Teletraffic analysis of ATM systems : symposium gehouden aan de Technische Universiteit Eindhoven op 15 februari 1993

Teletraffic Analysis of ATM Systems

Symposium gehouden aan de
Technische Universiteit Eindhoven
op 15 februari 1993

Organisatie:
Vakgroep Digitale Informatiesystemen
Faculteit Elektrotechniek
Technische Universiteit Eindhoven

Redactie:
Prof. ir. F. van den Dool
Ir. B.J. van Rijnsoever
Ir. M.J.M. van Weert

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Teletraffic

Teletraffic analysis of ATM systems : symposium gehouden
aan de Technische Universiteit Eindhoven op 15 februari
1993 / red. F. van den Dool, B.J. van Rijnsoever, M.J.M.
van Weert. - Eindhoven : Faculteit Elektrotechniek,
Technische Universiteit Eindhoven. - Fig., tab.
Symposium georganiseerd door: Vakgroep Digitale
Informatiesystemen, Faculteit Elektrotechniek, Technische
Universiteit Eindhoven. - Met lit. opg., reg.
ISBN 90-6144-994-4
NUGI 832
Trefw.: geïntegreerde communicatienetwerken /
telecommunicatieverkeer

Inhoudsopgave

Inleiding door de dagvoorzitter
F. van den Dool, PTT Research en Technische Universiteit Eindhoven

Heuristics for Finite-Buffer Queues
H.C. Tijms, Vrije Universiteit Amsterdam

A Simple Delay Variance Formula for ATM Queueing Analysis
G.A. Awater, Technische Universiteit Delft

An Approximate Model for the End-to-End Performance in an ATM Network
B.J. van Rijnsoever, Technische Universiteit Eindhoven

Optimization of ATM LDOLL Queueing in Case of Multiple Outlets
B. Stavrov, Technische Universiteit Delft

Queueing Analysis of an ATM Switch with Correlated Routing
H. Bruneel en S. Wittevrongel, Rijksuniversiteit Gent

On ATM Traffic Control
G.H. Petit, Alcatel-BELL Telephone

The Safety Margin in the Leaky Bucket Policing Function
M.J.G. Dirksen, PTT Research

Performance Analysis of a MAC Protocol for a Broadband Network Access Facility
C. Blondia, Katholieke Universiteit Nijmegen

Deelnemerslijst

INTRODUCTION
Symposium 'Teletraffic analysis of ATM systems'

F. van den Dool
Eindhoven University of Technology & PTT Research, Leidschendam

Introduction

This is the second Symposium on 'Teletraffic Analysis of ATM systems'. The problems that are facing us in ATM are attracting attention in various places throughout Belgium and the Netherlands. Furthermore people from different disciplines are working in this field. Therefore it is useful to facilitate that people meet and become aware of each others work. On the one hand this may prevent overlap, on the other hand collaboration may be the result. This is the main goal of the symposium.

The first symposium took place about 1½ year ago at PTT Research facilities in Leidschendam with the title 'Performance aspects of ATM'. The reactions were very positive. This stimulated the organisation of this second symposium.

ATM State of the art

We are now exactly 10 years from the advent of ATM. It was in 1983 that the first proposals for ATM related techniques were introduced, almost at the same time in France [Coudreuse 1983] and the United States [Turner 1983]. A lot has happened in these ten years. The unworldly technicians advocating ATM in the beginning are now joined by respectable telecommunication managers. Trials are being carried out on a large scale. A striking example of this is the European trial between (provisionally) 5 network operators. Also plans to introduce ATM in the operational network are becoming more and more serious: Nynex and IBM have announced plans to start introduction of ATM in the network within two years.

In the mean time also terminal and LAN manufacturers have recognized the possibilities of ATM. As a result of this, tumultuous developments have taken place in the area of cell based LANs. Although transfer methods of cell based LANs may be different from ATM, the cell definition of most proposals are in accordance with that of CCITT in order to facilitate the interworking with ATM (B-ISDN). An important development is that a number of companies have founded the ATM Forum. This Forum was created to stimulate the development and introduction of ATM products and services. In August 1992 already 52 principal members were involved.

