the Leaky Bucket based ATM usage parameter control process.

It is very impractical to handle a series of value pairs (i.e. $\{r_i, q(r_i)\}$), especially in case of drawing graphs. Thus I also define the *Leaky Bucket Envelope* curve that is a continuous function of $\hat{r}$, which intersects the points of the LB curve:

$$\hat{q}(\hat{r}) = q(r_B) - (r_B - \hat{r})\frac{q(r_A) - q(r_B)}{r_A - r_B}, \qquad \begin{array}{l} \frac{1}{T} \le \hat{r} = \frac{1}{\hat{t}} \le 1, \\ r_A = \frac{1}{\lfloor \hat{t} \rfloor + 1},\ r_B = \frac{1}{\lfloor \hat{t} \rfloor}. \end{array} \tag{2.7}$$

For the sake of simplicity, I will omit the notation of continuity "$\hat{}$" in the figure labels and in case of the service rate and service time.

Properties of The LB Curve

By analyzing the expressions (2.1), (2.3), (2.4) and (2.5), we can recognize the following properties of the LB envelope curve:

- non negative,
- monotone decreasing,
- and convex.

Several traffic patterns may have the same LB curve, because the max() function is not unambiguous. Moreover, the following equations hold for the edges of the LB curve:

$$\lim_{r \to 0} q(r) = N, \qquad \lim_{r \to 1} q(r) = 0. \tag{2.8}$$

The $\hat{q}(r)$ curve is non negative due to the $[\cdot]^+$ function. In order to prove that it is also monotone decreasing, let us analyze Equation (2.4). Denote $\mathcal{K}_n^+;\ n \in \{1, 2, \dots, k\},\ \bigcup_n \mathcal{K}_n^+ \subset \{1, 2, \dots, k\}$ the disjunct time intervals before k where the argument of the $[\cdot]^+$ function is non-negative, i.e. where $\xi(k) + \mu(k) - \eta\left(k, \frac{1}{r}\right) \ge 0$. First I consider one such a region and prove the monotonity. Since the queue length function $\xi(k)$ is zero in any other time slot $k \notin \mathcal{K}_n^+$, we can omit those terms and rewrite Equation (2.4):

$$\xi(k) = \begin{cases} 0, & \forall k:\ k \notin \mathcal{K}_n^+, \\ \sum\limits_{j \in \mathcal{K}_n^+} \left(\mu(j) - \eta\left(j, \frac{1}{r}\right)\right) = \sum\limits_{j \in \mathcal{K}^+} \mu(j) - \sum\limits_{j \in \mathcal{K}_n^+} \eta\left(j, \frac{1}{r}\right), & \forall k:\ k \in \mathcal{K}_n^+, \end{cases}$$

The last expression is monotone increasing with the service rate, because there are more service events in case of higher service rates even if the $\mathcal{K}_n^+$ regions may be shifted.

Thus the point series of $q(r)$, i.e. the maximum of the whole expression is monotone decreasing:

$$\eta(k, \tfrac{1}{r_1}) \ge \eta(k, \tfrac{1}{r_2}) \qquad \textit{if} \qquad r_1 \le r_2 \quad \Rightarrow \quad q(r_1) > q(r_2). \tag{2.9}$$

Illustration

The left side of Figure 2.1 illustrates the *Leaky Bucket* curve for two traces captured from video traffic. The continuous lines depict the LB envelope $\hat{q}(r)$, while the dots highlight the discrete $q(r)$ function. Three operating points are marked by diamonds. The two LB curves practically coincide in operating point A, highlighting that cell loss can be avoided by allocating a buffer of 300 cells and service rate of 4700 kbps for both of the sources. However, the two sources have very different buffer demand, if they are served at 2900 kbps (see operating points B and B', respectively).

I introduce the following notions for referring to the two end regions of the LB curve:

1. the *peak-end* is on the right where the service rate is close to the peak rate of the trace and the curve is almost horizontal,

2. the *mean-end* is on the left where the service rate is close to the mean rate of the trace and the curve is almost vertical.

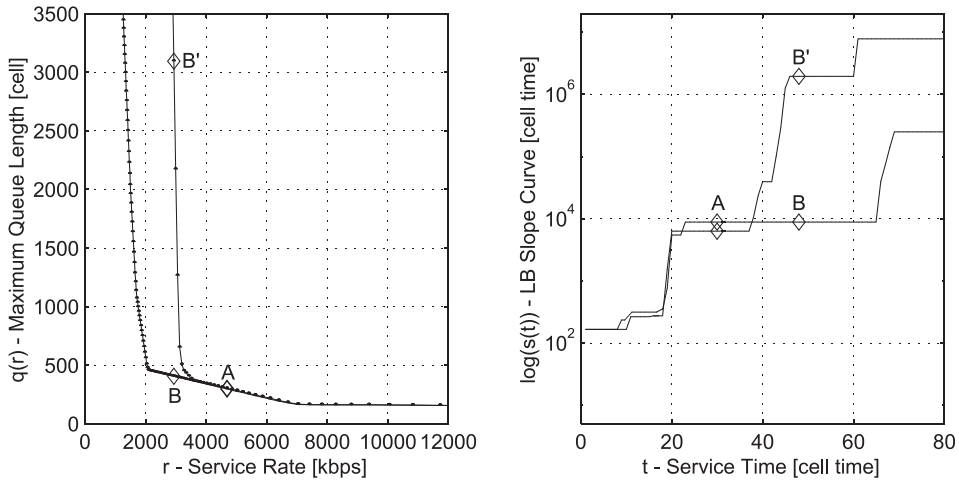

Figure 2.1: Leaky Bucket Envelope and Leaky Bucket Slope curves of two measured video traffic traces

It is very well pronounced in the figure, that the LB envelope curves consist of linear segments, with breaking points between them. For instance, the points between 7000 kbps $\leq r \leq$ 12000 kbps fit very well to a linear segment and there is a breaking point around $r = 7000$ kbps. In order to analyze this feature, let us express the slope of the LB Envelope

curve in each $r_i \in \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \ldots, \frac{1}{T}\}$ point, using (2.7):

$$\varrho(r_i) = \frac{q(r_i) - q(r_{i+1})}{r_i - r_{i+1}}, \qquad r_1 = 1, \quad r_{i+1} = \frac{1}{\frac{1}{r_i} + 1}, \qquad i = 1, 2, \ldots, T. \tag{2.10}$$

Since the LB curve is monotone decreasing, its slope is negative. Moreover, the computation of a metric is faster and easier, if it has a discrete argument. Thus I define the *Leaky Bucket Slope Curve* based on Equation (2.10) using t in the argument:

$$s(t_i) = \frac{\left| q\left(\frac{1}{t_i}\right) - q\left(\frac{1}{t_{i+1}}\right) \right|}{\frac{t_{i+1} - t_i}{t_i t_{i+1}}}, \qquad t_1 = 1, \quad t_{i+1} = t_i + 1, \qquad i = 1, 2, \ldots, T. \tag{2.11}$$

Using $q^t(t) = q(\frac{1}{t})$ and that $t_{i+1} = t_i + 1$ we get

$$s(t) = t(t+1) \left| q^t(t) - q^t(t+1) \right|, \qquad t = 1, 2, \ldots, T-1. \tag{2.12}$$

An important property of the LB slope curve is:

$$\lim_{t \to \infty} s(t) = T. \tag{2.13}$$

Since the slope of the LB curve usually changes with several orders of magnitude from the peak-end to the mean-end, I plot the LB slope on logarithmic scale. The right side of Figure 2.1 depicts the LB Slope curves of the previous two traces. The A, B and B' operating points are also noted on the LB slope curves. The slope curves emphasize the operating regions where the two sources have similar behaviour in the queue and also those where they differ. Moreover, the size and interarrival time of bursts in the trace can be directly read from the $\log(s(t))$ curve, as it will be described in Section 2.4.1.

Although plotting $q(r)$ and $s(t)$ with different X-axis may be confusing for the first glance, I have proposed that, because in this way the operating regions are represented by linear sections on the LB curve, and horizontal plateau on the LB slope curve.

The Leaky Bucket Analysis

The *Leaky Bucket Curve* and *Leaky Bucket Slope Curve* can be used for characterizing the rate and buffer demand and the burst structure of rate bounded, finite length traffic traces, $\mu(T, N)$. *Leaky Bucket Analysis* denotes traffic characterization based on these curves and on the presented framework.

In general, it is fairly complex to analyze the expression of the LB curve, i.e. Equations (2.4) and (2.5) mathematically. However, it is very straight forward to build a discrete event simulator, which can calculate the maximum queue length for different service rates based on these formulas. The equation of LB Envelope (2.7) is also complex, but any mathematical program can easily draw the $\hat{q}(r)$ graph using the point series of $q(r)$.

Related Metrics

There are further interesting metric definitions in connection with the Leaky Bucket curve. The *LB Distance* quantifies the distance of the LB curves of traces $A1$ and $A2$:

$$d(r) = \hat{q}_{A1}(r) - \hat{q}_{A2}(r), \qquad \frac{1}{T} \leq r \leq 1. \tag{2.14}$$

This factor can be used for analyzing how much more buffer is needed for one trace than for another or for validating traffic models (see section 3.4).

Maximum Delay is another practical metric, which can be easily retrieved from the LB curve by drawing trajectories to each working point (i.e. the line intersecting the origo and a certain point of the LB curve). The slope of the trajectory gives the maximum delay:

$$\delta(r) = \frac{\hat{q}(r)}{r}, \qquad \frac{1}{T} \leq r \leq 1. \tag{2.15}$$

Maximum Delay Ratio expresses the proportion of maximum delay of two traces:

$$\phi(r) = \frac{\hat{q}_{A1}(r)}{\hat{q}_{A2}(r)}, \qquad \frac{1}{T} \leq r \leq 1. \tag{2.16}$$

This metric can be used in advanced scheduling algorithms that are aiming for a fix proportion of delay among different traffic classes [86].

2.2.2 Leaky Bucket Analysis of Deterministic Traffic Traces

The aim of this section is to illustrate how the LB curve can characterize deterministic traffic traces.

I have synthesized several deterministic traffic traces, all with $T = 500500$ and $N = 1000$, and fed them into a G/D/1 queue. Than I have calculated and plotted the $q(r)$, $q^t(t)$ and $s(t)$ curves for the domain $1 \leq t \leq 1000$ therefore $0.001 \leq r \leq 1$.

Constant Bit Rate Trace

The arrival rate is constant in the Constant Bit Rate (CBR) trace. The cumulative arrival function $m(k)$ increases linearly, due to the regular arrivals $a_{i+1} = a_i + \alpha$, where $\alpha = \frac{T}{N}$ is constant[3] (see Figure 2.3). Figure 2.2 depicts the LB Envelope Curve, its dual pair and the LB Slope curve for the CBR trace. It can be seen in this figure that $q(0) = 1000$ and $s(t) \to T$.

[3]Since $\frac{T}{N} = 500.5$ is not integer, I repeated a pair of $\{\alpha_1 = 500, \alpha_2 = 501\}$ in this CBR sequence. That is the reason of having two peaks of equal size in the PMF(α) plot.

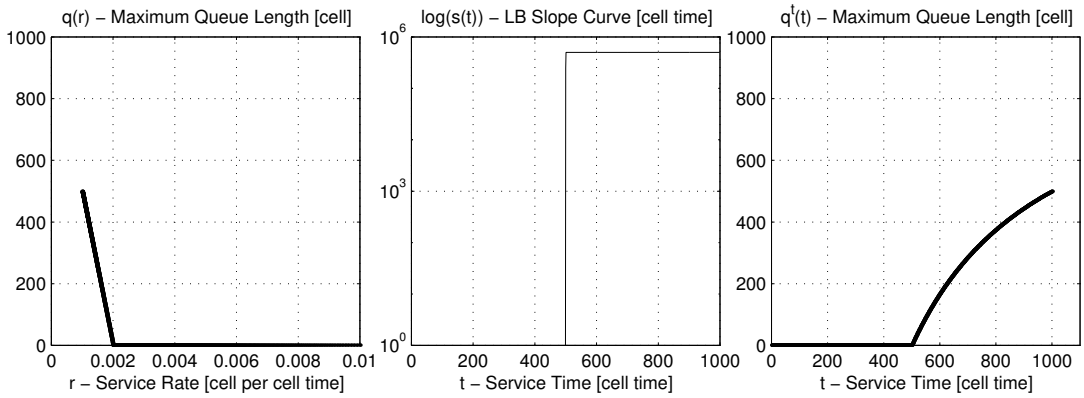

Figure 2.2: $q(r)$, $s(t)$ and $q^t(t)$ for a CBR Trace

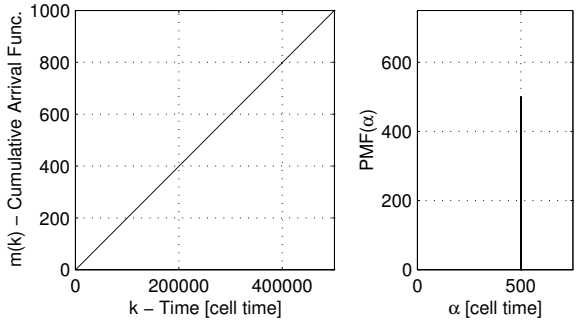

Figure 2.3: Cumulative arrival function and interarrival times of the CBR Trace

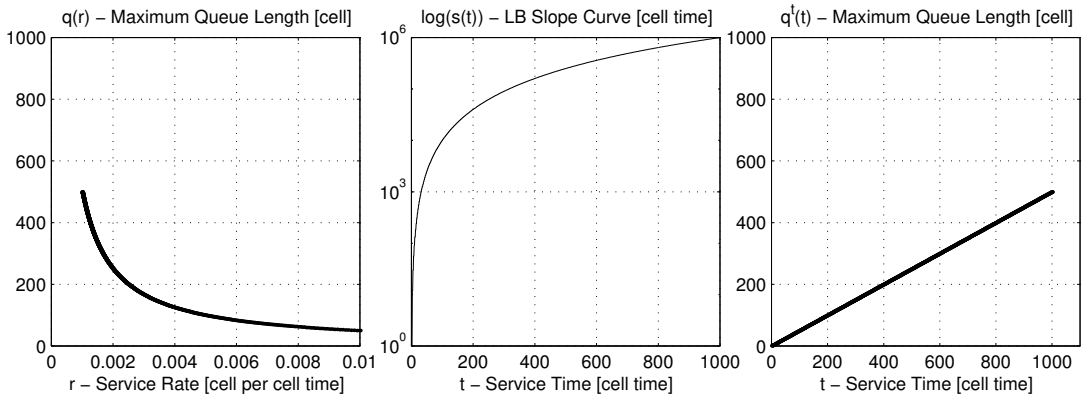

Figure 2.4: $q(r)$, $s(t)$ and $q^t(t)$ for the Accelerating Trace @ $\alpha_1 = 1,\ c = 1$

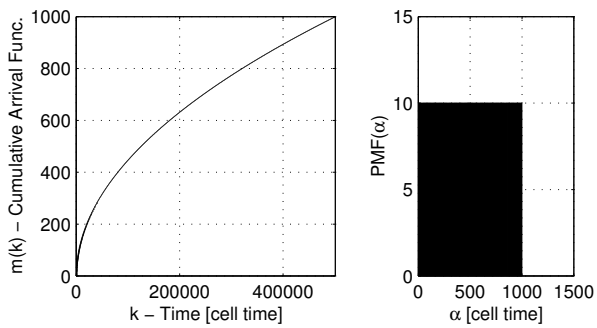

Figure 2.5: Cumulative arrival function and interarrival times of the Accelerating trace

Accelerating Traces

After the CBR trace, it is interesting to investigate other special traces, for which the histogram of interarrival time has several peaks of the same size (see Figure 2.5). Thus I constructed the *Accelerating Trace* family in which the interarrival time increases linearly:

$$\alpha_i = \alpha_1 + \lfloor ci \rfloor, \qquad i = 1, 2, 3, \ldots N, \tag{2.17}$$

where the constant c determines the decreasing rate of the slope of cumulative arrival function (i.e. the arrival rate). According to Figure 2.4, the $q^t(t)$ curve of the accelerating trace is linear, and the $q(r)$ curve is hyperbolic. Figure 2.12 depicts LB curves of several accelerating traces with different c constant (c=1, 2 and 5 for ACC1, ACC2 and ACC5, respectively). It is pronounced that as larger the c coefficient of an Accelerating trace is, as smaller is the slope of its $q^t(t)$ curve. One can also see that $\lim_{t\to\infty} s(t) = T$.

Simple On-Off Traces

A popular traffic pattern in teletraffic theory is the On-Off Trace [74, 75]. The illustrated On-Off trace contains 999 'back-to-back' cells with an interarrival time of $\alpha_{burst} = 1$ and one cell, which follows the burst $\alpha_{silence} = 499501$ cell times later. Both the $q(r)$ and $q^t(t)$ curves emphasize that this trace requires the most buffer space, i.e. this is the most bursty (see Figure 2.6). Noteworthy is that while the Accelerating Trace produced a linear $q^t(t)$ curve, the On-Off trace yields a linear $q(r)$ curve. Therefore these traces can be considered as "dual pairs" from the LB curve's point of view. Figure 2.7 demonstrates that the cumulative arrival function reaches its maximum suddenly.

Multilevel On-Off Traces

Based on the simple On-Off trace, I define another special traffic pattern. An *L-Level Multilevel On-Off Trace L-level MOO* is a deterministic trace, which consists of L levels of embedded bursts (see Figure 2.8). The burst on the first level consists of one cell and a silence period, which may be zero, if the peak rate equals to the line rate. The parameters of a MOO trace are N_i denoting the number of cells in the burst on the burst level i and T_i expressing the interarrival time of bursts on the burst level i ($i = 1, 2, \ldots, L$).

The basic equations for the parameters of MOO can be read from Figure 2.8:

$$T_1 = T_{B1} + T_{S1} = n_1 T_0 + T_{S1}, \tag{2.18}$$

where $n_1 = 1$ and $T_0 = 1$ by definition. Based on (2.18) the time factors can be retrieved for the other burst levels successively:

$$T_2 = n_2 T_1 + T_{S2} = n_2(n_1 T_0 + T_{S1}) + T_{S2} = n_2 n_1 T_0 + n_2 T_{S1} + T_{S2},$$

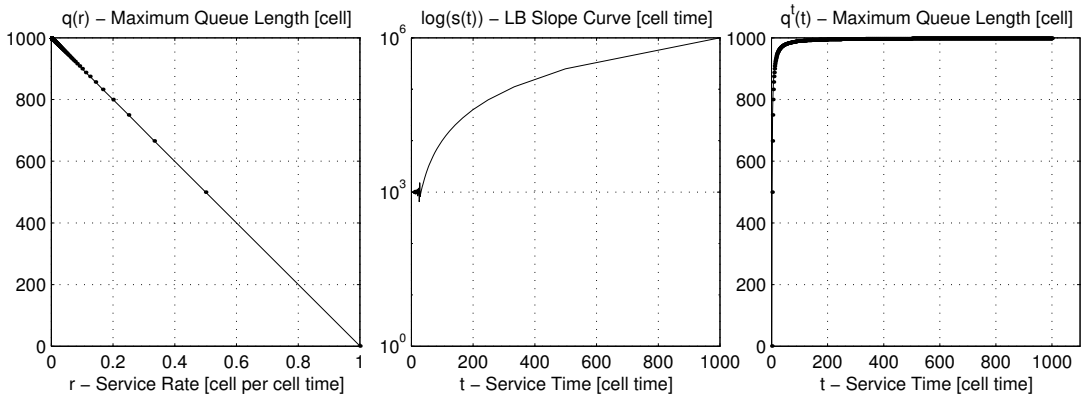

Figure 2.6: $q(r)$, $s(t)$ and $q^t(t)$ for an On-Off trace

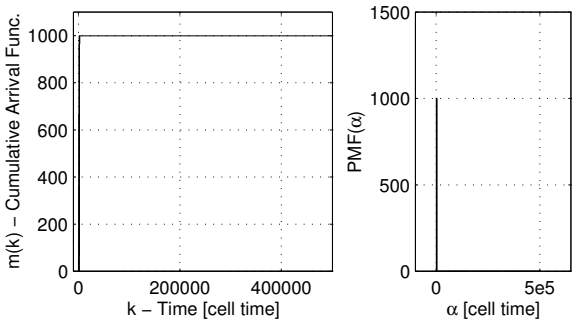

Figure 2.7: Arrival function and interarrival times of an On-Off trace

$$T_3 = n_3 T_2 + T_{S3} = n_3(n_2 n_1 T_0 + n_2 T_{S1} + T_{S2}) + T_{S3} = n_3 n_2 n_1 T_0 + n_3 n_2 T_{S1} + n_3 T_{S2} + T_{S3},$$

and so on for each burst level. These equations can be generalized:

$$\begin{aligned} T_i &= T_{Si} + T_{B1} N_i + \sum_{j=2}^{i} T_{Sj-1} \prod_{k=j}^{i} n_k = \\ &= T_{B1} N_i + \sum_{j=1}^{i} T_{Sj} \frac{N_i}{N_j}, \qquad\qquad i = 1, 2, \ldots, L. \end{aligned}$$

In order to simplify the formalism, I introduce $N_0 \doteq 1$ and $T_{s0} \doteq 1$:

$$T_i = \sum_{j=0}^{i} T_{Sj} \frac{N_i}{N_j}. \tag{2.19}$$

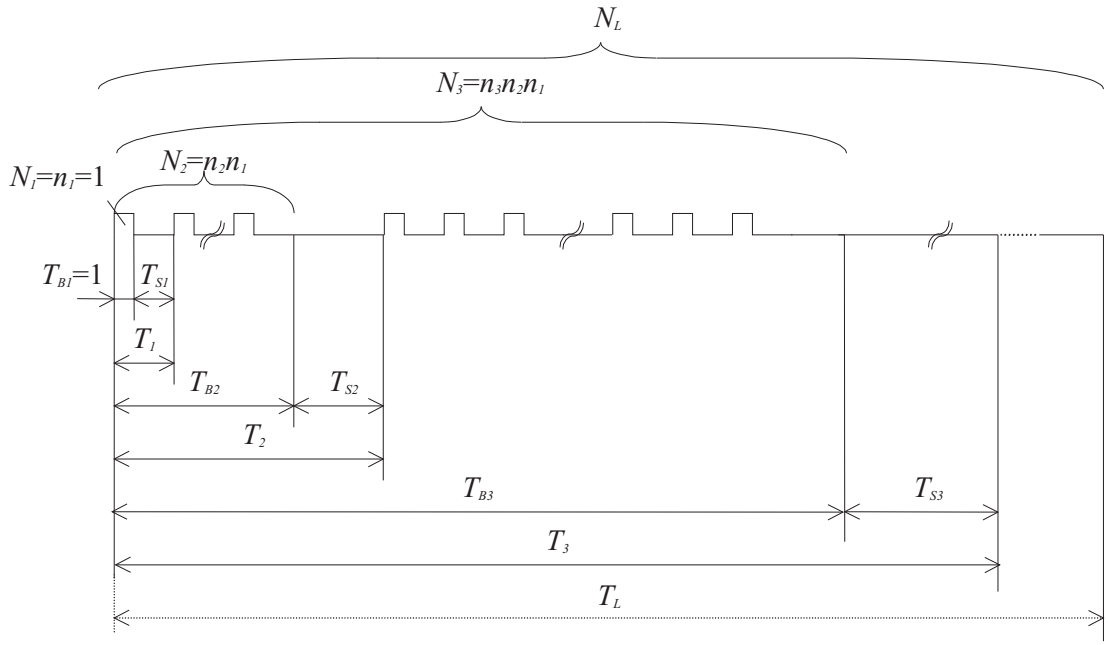

Figure 2.8: L-Level Multilevel On-Off Trace

The next step is to derive the equations of the LB Envelope curve for an L-level MOO. One can realize by looking at Figure 2.8 that the slope of the cumulative arrival function (i.e. the arrival rate) decreases step-wise from burst level to burst level. The average arrival rate is higher during the T_{B2} interval than during the T_{B3} interval. This also means that the rate of buffer build-up (i.e. the LB Slope) is constant, if the service rate is smaller than the average rate in the respective burst. Denote $\langle r_i \rangle$ the average rate in the first, second, etc. burst:

$$\langle r_1 \rangle = \frac{N_1}{T_1}, \qquad \langle r_2 \rangle = \frac{N_2}{T_2}, \qquad \dots \qquad \langle r_L \rangle = \frac{N_L}{T_L}. \tag{2.20}$$

The maximum queue length is a linear function of the service rate between two consecutive average burst rates. Therefore the LB curve $q(r)$ of a MOO trace is a piece-wise linear function with breaking points at each rate $r_i = \langle r_i \rangle$.

A typical LB Curve of an MOO trace is presented in Figure 2.9, indicating that the LB Curve of an L-level MOO consists of L linear sections with breaking points at $r_i = r_1, r_2, \dots, r_L$ and slopes $s_1, s_2, \dots, s_L$, where $s_i = \{s(t,i) : \frac{T_i}{N_i} \leq t \leq \frac{T_{i+1}}{N_{i+1}}\}$. Different sections represent different time-scales and different burst level of the MOO trace. Each burst level can be identified as a horizontal plateau on the LB Slope curve.

Denote $r_0 \doteq 1$ the full link rate. Let us determine the queue length in the breaking

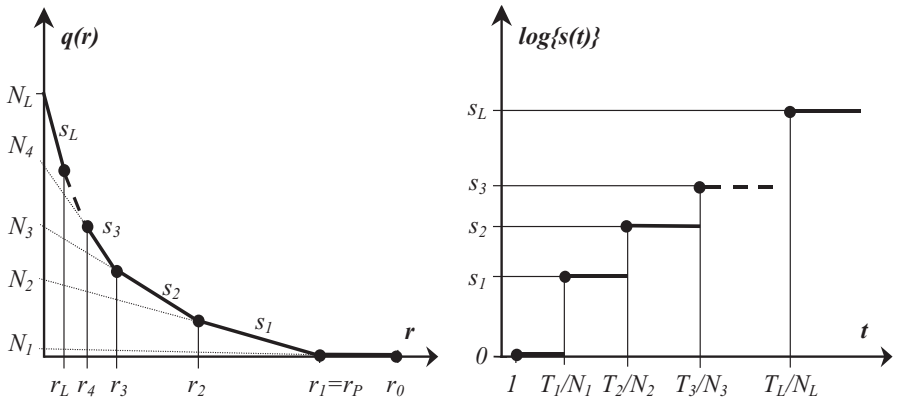

Figure 2.9: Leaky Bucket Envelope and LB Slope Curves of a Multilevel On-Off trace

points using (2.5) and Figure 2.8:

$$q(r_0) = 0, \qquad q(r_1) = N_1, \qquad q(r_2) = N_2 - \frac{r_2}{r_1}(N_2 - N_1), \qquad \ldots \tag{2.21}$$

The piece-wise linear LB curve can be deducted by describing each linear section, starting from the peak-end of the curve (i.e. $r = r_0$). The equation of the i^{th} linear LB envelope section is:

$$\hat{q}(r, i) = N_{i+1} - r \sum_{x=1}^{i} \frac{N_{x+1} - N_x}{r_x}, \qquad r_{i+1} < r \leq r_i, \quad i = 0, 1, 2, \ldots, L. \tag{2.22}$$

The equation of the LB slope curve is:

$$s^r(r, i) = \sum_{x=1}^{i} \frac{N_{x+1} - N_x}{r_x}, \qquad r_{i+1} < r \leq r_i, \quad i = 0, 1, 2, \ldots, L, \tag{2.23}$$

where r_L is the long term average rate of the trace $\mu(T_L, N_L)$. Similarly, the dual pair of $\hat{q}(r)$ and $s^r(r)$ as a function of service time t are:

$$\hat{q}^t(t, i) = N_{i+1} - \frac{1}{t} \sum_{x=1}^{i} T_x \left(\frac{N_{x+1}}{N_x} - 1 \right), \qquad \frac{T_i}{N_i} \leq t < \frac{T_{i+1}}{N_{i+1}}, \quad i = 0, 1, 2, \ldots, L, \tag{2.24}$$

$$s(t, i) = \sum_{x=1}^{i} T_x \left(\frac{N_{x+1}}{N_x} - 1 \right), \qquad \frac{T_i}{N_i} \leq t < \frac{T_{i+1}}{N_{i+1}}, \quad i = 0, 1, 2, \ldots, L. \tag{2.25}$$

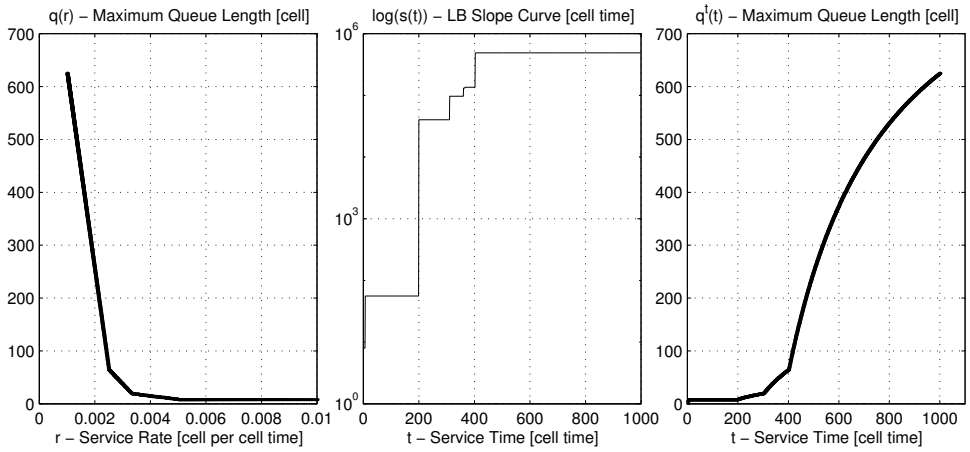

Figure 2.10: $q(r)$, $s(t)$ and $q^t(t)$ for a 5-level MOO trace

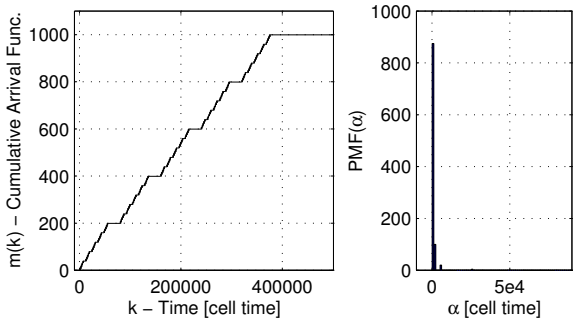

Figure 2.11: Cumulative arrival function and interarrival times of a 5-level MOO trace

I have performed the Leaky Bucket Analysis for a 5-level MOO trace. The results in Figure 2.10 validate the previous statements. Four linear (hyperbolic) sections can be seen on the $q(r)$ ($q^t(t)$) curve, while the LB slope curve highlights the four breaking points that separates the five burst levels.