	Release 1	*Release 2*	*Release 3*
Recommendations by:	1993	1995	1997
Network capabilities:	point-to-point connections	simple point-to-multi-point connections	broadcast connections
Bandwidth allocation based on:	peak rate	traffic characteristics enabling statistical multiplexing	← as in release 2
Signalling:	extension of ISDN signalling, B-ISUP and Q.93B	separation call and connection control, BAP	← as in release 2

Table 1: The three releases of B-ISDN protocols

For this symposium it is worth considering the situation of ATM standardization. The scheduling of traffic control issues is aligned with other standardization activities of B-ISDN capabilities, including signalling. Three releases of B-ISDN Recommendations have been identified: a short term solution (release 1), a medium term solution (release 2) and a long term (target) solution (release 3). Some important aspects of the three releases are presented in Table 1.

Traffic control is still one of the basic issues in ATM. The text on traffic control in the so called B-ISDN release 1 is now stable (CCITT 1992). In this release, algorithms for Usage Parameter Control (UPC) are not standardized, although two examples, the 'virtual spacer' and the 'continuous state leaky bucket' algorithm, are described in the Annex to the Recommendation I.371. However there is a growing consensus that this should be the case, or at least that parameters for traffic characterization should be defined in later releases.

The text on Connection Acceptance Control (CAC) in the draft Recommendation I.371 is very general and it is not to be expected that procedures will be standardized. However for the dimensioning and operation of the network these issues remain of vital importance.

The Symposium

The contributions in this symposium highlight the fact that people from a number of disciplines are looking at ATM because of the interesting problems that it presents. The contributions range from studies on the behaviour of individual queues, stand alone or in more complicated arrangements including queues in tandem, to the effects these studies might have on traffic control issues.

With respect to individual queue behaviour, the following topics are presented:

- 'Heuristics for loss probabilities in finite-buffer queues',
  H.C. Tijms, Vrije Universiteit, Amsterdam.
- 'A simple delay variance formula for ATM queuing analysis',
  G.A. Awater, Delft University of Technology

More complicated arrangements, e.g. in a switch, are highlighted with respect to:

- 'Optimization of ATM LDOLL queuing in case of multiple outlets',
  B. Stravrov, Delft University of Technology.
- 'Queuing analysis of an ATM switch with correlated routing',
  H. Bruneel and S. Wittevrongel, University of Ghent.

The behaviour of a number of queues in tandem is addressed in:

- 'Approximate end-to-end performance models in ATM networks',
  B.J. van Rijnsoever, Eindhoven University of Technology.

Using the results from queuing analysis, network operation like UPC and CAC are discussed in:

- 'Traffic control issues in ATM',
  G.H. Petit, Alcatel-Bell Antwerp.
- 'The safety margin in the leaky bucket policing function',
  M.J.G. Dirksen, PTT Research, Leidschendam.

Finally the performance of a MAC protocol for an ATM Passive Optical Network is analyzed in:

- 'Performance analysis of a MAC protocol for a broadband network access facility',
  C. Blondia et al, University of Nijmegen.

In this way a very broad spectrum of contributions will be presented, however all related to the central theme ATM. We trust that a lot of discussion will be the result of these stimulating presentations.

Acknowledgement

The author would like to thank J.C. van der Wal for his contribution on the status of ATM standardization activities.

References

[CCITT 1992] 'Traffic control and congestion control in B-ISDN', CCITT draft Recommendation I.371, Geneva 1992.

[Coudreuse 1983] 'Les reseaux temporel asynchrones: du transfert de données à l'image animée', Coudreuse J.P., L'echo des recherches, No 112, 2e trimestre 1983, pp 33-40.

[Turner 1983] 'A packet network architecture for integrated services', Turner J.S. and Wyatt L.F., Proceedings of Globecom 83, november 1983, pp 45-50.