Comparison

Let us compare the $q^t(t)$ curves of different deterministic traffic traces as a summary of this section. It can be seen in Figure 2.12 that the one level On-Off trace requires the most resources, i.e. it is the worst case, while the CBR trace is the best case. These traces represent the two extremes, which give an upper and lower bound for the LB curve of any other $\mu(T, N)$ trace. The CBR trace can be considered as a special Accelerating trace with $c = 0$ on one hand, or as a special one-level MOO trace with $T_1 = \alpha$ on the other hand.

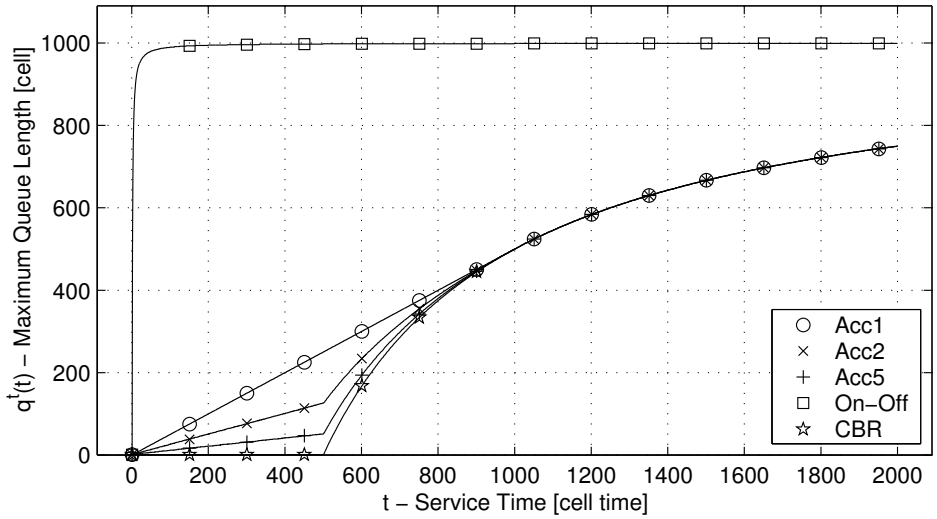

Figure 2.12: Comparison of deterministic traces

Moreover, the Accelerating trace can be considered as a special MOO trace with N burst levels, each consisting of one burst ($n_i = 1; \forall i = 1, 2, \ldots N$) and with linearly increasing burst interarrival time ($T_i = \lfloor ci \rfloor; \forall i = 1, 2, \ldots N$). I have started to extend the Leaky Bucket Analysis framework based on this observation and construct the linear space of MOO traces. My intuition is that this linear space always has such an element of which LB curve can approximate the LB curve of any arbitrary $\mu(T, N)$ trace with an accuracy of ε. Section A in the Appendix introduces this idea.

2.2.3 Leaky Bucket Analysis of Measured Traffic Traces

I have performed the Leaky Bucket Analysis on traffic traces captured in measurements. In particular, I have characterized the burst structure of more than 200 finite length, real traffic traces using the LB and LB Slope curves in case of 24 different configurations. This section presents the results for some of those traces. The measurement experiments and the main statistics of the captured traffic traces are summarized in Appendix C.

Multiplexed VBR Video Sources

Traffic of multimedia workstations connected via the Stockholm Gigabit Network (an ATM MAN) was multiplexed with CBR background traffic [C5]. Long traces of both traffic types were captured before and after multiplexing. LB curves of a shaped single multimedia source (a) and the aggregate traffic of four workstations (b) are depicted in Figure 2.13.

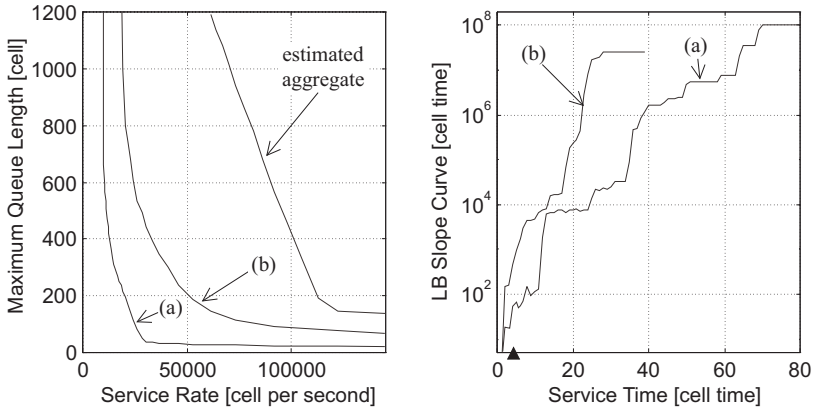

Figure 2.13: Multiplexing VBR video sources: (a) single VBR source before multiplexing (b) aggregate of four sources after multiplexing

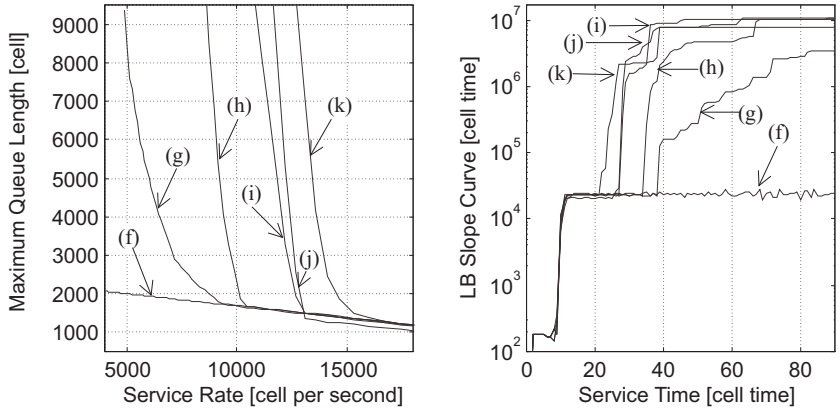

Figure 2.14: Leaky Bucket Analysis of several video sequences: (f) Girl with Toys, (g) Susie, (h) Table Tennis, (i) Tempest, (j) Flower Garden, (k) Popple

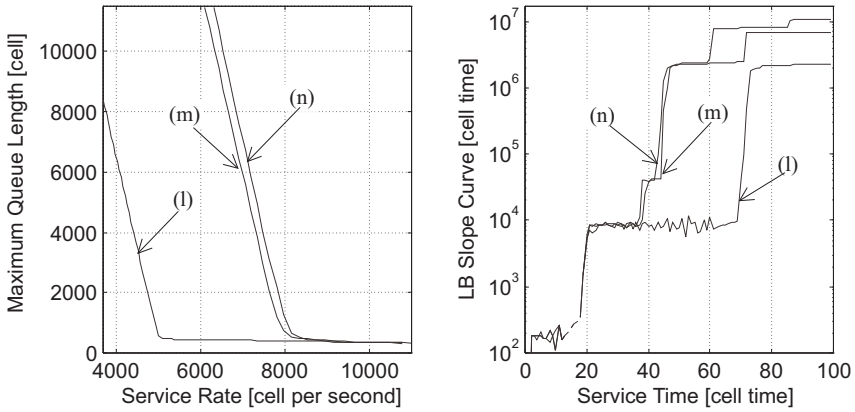

Figure 2.15: Effect of changing frame rate (l) @10 fps, (m) @20 fps, (n) @25 fps

The four sources were shaped to 34 Mbps (80354 cps) each, in order to avoid congestion in the first switch. Naturally, this peak rate limitation does not hold for the aggregate traffic (b) as it is pronounced in the LB slope graph where the first plateau is shifted to the left. The applied shaping rate can be identified by the place of the second jump in the LB slope curve of trace (a) (see marker). There were several ATM switches and SDH add/drop multiplexers on the way of the traffic from the source to the measuring point. The LB slope of trace (b) has more and shorter plateau than trace (a), which means that these active ATM and SDH devices destroyed the burst levels of the original VBR traffic by splitting and merging the bursts on different levels.

VBR Video Traffic

The Leaky Bucket Analysis is also applicable for tracking the effect of modifying the parameters of the multimedia traffic source. The impact of the input video sequence and the video frame rate on the LB curve is investigated in this section. Further analysis have been published in [B1, C3, C2, C1].

Figure 2.14 depicts the LB envelope curves of six VBR video traces. These traces have very different mean cell rate (see Table C.3 in Appendix A) and their LB curves are far from each other for service rates less than 16000 cps (6.7 Mbps). For instance, trace (f), which is an almost "still picture video", is less bursty than the other traces. On the other hand, the LB envelope curves converge for higher service rates, as it is also highlighted by the first two plateau of the LB slope curves that coincide for service time $t < 21$ cell time. A possible explanation is that the input video sequence influences only the long term (low service rate) burstiness of the traffic trace, while the multimedia terminal regulates the short term burstiness (for high service rates). Moreover, the video application has a very deterministic nature. It produces video frames on a given frame rate. These frames are sliced first into Maximum Transfer Units (MTU), than packed into IP packets and segmented into ATM cells by the protocol stack. This results in a multilevel burst structure, where only the size and timing of top-level burst (i.e. video frame) is different. This observation is further analyzed in Chapter 3. Another observation is that the number of cells in the trace (see Table C.3) determines the value of $s(T)$, as expected.

The effect of changing the frame rate as parameter of the video application can be seen in Figure 2.15. It is visible that the LB curves are shifted along the r-axis, but the level of plateau coincide in the LB slope graph. This indicates that the burst structure is similar and only the timing of bursts differ on the highest burst level, i.e. the video frame level. Another interesting observation is that the (m) and (n) curves are much closer to each other than to curve (l), although the elbow of the LB curves (which indicates the mean rate) is linearly proportional to the frame rate. A possible explanation to this phenomenon is that there is a finite upper limit of the performance of multimedia workstation, and it can not really support real-time pictures (25 fps) only about 22 fps.

Internet Traffic

Internet traffic is in the spotlight of many recent research activities. The Appendix lists several traffic traces from a number of traffic types. In this section the traffic of file transfer (FTP), Ping and aggregated LAN are analyzed.

A large size file was downloaded during the FTP session (Figure 2.16). The LB slope curves of trace (q) and (r) indicate the very simple on-off like bust structure. It is also interesting that repeated measurements yield traces with very similar characteristics. This observation is supported by a number of trials [C5].

The Ping UNIX command was used with four different sizes of transferred message (128, 256, 512 and 1024 cells). The queue length for the peak-end is proportional to the message size in case of the four traces (s), (t), (u) and (v). It is noticeable in Figure 2.17 that the LB slope curve of each Ping trace begins with a value which corresponds to the packet size setting of the trace, i.e. 164, 240, 475 and 1180 cell times for the (s-v) traces, respectively, although these values does not exactly follow the 1:2:4:8 ratio. The deviation is due to the fact that LBA captures the worst case behaviour of the traffic (exactly as the ATM switch).

Internet traffic of interconnected LANs was also measured on the Swedish University Network (SUNET). Since these traces contain more than 83 million cell arrivals, the $s(T)$ value is very large (see Figure 2.18). Although the traces were taken at different time, it is interesting to recognize the similarity among the curves from independent measurements in both graphs. The mean cell rate varied between 3000 and 9000 cps. The burst structure is very disperse, similarly to the other aggregate trace of four video sources (b) in Figure 2.13.

2.2.4 Relationship Between the Measured Traces and the MOO Traces

A pronounced similarity can be noticed by comparing the LB curve of measured traffic traces presented in the previous section with the LB curve of deterministic MOO traces in Section 2.2.2. Denote $q_R(r)$ and $q_{MOO}(r)$ the LB curve of the measured traces and that of the MOO traces, respectively.

Firstly, we can see that the basic properties of the LB curve (see section 2.2.1) are valid for both types. Specifically, both LB envelope curves ($\hat{q}_R(r)$ and $\hat{q}_{MOO}(r)$) are non negative, monotone decreasing and convex. Moreover, the extension of the curve's mean-end leads to the total number of cells in the trace (i.e. $\lim_{r \to 0} \hat{q}(r) = N$), while the peak-end approaches zero at the peak cell rate (r_P) of the source[4]. Another observation is, that both $\hat{q}_R(r)$ and $\hat{q}_{MOO}(r)$ are linear in the region where the service rate is below the long term average rate of the trace (i.e. $0 < r \leq r_L$). The slope of the corresponding linear section (s_L) is determined by the length of the traffic trace (T).

Secondly, there are two special properties that are also valid for both types:

[4]The peak cell rate is equal to the full link rate r_0 in case of unshaped traffic.

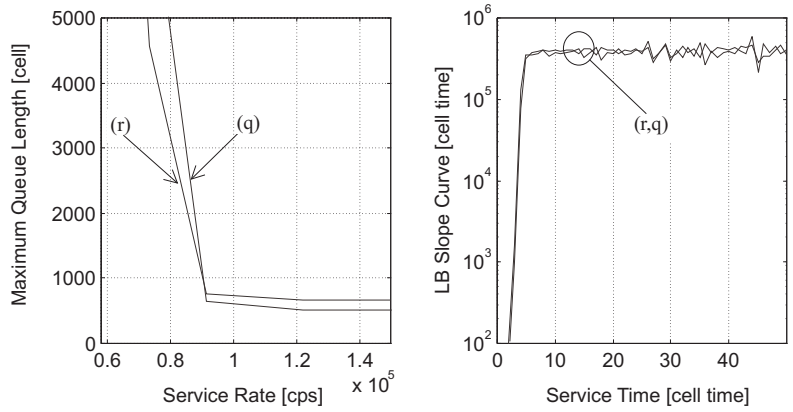

Figure 2.16: Leaky Bucket Analysis of file transfer sessions

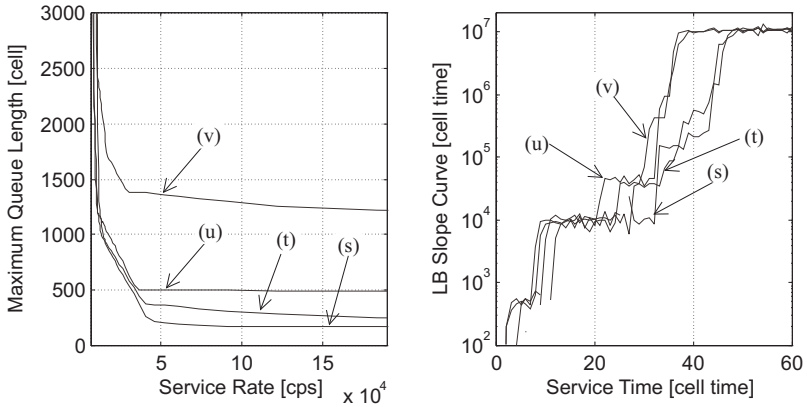

Figure 2.17: Leaky Bucket Analysis of traffic from Ping sessions with message size: (s) @128, (t) @256, (u) @512, (v) @1024 cells

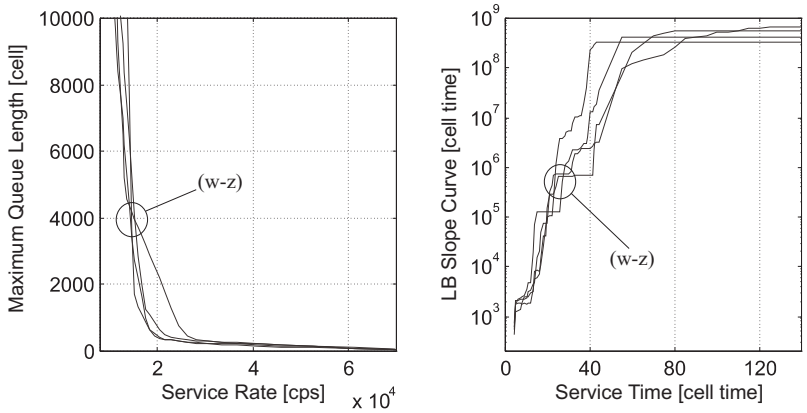

Figure 2.18: Leaky Bucket Analysis of aggregate traffic measured on SUNET

P1. Both $\hat{q}_R(r)$ and $\hat{q}_{MOO}(r)$ have a few breaking-points with semi-linear[5] sections among them. Each of these sections can be determined by its slope (s_i).

P2. The number of linear sections in $\hat{q}_R(r)$ and $\hat{q}_{MOO}(r)$ correspond to the number of burst levels (L) in the traffic trace.

As we have seen in Section 2.2.2, these two properties are not valid for every finite traffic trace $\mu(T, N)$. For instance the LB curve of the Accelerating Trace has no linear sections, and its slope is steadily increasing from the peak-end toward the mean-end. However, my measurement experiments on a broad set of traffic types validates these properties. The very likely reason of this is the 'On-Off nature' of active traffic devices, namely that network interface cards, routers, switches, shapers, etc. temporarily store the traffic in buffers of different sizes and read out traffic from these buffers at a fix rate. The size of different buffers determine the burst size on different burst levels, while the buffer drain rates determine the mean rate on the different burst levels. I utilize this observation for establishing a traffic model in Section 2.4.1.

2.2.5 The Leaky Bucket Analysis and Other Deterministic Bounds

The LB curve of the proposed LBA framework is a deterministic traffic constraint function that specifies an upper bound on the burstiness of the traffic entering a network node. Moreover, it can be used for burstiness analysis with the help of the LB slope curve. One of the major designing goal of LBA was to achieve a computationally simple approach. Thus there is only one bounding variable, namely the queue length, that has to be maintained for calculating this metric, unlike in case of other deterministic bounds, which are based on a different parameter set and thus require the maintenance of several variables [2, 89, 90]. The special selection of service rates (i.e. that $r \in \{1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{T}\}$) also targets simplicity, because floating point operations could be avoided in this way. According to my measurement based experiments, the loss of accuracy in the region $r \to 1$ due to this selection is negligible, because the mean rate of the trace (and thus the mean-end of $q(r)$) is far from the link rate in the practical cases. In the followings I compare the LBA to the 'arrival curve' and 'service curve' concepts of [3].

Cruz defines the $R(k), R(k) \in I^+$ function that denotes the number of packets flowing on the link during the discrete time slot k. Moreover, he defines the *arrival curve* $b(\cdot)$ as an upper bound on the arriving traffic volume:

$$R(j+1, k) \leq b(k-j) \qquad \forall j, k \in I, \quad j \leq k, \tag{2.26}$$

[5]Due to the discrete nature of $q_R(r)$, neighboring points do not exactly fit to a line only by allowing a small threshold of $\Delta q = \pm 2$ cells (this threshold is not the same as ε in section 2.4.1). But since the LB curve has usually very pronounced breaking points and the points beyond those breaking points deviate with one or more orders of magnitude from the regression line, it is straightforward to group the points into a few semi-linear sections.

where $R(j+1,k) \doteq \sum_{l=j+1}^{k} R(l)$.

For the packets departing from the network element, Cruz defines a non-decreasing non-negative function, called *service curve* $S(\cdot)$. A network element guarantees a service curve of S if for any k there exists $j \leq k$ such that the queue is empty $B(j) = 0$ and $R_{out}(j+1,k) \leq S(k-j)$.

Therefore Cruz considers both arriving and departing packets per network element using these concepts and aims for deriving end-to-end performance metrics for a whole network. The goal of LBA is different, because it aims for characterizing the arriving traffic with special respect to its burstiness structure in order to support resource allocation for traffic sources attached to a network element. By comparing the arrival curve $b(k-j)$ and the LB curve $q(r)$ bounds it is apparent, that the former limits the volume of traffic in a time window of size $k-j$, while the latter bounds the queue length in a G/D/1, FIFO system, which is evenly served at a service rate of r. The advantage of this queueing system is that the parameters of both the standardised ATM Usage Parameter Control and the ATM Connection Traffic Descriptor are based on that [35, 66], thus the LBA can directly provide those parameters, unlike the service curve concept.

2.3 Robustness Study of the Leaky Bucket Analysis

In the previous section, I used the Leaky Bucket Analysis for characterizing a single traffic trace. This section investigates how robust the LBA is, if it is used for characterizing different traces of the same traffic type.

An important feature of a practical characterization method is that it highlights the essential characteristics of the traffic and hides the unnecessary variations. Therefore an ideal tool gives basically the same descriptors to each traces of a specific traffic type, independently from the length of the trace and the time when it was captured. Let us investigate the impacts of these two factors on the LB curve.

2.3.1 Repeatability

VBR Video Traffic

Many traces were captured from the same traffic type, using the same quality settings in order to examine the repeatability of the LBA method. The traffic types are different from the perspective of determinism of cell generation. *Stored VBR video traffic* is easily reproducible, since the input video sequence can be repeated and most of the processes transforming and transferring the data inside of the multimedia workstation are of deterministic nature (see Chapter 3). That is why it can be assumed that traces taken from a VBR video traffic source have roughly the same characteristics. This fact is noticeable in Figure 2.19 which depicts the *relative error ratio* of $q^t(t)$ calculated for traces of the same

source from different trials. This metric for two of M measured traces can be expressed as:

$$E_{j,k} = 2\frac{|q_j^t(t) - q_k^t(t)|}{q_j^t(t) + q_k^t(t)} \qquad \forall j,k \in \{1,2,...M\}; \quad j \neq k. \tag{2.27}$$

Safety margins can be defined by giving *accuracy thresholds* and the service time regions where they are valid. The safety margins are indicated by bold lines and the maximum, minimum and mean error curves are drawn in the following figures.

The relative error has a peak between $20 < t < 30$ in case of video traffic (see Figure 2.19). This corresponds to 40 ms, i.e. the interarrival time between video frames (25 fps). As described later, the real frame rate is determined by the scheduling system and the upper limit of the terminal's performance. This is a reasonable explanation for the degradation of repeatability in this region.

Internet Traffic

In spite of VBR video traffic, one can only assume that the sources have similar behaviour in a statistical manner in case of *Internet traffic*. Therefore to have a robust statistical average, hundreds of traces should be taken from the same traffic. Repeating the measurement is quite easy in case of simple "Internet sessions" like Ping and FTP. According to Figure 2.20, the behaviour of these sources is also very deterministic. The only jump of relative error can be seen around the elbow of the LB curve (compare with Figure 2.17).

The safety region of acceptable error threshold (50%) is much shorter for *aggregated Internet traffic*, than for simple sources (see Figure 2.21). The main reason is the much higher number of independent processes which are generating or influencing the traffic. Considering the accuracy threshold of 200% it is interesting that any bandwidth can be allocated, if the buffer requirement is doubled.

2.3.2 Effect of Trace Length

I investigated the effect of trace length on the shape of the LB curve. Thus I have captured several traces from the same traffic type, but with different length and analyzed the LB envelope and LB slope curves. Figure 2.22 presents the LB slope curve of three traces on a linear scale. Each of these traces has been captured from the same type of traffic, but the second trace has half and the third trace has a quarter of the length of the first trace, respectively.

The results confirm Equation (2.13). The mean-end slope is proportional to the trace length for each trace. The most important observation is that the rest of the LB curves do not significantly differ. I have repeated this test for several other traffic types and got similar results. Based on my experiments, the LBA is robust against changing the trace length in the measurements, if the trace is longer than a threshold, which is in the range of 10^6–10^8 cell times, depending on the traffic type.

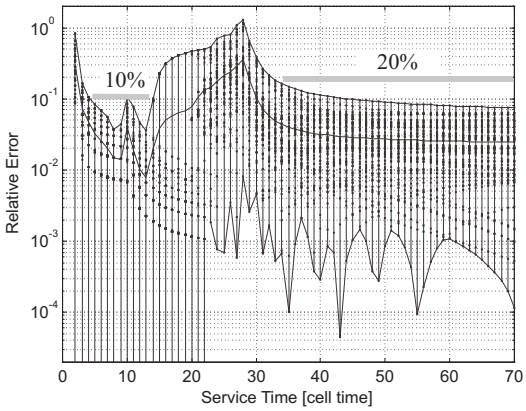

Figure 2.19: Repeatability of VBR video traffic

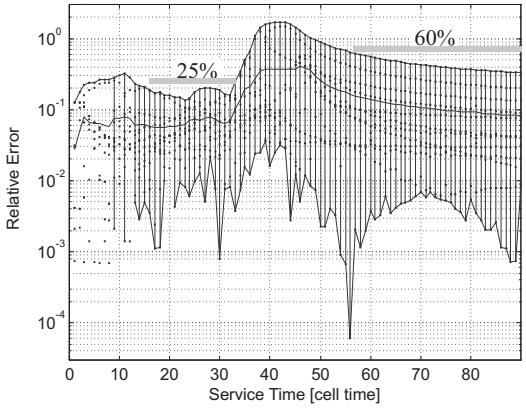

Figure 2.20: Repeatability of Ping sessions

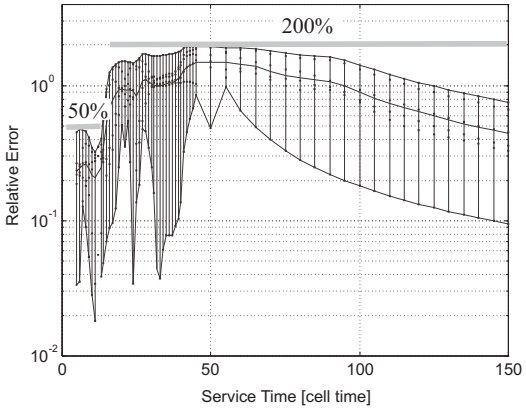

Figure 2.21: Repeatability of aggregate Internet traffic

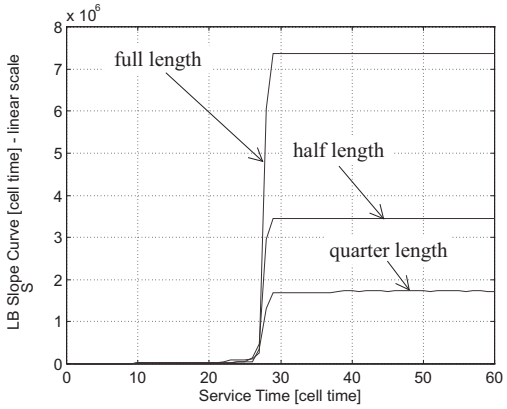

Figure 2.22: Effect of the number of cells in the measured trace

2.4 Models Based on the Leaky Bucket Curve

In this section I establish two traffic models using the Leaky Bucket Curve for finding the model parameters.

In the wide sense, the LB curve itself is a model of the traffic trace. However, this model represents only the worst case behaviour of the respective traffic source (see Section 2.5.3). Thus I used the LB curve as 'model fitting rule' and established two traffic models which can approximate the LB curve of measured traces. In the first one, I have selected a special subset of traces, namely the MOO traces. In the second one, I have used fluid flow modeling.

2.4.1 Multilevel On-Off Model

Based on the similarities between the LB curves of MOO traces and measured traces, a straightforward idea is to approximate the LB curve of the measured trace with a piece-wise linear function, which corresponds to an MOO trace. The resulted MOO trace is a model that captures the queuing behaviour of the measured trace.

I have developed a procedure for setting the parameters of a multilevel on-off model that can be used for approximating the LB and LB slope curves of a measured traffic trace $\mu(T, N)$ [B1, C1]. The steps of this procedure are illustrated on the GT10 trace (see Chapter 3):

1. *Capture a sample trace of length T from the traffic of the investigated traffic source.*

2. *Calculate the points of the LB curve $q(r)$ by post-processing the captured cell stream according to Equations (2.4) and (2.5).*

3. *Calculate the LB slope curve $s(t)$ according to Equation (2.12).*
   The LB and LB slope curves of the GT10 trace are presented in Figure 2.24.

4. *Identify the almost linear sections in the LB curve.*
   It was shown in Section 2.2.4, that the points of the LB curve can be grouped into a linear section between two breaking points r_i and r_{i+1}:

$$\hat{q}(r,i) = q(r_i) - (r_i - r)\frac{q(r_{i+1}) - q(r_i)}{r_{i+1} - r_i} \qquad r_{i+1} < r \leq r_i, \quad i = 0, 1, 2, \ldots, L, \quad (2.28)$$

   where L denotes the number of breaking points.

   These breaking points can be easily found, as they appear as vertical jumps on the LB Slope curve. This observation can be expressed by the following equations:

$$\begin{aligned} t_0 &= 1, \\ t_i &= \min_{t > t_{i-1}} \left\{ t : \frac{|s(t) - s(t_{i-1})|}{s(t)} > \varepsilon \right\}, \qquad i = 1, 2, \ldots, L, \end{aligned} \quad (2.29)$$

   where $t_i = \frac{1}{r_i}$ denote the breaking points, ε denotes an arbitrary, small threshold[6]. The general approximation process is illustrated by Figure 2.23. According to Figure 2.24, there are three such jumps in the $s(t)$ of GT10 – around $t \approx 1$, $t \approx 20$ and $t \approx 340$. This means that $L = 3$. By zooming out these graphs, one can precisely read the values of the $\{t_i\}$, $\{r_i\}$ series. I wrote these values in the first two columns of Table 2.1. I have chosen $\varepsilon = 20$ cell times for the GT10 trace.

Table 2.1: Parameters of the MOO model for the GT10 trace

i	$\mathbf{t_i}$ [cell time]	$\mathbf{r_i}$ [cell per cell time]	$\mathbf{N_i}$ [cell]	$\mathbf{T_i}$ [cell time]
1	1	1	172	172
2	20	0.05	581	11620
3	332	0.003	22872	7842534

5. *Plot the $\hat{q}_i(r)$ linear sections using Equation (2.22), the breaking points $\{r_i; q(r_i)\}$ and L retrieved in the previous step.*
   If we compare the empirical and approximated curves in Figure 2.24 and 2.25, respectively, we can see that the proposed approximation emphasizes the main burst levels without loosing the information represented by the $q(r)$ curve.

6. *Read the burst size N_i from the $\hat{q}(r)$ graph.*
   The first level burst consists of one cell $N_1 = 1$ by definition, while the burst size on the ith burst level is determined by the intersection of the extension of the $(i-1)$th

[6] If ε is very small, there are traces where the number of jumps equals to the number of cells in the trace, (i.e. $L \to N$), which is impractical.

linear section of the LB envelope and the vertical axis. This corresponds to inserting $r = 0$ in Equation (2.28):

$$N_i = \hat{q}(0, i-1), \qquad i = 1, 2, \ldots, L-1. \tag{2.30}$$

7. *Calculate the interarrival time of bursts on the ith burst level using:*

$$T_i = t_i \cdot N_i, \qquad i = 1, 2, \ldots, L. \tag{2.31}$$

Using these equations, the series of $\{T_i; N_i\}$ – i.e. the parameters of the MOO model –, can be successively obtained. The resulted values are presented in Table 2.1 for the GT10 trace.

Is the MOO Model Applicable for Every Traffic Type?

If we analyze the LB Slope curves of different traffic types (see Figures 2.14, 2.15, 2.16, 2.17 and 2.18), we can see that the traces captured directly at the traffic source yield smoother and more distinct plateau than the traces of aggregate traffic. Thus the presented modeling approach is more suitable for the former type. On the other hand, the latter type may be modeled with Accelerating traces (compare Figure 2.4 and 2.18).

2.4.2 Fluid Flow Model

Measured ATM traffic traces were analyzed in [C3] and a simple model was proposed for VBR video traffic based on the LB curve. Furthermore, a method was given for fitting the LB curve of the measured trace with the queue length of a two level fluid model.