HEURISTICS FOR FINITE-BUFFER QUEUES

by

Henk Tijms
Dept. of Econometrics, Vrije Universiteit, Amsterdam

Abstract The approximation of the loss probability in finite-buffer queues is a practical problem of considerable interest. A new heuristic is given. This heuristic is based on the equilibrium distribution of the corresponding infinite-buffer queue. Numerical results show an excellent performance of the heuristic.

Research Report 1991-29

1. Result

A practical problem of considerable interest is the calculation of the loss probability in a finite-capacity queueing system. Consider the GI/G/c/N+c queue with c servers and a finite buffer of capacity N, where any customer finding upon arrival N+c other customers in the system is lost. It is assumed that the traffic intensity $\rho = \lambda\mu/c$ is less than 1, where λ is the average arrival rate of customers and μ is the average service time per customer. The assumption $\rho<1$ guarantees that the corresponding infinite-capacity queue GI/G/c has an equilibrium probability distribution $\{\pi_j^{(\infty)}, j=0,1,\ldots\}$ with $\pi_j^{(\infty)}$ denoting the long-run fraction of customers finding upon arrival j other customers present.

A common heuristic for obtaining the loss probability is:

$$P_{appl} = \sum_{j=N+c}^{\infty} \pi_j^{(\infty)}, \tag{1}$$

i.e., the steady-state probability that in the infinite-capacity queue a customer finds upon arrival N+c or more other customers present. A more refined heuristic improving the crude heuristic (1) is

$$P_{app2} = \frac{\pi_{N+c}^{(\infty)}}{1 - \sum_{j>N+c} \pi_j^{(\infty)}}, \tag{2}$$

i.e., the conditional steady-state probability that in the infinite-capacity queue an arrival sees N+c other customers given that no more than N+c customers are present.

A quantitative justification of the heuristic (2) was given in Schweitzer and Konheim [1]. In particular, it was shown in this reference that for the loss probability the exact formula

$$P_{loss} = \frac{\lambda_{N+c}\, p_{N+c}^{(\infty)}}{\sum_{k=0}^{N+c} \lambda_k\, p_k^{(\infty)}}$$

applies to queueing systems with state-dependent Poisson input and exponential services, where λ_j is the arrival rate when j customers are present and $p_j^{(\infty)}$ denotes the time-average probability of having j customers present in the corresponding infinite-capacity queue. In case the arrival process is a Poisson process the formula for P_{loss} reduces to (2). Note that the heuristic (2) is exact for the M/M/c/N+c queue.

The purpose of this paper is to present a third heuristic that is in general an improvement upon the heuristic (2). The new heuristic is given by:

$$P_{app3} = \frac{(1-\rho) \sum_{j=N+c}^{\infty} \pi_j^{(\infty)}}{1-\rho \sum_{j=N+c}^{\infty} \pi_j^{(\infty)}} . \qquad (3)$$

What is the rationale behind this heuristic? For the two particular queueing systems M/G/c/N+c with c=1 and M/M/c/N+c, the long-run fraction of time the system is full equals:

$$\frac{(1-\rho) \sum_{j=N+c}^{\infty} p_j^{(\infty)}}{1 - \rho \sum_{j=N+c}^{\infty} p_j^{(\infty)}} .$$

A unifying proof of this known result is given in the appendix, see also chapter 4 in Tijms [3]. The key element of the proof is to establish that the first N+c-1 state probabilities in the finite-capacity model are proportional to the corresponding probabilities in the infinite-capacity model. Using the above result and using twice the property that Poisson arrivals see time averages, the heuristic (3) follows. It is emphasized that the heuristic (3) uses the customer-average probabilities rather than the time-average probabilities.