Motivation For Fluid Flow Modeling

Video traffic is generated by encoding subsequent video frames composed by several packets. The data rate is in the order of several Mbps, while the packet length is less than 100 kbits (in our situation a packet consisted of 172 cells). Thus, we can move from cell scale to larger time scales and the discrete nature of data can be ignored. Therefore I chose the fluid flow model for our modeling purposes.

The simplest model which can capture the basic characteristics of video traffic is a two state fluid flow model, where the source generates continuous bit streams at different data rates.

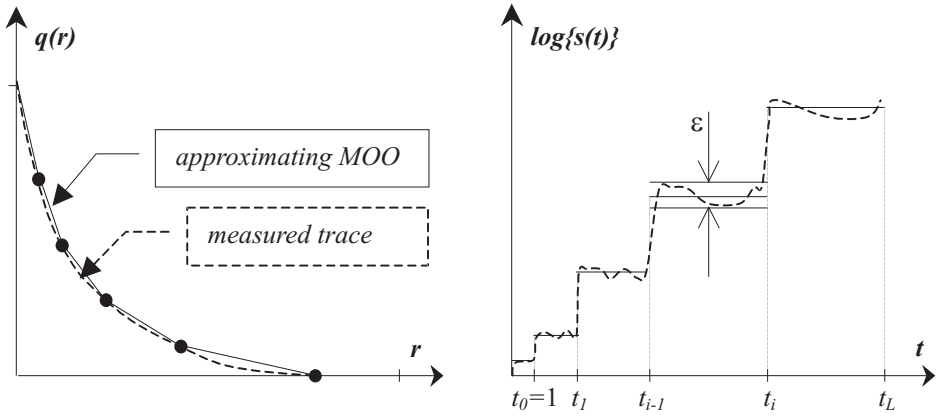

Figure 2.23: Multilevel On-Off Approximation

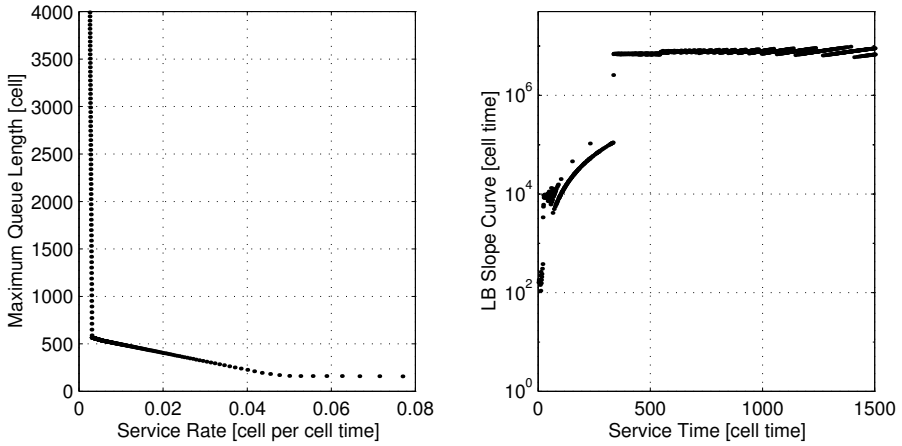

Figure 2.24: The empirical $q(r)$ and $s(t)$ curves of the GT10 trace

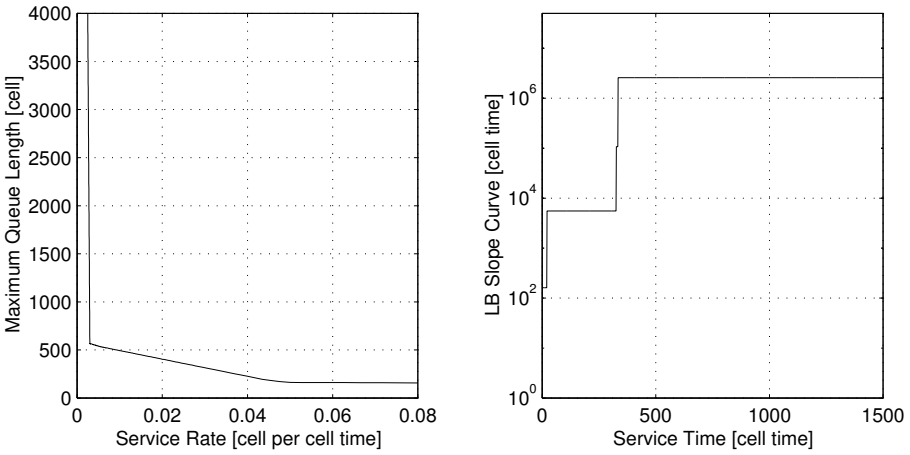

Figure 2.25: The approximated $q(r)$ and $s(t)$ curves of the GT10 trace

The Fluid Flow Model

In this model the fluid arrival rate at time t is denoted by $\Lambda(t)$ has the form $\Lambda(t) = r(Z(t))$ where $Z(t)$ is a Markov process and $r(z)$ is a deterministic function. Now assume that the Markov process $Z(t)$ has two states with generation rates r_0 and r_1, and $r_0 < c < r_1$ where c is the service rate. The transition intensities of $Z(t)$ are denoted by λ (from state 0 to state 1) and μ (from state 1 to state 0) as it is shown in Figure 2.26. By solving the Chapman-Kolmogorov equations for the partial distribution functions the complementary queue length distribution, i.e. an estimation of the cell loss probability, can be obtained [12]:

$$Q(x) = \Pr\{X > x\} = \frac{\lambda}{\lambda+\mu}\frac{r_1 - r_0}{c - r_0}e^{-\zeta x}, \qquad \zeta = \frac{(c-r_0)\mu - (r_1-c)\lambda}{(r_1-c)(c-r_0)}. \tag{2.32}$$

There are four parameters to set in the model: the two transition rates (λ and μ) and the two generation rates (r_0 and r_1). The purpose is to set these parameters such that the model can capture the LB curve. A problem arises from the nature of stochastic models, namely there is no point to speak about maximum queue length because there is a non zero probability of having any queue length in the model. Thus, instead of considering the maximum queue length curve, I relate the curve corresponding to a specified quantile of the queue length (i.e. the queue length, which is exceeded by a specified probability) in the model to the LB curve of the measured traffic trace. This curve can be expressed using a targeted cell loss rate (e.g. $l = 10^{-6}$) and the long term average rate r_M of the trace:

$$x(c) = -\frac{1}{\zeta}\ln\left(l\frac{c - r_0}{r_M - r_0}\right). \tag{2.33}$$

I have established the following model fitting algorithm for capturing the LB envelope curve with a two level fluid flow model [C3]:

1. The long term average rate (r_M) of the trace should be estimated by $r_M = \frac{N}{T}$, where N denotes the number of cell arrivals in the captured trace of length T.

2. The burst tolerance related buffer size (N_C) should be read from the LB curve of the measured trace using the sustainable cell rate (i.e. $N_C = q(r_S)$) and this point should be fitted with the queue length curve of the fluid model at a given cell loss probability (l) using Equation 2.32.

3. The parameters r_M, r_S, and N_C determine ζ (see Equation (2.32) and Appendix B) and the transition intensities can be derived from:

$$\lambda = \zeta\frac{(r_1 - r_S)(r_S - r_0)}{(r_S - r_0)\left(\frac{r_1 - r_M}{r_M - r_0}\right) - (r_1 - r_S)}, \tag{2.34}$$

$$\mu = \lambda\frac{r_1 - r_M}{r_M - r_0}. \tag{2.35}$$

4. The two remaining parameters, r_0 and r_1, can be tuned in order to get the best curve fitting for the rest of the curves. Alternatively, it is also possible to set them as r_0 to be zero and r_1 to be the peak cell rate.

The derivation of the parameter fitting equations are given in Appendix B. The above model has the advantage that it is a rather simple model with very simple fitting procedure. It also has the nice feature that it directly captures the important queuing behaviour (i.e. $q(r)$) instead of only the statistics of the arrival process coupled with a nontrivial queuing problem.

Validation of The Model

The Susie and Noise video traces were investigated for modeling purposes in case of high resolution, 10 fps performance setting (Fig. 5, 6). After performing the leaky bucket analysis of the measured video traces to get the maximum queue length curves and choosing the service rate related to the queue length $b = 500$ as sustainable cell rate with $q = 10^{-5}$ cell loss requirement, I obtained the model parameter settings presented in Table 2.2.

Table 2.2: Parameters of the fluid flow model for the Susie and Noise traces

	$\mathbf{r_M}$ [cps]	$\mathbf{r_S}$ [cps]	λ	μ	$\mathbf{r_0}$ [cps]	$\mathbf{r_1}$ [cps]
Susie	4 450	50 252	9.74	269.44	2 224	66 022
Noise	12 926	26 863	3.40	218.80	12 500	40 347

The LB curve of the Susie and Noise traces, and the queue length (related to the 10^{-5} quantile) of the model are plotted in Figures 2.27 and 2.28. After repeating the model validation for several different traces, we concluded that the model can accurately capture the curves and can be used as source traffic model of the investigated video traffic.

2.5 Applications of the Leaky Bucket Analysis

This section illustrates the applications of the LBA. The LB curve can directly provide us lots of practical information. For instance, we can select the 'ATM four-tuples' (i.e. Peak Cell Rate (PCR) – Cell Delay Variation Tolerance (CDVT), Sustainable Cell Rate (SCR) – Maximum Burst Size (MBS) pairs) using the LB curve, we can dimension the buffer of a shaper or of a switch, and determine appropriate shaping rate, as discussed below.

2.5.1 Resource Dimensioning

The previous sections illustrated how the LB curve can be directly used as a traffic descriptor. Apart from this, it can also be applied as a tool for determining the standard ATM

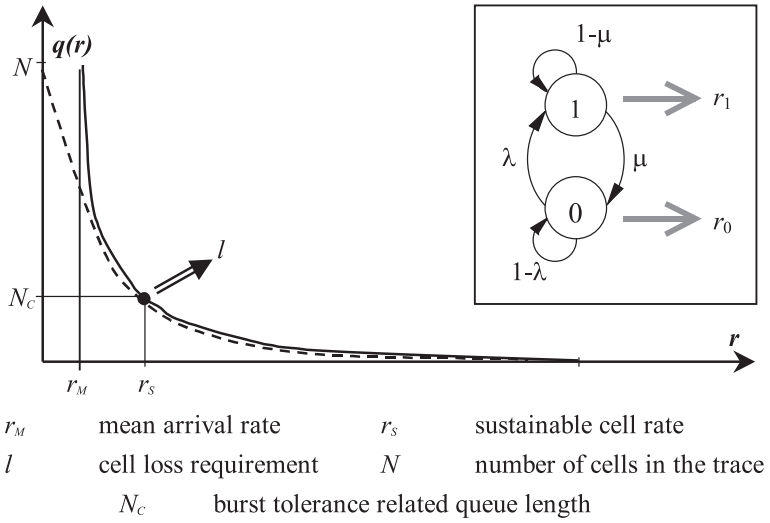

Figure 2.26: Fitting the LB curve of a measured trace (dashed line) and the queue length curve of the fluid flow model (solid line)

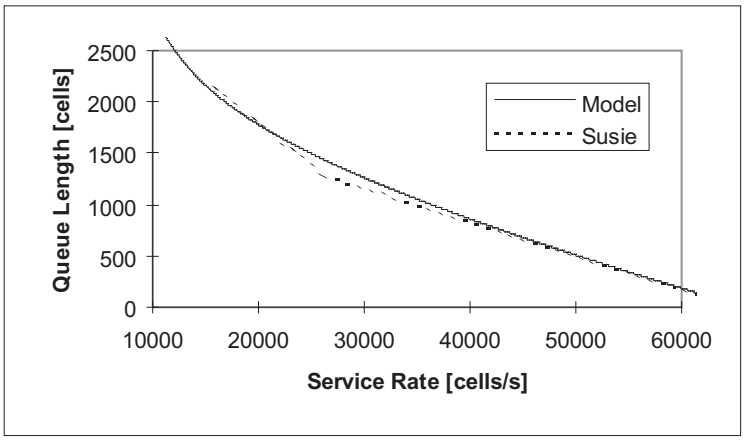

Figure 2.27: Fitting the fluid flow model to the Susie trace

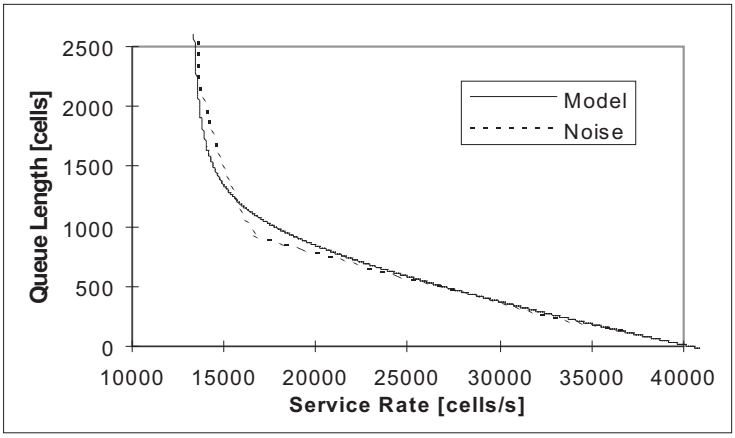

Figure 2.28: Fitting the fluid flow model to the Noise trace

Connection Traffic Descriptor [66, 34]. Let us compare first the constraint given by the latter and the demand represented by the LB curve of a traffic trace [B1]. The three first parameters of the ATM source traffic descriptor, i.e. the PCR (r_P), SCR (r_S) and MBS (N_S) and the CDVT (τ) determine only a line, while the LB curve gives several linear sections in most cases (except in the case of one-level on-off trace). In oder words, the traffic contract corresponds to a one-level MOO and thus it represents higher resource requirement than the LB curve (see bold line in Figure 2.29).

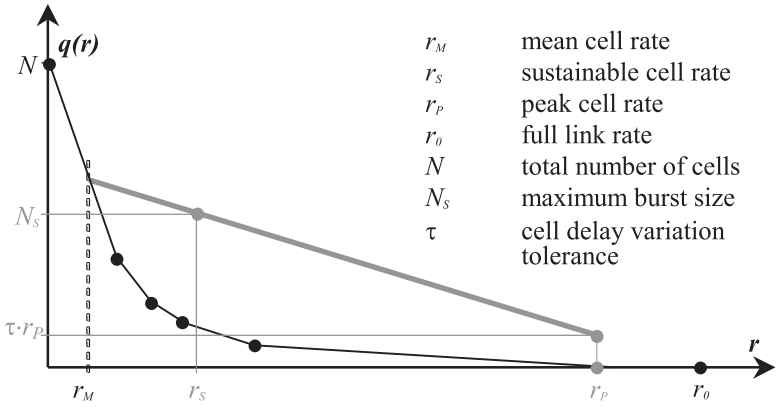

Figure 2.29: Leaky Bucket Curve of a $\mu(T, N)$ traffic trace and operating line determined by the ATM traffic descriptor (in gray)

Another interesting curve, the delay tolerance vs. service rate graph can be obtained by dividing the maximum queue length values in the graph in Figure 2.29 by the corresponding service rates (see definition of $\delta(r)$ in Equation (2.15)). In this way we get a similar, new graph that represents a series of tolerance – rate pairs for the Usage Parameter Control (UPC). From the $q(r)$ or $\delta(r)$ curves the traffic parameters of the ATM connection descriptor can be directly selected. The proper setting of these parameters are necessary for appropriate resource allocation, but the way how to chose them is far from being straightforward [46, 75].

We can use the LB curve for setting $\{r_S, N_S\}$ by choosing a service rate, buffer size pair $\{r, q(r)\}$ from or above the LB curve in the region $r_M \leq r < r_P$. The closer this point is to the LB curve, the less conservative is the selection. The selected maximum queue length value can be interpreted as a minimum buffer capacity needed in the network for avoiding cell loss for the given source. By choosing a point toward the peak-end, we can optimize for *saving buffer size*, while the points toward the mean-end yield larger buffer but *less link capacity* for the traffic source. Furthermore, we can consider a delay constraint by drawing trajectories into the LB graph (see d_s in Figure 2.30) and selecting a point below that line. The *targeted cell loss rate* can be a further parameter to consider. The LB curve

estimates the maximum queue length for zero loss, but it is possible to derive similar curves for specific loss rates (see Section 2.5.4).

The advantages of setting the traffic descriptor based on LBA are that (i) it is simple and intuitive to obtain the traffic parameters, (ii) it is straightforward to consider different various optimization criteria and (iii) it is clear how conservative the selection is, because the LB curve predicts the operation of the UPC (that also monitors the maximum queue length, but only for one or two service rates).

LB Curve or ATM Traffic Descriptor?

There is a tradeoff between the number of parameters for describing a connection and the efficiency of resource dimensioning. In most of the cases, the naive user is not able to determine even the simple parameters of the ATM traffic descriptor. However, for a well known traffic source (e.g. for a specific video sequence in a Video on Demand service), it is straight forward to estimate the characteristic points of the LB curve. Therefore the applicability of the LB curve as traffic descriptor depends on the situation.

2.5.2 Selection of Shaping Rate

Traffic shaping provides a means to control the burstiness [34, 12]. It is believed that such shaping will be needed in ATM networks for users to meet declared traffic agreement with the network, e.g. regarding a connection [67, 76]. A popular shaping method uses the leaky bucket algorithm to read out cells from the shaper's buffer [68, 46, 93, 95]. The shaper limits the maximum number of cells that can be transmitted over a certain time period. This section considers the LB curve and not the LB shaper and it demonstrates how the LBA can be used for determining the shaping rate.

A common approach for achieving acceptable link utilization of an ATM link is peak rate allocation combined with ATM traffic shaping at the source [93, 66]. The appropriate value of the ATM shaping rate for a source depends on which quantities are to be optimized. For instance, if the only parameter of interest is bandwidth utilization a value close to the average rate of the source would provide the optimum value for a shaping rate [94]. This simple approach does not take the shaping delay into consideration, although that is an important factor in case of multimedia applications. Thus another possible strategy is to consider both the shaping delay and the maximum utilization, therefore choose the lowest shaping rate which still gives acceptable delay. A further important aspect is that the shaper (i.e. which may be implemented in an edge node or in the interface card of the source terminal) should have enough buffer space for performing the shaping.

These three constrains can be taken into consideration for selecting a shaping rate by using the LBA. Figure 2.30 depicts the LB curve of the investigated traffic with trajectories F_1, F_2, F_s – representing the maximum delay in three working points –, and other important characteristics:

r_M the long term average rate of the traffic trace.

q_1 the maximum buffer size, which value is governed by the shaper or switch implementation.

r_2 the peak rate which represents the upper bound of shaping rate. It is determined by the required bandwidth utilization. Since the corresponding maximum queue length (q_2) is low, a low delay (d_2) belongs to this working point.

d_s the maximum of acceptable shaping or buffering delay. This value is determined by subjective evaluation of the application.

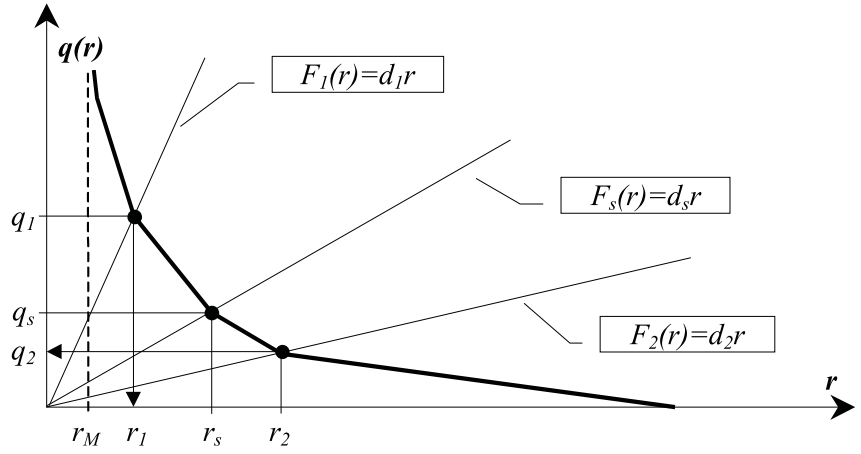

Figure 2.30: Selecting shaping rate

Using these constrains I propose the following guidelines for setting shaping rate [C2]:

- The buffer size should be dimensioned based on the delay constrain (d_s) and the upper buffer limit (q_1):

$$b \leq \min\{q_s, q_1\}.$$

- The actual shaping rate (r_s) should be selected considering the r_M, r_2 and q_1 ($\Rightarrow r_1$) constrains:

$$\max\{r_1, r_M\} < r_s < r_2.$$

The performance of the LBA based shaper design has been investigated in [C2]. The results have shown that the link utilization can be increased by a factor of 37-70 depending on the traffic type while still maintaining a shaping delay that is acceptable for the investigated desk-top multimedia application. Other measurement experiments (see Section 2.2.3) have shown that the LBA is applicable not only for selecting, but also for detecting the shaping rate.

2.5.3 Characterization of Multiplexing Gain

This section analyzes the performance of statistical multiplexing using the Leaky Bucket Analysis. [C1]

The goal is to characterize the multiplexing gain as a function of service rate. Assume that the traffic of N sources are multiplexed according to round-robin scheduling scheme. Denote $q_i(r);\ i = 1, 2, \ldots, N$ the LB curve of the input traces before the multiplexer and $q_{agg}(r)$ the LB curve of the traffic aggregate after multiplexing. In the worst multiplexing case, the output link is shared among the input sources and the buffer requirement is the sum of $q_i(r)$ (i.e. the largest bursts of each source arrive in a correlated manner). The LB distance between this worst case and the actual $q_{agg}(r)$ characterizes the multiplexing gain:

$$d(r) = \left(\sum_{n=1}^{N} q_n(\tfrac{r}{N}) \right) - q_{agg}(r) \tag{2.36}$$

I have characterized the multiplexing gain with this approach based on traffic measurements in [C1]. Figure 2.13 depicts the LB curve of a single VBR source before multiplexing (curve (a)), the calculated LB curve for the worst case and the LB curve of four multiplexed VBR sources (curve (b)). The area between the latter two curves emphasizes the multiplexing gain.

2.5.4 Cell Loss Estimation

With a small extension, the LBA can be used for analyzing the queuing behaviour of a traffic trace. In particular, the empirical queue length distribution can be calculated for a measured traffic trace on the top of the $q(r)$ curve which represents only the maximum queue length. We can express the empirical probability mass function of queue length $\varphi(\sigma)$ by considering Equation (2.4):

$$\varphi(\sigma, t) = \frac{1}{T} \sum_{k=1}^{T} 1\left(\xi(k, t) = \sigma\right), \qquad 1 \le \sigma \le N, \quad 1 \le t \le T, \tag{2.37}$$

where t denotes the service time, σ denotes an arbitrary queue length value and $1(\cdot)$ is the indicator function, which equals to one if the argument is true, otherwise it equals to zero.

The cell loss probability can be estimated using the overflow probability:

$$l(q, t) \approx \Pr\{\xi(k, t) > q\} \approx 1 - \sum_{\sigma=0}^{q} \varphi(\sigma, t), \qquad 1 \le q \le N, \quad 1 \le t \le T, \tag{2.38}$$

where the expression on the right corresponds to the empirical complementary probability distribution function (CPDF) of the queue length. Both of these equations are easy to compute using discrete event simulation. As an illustration, I have calculated the empirical probability mass function of queue length $\varphi(\sigma, t)$ for the 'Noise' trace. Figure 2.31 presents this three dimensional function for different service rates.

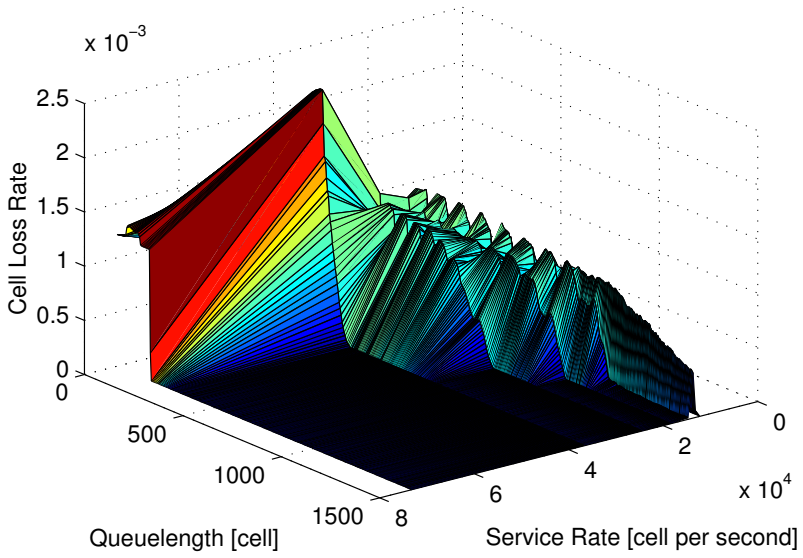

Figure 2.31: PMF of the queue length for different service rates

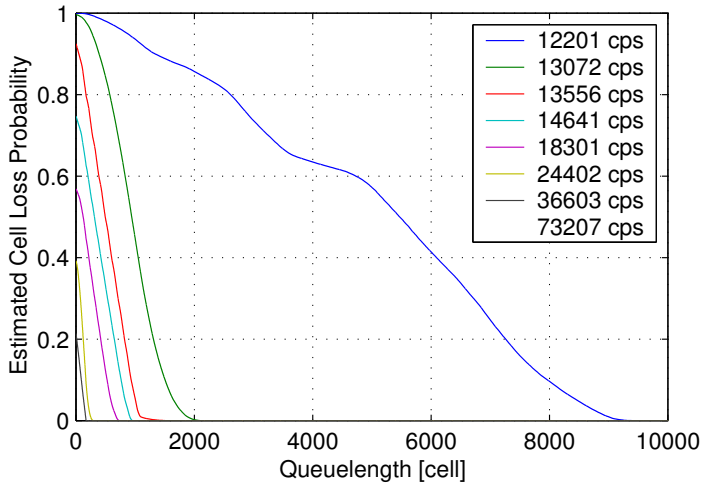

Figure 2.32: CPDF of the queue length for different service rates

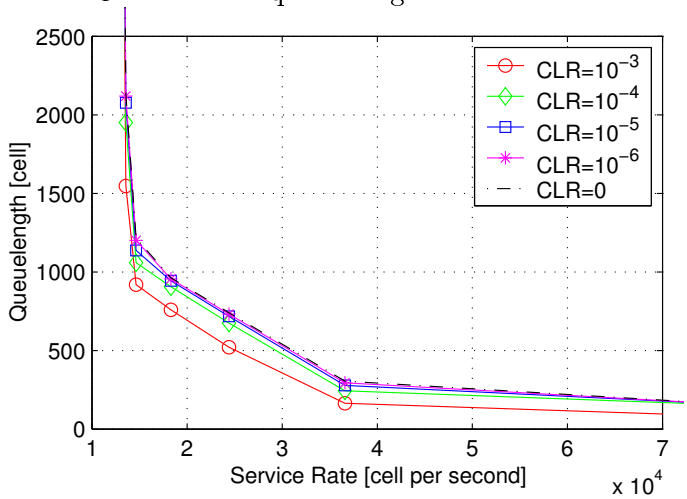

Figure 2.33: Queue length vs. service rate curves for different loss rates

Figure 2.32 presents the empirical CPDF of the Noise trace. The cell loss probability $l(q,t)$ can be estimated using Equation (2.38) and in this way a set of LB curves can be retrieved corresponding to different loss probabilities $l = \gamma$. Instead of the original zero loss curve (i.e. the LB curve), the adequate $q^{\gamma}(r)$ curves can be applied for the analysis. Figure 2.33 presents queue length vs. service rate curves for different cell loss values for the Noise trace. Noteworthy is that the $\gamma = 10^{-6}$ and $\gamma = 10^{-5}$ curves are very close to the LB curve ($\gamma = 0$) that not a single worst case pattern determines the LB curve in real cases. I made this observation for many other measured traffic traces.

In Section 3.4 I have applied this approach for validating a traffic model and numerically calculated cell loss probability for measured traffic traces and the traces generated by my model.

2.6 Summary

This chapter presented the Leaky Bucket Analysis – a framework for resource dimensioning and characterizing the burstiness of ATM traffic –, demonstrated its applicability on deterministic and measured traffic traces, and described its application for traffic modeling, resource dimensioning, shaper design, queuing behaviour analysis and analysis of multiplexing gain.

The evaluation of the Leaky Bucket curve $q(r)$, its dual pair $q^t(t)$ and the Leaky Bucket Slope curve $s(t)$ in Section 2.2.2 has shown that the LBA is capable to visualize the resource demand of a trace. For instance, I validated the well known result that the simple On-Off trace is the worst case, while the CBR trace is the best case among finite length traffic traces $\mu(T, N)$ regarding resource demand. It has been also discussed that other traffic traces can be considered as a special instance of the family of Multilevel On-Off traces or of Accelerating traces, which have linear $q(r)$ and $q^t(t)$ curve, respectively. Section 2.2.3 has demonstrated that the Leaky Bucket Analysis can characterize how the resource demand and the burst structure of VBR video traffic changes due to traffic shaping, multiplexing, changing the video sequence or the video frame rate. It has been illustrated that the LBA can also reflect the 'On-Off nature' of file transfer application and the size of a Ping message. The robustness of LBA against the variations in the traffic of a certain traffic type and the length of captured trace has been investigated in Section 2.3, and accuracy thresholds have been given for single and aggregated Internet and VBR video traffic using the relative error ratio of $q^t(t)$. This analysis emphasized that the LBA could characterize specific traffic types too, not only a single traffic trace.

The deterministic metrics of LBA have been fitted to two analytic models in Section 2.4. The first model has utilized that the burst structure of measured traffic traces and MOO traces are similar. Section 2.4.1 has proposed a procedure for setting the parameters of a MOO trace based on the breaking points in $q(r)$ and the place of plateau in $s(t)$. The second model has been a two-level fluid flow model. Section 2.4.2 has described how the model parameters can be determined using the long term average rate, the sustainable cell rate, the maximum burst size and the targeted cell loss probability, and it has presented numerical results for VBR video traffic.

Section 2.5 has described several applications of LBA. Section 2.5.1 has discussed how the LB curve can be used for selecting the parameter set of the ATM connection traffic descriptor considering different optimization criteria, such as saving buffer space, better link utilization, keeping a delay constraint or a targeted cell loss rate. Section 2.5.2 has proposed guidelines for determining traffic shaping rate based on the delay tolerance of the application, the size of buffer in the shaper device, the average traffic rate and the targeted link utilization. LBA was applicable both for selecting and detecting the shaping rate. Section 2.5.3 has illustrated that the LBA is applicable for queuing behaviour analysis by visualizing the gain of statistical multiplexing for four VBR video sources, while Section 2.5.4 has presented an extension of the LB curve for non zero cell loss rate.

The Leaky Bucket Analysis was designed for ATM networks. If we want to generalize the results for another networking technology, e.g. for IP/Ethernet encapsulation, the major

difficulty would be to handle the variable size of IP packets. Moreover, the benefits of the related MOO model would be also less pronounced due to the simpler multilevel burst structure of IP/Ethernet traffic.

The Leaky Bucket Analysis has the following advantages:

- directly provides the parameter 4-tuple of the ATM connection traffic descriptor,
- applicable for buffer and link dimensioning,
- characterizes traffic burstiness on several time scales,
- visualizes the burst structure,
- simple to compute for measured traffic traces,
- applicable for selecting the shaping rate based on different constrains,
- traffic modeling based on fitting directly these characteristics has the advantage that we avoid the classical way of modeling steps (statistical analysis of arrival process - modeling of arrival process - solving the related queuing problem) but rather we are focusing on directly to capture the queuing behavior.

Chapter 3

Hierarchical Model for Multimedia Traffic Sources

3.1 Introduction

This chapter presents a hierarchical source model for generating Variable Bit Rate (VBR) multimedia traffic. This model captures the inner operation of a multimedia workstation, in particular VBR encoding, process scheduling and protocol encapsulation.

The designers of modern packet switched networks require *traffic models for multimedia traffic sources*, in order to dimension the network and to achieve acceptable service quality and optimal usage of network resources [39, 40]. Variable Bit Rate video has an important role in multimedia teleservices. The measurement based characterization of VBR video traffic and the related networking problems have been of increasing interest in the last decade of teletraffic research [40, 42, 24, 46, C5]. A plenty of studies focus on the characteristics of the output of a particular video source in a practical context [106, 107, 108, 109, 110]. A number of different models have been proposed for VBR video traffic, ranging from *autoregressive processes* [41, 49] through different *Markov type models* [105, 38, 44] including the family of Markov Modulated Poisson Processes [10] to *fractal models* [45, 48]. An overview can be found in [11, 9, 8].

All of these approaches are common in the modeling technique that a specific stochastic process is chosen and the parameters are set by a particular method to fit some statistics of the real traffic source. This methodology can be regarded as a *black box modeling* approach, which is based merely on the characteristics of measured traffic. The black box analysis focuses on the traffic and not on the traffic source and tries to reproduce certain traffic characteristics (e.g. moments of interarrival time) by tuning the parameters of an independent generation process.

One disadvantage of the black box approach is that only very special video services can be characterized with a small number of parameters, and a large set of parameters are required in general [38, 42]. For instance, [105] analysis a special, low bit rate video

conference and uses the first two moments of the per frame bit rate and the coefficient of an assumed exponential autocorrelation function as fitting parameters to a Markovian model to evaluate the performance of a network multiplexer. However, such statistical parameters are not adequate in other cases, because the distribution of output rate can vary substantially for different minutes long segments of the same video sequence [107, 108].

Another drawback of the black box models is that it is difficult to modify the behaviour of the traffic source especially in a realistic way. In most of the cases the behaviour of the traffic source is determined by a few, simple parameter, such as frame rate, picture size, etc. The black box concept can not directly incorporate such simple parameters in the model.

An alternative method is the so-called *white box modeling* approach, which attempts to reproduce the detailed behavior of the source by imitating its inner working [43]. This approach utilizes the a' priori knowledge of the traffic generation process and focuses on the emulation of internal processes in the traffic source, yielding a more accurate source model. The white box approach has received little attention from the video modeling community so far. However, we believe that this non-traditional modeling concept can be very successful in practice, because it can capture the impact of encoding and encapsulation procedures on the generated data traffic.

I have combined the white and black box modeling approaches in previous work [J1, C7]. Apart from introducing a hierarchical traffic model, those papers give a comprehensive source analysis study and reveal the very nature of measured video traffic by performing traffic intensity analysis and correlation structure analysis. This chapter summarizes the essence of my previous work in this area.

The chapter is organized as follows. The main characteristics of the traffic are summarized in Section 3.2. The hierarchical source model is described in Section 3.3, while its validation is given in Section 3.4.

3.2 Multimedia Source Characterization

This section summarizes the main steps of the black box source analysis, which are necessary for the understanding of the hierarchical model. The complete source characterization is published in [J1, C7].

I have recorded the ATM cell stream generated by a multimedia workstation, which received standard video sequences from a video cassette recorder, encoded them using different VBR encoding schemes and transmitted the packetized video and audio information trough a multilayer network interface. In this chapter, the source analysis and the model are illustrated using the "Popple", "Susie" and "Girl with Toys" video sequences with 10 and 25 fps frame rate (i.e. the traces PL10, SU10, GT10, PL25, SU25 and GT25). The measurements and the main statistics of these traces are described in Appendix C.

3.2.1 Traffic Intensity Analysis

The most native way of traffic characterization is to measure its intensity, i.e. the amount of cell arrivals within a given time window. The traffic intensity of recorded ATM traffic traces is analyzed at different time scales, in order to investigate the burst structure. The following figures show traffic intensity of a short trace from the PL25 video sequence on different burst levels. Each column represents the number of arrivals in one time window of 58330, 750, 38 and one cell times in Figure 3.1,3.2, 3.3 and 3.4, respectively.

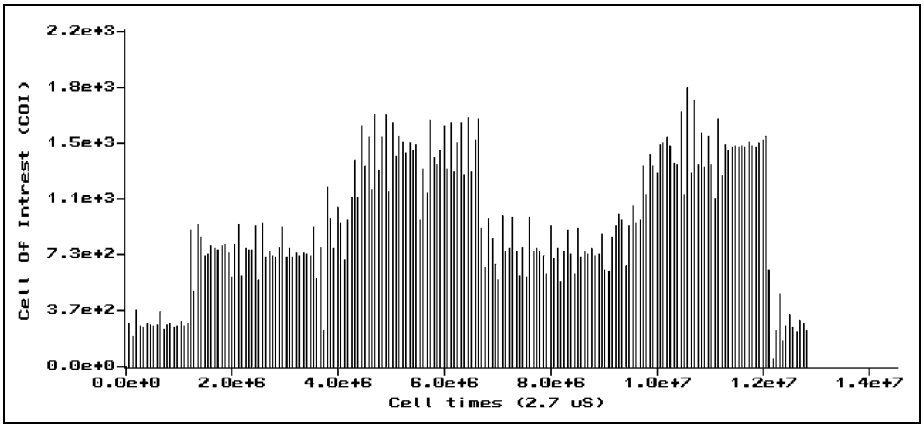

Figure 3.1: The cell arrival intensity of the PL25 sequence on scene level. Scale of the graph: 38 sec, mean rate in the graph: 2251 kbps, size of time window: 58330 cell times.

The complete PL25 sequence is shown in Figure 3.1. The two level shifts are caused by two high speed zoom periods with an intermediate period of partial motion in the picture field. Figure 3.2 magnifies the next time scale (i.e. video frame level) and shows the arrival pattern of seven video frames of varying size with three audio packets in between. The internal structure of a frame (i.e. IP packet level) is shown in Figure 3.3. Finally, we show a single packet in Figure 3.4 that contains 172 ATM cells arriving at practically full link rate.

Based on the traffic intensity analysis the multi-level burst structure of examined VBR traffic is well pronounced, as it is shown in other works [B1, 24, 30]. Observing these figures noteworthy is the regular arrival of frames, packets and cells at different time scales. Audio packets arrive also in a regular manner as it is illustrated in Figure 3.3. By plotting similar figures for the other video sequences we can say that the size of video frames depend on the content of video sequence while the structure of frame internal packets looks very much the same for each frame. Based on these observations our hypothesis is that the generation of frames and packets is independent thus these burst levels can be distinguished in our model. Based on my measurements, I have distinguished four burst levels in the ATM traffic stream generated by a VBR multimedia source (see Figure 3.5); namely the *cell*, *packet*, *frame* and *scene* levels. We introduce the following notation for describing the

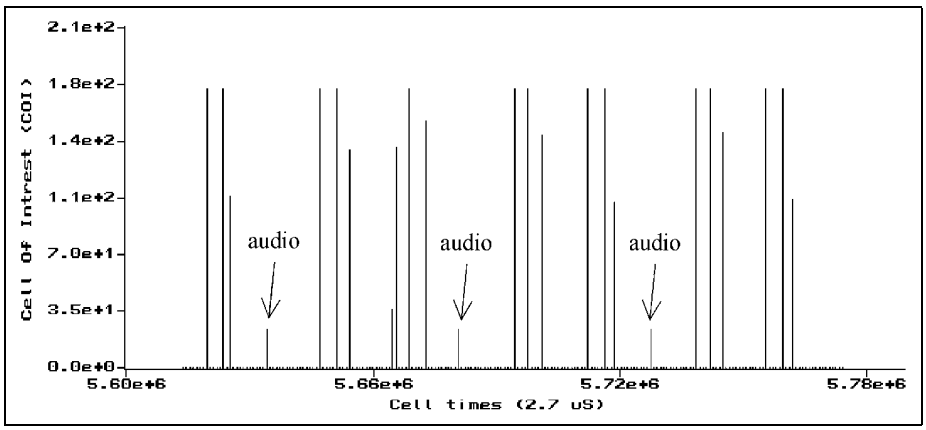

Figure 3.2: The cell arrival intensity of the PL25 sequence on video frame level. Scale of the graph: 490 ms, mean rate in the graph: 3.26 Mbps, size of time window: 750 cell times.

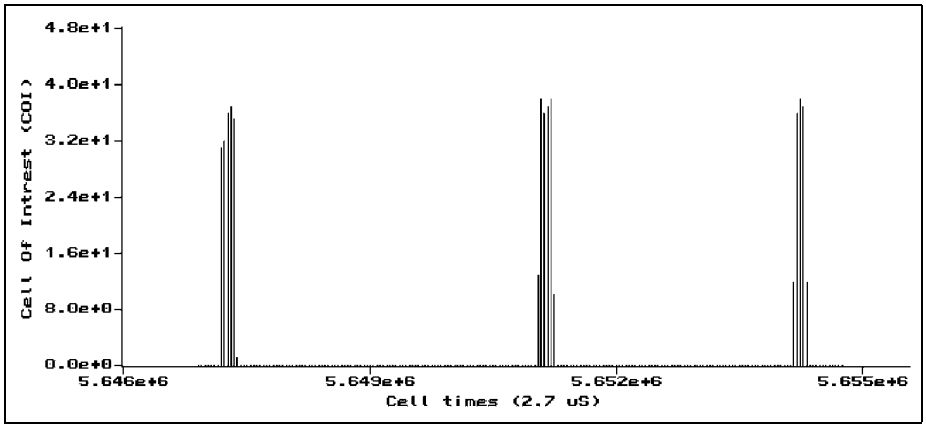

Figure 3.3: The cell arrival intensity of the PL25 sequence on IP packet level. Scale of the graph: 24.5 ms, mean rate in the graph: 9.45 Mbps, size of time window: 38 cell times.

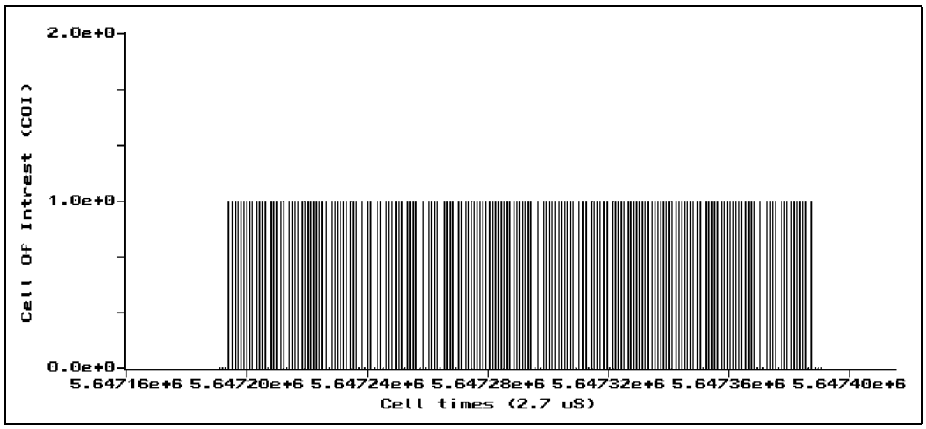

Figure 3.4: The cell arrival intensity of the PL25 sequence on cell level. Scale of the graph: 654 ms, mean rate in the graph: 133.4 Mbps, size of time window: 1 cell time.

multilevel burst structure:

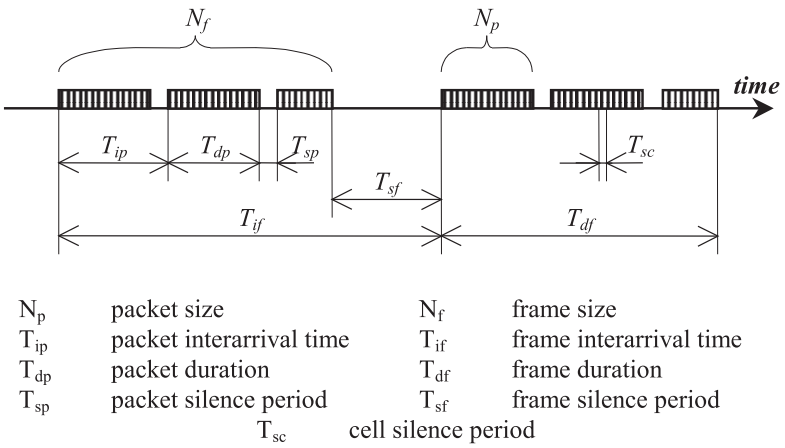

Figure 3.5: Notations regarding the burst levels in VBR ATM traffic

Figure 3.5 depicts the interarrival time and duration of frames (T_{if}, T_{df}) and packets (T_{ip}, T_{dp}) and the silence periods between frames and between packets (T_{sf} and T_{sp}), respectively. N_f and N_p denote the number of ATM cells in a frame and a packet, respectively. N_p^* denotes the size of the last packet in a frame, which is usually shorter than other packets (it contains the last fragment of the frame), while N_{paudio} refers to the packet containing an audio transfer unit (which is much smaller than a video packet).

3.2.2 Silence Period Analysis

Beside the traffic intensity analysis (i.e. analysis of busy periods) the other native way of gaining information about the traffic is to calculate the probability mass function (PMF) of the cell interarrival times (CIT). This latter method can be considered as analysis of the silence periods (T_{sf}, T_{sp} and T_{sc}). The probabilities are estimated by counting the occurrence of CITs of different lengths in the captured trace. The values are smoothed by a moving average technique before drawing figures. The PMF of CITs is depicted in Figure 3.6 and the complementary probability distribution function (CPDF) is presented in Figure 3.7.

The silence periods can be divided into three groups according to Figure 3.6, which characterizes the GT25, SU25 and PL25 sequences. The longest interarrival times (above 8000 cell times) represent the frame silence periods (T_{sf}). The medium values (around 4000 cell times) correspond to the silence periods within the video frames, i.e. between consecutive packets (T_{sp}) while the smallest values (below 10 cell times) express the short

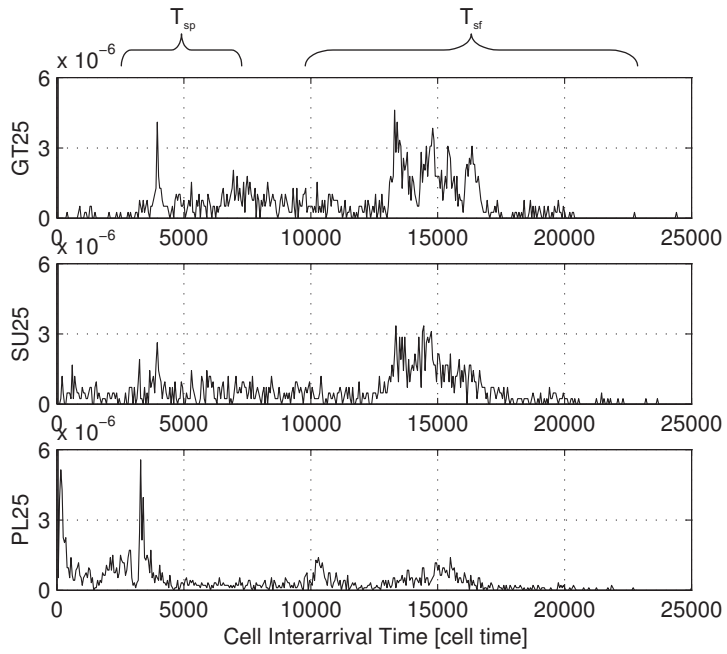

Figure 3.6: Probability mass function of cell interarrival times (GT25, SU25, PL25)

silent periods inside the packets (T_{sc}). The evaluation of each group is given below.

Frame Level

In case of GT-PL10 sequences, the video frame rate is 10 frame/sec thus the theoretical frame interarrival time (T_{if}) is around 36 679 cell time. For the GT-PL25 sequences with 25 frame/sec, the theoretical frame interarrival time (T_{if}) is 14 672 cell time. Two markers in Figure 3.7 indicate these values. Theoretically, the silence period between frames can not be longer than the frame interarrival time (since $T_{sf} = T_{if} - T_{df}$ from Figure 3.5). However, it is clearly shown in Figure 3.7 that there are several silence periods longer than 36 679 and 14 672 cell times for the GT-PL10 sequences and GT-PL25 sequences, respectively. That is in terms of frame rate, the investigated terminal platform is not able to produce frames on the theoretical rate (i.e. 10 and 25 fps). Moreover, the moderate declination of the CPDF curves in Figure 3.7 indicates that the frame generation time and duration varies. Another phenomenon to be noticed in Figure 3.7 is that the maximum silence period (i.e. $T_{sf\,max}$) is significantly shorter for the PL sequences than for the others, probably due to the higher traffic intensity and larger size of video frames.

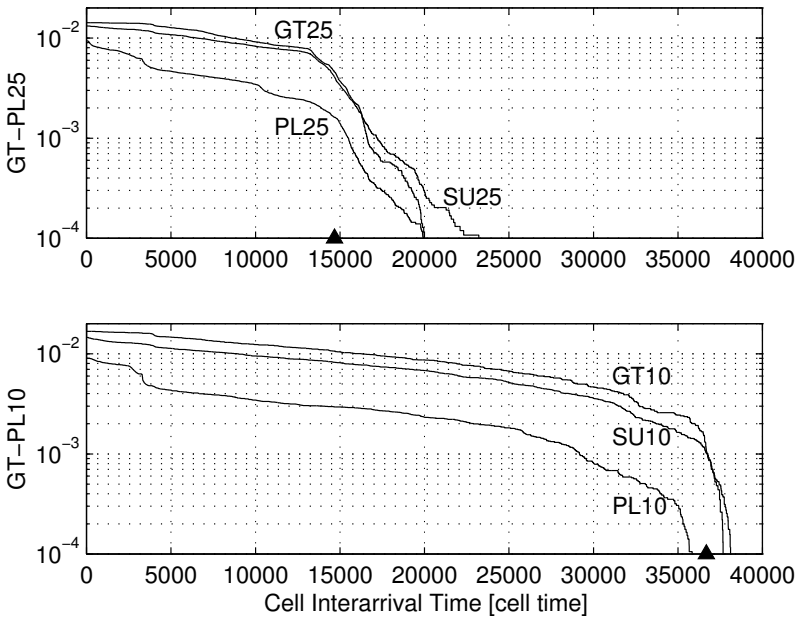

Figure 3.7: Complementary distribution function of cell interarrival times (GT10, SU10, PL10, GT25, SU25, and PL25)

Packet Level

In case of SU25 sequence, the probability of normal packet silence period ($3000 < T_{sp} < 5000$ cell time) is less than in case of PL25 (see Figure 3.6). The reason is that there are more frames, which consist of more than one packet in case of the more intensive PL25 sequence, thus there are more intra-frame packet silence periods in the captured cell stream. The mean packet silence period read from Figure 3.6 is around 4000 cell time for SU25, GT25 and around 3400 cell time for PL25. The packet silence period can be recognized also in Figure 3.7 in form of a sudden declination on the CPDF curves. By comparing the beginning of the CPDF curves in the CIT range of 2000-5000 cell time it is visible that the packet silence period is the same for the GT-PL10 and GT-PL25 sequences. Thus another hypothesis is that the frame rate setting (i.e. 10 fps or 25 fps in our paper) has no impact on the packet generation.

Cell Level

Figure 3.6 highlights that the probability of silence periods with length shorter than three cell times are high (see the peak at the left part of the graph). The reason is that most of the cells are transmitted back-to-back in the unshaped traffic stream. Although physical

layer information was discarded by the measurement instrument, the traces of SDH overhead information can be observed too in form of discontinuities within the packets, resulting silence periods of two–three cell times.

3.2.3 Packet Silence Period Regression

The relationship between the packet silence period (T_{sp}) and the packet size (N_p) is analyzed in this study on the packet level. One can observe in Figure 3.2 and 3.3, that the packet silence period is shorter for the last packet in the frame. Thus our next hypothesis is that the multimedia terminal can produce a shorter packet faster than a larger one. We can confirm this assumption by investigating the relationship between the length of the k^{th} silence period (denoted by T_{sp}^{k}) and the size of the packet generated right after that period (denoted by N_p^{k+1}). Figure 3.8 presents the relationship between these factors and the empirical distribution of the packet silence period for the PL25 sequence.

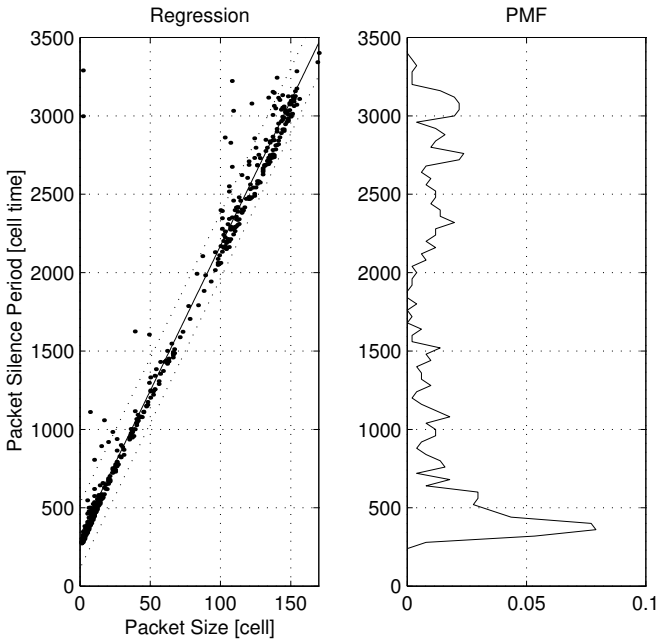

Figure 3.8: Linear regression between the packet size and packet silence period

It is very pronounced that there is a linear relationship between these factors, therefore we can establish a linear approximation:

$$T_{sp}^{k} = \alpha N_p^{k+1} + \beta \tag{3.1}$$

where T_{sp}^k denotes the length of the k^{th} silence period and N_p^{k+1} denotes the size of the packet generated right after that silence period. The α and β constants can be determined by regression from the corresponding T_{sp} and N_p value pairs. In our case, the value of α is 18,5 and β is 323 and at least 50% of the predictions are contained by the ± 200 cell times wide environment of the regression line (see dotted lines in the figure). I have got very similar values for the other sequences, proving that the packet level statistics are independent from the video content. Related works [29, 30] confirm this observation and Equation (3.1) for TCP/IP over ATM traffic.

The [J1, C7] papers provide further methods for black box traffic analysis. The next step is to map the observations of the black box analysis to the internal behaviour of the traffic source, i.e. perform a white box analysis.

3.3 Three-Level Hierarchical Model

After analyzing the VBR traffic, the next step is to look into the 'black box' and make a 'white box' analysis. The video and audio data from the video recorder is processed in three information processing stages in the multimedia workstation, before it is transmitted into the network. The first stage is video *encoding*, followed by the *scheduling*[1] and *encapsulation* of video frames, as it is highlighted in Figure 3.9. The traffic characteristics on the frame level influence the encoding and scheduling stages, while the packet and cell level characteristics have an impact on the encapsulation process. These relationships are described in this section.

For capturing the behaviour of the three information processing stages, I have established a three-level hierarchical model for VBR ATM traffic source. The three stages of this model are:

1. a two-state, discrete time, Markovian model for characterizing the scene level behaviour and the *encoding*,
2. a Gaussian noise model for characterizing the frame *scheduling*,
3. a deterministic, finite state machine for modeling the *encapsulation*.

The *Encoding Model* takes a sequence of captured frames $\{N_f^k\}$, the probability mass function of frame size $\mathrm{P}\{N_f\}$ and the long term average rate R as input, and generates video frames of different size as output. The *Scheduling Model* takes a synthetic frame sequence from the encoding stage as input and assigns timing to each frame according to the application's frame rate setting, i.e. produces a frame interarrival time series $\{T_{if}\}$. The *Encapsulation Model* receives frames of different size and timing from the scheduling stage and produces cell departures with certain packet and cell level characteristics.

In order to avoid using complex, high-order Markov models, I propose a simple, two state model for the encoding stage, because long-term correlation does not have significant

[1] Here I refer to the process scheduling in the operating system and not packet scheduling in a router.

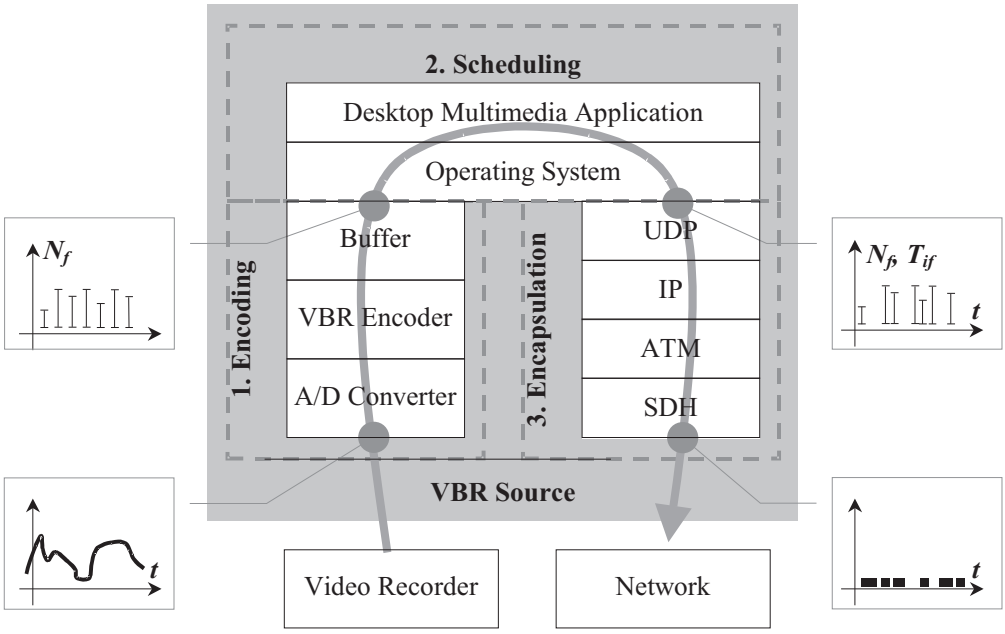

Figure 3.9: The internal operation of a VBR multimedia traffic source and the three stages of the hierarchical model

impact on cell loss in most switch scenarios [22, 24, 25, C6] and the short-term correlation can be efficiently captured by a two state Markov model. I have chosen a simple Gaussian model for the scheduling stage, because that was the simplest but suitable choice for capturing the stochastic nature of frame scheduling according to the frame interarrival time analysis on measured traffic traces [C7]. Finally, I modeled the encapsulation process with a deterministic process, because the network card that performs the encapsulation works as a state machine on one hand, and the structure of frame internal bursts is very deterministic according to the black box analysis on the other hand.

The three model stages are cascaded and the resulted hierarchical system is validated as one model (see Section 3.4).

3.3.1 Parameterization of the Encoding Model

The parameters of the encoding model can be set by analyzing traffic traces captured from the investigated VBR source. I propose the following algorithm:

1. *Capture a trace of cell interarrival times from the output of the VBR traffic source under test.*
   As an illustration, we investigate the PL25 trace.

2. *Perform a silence period analysis by classifying the silence periods into three groups based on their length.*
   I propose three groups, because we have seen in Section 3.2.2, that there are four burst levels in this type of traffic yielding three kinds of silence period (T_{sc},T_{sp} and T_{sf}). This three groups correspond to the cell, packet and frame levels (see Figure 3.5), respectively. The packet and frame silence periods are indicated in Figure 3.6.

3. *Determine the lower threshold of each silence period group.*
   For the PL25 trace, $\tilde{T}_{scmin} = 3$ celltimes, $\tilde{T}_{spmin} = 3500$ celltimes and $\tilde{T}_{sfmin} = 38000$ celltimes.

4. *Retrieve the parameters of the Markov model:*

   (a) *derive a series of video frames* $\{N_f^k\}$ *by filtering out silence periods which are shorter than* $\tilde{T}_{sfmin}$ *and joining individual cells to form a frame between the identified frame silence periods,*

   (b) *calculate the empirical probability mass function* $\mathrm{P}\{N_f\}$ *and the long term average rate* R *of the video trace.*

   For the PL25 trace $R = 2.25$ Mbps.

5. *Plot* $\mathrm{P}\{N_f\}$ *and determine a threshold* N_f^* *that divides the probability mass into an upper and lower quantiles with a proportion of 1:9. State 1 of the Markov model is assigned to the upper quantile, while state 2 is assigned to the lower quantile.*
   The threshold N_f^* is indicated by a marker on the $\mathrm{P}\{N_f\}$ graph of PL25 trace in Figure 3.10. Based on my experiments with numerous traffic traces I propose the '1:9' rule for setting up the threshold. However, the algorithm is not sensitive to this choice and any proportion from 3:7 to 1:19 are applicable.

6. *Set the transition probabilities* p_{12}, p_{21} *according to the ratio of the number of state transitions to the total number of transitions in the* $\{N_f^k\}$ *sequence:*

$$p_{12} = \frac{1}{K}\sum_{k=1}^{K-1} 1\left(N_f^k > N_f^* \wedge N_f^{k+1} < N_f^*\right), \qquad p_{11} = 1 - p_{12}, \tag{3.2}$$

$$p_{21} = \frac{1}{K}\sum_{k=1}^{K-1} 1\left(N_f^k < N_f^* \wedge N_f^{k+1} > N_f^*\right), \qquad p_{22} = 1 - p_{21}, \tag{3.3}$$

   where N_f^k *denotes the size of frame* k, K *denotes the total number of frames (that equals to the number of transitions) in the trace and* $1(\cdot)$ *is the indicator function, which equals to one if the argument is true, otherwise it equals to zero.*

7. *Calculate the frame size for both states.*
   The model generates N_{f1} cells in state 1, where N_{f1} is the sample mean of the upper quantile and N_{f2} cells in state 2:

$$N_{f2} = T_{if} R \frac{p_{11} + p_{22}}{p_{11}} - N_{f1} \frac{p_{22}}{p_{11}}. \tag{3.4}$$

The unit time of the Markov model is determined by the scheduling model, i.e. the next component of my hierarchical model.