Heuristics involving only the equilibrium probabilities for the infinite-capacity model are very useful for practical purposes, since these probabilities have to be computed only once. This is particularly convenient when selecting buffer sizes. Moreover, the customer-average probabilities for the infinite-capacity model are often easier to compute than the corresponding probabilities for the finite capacity model. A further simplification of the heuristics is possible when

$$\pi_j^{(\infty)} \sim \alpha\tau^j \quad \text{for } j \text{ large enough,} \tag{4}$$

for some constants $\alpha>0$ and $0<\tau<1$. This asymptotic expansion holds true in many queueing systems. Then the decay factor τ can often be computed simply as the root of a nonlinear equation in a single variable. The amplitude factor α is usually not so easy to obtain with the exception of the single-server queue with Poisson arrivals and the multi-server queue with exponential services. However, an approximation or simulation may be used to compute the amplitude factor α.

A numerical investigation of the three heuristics is given in Section 2. The numerical results show that the new heuristic (3) performs very well for practical purposes and usually improves the heuristic (2).

2. Numerical Results

A good battlefield for the heuristics (1), (2) and (3) are queueing systems with deterministic arrivals. In table 1 we give the approximate values (1)-(3) and the exact value of the loss probability for several single-server D/G/1/N+1 queues with Erlang-k services. Similarly, table 2 deals with the multi-server D/M/c/N+c queue. For both queueing systems the exact value of the loss probability was obtained by solving the balance equations of an embedded Markov chain. The infinite capacity versions of these queueing systems are easy to solve. For the D/M/c queue the state probability $\pi_j^{(\infty)}$ equals $\gamma\beta^{j+1-c}$ for $j\geq c-1$, where the constants γ and β are very easy to compute using explicit expressions for the G/M/c queue obtained in Takács [2]. A simple and fast algorithm to compute the state probabilities $\pi_j^{(\infty)}$ for the $D/E_k/1$ queue is given in Tijms and Van de Coevering [5]. This algorithm exploits the asymptotic expansion (4).

Table 1. The loss probability for the $D/E_k/1/N+1$ queue

		ρ=0.8		ρ=0.9		ρ=0.95	
		N=3	N=8	N=5	N=15	N=10	N=35
E_4	appl	1.87×10^{-3}	1.73×10^{-7}	9.84×10^{-3}	1.84×10^{-6}	1.36×10^{-2}	4.36×10^{-7}
	app2	1.57×10^{-3}	1.46×10^{-7}	5.69×10^{-3}	1.06×10^{-6}	4.65×10^{-3}	1.48×10^{-7}
	app3	3.74×10^{-4}	3.47×10^{-8}	9.92×10^{-4}	1.84×10^{-7}	6.88×10^{-4}	2.18×10^{-8}
	exact	5.50×10^{-4}	5.09×10^{-8}	1.23×10^{-3}	2.28×10^{-7}	7.69×10^{-4}	2.43×10^{-8}
E_2	appl	3.51×10^{-2}	3.38×10^{-4}	9.01×10^{-2}	1.23×10^{-3}	1.11×10^{-1}	6.30×10^{-4}
	app2	2.15×10^{-2}	2.05×10^{-4}	3.34×10^{-2}	4.31×10^{-4}	2.29×10^{-2}	1.18×10^{-4}
	app3	7.22×10^{-3}	6.76×10^{-5}	9.81×10^{-3}	1.23×10^{-4}	6.22×10^{-3}	3.15×10^{-5}
	exact	8.63×10^{-3}	8.05×10^{-5}	1.08×10^{-2}	1.35×10^{-4}	6.54×10^{-3}	3.30×10^{-5}
E_1	appl	1.56×10^{-1}	1.53×10^{-2}	2.76×10^{-1}	3.23×10^{-2}	3.20×10^{-1}	2.41×10^{-2}
	app2	6.43×10^{-2}	5.75×10^{-3}	6.86×10^{-2}	6.40×10^{-3}	4.43×10^{-2}	2.42×10^{-3}
	app3	3.57×10^{-2}	3.10×10^{-3}	3.67×10^{-2}	3.33×10^{-3}	2.30×10^{-2}	1.23×10^{-3}
	exact	3.86×10^{-2}	3.33×10^{-3}	3.84×10^{-2}	3.44×10^{-3}	2.36×10^{-2}	1.26×10^{-3}