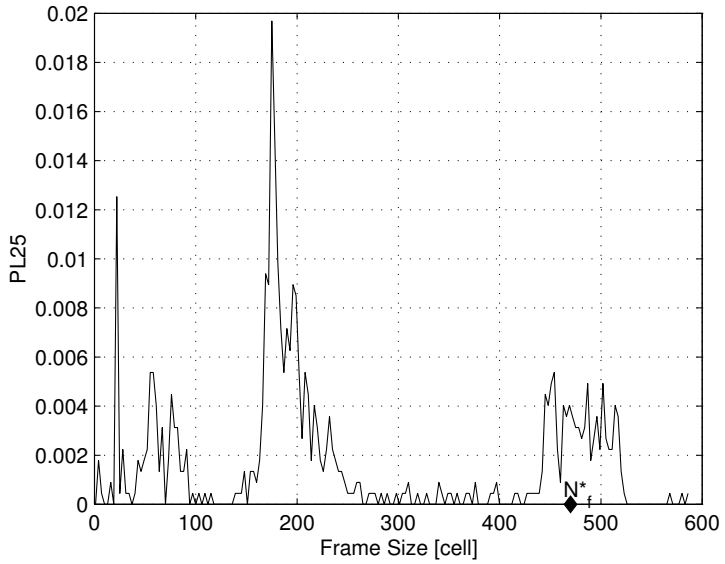

Figure 3.10: Probability mass function of frame size for PL25

3.3.2 Parameterization of the Scheduling Model

An application running in a non-real-time operating system can not transmit data into the network on a constant rate, due to the contention among concurrent applications [77]. The Gaussian nature of the scheduling process is well pronounced, by plotting the PMF of the frame interarrival time. Apart from the empirical PMF, the probability density function of the modeling Gaussian process is also depicted in Figure 3.11.

The parameters of a Gauss model can be set by fitting its probability density function to the empirical PMF. Denote T_{if} the normal variable which is my model for the frame interarrival time. The parameters of the Gauss model $G(m, \sigma)$, i.e. the sample mean and sample variance of T_{if} should be determined by taking the $\{T_{if}^k\}$ sequence resulted from the

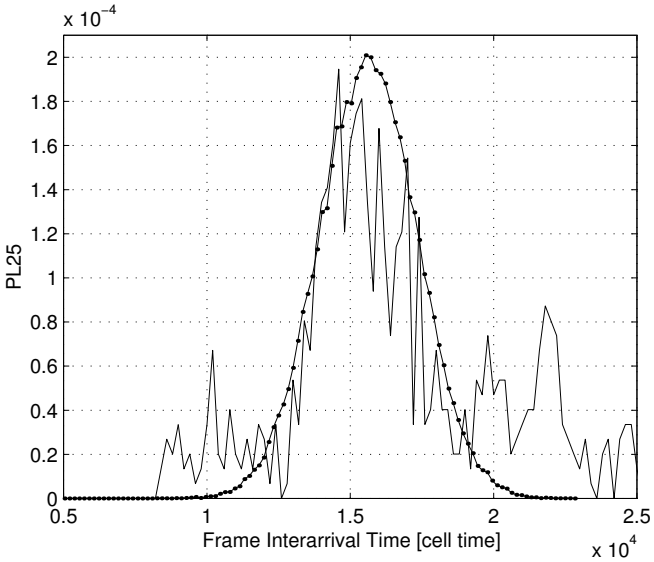

Figure 3.11: Probability mass function of frame interarrival times for PL25 (solid line) and the probability density function of the Gauss model (dotted line)

first three analysis steps in the previous section and applying:

$$m = \frac{1}{K}\sum_{k=1}^{K} T_{if}^{k}; \quad \sigma^2 = \frac{1}{K-1}\sum_{k=1}^{K}(T_{if}^{k} - m)^2, \tag{3.5}$$

where K denotes the total number of frames in the trace and T_{if}^{k} represents the interarrival time of frame k.

3.3.3 Parameterization of the Encapsulation Model

Based on the regular arrival of cells within packets, and packets within frames, I established a deterministic state machine for modeling the encapsulation. The parameters of this state machine can be determined by the following algorithm:

1. *Perform the first three steps of the algorithm described in Section 3.3.1.*

2. *Produce a packet stream* $\{N_p; T_{ip}\}$ *by filtering out silence periods, which are shorter than* $\tilde{T}_{spmin}$ *and joining individual cells to form a packet between the identified packet silence periods.*

3. *Determine* N_{pmax} *from the* $\{N_p\}$ *sequence.*
   For the PL25 sequence $N_{pmax} = 171$ cells.

4. *Finally, make a linear regression analysis for determining the α and β constants in Equation (3.1).*
   I have used the Matlab [104] program for performing linear regression between the $\mathbf{y} = \{N_p^{k+1}\}$ and $\mathbf{x} = \{T_{sp}^k\}$ vectors by finding the coefficients of a first degree polynomial $p(\mathbf{x})$ that fits the data $p(\mathbf{x}) \approx \mathbf{y}$ in the least squares sense[2]. The value of α is 18,5 and β is 323 for the PL25 sequence (see Figure 3.8).

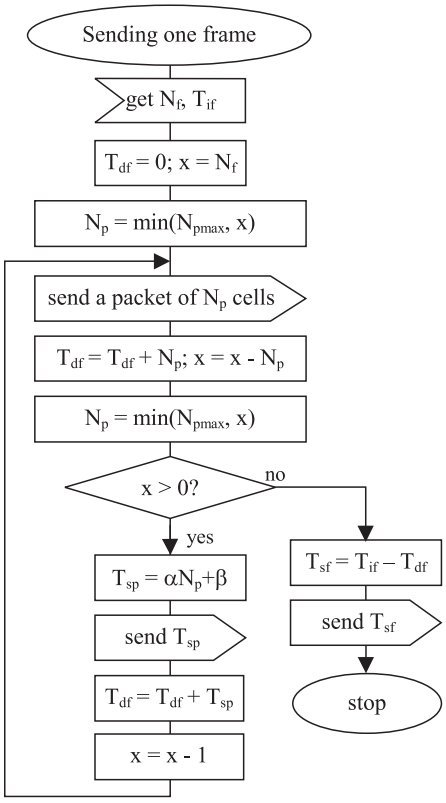

Figure 3.12: Deterministic model for the encapsulation of a video frame

The state machine reads a sequence of $\{N_f, T_{if}\}$ pairs as input. Figure 3.12 depicts the flowchart of sending one frame with this state machine. The parameters α, β and N_{pmax} are constant and can be set using the presented algorithm. N_p, T_{df}, T_{sf}, T_{sp}, and x are local variables of the state machine whose value is computed in runtime. The encapsulation model generates a cell interarrival time sequence, i.e. a synthetic traffic stream.

[2]Matlab uses the Vandermonde matrix for polynomial regression, but the method of regression is not critical from the algorithms' point of view.

3.4 Validation of the Hierarchical Model

I have validated the proposed hierarchical traffic model in terms of estimating the maximum queue length and the cell loss rate. The following tests have been performed on 27 measured video sequences and the corresponding synthetic traces generated by my hierarchical model.

3.4.1 Queue Length Estimation

I used the Leaky Bucket Algorithm for characterizing the queue length for the original and synthetic traffic traces. I use the metrics and notation introduced in Chapter 2. Results are plotted for the PL25 and GT25 traces in Figures 3.13 and 3.14.

We can see in these figures that my model very accurately captures the breaking points as well as the slope of the LB curve. The peak-end of the LB curves are fully matched, while the LB distance is relatively high toward the mean-end. The latter represents the scene level, which was not targeted by my hierarchical model.

Therefore my source model can be used for replacing the real VBR traffic source and in this way emulating the buffer build up in the ATM switch.

3.4.2 Cell Loss Estimation

It was presented in Section 2.5.4 that the LBA can be used for cell loss estimation. In order to show the performance of buffering, the CPDF of queue length is presented for both the PL25 traffic trace and the model-generated traffic in Figure 3.15.

This figure shows that the tail of the CPDF is close for the original and synthetic PL25 sequences in case of different service rates. In other words, my model can reproduce the CPDF of queue length curve of the investigated VBR video sources and thus it can be used for cell loss estimation. Note, that these results confirm that my model is able to capture those characteristics of the traffic which are important from the queuing point of view. I infer the general validity of this modeling technique based on our study of a relatively large number of video sequences and performance settings.

Is this Model Platform Dependent?

An interesting question is, whether the presented model building technique is applicable for other multimedia platforms. One of the key issues in my modeling concept was that different burst levels in the traffic correspond to different stages of the traffic generation process within the source host. Thus I constructed a hierarchical model that consists of three independent model stages for the encoding, scheduling and encapsulation, respectively. I designed and parameterized these models based on the analysis of a particular multimedia platform. However, most of the common VBR encoders, operating systems and hardware/software components (constituting the protocols stack) influence the traffic generation process in a similar way.

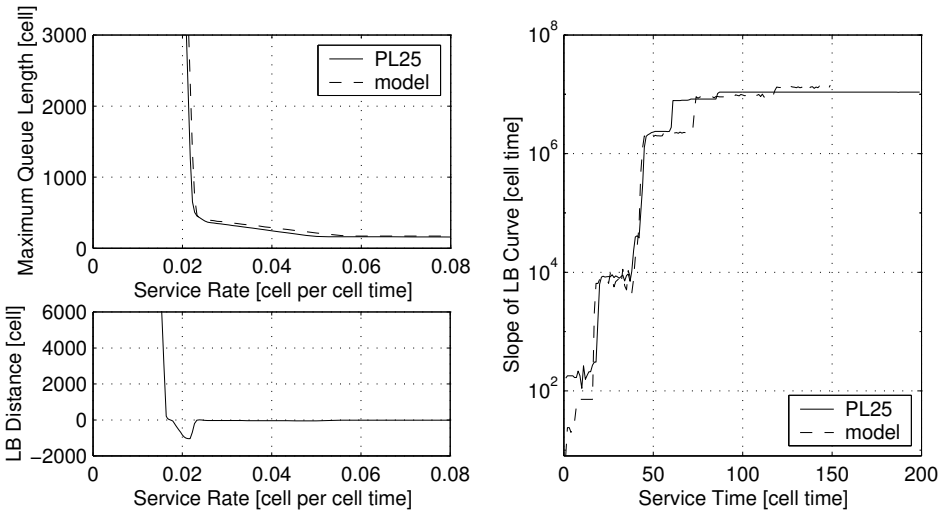

Figure 3.13: Leaky Bucket Analysis of the measured and synthetic PL25 traffic trace (plotted by solid and dashed lines, respectively)

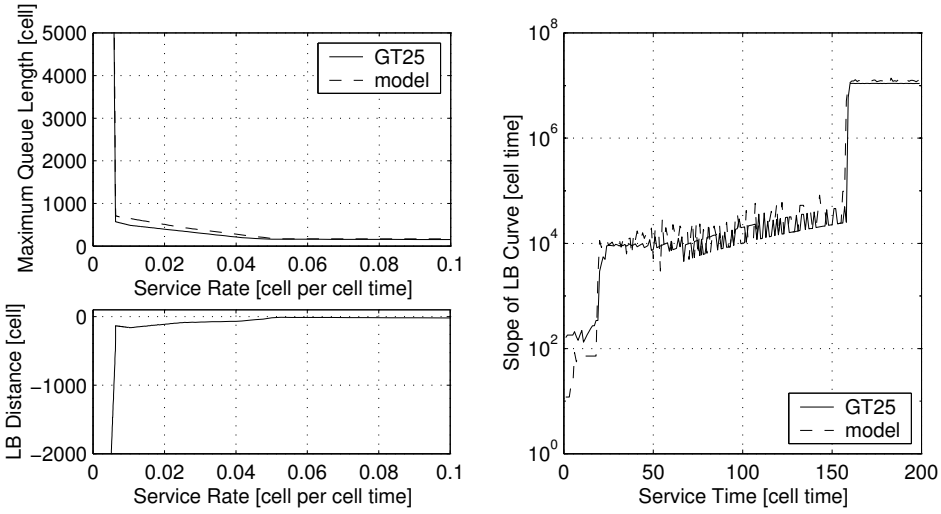

Figure 3.14: Leaky Bucket Analysis of the measured and synthetic GT25 traffic trace (plotted by solid and dashed lines, respectively)

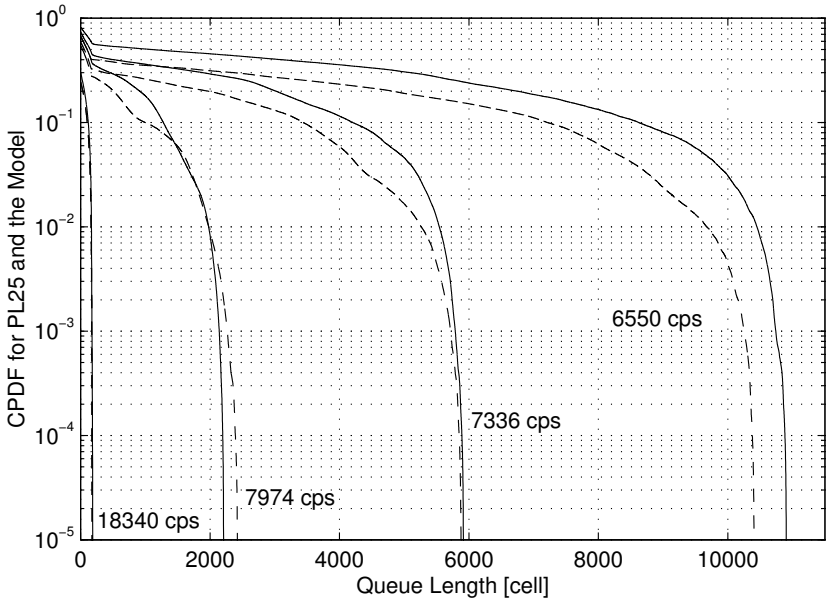

Figure 3.15: Complementary distribution function of queue length for the measured and synthetic PL25 traffic trace (plotted by solid and dashed lines, respectively) in case of different service rates

Therefore one can repeat the black box analysis steps for the traffic generated by the new platform, and modify the encoding, scheduling and encapsulation models accordingly. For instance, the multimedia terminal had UDP/IP/ATM protocol stack for a set of the traffic traces, which I have analyzed. Although the payload size of ATM cells (48 bytes) differs from that of Ethernet packets (1500 bytes), the 8192 bytes large Maximum Transfer Units generated by the multimedia application [J1] have to be divided into several segments in both cases. This affects probably only the N_p, N_{pmax}, α and β parameters of the encapsulation model, and the rest of my hierarchical model does not need to be changed.

Therefore I think that my model building technique can be successfully adapted to other coding schemes (e.g. M-JPEG or MPEG), other operating systems and other network protocol stacks (e.g. IP over Ethernet).

3.5 Summary

This chapter proposed a hierarchical source model based on the white box modeling concept. This model can synthesize the traffic of a VBR traffic source by imitating the operation of VBR encoding, process scheduling and protocol encapsulation in a multimedia workstation.

The main characteristics of the multimedia traffic have been evaluated in Section 3.2 by performing traffic intensity analysis and silence period analysis on different burst scales ranging from cell level to scene level. This black box analysis has emphasized that there are four burst levels in the multimedia traffic – scene, video frame, packet and cell levels –, which appear periodically in a regular manner. Another observation has been that, while the scene and video frame level characteristics strongly depend on the video content, the packet and cell level characteristics (such as packet size, packet interarrival time, etc.) are unchanged. This section also highlighted that the frame interarrival time varies and its mean is determined by the video frame rate setting. Moreover, a linear relationship has been found between the packet silence period and the size of packet generated after that period.

The observations and numerical results of the black box analysis have been extended based on the knowledge of the multimedia terminal's internal operation resulting in a hierarchical model. This model consists of three model stages according to the terminal's three independent traffic generation procedures, i.e. encoding, scheduling and encapsulation. The video encoding procedure has been modeled with a two-state Markov process, while the scheduling of application transfer units has been modeled with a simple Gaussian process and a deterministic model has imitated the data encapsulation in the protocol stack. Apart from describing the hierarchical model, Section 3.3 has also proposed algorithms for setting the parameters of the three models based on a measured trace from the investigated traffic source.

The model has been validated by LBA and cell loss analysis in Section 3.4. The results have shown that the queuing performance of original traces (in terms of maximum queue length and estimated cell loss) is successfully reproduced by the synthetic traffic of our model.

Applications

A possible application of our model is to emulate the behavior of a particular VBR source and reproduce its traffic. In this case, the only variable input parameter of our model is the $\{N_f^k\}$ sequence, which represents the scene level characteristics of particular video content, while other parameters are fixed. Such emulated sources can be utilized in large simulation or measurement scenarios. Alternatively, one can determine the required ATM traffic contract parameters for a given source, perform multiplexing analysis using real and emulated sources, or predict cell loss rate using this model. Network dimensioning and designing control methods (e.g. Call Admission Control) are further applications of this model.

Advantages

The main advantages of the proposed model building technique compared to previous models are the following:

- We map the results of black box traffic analysis to our knowledge of the traffic generation procedure. This white box approach yields a model which behavior is very close to that of the real source.
- We present a relatively simple algorithm for traffic generation where the parameters of our model can be easily set based on measurements.
- The model targets to capture directly the queuing behavior (i.e. leaky bucket curve, queue length and cell loss) of the real source. We avoid the complexity due to fitting different statistical characteristics and investigating a rather complex queuing model.
- We restricted the use of statistical assumptions about the traffic and set our model parameters directly from measurements. The modeling concept is verified by comparing queuing performance of the synthetic and captured traffic traces.
- The proposed white box model represents a good compromise between stochastic source models and measurement-based source emulation. Stochastic models are relatively simple to parameterize, but the synthesized traffic is very far from the original and it usually captures only the scene level characteristics. Measurements yield very good accuracy, but they are too expensive and require the storage and play back of a huge amount of data.

Chapter 4

Resource Management for Multimedia Services

4.1 Introduction

The previous chapters presented methods for characterizing the resource demand and designing source model for multimedia traffic. In this chapter I focus on the management of network resources for multimedia services.

The key issue in resource management is how to assign the proper amount of network resources (such as bandwidth, buffer, scheduling priority, drop precedence, etc.) to the traffic streams in order to achieve the targeted quality of service. This topic is addressed in both the Internet and ATM communities; especially in case of multimedia applications.

The usage of *signaling protocols* for reserving the resources before the beginning of data transfer is widespread (see [13, 14, 36, 15, 16, 17, T5]). One part of resource reservation protocols, such as the Stream Protocol family (ST [62], ST-II [14] and ST2+ [63]), sets up permanent, "hard", reservation states in routers. Main drawbacks of *hard-state* reservation are the complex failure recovery and limited robustness [16]. Other protocols use "soft states" (e.g. the ReSerVation Protocol RSVP [57], or Dynamic Reservation Protocol DRP [16]), which are released if no refresh message arrives within a certain period of time. Some *soft-state* protocols are "lightweight" regarding the complexity of allocation scheme and the design of protocol messages (e.g. YESSIR [17], Boomerang [C11, T5]). In a further reservation concept, known as Ticket protocol, the amount of already reserved resources is estimated by counting the refresh messages [18].

Reservation protocols can be differentiated based on their strategy for reservation setup as well. Earlier schemes (such as CSS7 [78] and q.2931 [13]) basically assume that the destination terminal is always available, thus reservation shall be started in the network, hop by hop. In spite of this *sender-oriented* approach, some newer reservation schemes (e.g. RSVP) also consider multicast sessions, where the destinations (or receivers) can gradually join to

a reservation session. *Receiver-oriented* protocols also allow the destination host to determine the resources for the reservation or reject it. From all these and many other IP resource reservation protocols, only RSVP could attract real industrial interest, although the Integrated Services (IntServ) architecture [36] is not hardly coupled to RSVP and it can cooperate with other mechanisms as well [64, 65]. A common property of these reservation protocols is that the amount of resources to be reserved is specified in simple terms, typically with the peak bit rate or with the parameters of a leaky bucket (or token bucket) [J3].

Another fundamental way of resource management is to assign resources to the traffic during the forwarding of packets instead of preliminary reservation. The Differentiated Services (DiffServ) architecture [69] of the Internet Engineering Task Force (IETF) describes the basic concept. While the IntServ architecture provisions resources per hop and per end-to-end traffic flow, the traffic is ordered into a few traffic classes (called Behaviour Aggregate) in DiffServ, which traffic classes have pre-established resources per network domain. Instead of explicit signaling, the *QoS demand is marked in the packet header* [70]. There are several extension proposed to this basic concept. Some of them would extend the scope and interpretation of the marking [19, 20, 21], others would combine signaling and DiffServ [64] or propose an overlay architecture that dynamically provisions the resources for the traffic classes using a *centralized resource manager* (often called oracle or Bandwidth Broker [31]). This latter approach is similar to other methods of separating the service management and data forwarding layers, especially in case of distributed multimedia services [55, 54, J3, C16].

The critical issue is to determine the scope and scalability of these alternatives, and it requires a *performance evaluation framework*. I describe a network scenario and summarize traditional and new performance metrics for such a framework in Section 4.2. I evaluate and compare the performance of different signaling and centralized resource reservation schemes in Section 4.3. Finally, I analyze the service specific information and identify additional parameters – namely the downgrade vector and the preference function – for the reservation protocol, which parameters can enhance the efficiency of resource reservation and evaluate different schemes that utilize such information in 4.4.

4.2 Performance Evaluation Framework

4.2.1 Network Scenario

I consider a network of N nodes in which *host A* initiates I resource (bandwidth) reservation sessions toward *host B* on a fixed route (see 4.1). Moreover, each node handles *background reservation* messages originated from x background hosts. The background requests have a different route than N1-NN and each group of background sources loads only one node. Furthermore, background requests can not be downgraded only blocked.

I distinguish two traffic classes, namely reservations are performed for *Premium traffic*,

while no reservation is made for *Best Effort traffic*. The network nodes and host B can also produce signaling messages as a consequence of receiving a message. A node may handle several atomic messages regarding the same session. Thus I introduce the $k = 1, 2, \ldots, K^{n,i}$ index that specifies the sequence number of a message on node n for session i. Although the atomic message primitives can have many different interpretation (e.g. reserve, probe, release, renegotiate) each of these can be described with a parameter triple:

$$\{b_k^{n,i}; \alpha_k^{n,i}; \sigma_k^{n,i}\} \qquad b_k^{n,i} \in \{1, 2, \ldots, b_{max}\}; \quad \alpha_k^{n,i}, \sigma_k^{n,i} \in \mathbb{R}^+ \qquad (4.1)$$
$$i = 1, 2, \ldots, I$$
$$n = 0, 1, \ldots, N+1,$$

where $b_k^{n,i}$ denotes the reserved bandwidth at node n after processing the message regarding the reservation session i; $\alpha_k^{n,i}$ is the arriving time and $\sigma_k^{n,i}$ is the departure time of the message (see Figure 4.1). The maximum bandwidth, i.e. the full link rate is denoted by b_{max}. Messages from Host A and Host B are indexed with $n = 0$ and $n = N+1$, respectively.

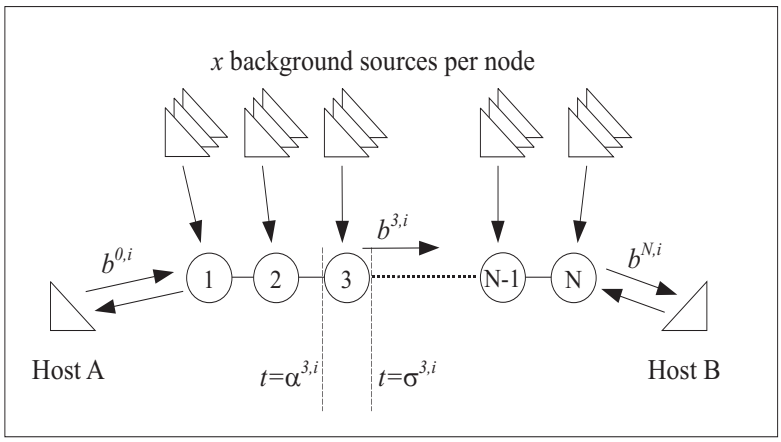

Figure 4.1: The investigated network scenario

I assume *downgrading*, i.e. that the amount of demanded bandwidth is gradually decreased during the negotiation process in a reservation session, therefore subsequent signaling messages request less and less:

$$b_k^{n,i} > b_l^{n,i} \quad \forall \quad k < l; \qquad k, l = 1, 2, \ldots, K^{n,i}. \qquad (4.2)$$

The nodes have a fix message handling time denoted by t_k^n $(n = 1, 2, \ldots, N)$. Peak rate allocation is assumed in network nodes with no priority scheduling. Message handling time for host A and host B are denoted by t_k^0 and t_k^{N+1}, respectively. The propagation delay of messages is neglected. Peak rate allocation is assumed in the nodes with no priority scheduling. Both the foreground and background sources generate requests according to a Poisson process with arrival rate λ and mean holding time μ. Poisson arrival and exponentially distributed service time has been shown to be reasonable approximation for measured

telephony traffic and this assumption has been used both for modeling personal communication services [58, 59, 52] and integrated voice, video and data calls [60, 53]. There is only one requested network resource, namely the required bandwidth (i.e. Peak Cell Rate) considered in this simplified scenario and the others, such as buffer size and number of ports in switching elements are neglected.

4.2.2 Performance Metrics

This section introduces performance metrics for benchmarking resource reservation protocols using the presented notations. On one hand, the local impact of signaling can be characterized using *per hop performance metrics*. On the other hand, *per reservation performance metrics* can characterize the entire resource reservation setup process.

Per Hop Performance Metrics

Processing signaling messages loads the router. This effect can be quantified by different per hop performance metrics, such as (i) the message handling time, (ii) the memory consumption of the signaling message or (iii) the number of code lines in the corresponding handler software [32]. The per hop *message handling time* can be expressed using the arrival and departure time of a message to/from the router. Using the aforementioned notation the average message handling time is:

$$t_k^n = \frac{1}{I}\sum_{i=1}^{I}(\sigma_k^{n,i} - \alpha_k^{n,i}), \qquad n = 1, 2, \ldots, N; \quad k = 1, 2, \ldots, K^{n,i}. \tag{4.3}$$

Extending the list of per hop performance metrics, I propose the *Signaling Intensity* metric for quantifying the impact of the reservation protocol on the signaling handling node:

$$J^n = \frac{1}{I}\sum_{i=1}^{I} K^{n,i}, \qquad n = 1, 2, \ldots, N. \tag{4.4}$$

The signaling intensity gives the total amount of signaling messages handled by a node divided by the total number of reservation sessions. Although Signaling Intensity is more a design issue that puts requirement on the router's control plane performance, it also has some impact on reservation setup time. The unique feature of this metric is, that it takes the number of atomic messages per reservation trial (i.e. $K^{n,i}$) into account, and in this way characterizes the effectiveness of the protocol.

Apart from load, it is important to quantify the greediness of the reservation algorithm, because it has an indirect impact on the Best Effort traffic. In particular, the router may reject (drop or delay) traffic from the Best Effort class, if there is a temporary reservation for Premium traffic. If this reservation is partially released at a later stage, the dropping of Best Effort traffic was exaggerated. I propose the *Over-Provisioning Factor*, or for short

the OPF, for characterizing the impact of the greediness of reservation protocol on the Best Effort traffic. The OPF gives the amount of excess resources, which are allocated and later released (e.g. due to the lack of resources on subsequent nodes) before the end of the reservation session:

$$O^n = \frac{1}{I}\sum_{i=1}^{I} O^{n,i} = \frac{1}{I}\sum_{i=1}^{I}\sum_{k=1}^{K^{n,i}-2}(b_k^{n,i} - b_{k+1}^{n,i})(\alpha_{k+1}^{n,i} - \alpha_k^{n,i}), \qquad n = 1, 2, \ldots, N. \tag{4.5}$$

The OPF also describes how competitive is the resource reservation process in the Premium class. The more greedy a reservation scheme is, the less chance other request have for being accepted by a specific network node. The main difference between the message handling time t_k^n and OPF metrics is, that the former characterizes the capabilities of a single node independently from the network load, while the latter describes the interaction of a reservation scheme and the resources in a node in case of a certain load situation. The OPF provides more information, because it considers the timing and resource aspects together.

Per Reservation Metrics

Traditional per reservation metrics are (i) the blocking probability, (ii) the reservation setup time, (iii) the signaling overhead, (iv) the number of reservation trials and (v) different fairness metrics. These metrics are of different significance in relation to overall network performance and user satisfaction. Blocking probability and reservation setup time are the most important, basic metrics of signaling performance [58]. A longer reservation setup time or higher probability of blocking is less acceptable from the user's perspective. Although these factors are determined by the network load, a more effective reservation protocol (e.g. which works faster) results less blocking probability [C9].

The *reservation setup time* quantifies the time that host A shall wait between issuing a request and receiving the corresponding acknowledgement from host B:

$$T^i = \alpha_{K^{0,i}}^{0,i} - \sigma_0^{0,i}. \tag{4.6}$$

Signaling overhead quantifies the bandwidth, which is taken by signaling messages from the data traffic. It is a common belief that this factor is critical, but our results disprove it [C11, C10]. Although the size of subsequent signaling messages may differ, higher signaling intensity results larger signaling overhead. In spite of the previous metrics, the *number of reservation trials and fairness* metrics can directly characterize the effectiveness of the resource reservation protocol. The average number of reservation trials can be given as:

$$L = \frac{1}{I}\sum_{i=1}^{I} K^{0,i}. \tag{4.7}$$

By comparing this expression with (4.4), it is noticeable that L can be considered as the signaling intensity for host A. As a consequence, it do not characterize the load on network nodes unlike J^n.

Based on my new per hop metrics, I propose the *Over-Provisioning Balance* which characterizes how much the over-provisioning factor depends on the topological location of a node:

$$\theta_O = \max_{1 \leq n \leq N} \left| \frac{O^n - \frac{1}{N} \sum_n O^n}{\frac{1}{N} \sum_n O^n} \right|. \tag{4.8}$$

This is a maximum, relative error type of metric which equals to zero in the best case. Similarly, I propose the *Signaling Intensity Balance* for characterizing how much dependent the signaling intensity is on the topological location of a node:

$$\theta_J = \max_{1 \leq n \leq N} \left| \frac{J^n - \frac{1}{N} \sum_n J^n}{\frac{1}{N} \sum_n J^n} \right|. \tag{4.9}$$

4.2.3 Application

Pudding's probe is eating. Thus I have applied the proposed performance metrics for analyzing different resource reservation schemes in both a simulation system and laboratory measurements. [J3, J2, C14, C11, C10, C9, C8]

The results presented in this chapter are based on performance studies using a *discrete event simulator* [T3] and an OPNET simulation module [72] that emulates the necessary signaling and resource contention in a multi-rate network environment. Numerical results were generated for the aforementioned network scenario. The confidence level of each simulation run was set according to model I =100,000 to 1,000,000 foreground reservation sessions. Moreover, we have measured and calculated some of these metrics on a *test network* consisting of Linux based router prototypes and real router products [C10, C8].

During these tests I found that the over-provisioning factor is a proper performance metrics for capturing the local and topology dependent impact of reservation on Best Effort traffic classes and that the signaling intensity can characterize the effect of reservation signaling on the network nodes. Some of my results are summarized in the following sections.

4.3 Resource Reservation Schemes

The introduction provided a summary of distributed and centralized resource management techniques, the former based on signaling protocols and the latter based on a central resource manager. This section analyzes these approaches on concrete examples, and presents a new allocation scheme, which aims for minimizing the Over-Provisioning Factor.

4.3.1 Investigated Schemes

I have defined a new resource reservation scheme, called *Hybrid Allocation* that aims for minimizing the OPF. This scheme is based on intelligent network nodes, which can interpret

a downgrade vector[1] and send signaling messages both upstream and downstream in order to immediately release the unnecessary resources. Furthermore, by keeping the basic features of fundamental reservation techniques [13, 15, 31, C18] and modifying others, I have also defined a sender-oriented and a receiver-oriented resource reservation scheme, called *Forward Allocation* and *Backward Allocation*, respectively [J3]. The reservation strategy of these schemes is the following:

- The *Forward Allocation* scheme reserves network resources upstream, from host A to host B. This corresponds to a sender-oriented approach, such as ITU Q.2931.
- In the *Backward Allocation* scheme, host A just informs host B about his resource demand and than host B makes the resource reservation, downstream. This scheme can be attributed to receiver-oriented, like RSVP.
- In the *Hybrid Allocation* scheme, each node receives a reservation setup message from a central node. This strategy is somewhat similar to centralized schemes, such as the SIGNE [C16] or Bandwidth Broker concept [31].

Forward Allocation

In case of Forward Allocation the protocol tries to grab the resources in the network before reaching the destination host:

1. Host A generates a bandwidth request and sends it into the network.
2. Each network node allocates the requested bandwidth if it is available. If only less bandwidth is available, it reserves that amount and downgrades the request to this new value. Therefore the *downgrading* is independent from the downgrade vector (see Section 4.3.2).
3. When the request message arrives to host B, it checks whether the implied bandwidth level is meaningful for the application and downgrades it according to the downgrade vector if necessary. If the bandwidth request is downgraded to zero, the reservation setup is failed. If not, then the terminal resources are allocated in host B.
4. Host B sends back a confirmation message into the network in order to confirm the reserved resources for the session and to release the extra bandwidth allocations along the link.
5. When this message reaches the host A, the reservation setup is finished and the application can be started.

[1] *I define the downgrade vector in Section 4.3.2.*

Backward Allocation

In spite of the previous scheme, host B is considered as the most critical resource of the teleservice. Moreover, and very simple nodes are assumed which are not able to downgrade a request (like in RSVP).

1. Host A sends an informative request message to host B asking for the reservation of guaranteed service.
2. The nodes forward the message without making reservation and host B checks and allocates its available terminal resources. It selects a quality level according to its allocated terminal resources and generates a reservation request based on that.
3. The network nodes try to allocate the requested bandwidth. If all nodes along the path can allocate it and the message reaches host A, the reservation is set up and the application can be started. However, if the available bandwidth on any node is less than the requested, that node sends back a reject message to host B.
4. Host B downgrades the request according to the downgrade vector, releases the extra terminal resources and sends a new, downgraded request. The procedure goes on from step 3 until the reservation is established or failed (i.e. the requested bandwidth reaches zero).

Hybrid Allocation

The aim of this allocation scheme is to *minimize the over-provisioning factor* in the network and equalize the signaling intensity among the nodes. Thus intelligent network nodes are assumed, which can interpret a downgrade vector and send signaling messages both upstream and downstream, in order to immediately release the unnecessary resources. Another difference to the previous schemes is that a *Central Resource Manager* (CRM) is involved in the reservation setup, which has information about the teleservice[2]. In this simplified scenario, this service-specific information is equivalent to the downgrade vector.

1. Firstly, host A sends a session request to CRM using an application-layer signaling protocol.
2. The CRM checks the policy and calculates the downgrade vector which will be sent to the first node (N1).
3. The nodes check their resources and allocate the requested bandwidth if it is available. If not, they downgrade the request according to the vector; forward the new, downgraded request to the next node and simultaneously send back a release message for the extra reserved bandwidth to the previous nodes.

[2] Apart from the Bandwidth Broker concept, the idea of central resource manager is reflected in the Policy Based Networking framework [71] where the service specific information is expressed in a policy.

4. When the last node (NN) is ready with the reservation, it informs the CRM about the final bandwidth value. The CRM checks the resulted session using the policy information and it sends an application-layer message to both host A and host B.

5. When this message reaches the hosts, the reservation setup is finished and the application can be started.

It is apparent that the proposed Hybrid allocation scheme requires more complex network nodes. However, the numerical results will show that it yields more economic usage of the link capacity.

4.3.2 Numerical Results

I have compared the performance of these schemes using the aforementioned performance framework and network scenario [J3]. This section summarizes the main results.

Movie on Demand Service

In [J3] I describe a simple Movie on Demand service and a central resource management system, called SIGNE. In order to avoid the replication of that work, I summarize here only the main parameters that I used in the simulation study (see Table 4.1).

Table 4.1: Parameters for the movie on demand service

Parameter	**Notion**	**Value**	**Comment**
link rate	b_{max}	155.2 Mbps	
number of foreground sessions	I	$10^6 - 10^7$	
number of nodes	N	10	
signaling handling time for hosts	t^0, t^{N+1}	100 ms	
signaling handling time for nodes	t^n	10 ms	$n = 1, 2, \ldots, 10$
reservation arrival rate	λ	$0.001\frac{1}{s}$	Poisson process
reservation holding time	$\frac{1}{\mu}$	100 s	Poisson process

The mapping between user quality settings and the resource demand is described in Section 4.4. Here is only the downgrade vector is given that specifies the discrete levels of teleservice resources (i.e. bandwidth in this case) which are meaningful for the application:

$$\mathbf{d} = [248\ 244\ 242\ 241\ 104\ 100\ 98\ 97\ 68\ 64\ 62\ 61\ 32\ 28\ 26\ 25] * 64\ kbps. \tag{4.10}$$

Reservation Setup Time

The reservation setup time depends on the allocation scheme. It is a constant value for Forward and Hybrid Allocations:

$$T^i = t^{N+1,i} + 2\sum_{n=1}^{10} t^{n,i} = 300\ ms, \tag{4.11}$$

while it is changing for the Backward scheme proportionally to the network load. It is well pronounced in Figure 4.2, that the mean reservation setup time equals to that of the other schemes for moderate network load ($x = 50$), while a much longer time (up to factor four) is required in case of heavy congestion ($x \geq 500$), due to the repeated trials of the Backward scheme. The probability mass function of the reservation setup time is also depicted in four plots. The probability mass disperses as the congestion increases. There are discrete time values which occur more than others, according to the basic time parameters of the simulation model (such as switching time of 10 ms, terminal response time of 100 ms, etc.).

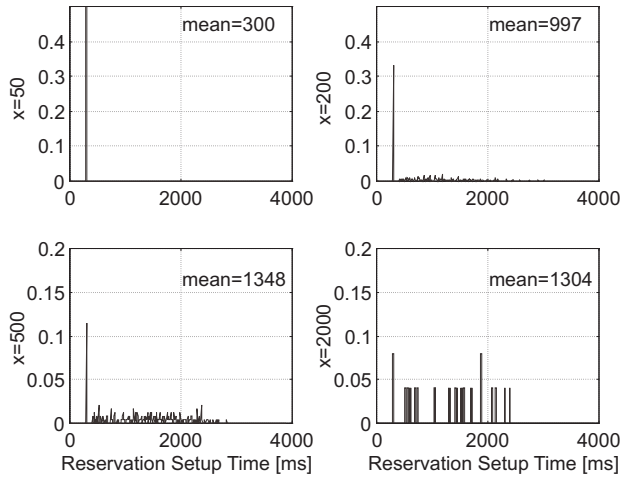

Figure 4.2: Probability mass function of reservation setup time for Backward Allocation

Blocking Probability of Foreground Sources

The blocking probability is calculated for different load situations, i.e. different number of background sources connected to each node. I defined blocking probability in this case as the ratio of completely unsuccessful reservations and sum of reservation trials. Since resource requests can be downgraded, reservations may be partially unsuccessful. By taking the amount of requested and resulted resource into account, further performance metrics could be defined (e.g. user satisfaction), which are somewhat similar to blocking probability.

Three network load regions can be differentiated in Figure 4.3; the regions of small, medium and large congestion. The blocking probability of the requests of foreground source is practically the same in case of Forward and Hybrid Allocation schemes in each region. The Backward Allocation performs worse in the region of small congestion ($0 < x < 90$), but it has better results than the other two allocation schemes in case of medium congestion level ($90 < x < 200$) and each curves coincide in the large congestion region ($x > 200$). Since a blocking level larger than 1% is hardly acceptable by quality oriented applications and each scheme has a larger probability of blocking in the second and third regions, one could conclude that the Forward and Hybrid schemes are somewhat better in the important cases.

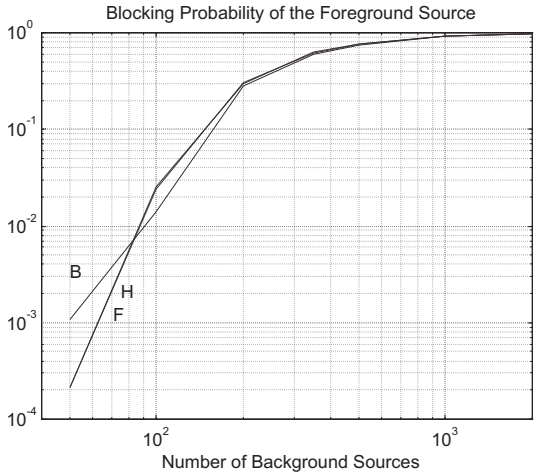

Figure 4.3: Blocking probability of the foreground source vs. the number of background sources

Blocking Probability of Background Sources

Each background source generates reservation requests for its node in an independent manner. Therefore the average blocking probability of background requests connected to the same node is evaluated in case of every node and different congestion situations. Results for the first and last nodes (N1, N10) are depicted in Figure 4.4. There is no noticeable difference between the blocking characteristics except the region of small congestion. The Forward and Hybrid allocation schemes performs very similarly, since both method downgrades the request according to the downgrade vector (although downgrading takes place in the destination terminal or in the network nodes, respectively). Interesting is the phenomena, that the Backward scheme has better results on the first node (N1) while higher blocking probabilities on the last node (N10). The explanation is that the foreground source loads the network nodes with atomic signaling messages in an unbalanced way in case of

Backward allocation. The nodes closer to the destination terminal should process much more messages than the first nodes due to the iterative allocation scheme, i.e. this is a location biased protocol using the terms presented in [61]. The over-provisioning and signaling intensity balance metrics also highlight this location bias (see later). The general conclusion regarding the results of blocking probability is that the three schemes have a very similar performance.

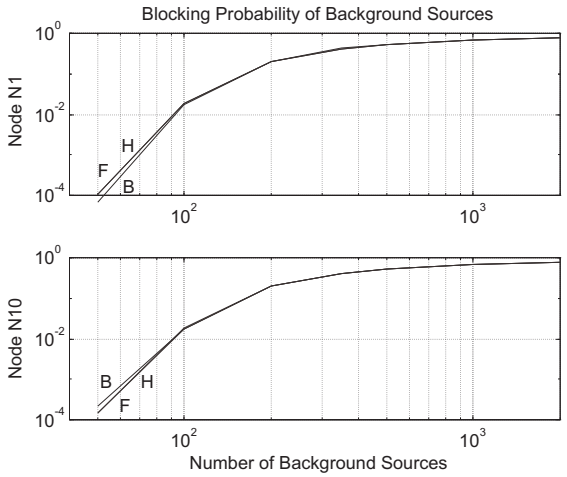

Figure 4.4: Blocking probability of background sources vs. the network load

Over-Provisioning Factor

In Figure 4.5, the OPF is drawn for the second and eighth node as a function of the network load. The practical interpretation of this graph is, how much amount of data could have been transferred through the network in an optimal case, if the greedy reservation protocol would have not blocked some resources. One could see previously, that the first and last nodes are special either form the Backward or the Hybrid Allocation's point of view. The Backward approach does not allow downgrading in the nodes, thus there is no over-provisioning on node N1. This location bias is the reason for taking N2 and N9 instead of the edge nodes. Although the curves get closer to each other in the large congestion region ($x > 200$), the Backward Allocation has clearly the worst and the Hybrid allocation has the best results. The declination above $x = 100$ background nodes can be explained by saturation in the network. It is well expressed by this figure, that the Hybrid scheme can use the knowledge of the downgrade vector very well (i.e. the CRM can provide useful information for reservation) and the immediate release messages yield less over-provisioning.

The location bias of the protocols can be characterized using the over-provisioning balance θ_O metric. Table 4.2 presents the mean OPF (i.e. $\langle O^n \rangle = \frac{1}{N} \sum_n O^n$) and θ_O for two

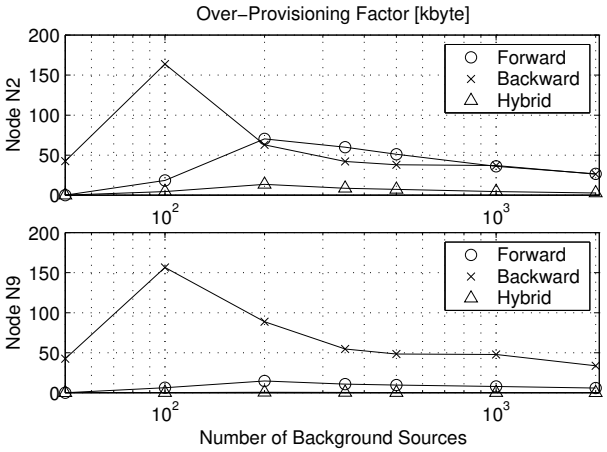

Figure 4.5: Average over-provisioning as a function of background source number

Table 4.2: Mean OPF and over-provisioning balance

Over-Provisioning				
Scheme	x=100		x=1000	
	$\langle O^n \rangle$	θ_O	$\langle O^n \rangle$	θ_O
Forward	11.790	0.613	15.721	1.288
Backward	157.035	0.043	42.356	0.270
Hybrid	1.950	1.327	1.499	2.033

load cases. Although the Hybrid scheme yields the least mean OPF in both cases, θ_O indicates the largest location bias for this scheme. These metrics show the opposite for the Backward Allocation, which has very big $\langle O^n \rangle$, but in a balanced manner.

Signaling Intensity

Since the signaling intensity is also dependent on the node's position in the network, results are depicted for the two edge nodes (N1, N10) in Figure 4.6. In case of the first node (N1), the number of signaling messages is the highest for the Hybrid scheme, but only in the medium congestion region ($90 < x < 200$), and each curve declines in case of high congestion. The reason of former observation is that the number and frequency of downgrading and reallocation actions increases when the load is higher in case of Hybrid Allocation, since the first node should process every reallocation message coming from the upper nodes besides the three basic messages (reserve, allocate and release). The latter phenomena is caused by the higher number of rejected reservation requests (i.e. shorter setup procedures) in case of higher congestion. In spite of the above plot, the Forward and Hybrid schemes result exactly the same characteristics for the last node (N10), while the

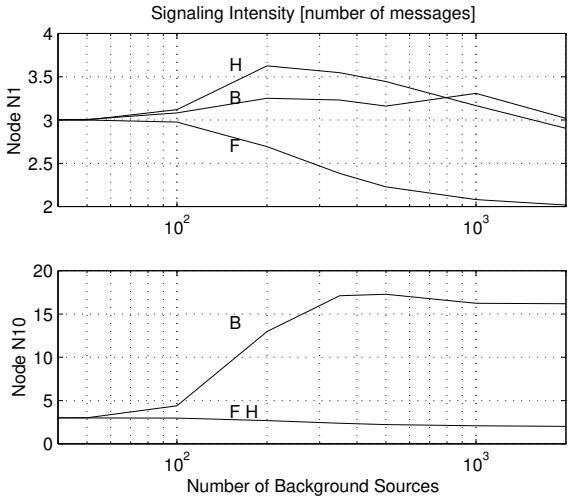

Figure 4.6: Average signaling intensity as a function of network load

Backward curve has a jump in the medium congestion region. This observation underlines the aforementioned fact, that the Backward allocation causes a very unbalanced load for the nodes from the point of view of blocking probability and signaling intensity. Although the Hybrid allocation requires somewhat more processing of atomic signaling messages in case of the first node, this difference is very moderate.

Table 4.3: Mean signaling intensity and signaling intensity balance

Signaling Intensity				
Scheme	x=100		x=1000	
	$\langle J^n \rangle$	θ_J	$\langle J^n \rangle$	θ_J
Forward	2.976	0	2.080	0
Backward	3.786	0.162	7.675	1.246
Hybrid	3.038	0.022	2.312	0.211

I have calculated the mean signaling intensity $\langle J^n \rangle = \frac{1}{N} \sum_n J^n$ and the signaling intensity balance θ_J for two load cases with $x = 100$ and $x = 1000$ active background sources. The results in Table 4.3 show that the Forward Allocation performs the best from this point of view. It loads the nodes in an absolutely balanced way and it requires the least number of atomic messages for a successful reservation (notice that $J^n = 2$ is the optimum with that scheme). The Backward scheme has the worst performance, with the largest location bias and almost 8 messages per successful reservation for $x = 1000$. This latter result can be explained by the increasing number of retries in case of high blocking.

4.3.3 Conclusions

Let us summarize the main results of this investigation, using the *Forward Allocation* scheme as baseline and comparing the performance of the other two schemes to that.

The *Backward Allocation* scheme has the same mean reservation setup time in case of small network load, but it requires three-four times longer setup time in case of large congestion. Moreover, the reservation setup time is not a fixed value, but it disperses according to the message handling time and topology of network nodes. The blocking probability of both foreground and background reservation requests is practically the same for the Forward and Backward schemes. The Backward Allocation yields somewhat less blocking on the nodes close to host A, and more blocking on nodes close to host B. This scheme produces higher signaling intensity than the Forwarding Allocation scheme. This scheme is the least location biased according to the over-provisioning balance metric, while the most location biased from the signaling intensity's point of view.

The *Hybrid Allocation* scheme has the same, fix reservation setup time as the Forward Allocation. Moreover, it results very similar blocking probability than the Forward scheme for both foreground and background reservation requests, in a balanced manner. It performs better regarding the over-provisioning factor and signaling intensity, but the over-provisioning and signaling intensity balance metrics are somewhat worse for the Hybrid Allocation scheme than for the Forward Allocation. On the other hand, if we compare the operation of these allocation schemes, it is obvious that the Hybrid scheme uses a more complicated protocol than the other two, and it requires a central resource manager and more intelligent network nodes. This approach tries to meet a trade-off between complexity and over-provisioning. The centralized resource management promotes optimization, which requires many iteration to achieve in the other cases.

Therefore the Hybrid Allocation has achieved its goal, because it yielded the least over-provisioning factor in the network producing a relatively low signaling intensity. This section has also illustrated that the OPF and signaling intensity metrics are useful, because they provide different information than the reservation setup time and blocking probability.

The proposed Hybrid allocation scheme yields more economic usage of the link capacity, but it requires more complex network nodes. Therefore the primary application of such a scheme can be in network-bottlenecks (e.g. wireless access network), where the bandwidth is more a scarce resource than the processing capacity.

4.4 Extensions Using Service Specific Information

This section investigates an intelligent allocation approach that utilizes the knowledge of multimedia application and user behavior for signaling based resource reservation.

First the service specific information is analyzed and a mapping is defined for expressing the demanded network resources as a function of the quality parameters of the application. Based on this mapping, different resource allocation schemes are proposed in which the signaling message carries service specific information, such as preference function and

Table 4.4: Quality parameters and levels of video

Video				
Quality Parameter	Resolution [Pixels]		Frame Rate [fps]	
Level	"Small"	"Large"	"Slow"	"Quick"
Value	192x144	384x288	10	25

Table 4.5: Quality parameters and levels of audio

Audio				
Quality Parameter	Audio Channel		Sampling Rate [kHz]	
Level	"Mono"	"Stereo"	"Low"	"High"
Value	1	2	24	48

downgrade vector.

The performance of these schemes is evaluated and benefits are highlighted in terms of reservation setup time, blocking probability, over-provisioning and signaling intensity based on simulation study [C14, J2].

4.4.1 Resource Vector of the Multimedia Application

Multimedia applications usually have several parameters, which influence the amount of network resources they require for the targeted QoS. For instance, by altering video coding parameters of a scalable MPEG application, its bandwidth requirement scales from 2.7, 3.085, 3.6, 4.32, 5.4, 7.2 up to 10.8 Mbps in discrete steps [56]. For describing a general case, let us denote the ith independent *quality parameter* of the application by q_i; the number of independent parameters by N and the number of quality levels each parameter can take by N_{Q_i}, i.e.,

$$q_i \in \mathcal{Q}_i = \{q_{i,1}, \ldots, q_{i,N_{Q_i}}\}, \quad i = 1, \ldots, N. \tag{4.12}$$

As an illustration, let us consider a video-phone service that has two media, video and audio, both having two independent quality parameters, such as resolution, frame rate, number of audio channels, and sampling rate (see Table 4.4, 4.5).

Denote K the number of independent *network resource types* (e.g. bandwidth, token bucket size, route, priority, ...) that applications are competing for. Let us also define a mapping function between the quality parameters and the network resource types ($\mathbb{R}$) in the following way:

$$r_j \in \mathcal{R}_j := \mathbb{R}_j\{\mathcal{Q}_1 \times \mathcal{Q}_2 \times \ldots \times \mathcal{Q}_N\}, \quad j = 1, \ldots, K, \tag{4.13}$$

Table 4.6: Mapping between quality settings and resource set of video-phone service

User Quality Levels for Video	**User Quality Levels for Audio**			
	low sampling rate		high sampling rate	
	mono	stereo	mono	stereo
small size simple frame rate	25	26	28	32
small size double frame rate	61	62	64	68
large size simple frame rate	97	98	100	104
large size double frame rate	241	242	244	248

where r_j is the value of the jth network resource type and $\mathcal{R}_j$ denotes the *resource set* which contains the possible resource values that the application may take from the jth resource type in case of different quality settings. The number of elements in each $\mathcal{R}_j$ is at most $L = \prod_{i=1}^{N} N_{Q_i}$, but it is much less in practical cases, because $\mathbb{R}_j$ does not define a one-to-one mapping and several combinations of the N parameters may result the same resource requirement, i.e. the same value.

The elements can be sorted in descending order yielding K *resource vectors*:

$$\mathbf{r}_k = [r_{k,1}, r_{k,2}, \ldots, r_{k,L}], \qquad r_{k,i} \geq r_{k,j} \Longleftrightarrow i < j, \quad i,j = 1,2,\ldots,L, \quad k = 1,2,\ldots,K. \tag{4.14}$$

In the previous example, the combinations of the two parameters and values yield four discrete quality levels for both media, which levels require different amount of network resources from the bearer network. These requirements are summed and given in the resource vector. The peak cell rate demand for each quality settings can be determined from Table 4.4-4.5, and are shown in Table 4.6.

Although these calculations are very simple, the resultant bandwidth requirements are in the range of a real audio and a coded video channel [51, C5]. Instead of a single peak, mean or minimum cell or bit rate (which are commonly used in standards), a set of bandwidth values are given which the application can exactly use for data transport. But are these values equally preferred by the service user?

4.4.2 Preference Function of the User

The $\mathbf{r}_k$ resource vectors represent one type of special *service specific information*, which is determined based on the capabilities of the multimedia application. Apart from this, another type of service specific information can be retrieved by considering how much the

user prefers the application's quality resulted by the different parameter settings. This can be expressed by a *preference index*, i.e. a positive number which is assigned to each combinations of the quality parameters. These preference values can be expressed by an N dimensional vector-scalar function, called *preference function*:

$$p = P(q_1, \ldots, q_N) \in \{0, 1, \ldots, N_P\}, \tag{4.15}$$

where N_P denotes the maximum preference index (e.g. 255).

One can assume in case of a video-phone service that users prefer the similar quality settings for audio and video. It happens rather rarely that a user requests a high quality video channel with mono, low sampling rate audio or vice-versa. Thus the 16 quality settings given in Table 4.6 are not equally pleasant to the users. We made a small Gallup poll among a group of colleagues in order to determine the preference function for the video-phone service. Results are given in Table 4.7.

Table 4.7: User's preference values

Video	**Audio**			
	worse	**bad**	**good**	**best**
worse	4	2	2	1
bad	3	4	2	2
good	3	3	6	4
best	2	3	5	8

The goal of user's preference function is three-fold in our case; (i) reservations are initiated with resource requests selected from the available settings through the distribution given by P, (ii) the user's decision is also emulated through P, if downgrading happens inside the network and (iii) the network node determines the next downgrading step using P in the most advanced allocation scheme.

4.4.3 Downgrading the Reservation Request

Both types of the service specific information, i.e. the resource vector and the user preference function, can be used for intelligent downgrading of reservation requests.

Decision on Downgrading

A subset of the elements of the $\mathbf{r}_k$ vector can be ordered into a *Downgrade Vector* $\mathbf{d}$, that specifies the discrete resource downgrading steps that the application may require from the

network, in a descending order:

$$\mathbf{d}_k = [d_{k,1}, d_{k,2}, \ldots, d_{k,M_k}] \qquad \begin{array}{l} d_{k,i} \in \mathcal{R}_k \\ i = 1, 2, \ldots, M_k \\ k = 1, 2, \ldots, K, \end{array} \tag{4.16}$$

where M_k denotes the number of elements in the downgrade vector belonging to the kth resource type. The $\mathbf{d}_k$ downgrade vector expresses information that network nodes shall consider during the reservation for the kth resource. The allocation of either more or less resource than the quantities described in $d_{k,i}$'s is just a waste of resource.

Another way for optimizing resource reservation can be achieved by taking the *preference function* into account. The main idea is that even if the network downgrades to those quality levels that bear with rational quality settings for the application, the user does not necessarily accept the established reservation, if that quality level is less preferred than another. Therefore a weight shall be assigned to each element of the downgrade vector using the P function:

$$\mathbf{w}_k = [w_{k,1}, w_{k,2}, \ldots, w_{k,M_k}] \qquad \begin{array}{l} w_{k,i} = P(\mathbb{R}_k^{-1}\{d_{k,i}\}) \\ i = 1, 2, \ldots, M_k \\ k = 1, 2, \ldots, K. \end{array} \tag{4.17}$$

If a request can not be admitted, the node considers both the downgrade vector and the preference function for downgrading the request and selects that element of the $\mathbf{w}_k$ whose weight is the largest.

Admission Control

The basic idea of our admission control algorithm is to reject the incoming reservation request if the available bandwidth in the node is less than the smallest element of the $\mathbf{d}$ vector, otherwise accept it and allocate bandwidth according to the largest element of the $\mathbf{d}$ vector that fits into the available capacity. If the available bandwidth at the jth node is denoted by f_j, then the admission control mechanism applied at the jth node can be expressed as follows:

$$\begin{array}{rcl} f_j < \min\limits_i(d_i) & \Longrightarrow & \text{reject request} \\ \exists\ d_i \le f_i & \Longrightarrow & b_j = \max\limits_i\{d_i | d_i \le f_j;\ i = 1, \ldots, M\}, \end{array} \tag{4.18}$$

where b_j denotes the bandwidth allocated on node j and I use the notion $d_i = d_{k,i}$ because only one resource type, bandwidth is concerned here. If downgrading happens, a reservation tear message is sent backward to the up-stream nodes in order to release excess reservations resulting from recent downgrading. The tear message is also propagated along the backward path.

User Revision

The preference function is also utilized for modeling the behaviour of the user. If downgrading happens inside the network, the user will decide whether the quality level of the established service is still acceptable or not. The preference function can characterize this decision, further referred as *User Revision* in this section.

4.4.4 Resource Allocation Schemes

This section presents two resource allocation schemes that uses the downgrade vector and the preference function in different ways. These schemes are compared to an ordinary resource allocation scheme, which is a simplification of RSVP [57, 15, C11]. For the sake of fair comparison, I assume that admission control (AC) takes place in nodes and host A initializes all reservation. Moreover, only one network resource is considered in the following, namely the bandwidth. Others like switching capacity and buffer size are excluded from the recent investigation.

Apart from the "baseline" allocation scheme (type 0), two enhanced allocation schemes are proposed. The first (type 1) uses only the downgrade vector, while the second (type 2) utilizes the preference function too, as service specific information. In spite of this principal difference, the reservation request is processed for both according to the following rules: (i) a reservation request is sent to the network with a downgrade vector; (ii) in the nodes along the path AC is performed as described previously; (iii) if downgrading happens, a tear message is sent backward along the path. The three basic allocation types are further modified by disabling the user revision function (type 0* and type 1*). The different allocation types are summarized in Table 4.8.

Allocation Type 0 This is the reference reservation model without any service specific information for processing reservations. The reservation request is launched into the network, and progresses forward until it finds a bottleneck link or reaches the destination. If in a certain node there is not enough available bandwidth, it stops the reservation message and sends a tear message to the previous nodes for releasing the total requested bandwidth. This tear message also contains the bottleneck bandwidth. Host A, receiving this tear message either tries a downgraded request considering both the preference function and the bottleneck capacity or considers the request to be blocked.

Allocation Type 0* This is a variant of Type 0, where the user always accepts an established reservation when it meets any of the elements of the downgrade vector (i.e. user revision is disabled).

Allocation Type 1 In this case, after a reservation request has been launched to the network host A either gets a release message, that means the failure of the reservation request, or an acknowledgment message with the reserved bandwidth. As downgrading could happen inside the network, the acknowledged bandwidth expresses the effective reservation.

Table 4.8: Resource allocation schemes

Allocation Type	**Service Specific Information**	**Downgrading in the Network**	**User Revision**
Type 0	No	No	Based on **P**
Type 0*	No	No	No
Type 1	**d**	Based on **d**	Based on **P**
Type 1*	**d**	Based on **d**	No
Type 2	**d and P**	**d and P**	No

Now the user - i.e. the preference function in our case - decides whether the acknowledged allocation is acceptable or not. If no downgrading happened then the user accepts the established reservation with probability 1. However, in case of downgrading, the user may reject the reservation or choose a more preferred resource setting according to the preference function. In either case, the excess bandwidth reservation is released and the necessary bandwidth is allocated for the application by issuing allocate and tear excess[3] messages.

Allocation Type 1* This type is similar to Type 1 except that user revision is disabled, i.e. any non-zero allocation is accepted by the initiator user.

Allocation Type 2 In this case, not only the downgrade vector but also the preference function is included in the reservation request message. Hence, the user behavior is modeled within the network nodes, so user revision is not required.

4.4.5 Numerical Results

I used the generic performance evaluation framework and simulation scenario that is described in Section 4.2 with the downgrade vector and preference function given in the previous sections, in order to evaluate these allocation schemes. This section summarizes the main results of the simulation study [C14, J2].

Reservation Setup Time

Figure 4.7 shows the reservation setup time distribution for the type 0 scheme. It always starts from 300 ms, that is the minimum reservation setup time, and flattens with the increasing number of background sources. It is highly dependent on the actual network load contrary to the type 2 scheme which gives a constant setup delay of 300 ms whatever the network load is.