The final table 3 gives the approximate values and the exact value of the loss probability for the M/G/c/N+c queue. Here we substituted in the approximations (1)-(3) the asymptotic expansion (4) for the state probabilities $\pi_j^{(\infty)}$, where the decay factor τ was computed from the characteristic equation for the M/G/c queue and the amplitude factor α was computed from a simple approximation, see pp. 350-351 in Tijms [3]. For small loss probabilities and nonlight traffic, the use of the asymptotic expansion is justified. It should be noted that the larger the traffic intensity ρ, the earlier the asymptotic expansion (4) applies. The calculations were done for Erlang-2 services (c_s^2=0.5) and hyperexponential services with the normalization of balanced means (c_s^2=2, 4). Here c_s^2 denotes the squared coefficient of variation of the service time.

Table 2. The loss probability for the D/M/c/N+c queue.

		ρ=0.8		ρ=0.9		ρ=0.95	
		N=5	N=30	N=10	N=70	N=15	N=140
c=5	appl	3.55×10^{-2}	3.24×10^{-7}	7.42×10^{-2}	1.90×10^{-7}	1.71×10^{-1}	4.11×10^{-7}
	app2	1.35×10^{-2}	1.20×10^{-7}	1.52×10^{-2}	3.68×10^{-7}	1.98×10^{-2}	4.05×10^{-8}
	app3	7.31×10^{-3}	6.47×10^{-8}	7.95×10^{-3}	1.90×10^{-8}	1.02×10^{-2}	2.06×10^{-8}
	exact	7.84×10^{-3}	6.93×10^{-8}	8.24×10^{-3}	1.97×10^{-8}	1.04×10^{-2}	2.09×10^{-8}
c=10	appl	2.21×10^{-2}	2.01×10^{-7}	6.11×10^{-2}	1.57×10^{-7}	1.56×10^{-1}	3.76×10^{-7}
	app2	8.31×10^{-3}	7.47×10^{-8}	1.24×10^{-2}	3.03×10^{-8}	1.79×10^{-2}	3.70×10^{-8}
	app3	4.49×10^{-3}	4.03×10^{-8}	6.46×10^{-3}	1.57×10^{-8}	9.16×10^{-3}	1.88×10^{-8}
	exact	4.82×10^{-3}	4.31×10^{-8}	6.70×10^{-3}	1.62×10^{-8}	9.34×10^{-3}	1.91×10^{-8}
c=25	appl	7.35×10^{-3}	6.70×10^{-8}	4.03×10^{-2}	1.03×10^{-7}	1.30×10^{-1}	3.13×10^{-7}
	app2	2.74×10^{-3}	2.49×10^{-8}	8.03×10^{-3}	1.99×10^{-8}	1.45×10^{-2}	3.08×10^{-8}
	app3	1.48×10^{-3}	1.34×10^{-8}	4.18×10^{-3}	1.03×10^{-8}	7.42×10^{-3}	1.57×10^{-8}
	exact	1.59×10^{-3}	1.44×10^{-8}	4.33×10^{-3}	1.07×10^{-8}	7.56×10^{-3}	1.59×10^{-8}

Table 3 The loss probability for the M/G/c/N+c queue with c=10.