The mean reservation setup time of the five allocation schemes is presented in Figure 4.8. Comparing type 0 to type 0*, a moderate improvement can be noticed, which means that

[3] note that the allocated bandwidth may be zero

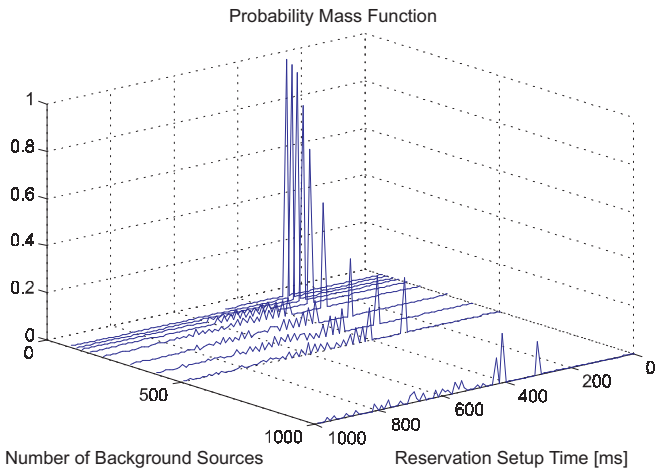

Figure 4.7: PMF of reservation setup time for type 0

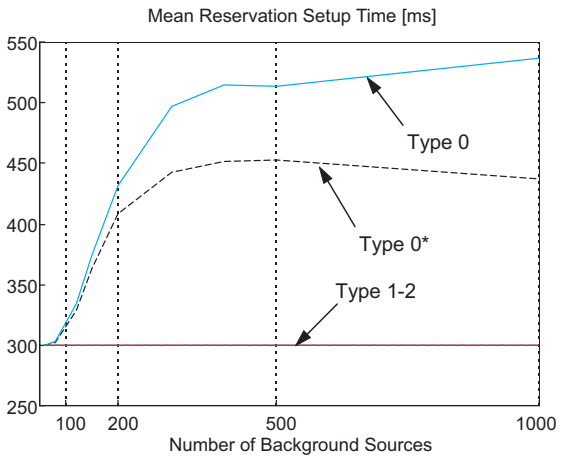

Figure 4.8: Mean reservation setup time for the five allocation types

the free bandwidth determined by the random load situation in the network rarely matches the resource demand of quality settings preferred by the user. Due to downgrading in the nodes, setting up a reservation takes always a fix time (if it successes) for type 1, type 1* and type 2.

Setup Retries

The difference between the performance of type 0 and type 0* can be further stressed by visualizing the number of reservation setup retries (Figure 4.9,4.10). It can be seen that more retries are needed to make a successful reservation, if there is a higher network load. Moreover, type 0* requires much less retries for reservation setup than type 0. The reason of this is that the user does not check the preference function, but accepts everything in case of type 0*.

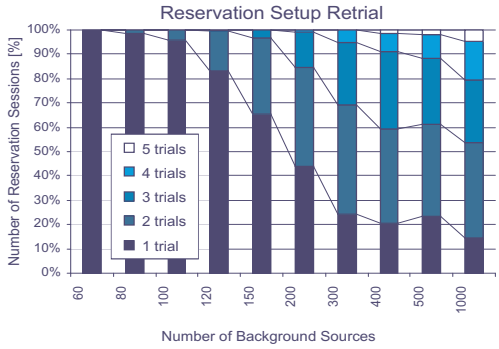

Figure 4.9: Reservation setup retries for (type 0)

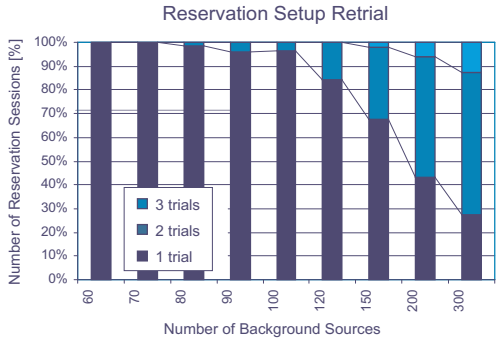

Figure 4.10: Reservation setup retries without user revision (type 0*)

Call Blocking Probability

In Figure 4.11 we show the foreground source's blocking probability as a function of the number of background sources. It can be seen that the blocking probability is an increasing function of the number of background sources for each type, and the curvatures are almost identical. It is also shown, that the advantage of instant downgrade capability of type 1 and type 2 does not really influence the blocking probability, however it slightly remains below the blocking probability of type 0 in case of large network load, while it is a little higher in case of small network load.

Over-Provisioning Factor

Figure 4.12 shows the over-provisioning on node 2 and node 9 for allocation schemes type 0* and type 1*. It can be seen, that with the increasing number of reservation setup retries (see Figure 7) the over-provisioning of the type 0* allocation increases. Moreover, the curve of type 1* is always below of the type 0*. It means less waste of bandwidth which could result in higher efficiency in sharp situations. It is obvious that the worst case node is the first node next to host A, where the unnecessary reservations are kept longest. For node 8 – close to the destination –, the over-provisioning factor straightens and is smaller with two magnitude for both allocation schemes. This unbalanced way of loading the network nodes is often called location bias [J3].

Over-provisioning is also plotted for the allocation types in which the host A revises the acknowledged bandwidth. Reservation types 1 and 2 resulted in smaller over-provisioning than type 0, as it is pronounced in Figure 4.13. It is interesting that type 2 can utilize the additional service specific information (i.e. the user preference function) for sparing with excess bandwidth in case of small background load ($n < 200$), while type 1 performs better than type 2 in case of more background calls. The over-provisioning on node 1 has the same order of magnitude as in the previous case.

Signaling Intensity

Figure 11 presents the signaling intensity on the first (N0) and last node (N9) in case of type 0* and type 1*. Node 0 should handle more atomic signaling messages in case of allocation type 0, while this difference is very small on node 9. Allocation type 1 causes less location biased load regarding signal handling.

The same performance measure is plotted for the types implementing user revision (Figure 4.15). Only a small improvement can be observed in case of this variant of the baseline allocation type (i.e. type 0). Type 2 has the smallest location bias and less intensive signaling.

It is noticeable, that the presented allocation schemes require signaling messages of different size, because types 1 and 2 include the Downgrade Vector, type 2 the Preference Function too, but type 0 does not transfer service specific information. Thus the overhead due to signaling is maximal for type 2 and minimal for type 0.

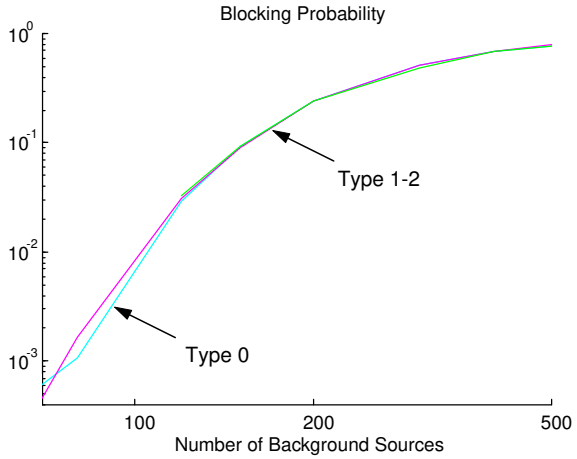

Figure 4.11: Blocking probability without user downgrade

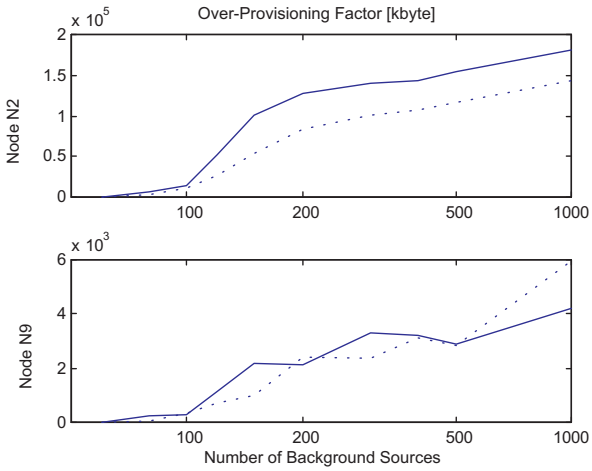

Figure 4.12: Over-provisioning factor for type 0* and type 1* (solid and dotted lines)

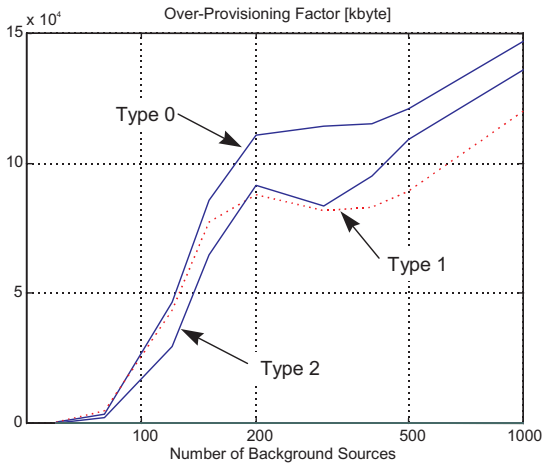

Figure 4.13: Over-provisioning factor on node N2

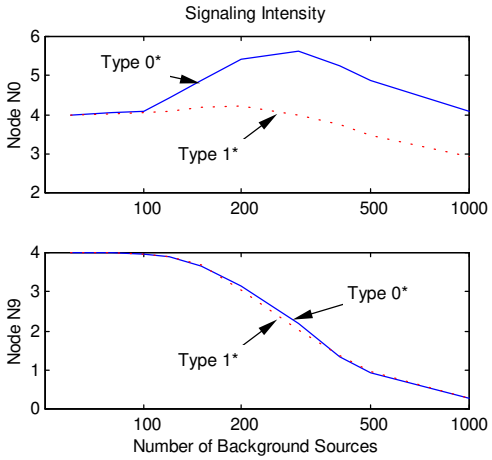

Figure 4.14: Signaling intensity on nodes N1 and N10

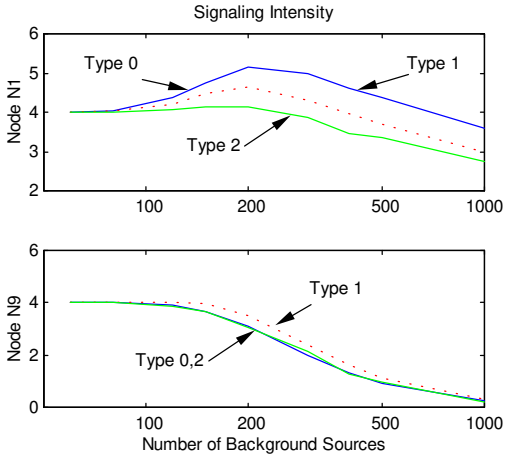

Figure 4.15: Signaling intensity on nodes N1 and N10 in case of user revision

4.4.6 Conclusions

Two intelligent resource allocation schemes were introduced and analyzed in this section, which utilize service specific information, specifically the downgrade vector and the user preference function, in order to minimize over-provisioning and reservation setup time. The numerical results highlight that the service specific information results in (i) a fix and shorter call setup time, (ii) less reservation trials, (iii) less over-provisioning and (iv) less intensive signal handling in the nodes by maintaining (v) the same blocking probability [C14, J2].

Where is the Optimum?

Although the more intelligent schemes yielded better performance, the usage of further service specific information is questionable. The more information is carried in signaling messages the more complex their handling is in the hosts and network nodes. This effect can be quantified by the *Memory and CPU cycle consumption* or the *Signaling Message Processing Time* metrics. According to [C10] and our newer results, the admission control process takes only about 8% of the total processing time of an RSVP message. Thus even if we assume that the computational complexity (i.e. the time needed for admission control) grows linearly with the number of resource values in a downgrade vector, this factor is negligible. On the other hand, more information yields longer signaling messages, which can be indicated by the *signaling overhead* metric.

4.5 Summary

This chapter proposed a performance evaluation framework for resource reservation schemes including traditional and novel performance metrics. Moreover, it demonstrated the application of this framework on different reservation schemes, such as sender-oriented, receiver-oriented protocols and others using a central resource manager or service specific information.

The network scenario and notation of this framework have been introduced in Section 4.2. Performance metrics have been discussed in two groups, differentiating *per hop metrics* and *per reservation metrics*. The former describes the local impact of signaling messages on a network node, while the latter characterizes the entire resource reservation setup process in the network.

Besides (i) message handling time, (ii) memory consumption and (iii) software complexity, the *Signaling Intensity* metric has been proposed for quantifying the impact of the reservation protocol on the signaling handling node. This metric takes the number of atomic messages per reservation trial into account, and in this way characterizes the effectiveness of the protocol. Moreover, I have proposed the *Over-Provisioning Factor* OPF for characterizing the greediness of the reservation algorithm, which has an indirect impact on the Best Effort traffic. This metric also describes how competitive is the resource reservation process in the Premium traffic class.

Extending the list of traditional per reservation metrics, such as (i) blocking probability, (ii) signaling overhead, (iii) reservation setup time, (iv) number of reservation trials and (v) different fairness metrics, I have proposed the *Over-Provisioning Balance* and *Signaling Intensity Balance* metrics for characterizing how much the OPF and signaling intensity metrics, resp., depend on the topological location of a node.

I have proposed a new resource reservation scheme (called Hybrid Allocation) in Section 4.3, which aims for minimizing the over-provisioning factor in the network by downgrading the reservation request and sending release messages from every network node. I have compared the performance of this new scheme with a sender-oriented, and a receiver-oriented scheme and I have proven that the Hybrid Allocation has achieved its goal. In particular, it has yielded the least OPF in the network producing a relatively low signaling intensity.

The service specific information has been analyzed in Section 4.4 and a mapping has been defined for expressing the demanded network resources as a function of the quality parameters of the application. The main idea is that why shall we reserve a particular set of resources, if the application can not utilize it or the user does not prefer the resulted quality. Thus I have proposed the *Downgrade Vector* for expressing the multimedia application's capabilities and the *Preference Function* for characterizing the user's behaviour. Two intelligent resource allocation schemes have been introduced and analyzed in this section, which utilize service specific information, specifically the downgrade vector and the user preference function, in order to minimize over-provisioning and reservation setup time. The numerical results has shown that the service specific information results in (i) a fix and shorter call setup time, (ii) less reservation trials, (iii) less over-provisioning and (iv) less intensive signal handling in the nodes by maintaining (v) the same blocking probability.

Chapter 5

Summary of the Dissertation

This dissertation covers various fields of traffic characterization, traffic modeling and traffic managements for multimedia teleservices.

Chapter 2 presents the Leaky Bucket Analysis – a framework for resource dimensioning and characterizing the burstiness of ATM traffic –, demonstrates its applicability on deterministic and measured traffic traces, and describes its applications.

The Leaky Bucket Analysis (LBA) provides the Leaky Bucket curve $q(r)$, its dual pair $q^t(t)$ and the Leaky Bucket Slope curve $s(t)$ as deterministic bounds for quantifying and visualizing the resource demand of a trace. Several deterministic traffic traces are analyzed using the LBA, and the family of Multilevel On-Off (MOO) traces is introduced. The analysis of measured traces demonstrates that the Leaky Bucket Analysis can characterize how the resource demand and the burst structure of VBR video traffic changes due to traffic shaping, multiplexing, changing the video sequence or the video frame rate. The robustness of LBA against the variations in the traffic of a certain traffic type and the length of captured trace is investigated and accuracy thresholds are given for single and aggregated Internet and VBR video traffic using the relative error ratio of $q^t(t)$. This analysis emphasizes that the LBA can characterize specific traffic types too, not only a single traffic trace.

The deterministic metrics of LBA are fitted to two analytic models. The first model utilizes that the burst structure of measured traffic traces and MOO traces are similar. The parameters of the MOO model are set based on the breaking points in $q(r)$ and the place of plateau in $s(t)$. The second model has been a two-level fluid flow model.

Moreover, I describe several applications of the LBA, such as selecting the parameter set of the ATM connection traffic descriptor considering different optimization criteria, selecting and detecting the shaping rate, queuing behaviour analysis by visualizing the gain of statistical multiplexing and cell loss rate estimation.

Chapter 3 presents a hierarchical source model based on the white box modeling concept. This model can synthesize the traffic of a VBR traffic source by imitating the operation of VBR encoding, process scheduling and protocol encapsulation in a multimedia workstation.

First the main characteristics of the multimedia traffic are evaluated by performing traffic intensity analysis and silence period analysis on different burst scales ranging from cell level to scene level. This black box analysis emphasizes that there are four burst levels in the multimedia traffic – scene, video frame, packet and cell levels –, which appear periodically in a regular manner. Moreover, while the scene and video frame level characteristics strongly depend on the video content, the packet and cell level characteristics (such as packet size, packet interarrival time, etc.) are unchanged. That is also emphasized that the frame interarrival time varies and its mean is determined by the video frame rate setting. Furthermore, a linear relationship is found between the packet silence period and the size of packet generated after that period.

The observations and numerical results of the black box analysis are extended based on the knowledge of the multimedia terminal's internal operation resulting a hierarchical model. This model consists of three model stages according to the terminal's three independent traffic generation procedures, i.e. encoding, scheduling and encapsulation. The video encoding procedure is modeled with a two-state Markov process, while the scheduling of application transfer units is modeled with a simple Gaussian process and a deterministic model imitates the data encapsulation in the protocol stack.

The model is validated using the LBA and the results show that the queuing performance of original traces (in terms of maximum queue length and estimated cell loss) is successfully reproduced by the synthetic traffic of our model.

Chapter 4 proposes a performance evaluation framework for resource reservation schemes including traditional and novel performance metrics. Moreover, it demonstrates the application of this framework on different reservation schemes.

Performance metrics are discussed in two groups, differentiating *per hop metrics* and *per reservation metrics*. The former describe the local impact of signaling messages on a network node, while the latter characterize the entire resource reservation setup process in the network. Besides (i) message handling time, (ii) memory consumption and (iii) software complexity, the *Signaling Intensity* metric is proposed for quantifying the impact of the reservation protocol on the signaling handling node. This metric takes the number of atomic messages per reservation trial into account, and in this way characterizes the effectiveness of the protocol. Moreover, the *Over-Provisioning Factor* OPF is proposed for characterizing the greediness of the reservation algorithm, which has an indirect impact on the Best Effort traffic. This metric also describes how competitive the resource reservation process is in the Premium traffic class. Extending the list of traditional per reservation metrics, such as (i) blocking probability, (ii) signaling overhead, (iii) reservation setup time, (iv) number of reservation trials and (v) different fairness metrics, the *Over-Provisioning Balance* and *Signaling Intensity Balance* metrics are proposed for characterizing how much the OPF and signaling intensity metrics, respectively, depend on the topological location of a node.

A new resource reservation scheme is introduced for minimizing the over-provisioning factor and its performance is compared with a sender-oriented, and a receiver-oriented scheme. It is shown that the new scheme yields the least OPF in the network producing a

relatively low signaling intensity.

Moreover, the service specific information is analyzed and a mapping is defined for expressing the demanded network resources as a function of the quality parameters of the application. The *Downgrade Vector* and the *Preference Function* are proposed for expressing the multimedia application's capabilities and for characterizing the user's behaviour, respectively. Furthermore, two intelligent resource allocation schemes are introduced and analyzed, which can utilize the service specific information for minimizing the over-provisioning and reservation setup time, according to the numerical results.

Bibliography

[B1] **I. Cselényi**, N. Björkman, and S. Molnár. How Does an ATM Switch See the Traffic Through the Leaky Bucket? *4th IFIP book on ATM Networks, Performance Modelling and Analysis*, (edited by D. Kouvatsos), Chapman and Hall, London, UK, 1998, pp. 93-116.

[B2] G. Fehér, and **I. Cselényi**. Adaptive Resource Broker for Multimedia Teleservices, accepted for publication in *Multimedia Internet Broadcasting Book*, (edited by A. Sloane and D. Lawrence), Springer, August 2000.

[J1] **I. Cselényi**, and S. Molnár. VBR Video Source Characterization and a Practical Hierarchical Model, accepted for publication in *Telecommunication System Journal*, September 2000.

[J2] **I. Cselényi**, and R. Szabó. Service Specific Information Based Resource Allocation for Multimedia Applications, *Informatica Journal*, September 1999, pp. 317-324.

[J3] **I. Cselényi**, R. Szabó, I. Szabó, N. Björkman, and A. Latour-Henner. Experimental Platform for Telecommunication Resource Management, *Computer Communications Journal*, special issue on Stochastic Analysis and Optimal Management of Computer Communication Systems, Vol. 21, No. 17, November 1998, pp. 1624-1640.

[J4] **I. Cselényi**, P. Haraszti, and S. Székely. The First Hungarian ATM Pilot Network, *Magyar Távközlés*, February 1997, pp. 6-10. (in English)

[J5] **I. Cselényi**, P. Füzesi and K. Németh. The Evolution of Quality of Service in Internet, *Magyar Távközlés*, February 2000, pp. 26-31. (in Hungarian)

[J6] P. Füzesi, L. Malicskó and **I. Cselényi**. Billing of services with guaranteed quality, *Magyar Távközlés*, June 2000, pp. 34-37. (in Hungarian)

[J7] Cs. Antal, and **I. Cselényi**. Distributed Multimedia Applications: Teleworking and Teleeducation, *Híradástechnika*, December 1995, pp. 58-62. (in Hungarian)

[J8] A. Bortnyák, **I. Cselényi**, Z. Gimesi, and I. Illés. Dactiloscopic Investigations with Neural Networks, *Floppy.lap*, January 1992, pp. 31-33. (in Hungarian)

[C1] **I. Cselényi**, S. Molnár, and N. Björkman. Leaky Bucket Analysis as a Practical Traffic Characterisation Tool, *5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley, England, 21-23 July 1997, paper number 28.

[C2] A. Latour-Henner, N. Björkman, and **I. Cselényi**. ATM Traffic Shaping at the Multimedia Terminal, *International Conference on Telecommunications, ICT 97*, Melbourne, Australia, 2-5 April 1997, pp. 1193-1198.

[C3] S. Molnár, **I. Cselényi**, and N. Björkman. ATM Traffic Characterisation and Modelling Based on the Leaky Bucket Algorithm, *IEEE International Conference on Communication Systems, ICCS'96*, Singapore, November 1996, Vol. 1, pp. 185-189.

[C4] **I. Cselényi**, S. Molnár, and N. Björkman. On Measurements of Multimedia Traffic in ATM Networks, *International Conference on Multimedia Technology and Digital Telecommunication Services*, ICOMT '96, Budapest, Hungary, 28-30 October, 1996, pp. 191-199.

[C5] S. Molnár, **I. Cselényi**, N. Björkman, C.G. Perntz, and M. Boda. ATM Traffic Measurements and Analysis on a Real Testbed, *10th International Teletraffic Congress, Specialist Seminar on Control in Communications*, Lund, Sweden, 17-19 September, 1996, pp. 237-250.

[C6] A. Vidács, S. Molnár, G. Gordos, and **I. Cselényi**. The Impact of Long Range Dependence on Cell Loss in An ATM Wide Area Network, *IEEE Conference on Global Communications*, GLOBECOM'98, Sydney, Australia, 8-12 November, 1998.

[C7] **I. Cselényi**, S. Molnár, N. Björkman, and A. Latour-Henner. Hierarchical VBR Video Source Model, *5th International Conference on Telecommunication Systems, Modeling and Analysis*, Nashville, Tenesse, 20-23 March, 1997, pp. 401-410.

[C8] G. Fehér, K. Németh, and **I. Cselényi**. Router Benchmarking Framework for QoS Signaling, 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, Ilkley, England, 17-19 July, 2000, paper number 24.

[C9] K. Németh, G. Fehér, and **I. Cselényi**. Simulation Study for IP Resource Reservation, 8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks, Ilkley, England, 17-19 July, 2000, paper number 64.

[C10] **I. Cselényi**, G. Fehér, and K. Németh. Benchmarking of Signaling Based Resource Reservation in Internet, *Networking 2000 Conference*, Paris, France, 14-19 May, 2000.

[C11] G. Fehér, K. Németh, M. Maliosz, **I. Cselényi**, J. Bergkvist, D. Ahlard, and T. Engborg. Boomerang – A Simple Protocol for Resource Reservation in IP Networks, *IEEE Workshop on QoS Support for Real-Time Internet Applications*, Vancouver, Canada, 1 June, 1999.

[C12] M. Maliosz, K. Farkas, and **I. Cselényi**. Agent Based Interactive Television Service For An Experimental Multimedia System, *6th Singapore International Conference on Communication Systems, ICCS'98*, Singapore, November 1998. pp. 864-868.

[C13] K. Farkas, **I. Cselényi**, and G. Fehér. Rule Based Resource Reservation Scheme For Multiparty, Multimedia Teleservices, *6th Singapore International Conference on Communication Systems, ICCS'98*, Singapore, November 1998. pp. 855-858.

[C14] R. Szabó, and **I. Cselényi**. Performance Evaluation of an Intelligent Resource Allocation Scheme for Multimedia Applications, *Symposium on Performance Evaluation of Computer and Telecommunication Systems, SPECTS'98*, Reno, Nevada, 19-22 July, 1998, pp. 302-306.

[C15] N. Björkman, **I. Cselényi**, A. Latour-Henner, and G. Záruba. The EMMA Multimedia Conference Service, *International Conference on Information, Communications and Signal Processing*, ICICS'97, Singapore, 9-12 September, 1997, Vol. 3, pp. 1691-1995.

[C16] **I. Cselényi**, I. Szabó, P. Haraszti, N. Björkman, and C. Gisgard. A Versatile Multimedia Platform, *15th IASTED International Conference on Applied Informatics*, Innsbruck, Austria, 18-20 February, 1997, pp. 276-279.

[C17] **I. Cselényi**, C. Gisgard, P. Haraszti, A. Latour-Henner, I. Szabó, and G. Záruba. Multipurpose Middleware for Broadband Multimedia Applications, *ConTEL Brodband & Multimedia Workshop*, Zagreb, Croatia, 11-12 November, 1996, pp. 155-164.

[C18] **I. Cselényi**, C. Gisgard, P. Somos, and L. Taska. Exploration of Broadband Multimedia Services, *Czech-Hungarian-Polish Workshop*, Prague, Czech Republic, 3-5 September, 1996, pp. 33-37.

[C19] **I. Cselényi**, N. Björkman, F. Baumann, P. Haraszti, T. Henk, A. Latour-Henner, and A. Miah. Service User Emulator for Generating a Repeatable Network Load, *IASTED International Conference on Networks*, Orlando, Florida, 8-10 January, 1996, pp. 229-232.

[T2] **I. Cselényi**, D. Ahlard, and J. Bergkvist. Downgrade Vector for Resource Reservation, *Telia Research AB*, pending patent application, No. PCT/SE00/01341.

[T3] R. Szabó, and **I. Cselényi**. Session Level Network Simulator, *Telia Research AB*, technical report, No. 9/0362FCPA1090150, November 1997.

[T4] D. Ahlard, J. Bergkvist, **I. Cselényi**, and T. Enborg. Boomerang - A Simple Resource Reservation Framework for IP, *Internet Engineering Task Force, IETF*, Internet Draft, February 1999. <draft-bergkvist-boomerang-framework-00.txt>

[T5] J. Bergkvist, **I. Cselényi**, and D. Ahlard. Boomerang Protocol Specification, *Internet Engineering Task Force, IETF*, Internet Draft, June 1999. <draft-bergkvist-boomerang-spec-00.txt>

[T6] G. Fehér, **I. Cselényi**, and K. Németh. Benchmarking Methodology for QoS Signaling, *Internet Engineering Task Force, IETF*, Internet Draft, July 2000. <draft-feher-benchresres-00.txt>

---

[1] R.L. Cruz. Service Burstiness and Dynamic Burstiness Measures: A Framework *Journal of High Speed Networks*, Vol. 1, No. 2, pp. 105-127, 1992.

[2] R.L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation, Part II: Network Analysis *IEEE Transactions on Information Theory*, Vol. 37, No. 1, Jan. 1991, pp. 114-131 and pp. 132-141.

[3] R.L. Cruz. Quality of Service Guarantees in Virtual Circuit Switched Networks, *IEEE Journal of Selected Areas in Communication, special issue on "Advances in the Fundamentals of Networking"*, Vol. 13, No. 6, August, 1995, pp. 1048-1056.

[4] J.Y. Le Boudec. Application of Network Calculus To Guaranteed Service Networks, *IEEE Transactions on Information Theory*, Vol. 3, No. 44, May 1998

[5] F.P. Kelly. Effective Bandwidths at Multi-Class Queues, *Queueing Systems*, Vol. 9, 1991, pp. 5-16.

[6] S. Molnár, and Gy. Miklós. Peakedness Characterization in Teletraffic, *IFIP TC6, WG6.3 Conference on Performance of Information and Communications Systems*, PICS'98, Lund, Sweden, 25-28 May, 1998.

[7] S. Molnár, and Gy. Miklós. On Burst and Correlation Structure of Teletraffic Models, *5th IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, Ilkley, England, 21-23 July 1997.

[8] H. Zhang. Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks, *IEEE*, October 1995, pp. 1-23.

[9] P. J. Kühn. Modelling of new services in computer and communication networks, *Computer Networking and Performance Evaluation*, (edited by T. Hasegawa), Netherlands, 1986, pp. 283-303.

[10] G. D. Stamoulis, M. E. Anagnostou, and A. D. Georgantas. Traffic Source Models For ATM Networks: a Survey, *Computer Communications Journal*, Vol. 17, No. 6, June 1994, pp. 428-438.

[11] V. S. Frost, and B. Melamed. Traffic Modeling For Telecommunications Networks, *IEEE Communications Magazine* March 1994, pp. 70-81.

[12] J. Roberts, U. Mocci, and J. Virtamo, (Eds.). *COST 242 Performance Evaluation and Design of Broadband Multiservice Networks*, Final Report. Springer-Verlag, Berlin, 1996.

[13] Broadband Integrated Services Digital Network (B-ISDN) - Digital Subscriber Signalling System No. 2 (DSS 2) - User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control, *ITU-T*, Recommendation, February 1995.

[14] C. Topolcic. Experimental Internet Stream Protocol, Version 2 (ST-II), *IETF RFC 1190*, October 1990.

[15] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource Reservation Protocol (RSVP) Version 1, Functional Specification, *Internet Engineering Task Force, IETF*, RFC 2205, Proposed Standard, September 1997.

[16] Jon P. White, J. Crowcroft. A Case for Dynamic Sender-Initiated Reservation in the Internet, *Journal on High Speed Networks, Special Issue on QoS Routing and Signaling* Vol. 7 No. 2, 1998.

[17] P. Pan and H. Schulzrinne. YESSIR: A Simple Reservation Mechanism for the Internet, *ACM Computer Communication Review*, Vol. 29, No. 2, April 1999, pp. 89-101.

[18] A. Eriksson, and C. Gehrmann. Robust and Secure Light-weight Resource Reservation for Unicast IP Traffic, *International Workshop on QoS*, IWQoS'98, 18-20 May, 1998.