		ρ=0.8		ρ=0.9		ρ=0.95	
		N=10	N=25	N=20	N=50	N=25	N=100
c_s^2=0.5	appl	2.37×10^{-2}	2.89×10^{-4}	4.26×10^{-2}	6.45×10^{-4}	1.53×10^{-1}	9.18×10^{-4}
	app2	6.15×10^{-3}	7.35×10^{-5}	5.77×10^{-3}	8.42×10^{-5}	1.18×10^{-2}	6.06×10^{-5}
	app3	4.84×10^{-3}	5.78×10^{-5}	4.43×10^{-3}	6.46×10^{-5}	8.94×10^{-3}	4.60×10^{-5}
	exact	5.12×10^{-3}	6.11×10^{-5}	4.55×10^{-3}	6.62×10^{-5}	9.08×10^{-3}	4.67×10^{-5}
c_s^2=2	appl	8.20×10^{-2}	9.95×10^{-3}	1.56×10^{-1}	1.96×10^{-2}	3.42×10^{-1}	2.69×10^{-2}
	app2	1.18×10^{-2}	1.28×10^{-3}	1.22×10^{-2}	1.34×10^{-3}	1.70×10^{-2}	9.19×10^{-4}
	app3	1.75×10^{-2}	1.90×10^{-3}	1.81×10^{-2}	2.00×10^{-3}	2.53×10^{-2}	1.38×10^{-3}
	exact	1.46×10^{-2}	1.51×10^{-3}	1.63×10^{-2}	1.82×10^{-3}	2.37×10^{-2}	1.32×10^{-3}
c_s^2=4	appl	1.33×10^{-1}	3.73×10^{-2}	2.60×10^{-1}	7.59×10^{-2}	5.56×10^{-1}	1.22×10^{-1}
	app2	1.24×10^{-2}	3.14×10^{-3}	1.39×10^{-2}	3.29×10^{-3}	2.44×10^{-2}	2.77×10^{-3}
	app3	2.98×10^{-2}	7.69×10^{-3}	3.39×10^{-2}	8.15×10^{-3}	5.89×10^{-2}	6.90×10^{-3}
	exact	2.14×10^{-2}	5.10×10^{-3}	2.72×10^{-2}	6.70×10^{-3}	3.64×10^{-2}	5.18×10^{-3}

Summarizing, the numerical results indicate that the new heuristic (3) performs very well for practical purposes. Also, note that the crude heuristic (1) commonly used in practice may perform very poorly. In all cases the value of the new heuristic is of the same order of magnitude as the exact value of the loss probability. This is what is typically needed when a heuristic is used for dimensioning the buffer size.

References

1. Schweitzer, P.J. and Konheim, A. (1978). Buffer overflow calculations using an infinite-capacity model. Stoch. Proc. and Appl., 6: 267-276.
2. Takacs, L. (1962). *Introduction to the Theory of Queues*, Oxford University Press, New York.
3. Tijms, H.C. (1986). *Stochastic Modeling and Analysis: A Computational Approach*, Wiley, New York.
4. Tijms, H.C and Van Ommeren, J.C.W. (1989). Asymptotic analysis for buffer behavior in communication systems, Prob. Eng. Inf. Sci., 3: 1-12.
5. Tijms, H.C. and Van de Coevering, M.C. (1991). A simple numerical approach for solving infinite Markov chains, Prob. Eng. Inf. Sci., 5: 285-295.
6. Wolff, R.W. (1982). Poisson arrivals see time averages, Oper. Res. 30: 223-231.

A Simple Delay Variance Formula for ATM Queueing Analysis

Geert A. Awater
Delft University of Technology
Telecommunications and Traffic Control Systems Group *

Abstract

An important QoS parameter of ATM supported connections is delay jitter. One measure of jitter is the statistical variance of the delay. Often, ATM queueing analysis is performed in terms of the total number of customers in the queue, not individual waiting times. For simple queueing systems the probability distribution function of customer delays, and hence the delay variance is easily found. This is not viable anymore for the more complex service disciplines that arise in the context of prioritized queueing. Little's result provides the expected delay for almost any queueing system in terms of the expected queue population. This paper proposes a formula that gives the delay variance in terms of the population mean and population variance for a wide class of queueing systems.

1 Introduction

In this paper we will be concerned with the variable delay that ATM cells experience when they are sent through a network for fast packet switching. This delay consists of three components, the transmission delays that occur every time a cell is being put on a transmission line, the propagation delay caused by the finite speed of light and the total waiting time, which is the time that cells spend waiting for transmission links to become available.