[19] L. Zhang. VirtualClock: A New Traffic Control Algorithm for Packet Switching Networks, *SIGCOMM Symposium on Communications Architectures and Protocols*, Philadelphia, USA, September 1990, pp. 19-29.

[20] I. Stoica, and H. Zhang. Providing Guaranteed Service Without Per Flow Management, *ACM Computer Communication Review*, Vol. 29, No. 4, October 1999. pp. 81-94.

[21] L. Westberg, Z. Turányi, and D. Partain, Load control of real-time traffic, *Internet Engineering Task Force, IETF*, Internet Draft, April 2000.

[22] J. Beran, R. Sherman, M. S. Taqqu, and W. Willinger. Variable-Bit-Rate Video Traffic and Long Range Dependence *IEEE Transactions on Communications*, Vol. 43, No. 2/3/4, 1995, pp. 1566-1579.

[23] W.E. Leland, M.S. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic, *IEEE/ACM Transactions on Networking*, Vol. 1, No. 2, February 1994.

[24] P. R. Jelenkovic, A. A. Lazar, and N. Semret. The Effect of Multiple Time Scales and Subexponentiality in MPEG Video Streams on Queuing Behaviour, *IEEE JSAC*, Vol. 15., No. 5, May 1997.

[25] B.K. Ryu, and A. Elwalid. The Importance of Long-Range Dependence of VBR Video Traffic in ATM Traffic Engineering: Myths and Realities, *ACM SIGCOMM '96*, 1996, pp. 3-17.

[26] N. Björkman, and A. Latour-Henner. High time-resolution cell loss measurements of mixed CBR and VBR traffic, *IFIP Broadband Communications '94 Conference*, March 1994.

[27] The Adtech AX/4000 Broadband Test System, User's Manual, *Adtech Systems Inc.*, Hawaii, 1996. http://www.adtech-inc.com/products/ax4000.asp

[28] HP Broadband Series Test System HP 75000, User's Manual, *Hewlett-Packard*, California, 1997.

[29] A. Rindos, S. Woolet, D. Cosby, L. Hango, and M. Vouk. Factors Influencing ATM Adapter Throughput, *IBM*, Technical Report, No. 29.2099, can be downloaded from http://www.networking.ibm.com/per/per1.html

[30] G. Seres, T. Éltetö, and A. Oláh, Measurement-Based Simulation Model for TCP over ATM, accepted for publication in *Annals of Telecommunications*, May - June 2000, pp. 278-287.

[31] R. Neilson, J. Wheeler, F. Reichmeyer, and S. Hares. A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment, Version 0.7, *Internet2 QBone BB Advisory Council*, August 1999.

[32] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. Resource ReSerVation Protocol (RSVP) Version 1 – Applicability Statement, Some Guidelines on Deployment *Internet Engineering Task Force*, IETF Informational RFC 2208, September 1997

[33] *ATM Forum*, ATM User-Network Interface Specification Version 3.1, Prentice Hall, Upper Saddle River, NJ, 1995.

[34] *ATM Forum*, Traffic Management Specification Version 4.0, Specification No. af-tm-0056.000, Prentice Hall, Upper Saddle River, NJ, April 1996.

[35] *ITU-T*, Recommendation I.371, Traffic control and congestion control in B-ISDN.

[36] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: An Overview, *Internet Engineering Task Force*, IETF RFC 1633, July 1994.

[37] Y.I. Manolessos, and M.E. Theologou. Dynamic application scaling as a means for QoS management, *Computer Communications* No. 20, 1997, pp. 393-405.

[38] C. Blondia, and O. Casals. Statistical Multiplexing of VBR Sources: A Matrix Analytic Approach, *Performance Evaluation*, Vol. 16, No. 5, 1992, pp. 5-20.

[39] S. Blaabjerg, and S. Molnár. Methods for UPC Dimensioning of a CDV Perturbated Cell Stream, *RACE BRAVE Workshop*, Milano, Italy, June 1995.

[40] N. G. Duffield, K. K. Ramakrishnan, and A. R. Reibman. Issues of Quality and Mulitplexing when Smoothing Rate Adaptive Video, *International Workshop on Network and Operating System Support for Digital Audio and Video*, NOSSDAV, Cambridge, England, July 1998, pp. 265-278.

[41] D.P. Heyman, A. Tabatabai, and T.V. Lakshman. Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, No. 1, 1992.

[42] M. Hamdi, J. W. Roberts, and P. Rolin. Rate Control for VBR Video Coders in Broad-Band Networks, *IEEE JSAC*, Vol. 15, No. 6, August 1997, pp. 1040-1051.

[43] P.E. Heegaard, and B.E. Helvik. Establishing ATM Source Models for Traffic Measurement, 11^th *Nordic Teletraffic Seminar*, Stockholm, August, 1993.

[44] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance Models of Statistical Multiplexing in Packet Video Communications, *IEEE Transactions on Communications*, Vol. 36, No. 7, 1988, pp. 834-844.

[45] S. Molnár, and I. Maricza. Source Characterization in Broadband Networks, *COST 257 Interim Report*, January, 1999.

[46] E.P. Rathgeb. Policing of Realistic VBR Video Traffic in an ATM Network, *International Journal of Digital and Analog Communication Systems*, Vol. 6, 1993, pp. 213-226.

[47] E.P. Rathgeb. Policing Mechanisms for ATM Networks: Modelling and Performace Comparison, *International Teletraffic Congress, Seventh Specialist Seminar*, Morristown, USA, October 1990, pp. 10.1.1–10.1.10.

[48] W. Willinger, M. S. Taqqu, and A. Erramilli. A bibliographical guide to self-similar traffic and performance modeling for high speed networks, *Stochastic Networks, Theory and Applications*, Oxford University Press, 1996, pp. 339-366.

[49] G. Ramamurthy, and B. Sengupta. Modeling and Analysis of Variable Bit Rate Video Multiplexer, *INFOCOM '92*, 1992, 6.C.1.1-11.

[50] C. Parris, and D. Ferrari. The implementation of a dynamic management scheme for guaranteed-performance connections, *Computer Communications* No. 21/1, 1998, pp. 1-23.

[51] H. Haapasalo, I. Norros, and T. Raij. Some experiments from ATM traffic measurements. In *The Twelfth Nordic Teletraffic Seminar*, Aug 1995. Espoo, Finland.

[52] Y.B. Lin and I. Chlamtac. Effective call holding times for PCS network. submitted for publication, available at http://liny.csie.nctu.edu.tw.

[53] D. Medhi and S. Guptan. Network dimensioning and performance of multiservice, multirate loss networks with dynamic routing. *IEEE/ACM Transaction on Networking*, 5(6):944–957, December 1997.

[54] N. Natarajan. *Network Resource Information Model Specification*, 1997. Version 3.0.

[55] RACE Project R2044. *MAGIC-Multiservice Applications Governing Integrated Control, Service Descritption Framework and B-ISDN Service Descripton*, 3rd deliverable edition, 1992.

[56] Litton Network Access Systems. *CAMVision-2 7615, User's Guide*, October 1998.

[57] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A new resource reservation protocol. *IEEE Networks Magazine*, September 1993.

[58] D. Niehaus. Performance benchmarking of signaling in ATM networks. *IEEE Communication Magazine*, August, 1997, pp. 134-143.

[59] W.C. Wong. Packet reservation multiple access in a metropolitan microcellular radio environment, *IEEE Jounral on Selected Areas of Communication*, Vol. 11, No. 6, 1993, pp. 918-925.

[60] M.C. Yuang, and Y.R. Haung. Bandwidth assigment paradigms for broadband integrated voice/data networks, *Computer Communications*, Vol. 21, No. 3, 1998, pp. 243-253

[61] L. Barnett. The Space Division Multiple Access Protocol: asynchronous implementation and performance, *Computer Communications*, Vol. 21, No. 3, 1998, pp. 227-242

[62] J. Forgie, ST - A Proposed Internet Stream Protocol, *IEN 119, M.I.T. Lincoln Laboratory*, September 1979.

[63] L. Delgrossi, L. Berger, Internet Stream Protocol Version 2 (ST2) Protocol Specification - Version ST2+, *IETF RFC 1819*, August 1995.

[64] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski and E. Felstaine. A Framework For Integrated Services Operation Over Diffserv Networks Internet Draft, September, 1999. <draft-ietf-issll-diffserv-rsvp-03.txt>

[65] A. Ghanwani, J. W. Pace, V. Srinivasan, A. Smith and M. Seaman A Framework for Integrated Services Over Shared and Switched IEEE 802 LAN Technologies Internet Draft, June 1999. <draft-ietf-issll-is802-framework-07.txt>

[66] *ATM Forum*, ATM User Network Interface Specification 3.1 (1994).

[67] J.J. Bae, and T. Suda. Survey of Traffic Control Protocols in ATM Networks, *IEEE Conference on Global Communications*, GLOBECOM'90, San Diego, California, December 1990, pp. 1-6 (300.1)

[68] I. Khan, and V.O.K. Li. Traffic control in ATM Networks, *Computer Networks and ISDN Systems*, Vol. 27, 1994, pp. 85-100.

[69] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, *Internet Engineering Task Force*, IETF RFC 2475, December 1998.

[70] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *Internet Engineering Task Force*, IETF RFC 2474, December 1998.

[71] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry. The COPS (Common Open Policy Service) Protocol, *Internet Engineering Task Force*, IETF RFC 2748, January 2000.

[72] Optimum Network Performance – Modeler, *OPNET Technologies*, http://www.opnet.com

[73] A. Neogi, et. al. Performance Analysis of an RSVP-Capable Router, *IEEE Network*, Vol. 14, No. 5, September/October 1999.

[74] M. Butto, E. Cavallero, and A. Tonietti. Effectiveness of the "leaky bucket" policing mechanism in ATM networks, *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, April 1991, pp. 335-342.

[75] A.I. Elwalid, and D. Mitra. Analysis and design of rate-based congestion control of high speed networks, I: stochastic fluid models, access regulation, *Queueing Systems*, Vol. 9, No. 1-2, October 1991, pp. 29-64.

[76] D.E. McDysan, and D.L. Spohn. ATM – Theory and Application, *McGraw-Hill, Inc.*, 1995.

[77] E.C. Lin. On the Effects of Scheduling Contention on ATM Cell Output, *Kungliga Tekniska Hgskolan*, licentiate, No. TRITA-IT R 95:20, October 1995.

[78] Specifications of Signalling System No. 7 – Telephone user part, CCITT, Recommendation Q.724, 1988.

[79] Z. Wang, and J. Crowcroft. Analysis of Burstiness and Jitter in Real-Time Communications, *SIGCOMM93*, Ithaca, USA, September, 1993, pp. 13-19.

[80] R. Gusella. Characterizing the variability of arrival processes with indexes of dispersion, *IEEE Journal on Selected Areas in Communications*, Vol. COM-30, No. 4, 1982, pp. 728-738.

[81] K. Srivam, and W. Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data, *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, September, 1986.

[82] A.E. Eckberg. Generalized peakedness of teletraffic processes, *10th International Teletraffic Congress*, ITC-10, 1984.

[83] S. Molnár. Evaluation of Quality of Service and Network Performance in ATM Networks, *PhD dissertation*, Technical University of Budapest, 1995.

[84] W.E. Leland, M.S. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic, *IEEE/ACM Transactions on Networking*, Vol. 1, No. 2, February 1994.

[85] S. Molnár, and A. Vidács. On modeling and shaping self-similar ATM traffic, *15th International Teletraffic Congress*, ITC-15, Washington D.C., USA, June 1997.

[86] A.K. Parekh. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks, *PhD Thesis*, MIT, 1992.

[87] A.K. Parekh, and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks – the multiple node case, *Conference on Computer Communications (IEEE Infocom)*, San Francisco, USA, Vol. 2, March/April 1993, pp. 521–530. (5a.1)

[88] D. Clark, S. Shenkar, and L. Zhang, Supporting Real-Time Applications in an Integrated Services Packet Network Architecture and Mechanism, *SIGCOMM92*, 1992.

[89] D. Ferrari, and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, Vol. 3, No. 8, April 1990, pp. 368-379.

[90] S. Golestani. Congestion-free transmission of real-time traffic in packet networks, *IEEE INFOCOM'90*, San Francisco, USA, June 1990, pp. 527-542.

[91] C. Kalmanek, H. Kanakia, and S. Keshav. Rate controlled servers for very high-speed networks, *IEEE Global Telecommunications Conference*, San Diego, USA, December 1990, pp. 299.3.1–200.3.9.

[92] E. Knightly, and H. Zhang. Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models, *IEEE INFOCOM'95*, Boston, USA, April 1995.

[93] M.W. Beckner, T.T. Lee and S.E. Minzer. A protocol and prototype for broadband subscriber access to ISDNs, *International Switching Symposium*, Phoenix, USA, May 1987, pp. 462-469. (B6.3)

[94] P. Boyer, F.M. Guillemin, M.J. Servel, and J.P. Coudreuse. Spacing cells protects and enhances utilization of ATM network links, *IEEE Network*, Vol. 6, No. 5, September 1992, pp. 38-49.

[95] L.K. Reiss, and L.F. Merakos. Shaping of Virtual Path Traffic for ATM BISDN, *IEEE Infocom*, San Francisco, USA, March/April 1993, Vol. 1, pp. 168-175.

[96] J.S. Turner. New directions in communications – or which way to the information age?, *IEEE Communications Magazine*, Vol. 24, No. 10, October 1986, pp. 8-15.

[97] A.E. Eckberg, D.T. Luan, and D.M. Lucantoni. An Approach to Controlling Congestion in ATM Networks, *International Journal of Digital and Analog Communication Systems*, Vol. 3, No. 2, 1990, pp. 199-209.

[98] W. Whitt. Approximating a point process by a renewal process: the view through a queue – an indirect approach, *Manage. Sci*, Vol. 27, No. 6, June 1981, pp. 619-636.

[99] C.S. Chang, R.L. Cruz. A time varying filtering theory for constrained traffic regulation and dynamic service guarantees, *IEEE Communications*, 1999.

[100] T. Konstantopoulos, and V. Anantharam. Optimal Flow Control Schemes that Regulate the Burstiness of Traffic, *Trnasaction on Networking*, Vol. 3, No. 4, August 1995. pp. 423-432.

[101] V. Jacobson. Congestion avoidance and control, *ACM SIGCOMM'88*, August 1988, pp. 314-329.

[102] K. Ramakrishnan, D. Chiu, and R. Jain. Congestion avoidance in computer networks with a connectionless network layer, *ACM SIGCOMM'88*, August 1988, pp. 303-313.

[103] A. Lazar, and P. Pacifici. Control of resources in broadband networks with quality of service guarantees. *IEEE Communication Magazine*, October 1991, pp. 66-73.

[104] MATLAB – The language of Technical Computing, Functional Reference, Release 11, *The MathWorks, Inc.*, January 1999

[105] A. Elwalid, D.P. Heyman, T.V. Lakshman, A. Weiss, and D. Mitra. Fundamental bounds and approximation for ATM multiplexers with application to video teleconferencing, *IEEE Journal on Selected Areas in Cummunication*, Vol. 13, August 1995, pp. 1004-1016.

[106] M.R. Frater, P. Tan, and J.F. Arnold. Variable bit rate video traffic on the broadband ISDN: Modelling and verification, *ITC-14* Elsevier Science Publisher B.V., North-Holland, 1993, pp. 1351-1360.

[107] M.W. Garrett, and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic, *SIGCOMM Symposium on Communications Architectures and Protocols*, London, UK, September 1994, pp. 269-280.

[108] H. Heeke. A Traffic-Control Algorithm for ATM Networks, *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 3, No. 3, June 1993, pp. 182-189.

[109] D.S. Lee, B. Melamed, A.R. Reibman, and B. Sengupta, TES modeling for analysis of a video multiplexer, *Performance Evaluation*, No. 16, 1992, pp. 21-34.

[110] M. Nomura, T. Fujii, and N. Ohta, Basic characteristics of variable rate coding in ATM environment, *IEEE Journal on Selected Areas in Cummunication*, Vol. 7, June 1989, pp. 752-760.

Appendix A

Extension of the Leaky Bucket Analysis

I have presented in Section 2.2.2 that the CBR trace and one-level On-Off traces result the least and most bursty LB curve, respectively, that is these traces constitute two extremes. I have also stated that any traffic trace $\mu(T, N)$ can be considered as a special MOO trace. Thus an interesting idea is to define a linear space with the MOO traces as base vectors and project other traffic traces to the vectors of this space.

Let Ω denote the space of traces $\mu(T, N)$ and Ξ denote a subspace of Ω, which contains every MOO trace. The 'addition' of two traces μ_A and μ_B is defined in the Ξ space as a "slot-wise" logical OR operation between the traces for all k:

$$\mu_{A+B} = \mu_A + \mu_B, \qquad \mu_A, \mu_B, \mu_{A+B} \in \Xi \tag{A.1}$$

$$\mu_{A+B}(k) \doteq \mu_A(k) \text{ OR } \mu_B(k), \qquad k = 1, 2, \ldots, T.$$

The 'product' of message μ with a scalar λ is defined as:

$$\mu_{\lambda C} = \lambda \mu_C, \qquad \mu_C \in \Xi; \quad \lambda \in I^+ \tag{A.2}$$

$$\mu_{\lambda C}(k) \doteq \mu_C(k + \lambda \% T) \qquad k = 1, 2, \ldots, T,$$

where % denotes the residuum function.

The 'null' element of this linear space is defined as:

$$\mu_0(k) \doteq 0, \qquad k = 1, 2, \ldots, T; \quad \mu_0 \in \Xi. \tag{A.3}$$

And finally, the 'base vectors' of this space are special MOO traces:

$$\mu_i(k) \doteq \epsilon(k - i), \qquad k = 1, 2, \ldots, T; \quad i = 1, 2, \ldots, T; \quad \mu_i \in \Xi. \tag{A.4}$$

Section 2.4.1 presented how an arbitrary, measured trace $\mu(T, N)$ can be modeled with an MOO trace targeting a good fit of LB curves. That modeling approach can be considered as a transformation from the Ω space to the Ξ space. This I call the *MOO Transformation*. The resulted MOO trace can be decomposed from the linear combination of MOO base vectors, yielding a canonical description of the trace. The LBA can promote the practical implementation of this transformation.

The linear space of MOO traces and the MOO transformation is subject of further research.

Appendix B

Equations to the Fluid Flow Model

This section provides a summary of the derivation of the parameter fitting equations for the two-level fluid flow model, presented in Section 2.4.2. Using the notations of that section, the basic equations for the two-level fluid flow model are:

$$r_0 < c < r_1, \qquad \Lambda(t) = r\left(Z(t)\right), \tag{B.1}$$

$$\pi_0 = \frac{\mu}{\lambda+\mu}, \qquad \pi_1 = \frac{\lambda}{\lambda+\mu}. \tag{B.2}$$

Denote $X(t)$ the amount of fluid in the buffer (i.e. the queue length). Its partial distribution function in state i is:

$$F_i(x) \;=\; \Pr\{Z = i, X \leq x\}. \tag{B.3}$$

The stationary first and second moments of the arrival rate are [12]:

$$\begin{aligned} \mathrm{E}\left[\Lambda\right] &= r_0 + (r_1 - r_0)\pi_1, \\ \mathrm{Cov}\left[\Lambda_u, \Lambda_v\right] &= (r_1 - r_0)^2 \pi_0 \pi_1 e^{(\lambda+\mu)|u-v|}. \end{aligned} \tag{B.4}$$

We can establish the following set of equations for the two states:

$$\begin{aligned} (r_0 - c)F_0^{'} &= -\lambda F_0 + \mu F_1, \\ (r_1 - c)F_1^{'} &= \lambda F_0 - \mu F_1. \end{aligned}$$

The solutions for the partial distribution functions can be obtained by using the following boundary conditions:

$$\begin{aligned} F_1(0) &= 0, \\ F_1(\infty) &= \pi_1, \end{aligned}$$

and we can get:

$$F_0(x) = \pi_0 - \pi_1 \frac{r_1 - c}{c - r_0} e^{-\zeta x}, \tag{B.5}$$

$$F_1(x) = \pi_1 \left(1 - e^{-\zeta x}\right), \tag{B.6}$$

where

$$\zeta = \frac{(c - r_0)\mu - (r_1 - c)\lambda}{(r_1 - c)(c - r_0)}. \tag{B.7}$$

The complementary queue length distribution, i.e. the estimation for the cell loss probability is:

$$Q(x) = \Pr\{X > x\} = \pi_1 \frac{r_1 - r_0}{c - r_0} e^{-\zeta x} = \tag{B.8}$$

$$= \frac{\lambda}{\lambda + \mu} \frac{r_1 - r_0}{c - r_0} e^{-\zeta x}.$$

Our aim is to fit the mean rate r_M to the LB curve and fit the cell loss rate l to the $Q(x)$ function. The former can be expressed from Equation (B.4) using (B.2), while the latter from Equation (B.8) using the working point r_S, N_S (see Figure 2.26):

$$r_M = r_0 + (r_1 - r_0)\frac{\lambda}{\lambda + \mu}, \tag{B.9}$$

$$l = \frac{\lambda}{\lambda + \mu} \frac{r_1 - r_0}{r_S - r_0} e^{-\zeta N_S}. \tag{B.10}$$

From Equation (B.9) we can get:

$$\frac{\lambda + \mu}{\lambda} = \frac{r_1 - r_0}{r_M - r_0}, \tag{B.11}$$

$$1 + \frac{\mu}{\lambda} = \frac{r_1 - r_0}{r_M - r_0},$$

$$\mu = \left(\frac{r_1 - r_0}{r_M - r_0} - 1\right)\lambda,$$

and finally we have:

$$\mu = \left(\frac{r_1 - r_M}{r_M - r_0}\right)\lambda. \tag{B.12}$$

Let us use $c = r_S$ in Equation (B.7) and insert μ from Equation (B.12):

$$\zeta = \frac{(r_S - r_0)\mu - (r_1 - r_S)\lambda}{(r_1 - r_S)(r_S - r_0)}, \tag{B.13}$$

$$= \frac{(r_S - r_0)\left(\frac{r_1 - r_0}{r_M - r_0} - 1\right)\lambda - (r_1 - r_S)\lambda}{(r_1 - r_S)(r_S - r_0)},$$

$$= \lambda \frac{(r_S - r_0)\left(\frac{r_1 - r_0}{r_M - r_0} - 1\right) - (r_1 - r_S)}{(r_1 - r_S)(r_S - r_0)}.$$

Now we can express λ from this expression:

$$\begin{aligned} \lambda &= \zeta \frac{(r_1 - r_S)(r_S - r_0)}{(r_S - r_0)\left(\frac{r_1 - r_0}{r_M - r_0} - 1\right) - (r_1 - r_S)}, \\ &= \zeta \frac{(r_1 - r_S)(r_S - r_0)}{(r_S - r_0)\left(\frac{r_1 - r_M}{r_M - r_0}\right) - (r_1 - r_S)}. \end{aligned} \tag{B.14}$$

We can get ζ also from Equation (B.10):

$$\zeta = -\frac{1}{N_S} \ln\left(l \frac{\lambda + \mu}{\lambda} \frac{r_S - r_0}{r_1 - r_0}\right),$$

and now use Equation (B.11) to insert $\frac{\lambda+\mu}{\lambda}$:

$$\begin{aligned} \zeta &= -\frac{1}{N_S} \ln\left(l \frac{r_1 - r_0}{r_M - r_0} \frac{r_S - r_0}{r_1 - r_0}\right), \\ \zeta &= -\frac{1}{N_S} \ln\left(l \frac{r_S - r_0}{r_M - r_0}\right). \end{aligned}$$

Therefore the three parameters for fitting the fluid flow model are:

$$\zeta = -\frac{1}{N_S} \ln\left(l \frac{r_S - r_0}{r_M - r_0}\right),$$

$$\lambda = \zeta \frac{(r_1 - r_S)(r_S - r_0)}{(r_S - r_0)\left(\frac{r_1 - r_M}{r_M - r_0}\right) - (r_1 - r_S)}, \qquad \mu = \lambda \frac{r_1 - r_M}{r_M - r_0}.$$

Appendix C

Traffic Measurements

Several hundreds of traces from many different ATM traffic types were captured by measurements in Telia Research, Sweden during the last couple of years [B1, C5, C3, C2, C19, C6, C4, 26]. This section shortly summarizes the measurement and describes the type of measured traffic. Moreover, it presents the main parameters and traditional traffic characteristics – such as mean cell rate and burstiness (i.e. squared coefficient of variation of the cell interarrival time) – of a subset of measured traces, which are mentioned in this dissertation.

C.1 Measurements on the Stockholm Gigabit Network

Traffic of multimedia workstations connected via the Stockholm Gigabit Network (an ATM MAN) was multiplexed with CBR background traffic [C5]. Long traces of both traffic types were captured before and after multiplexing. Traffic of different single multimedia sources and the aggregate traffic of four workstations were captured in case of different load conditions (*see traces (a)–(e)*).

C.2 Measurements on the "Internet" Backbone

Different parts of the Swedish University Network (SUNET) are attached to the Swedish ATM WAN. The aggregated traffic on the SUNET were analyzed during summer 1996 in the framework of a common trial between the SUNET community and Telia Research. The LAN traffic of universities in the northern region, around Uppsala, was aggregated on an FDDI backbone, which was connected via a couple of routers and a 34 Mbps PDH link to the ATM backbone in Stockholm. This network joins the northern LANs of SUNET to the international Internet backbone and to the southern university networks around Göteborg.

Table C.1: Main characteristics of VBR traffic traces captured in a MAN measurement

Trace	*Trace Length (s)*	*Total Number of Cells*	*Mean Cell Rate (cps)*	*Burstiness*	*Comments*
a	294.42	1 627 720	5500	140.3063	shaped
b	68.92	1 000 000	14 500	133.3298	aggregate
c	19.54	253 627	12 900	700.6972	bgr. 100 Mbps
d	20.38	264 258	12 900	693.0733	bgr. 120 Mbps
e	20.14	260 080	12 900	680.9803	bgr. 140 Mbps
f	51.00	100 309	1960	179.8622	10 fps
g	45.60	138 453	3030	258.6941	10 fps
h	36.86	151 544	4110	345.9633	10 fps
i	45.68	307 471	6720	483.3476	10 fps
j	38.27	250 142	6540	486.7423	10 fps
k	49.67	293 180	5890	452.2486	10 fps
l	35.73	87 353	2440	197.9074	10 fps
m	32.41	117 899	3640	245.1639	20 fps
n	45.29	186 966	4120	226.8030	25 fps

That is the captured traffic traces represent the traffic in the core network (*see traces (w)–(z)*). These are the longest traces that I have analyzed, capturing more than $8 \cdot 10^6$ cell arrivals. A good assumption is that the traffic was an ordinary mix of common Internet traffic types such as HTTP, FTP, Telnet, Chat, IP-phone etc.

C.3 Measurements in the Lab

Most of our laboratory tests have been carried out in Telia Research, Sweden, but I have also participated in measurements at Ellemtel Research in Älvsjö and Ericsson Traffic Lab in Budapest.

In Telia, we have measured the ATM traffic characteristics of several standard CCIR video sequences, which were encoded in hardware using a lossy compression algorithm. These sequences had equal duration but different traffic intensity and burst structure corresponding to the picture content. The video sequences were played from a video cassette recorder, which was connected to a multimedia workstation. The optical signal from the ATM interface card of the workstation was tapped to the ATM test equipment by means of an optical splitter. In this way we captured an exact copy of the cell flow between terminals without effecting the behaviour of the application in use. The arrival time of ATM cells of interest was recorded in real-time by a module developed by Telia AB, Sweden. It resides

Table C.2: Main characteristics of VBR traffic traces captured in LAN measurements

Trace	*Trace Length (s)*	*Total Number of Cells*	*Mean Cell Rate (cps)*	*Burstiness*	*Name of Video Sequence*	*Frame Rate (fps)*	*Resolution (pixels)*
GT10	21.41	22 871	1208	77.21	Girl with Toys	10	384x288
SU10	37.17	51 089	1554	98.52	Susie	10	384x288
PL10	24.04	85 060	4000	215.38	Popple	10	384x288
GT25	35.89	78 066	2459	78.82	Girl with Toys	25	384x288
SU25	34.55	83 696	2738	92.53	Susie	25	384x288
PL25	35.19	186 827	6003	171.93	Popple	25	384x288
Noise	20.18	263 570	12 926	710.26	Noise	10	768x576

in an ATM test instrument developed in the RACE PARASOL project. This instrument is capable of recording about 8 million cell arrivals [26].

Table C.3 presents traces from *six different video sequences*; namely the (f) Girl with Toys, (g) Susie, (h) Table Tennis, (i) Tempest, (j) Flower Garden, and (k) Popple sequences, while Table C.2 provides the main characteristics of seven traces which correspond to the Girl with Toys, Susie, Popple and Noise sequences with different frame rates and resolution.

Apart from VBR video, we have also measured the traffic of fundamental TCP/IP applications, such as WWW browsing, FTP and Ping. During capturing *traces (o,p)* a user has downloaded a large size image with its WWW browser several times. First we measured the traffic in case of a real user, than we played back the user's mouse moving and mouse click events by the "Service User Emulator" (SUE) tool, and in this way repeated the WWW session, resulting in trace (p). *Traces (r) and (q)* represent file transfer, while the traffic of *traces (s)–(v)* was generated by the UNIX Ping command, using different message size.

C.4 Measurement Configurations

We have repeated the measurements for many, different configurations, in which we have changed the type of source application (video, WWW, FTP, Ping, IP phone, White Board, etc.), the video sequence (see above), the VBR encoding technique (Cell-B and M-JPEG), the application level quality of service (10, 20 and 25 fps frame rate) and the protocol stack (AAL3/4, AAL5, and LLC or null encapsulation). In the investigated configurations, we have used numerous ATM measurement tools such as the 'Parasol' instrument [26], different HP devices [28], and the AdTech measurement tool [27]). The tests have been repeated for various terminal platforms (different SUN Sparc workstation models, Silicon Graphics Indigo, dedicated ATM hardware from AVA) equipped with different ATM cards (Fore Runner SBA 100–200E , SAHI-2 , Interphase with either SDH or TAXI physical layer).

Table C.3: Main characteristics of Internet traffic traces

Trace	*Trace Length (s)*	*Total Number of Cells*	*Mean Cell Rate (cps)*	*Burstiness*	*Comments*
o	11.45	22 889	1900	142.7399	real user
p	12.20	22 888	1880	150.4459	SUE
q	1.51	81 117	6843	680.8446	
r	1.48	81 116	6718	759.9904	
s	30.37	230 732	7573	448.6227	128 cells
t	29.94	286 945	9583	444.6371	256 cells
u	29.98	233 723	7772	471.4246	512 cells
v	29.83	290 948	9726	458.3453	1024 cells
w	912	8 386 560	9193		
x	1140	8 386 560	7355		
y	2301	8 386 390	3643		
z	1837	8 386 560	4565		

Furthermore, we have measured both single sources and representative traffic aggregates after different number of multiplexing stages. Finally, there were various ATM switches involved in my tests (AT&T, 3COM, Ericsson, Fore).