For the lowest bitrate proposed for ATM, 149.95 Mbit/s (the SONET OC-3 payload throughput) a cell's transmission time is less than 2.85 μs. This bit rate is typically intended for access lines. Inside the network transmission time will typically be shorter. The propagation delay is determined by the speed of light, that travels 1 km at most in 3.33 μs. The mean waiting time of a cell depends on the dimensions of the buffers inside the network switches which the cell passes, and on the transmission speeds of the links that serve those buffers.

The delay jitter, on the other hand, is solely the result of the variable waiting times that cells experience. Jitter is to be kept as small as possible for real-time traffic such as voice or compressed video, where the time relation of the data has to be restored at the receiver's end. This means that delay jitter has to be filtered out without disturbing variations in the bit rate. The smaller the delay variance is, the smaller the dejitter buffers, the more simple or stable clock recovery (phase locked loop) circuitry can be.

Thus, study of the delay jitter is a queueing problem. In queueing analysis the state of the queue is preferably described by the number of cells or —in queueing theoretical terms— customers in the queue, rather than by the individual waiting times of the customers. As a result, waiting time statistics are not obtained directly from the analysis. For simple queueing systems the waiting time distribution is almost always found easily from the customer distribution. In these cases the total workload which has to be processed before the cell under consideration (or *tagged* cell) can be transmitted, is straightforwardly obtained from the number of customers in the system upon the cell's arrival. If the governing arrival process is assumed to be memoryless, the individual customers have a 'random view' [6, p. 175], [10] of the system, and thus we arrive at the individual waiting time distribution.

As more and more authors indicate the necessity of prioritized buffering in ATM systems, ATM queueing analysis becomes increasingly difficult. More specifically; the waiting time of a customer arriving at a

*Rm. 19.11, P.O. Box 5031, 2600 GA Delft, The Netherlands. e-mail: g.a.awater@et.tudelft.nl

queue comes to depend on future arrivals, since cells of a higher priority bypass the tagged cell. Now, it is no longer easy —if possible at all— to obtain the waiting time distribution. For cases like this it is convenient to have a simple formula that translates the customer distribution into waiting time variance, just like Little's law provides the expected delay from the expected queue population, without the need to obtain the complete delay distribution.

To the author's knowledge, the only extensions to Little's law that obtain moments of the waiting time of customers from moments of the queue population are concerned with queues that are fed with a non-compound, memoryless customer arrival process. Whitt [9] gives a broad review of these and other, interesting extensions of Little's law. Keilson and Servi study the conditions under which a distributional form of Little's law applies in [4]. One predominant condition is that the customer (cell) arrival process has to be Poissonian. Although they focus on continuous time systems, their results are easily transferred to the discrete time domain. The complication of queueing analysis in the ATM context, is that at ATM buffers, typically, multiple cells can arrive in a single time slot. This means that we have a compound arrival process, where customers may arrive in batches. In this paper we shall derive a formula that provides a relationship between the variance of the waiting time of customers and the first two moments of the customer population for discrete time queues with a memoryless, compound arrival process.

The main part of this paper is concerned with the derivation of the formula. Section 2 introduces some conventions and techniques that are required for this derivation. Section 3 discusses an extension to the principles of discrete time embedded Markov chains, if multiple arrivals per time slot —which is common practice in ATM queueing systems— are possible. In section 4 we relate the probability generating function (PGF) of the waiting time with that of the queue population. This relation allows us to relate their moments, which is demonstrated in section 5. Finally, in section 6 the formula is applied to a sample queueing system for which no explicit waiting time density function has been found yet.

2 Theoretical preliminaries

We will start this section by defining the queueing system and introducing some random variables. First we will narrow down the analysis to discrete time systems, since that is most appropriate for ATM systems. However, the analysis is readily extendible to continuous time systems. Time is marked with the positive integer numbers $t, t \in \{0, 1, \ldots\}$. We will refer to the number of customers in system at time mark t as the queue population, represented by the random variable $Q(t)$. The equilibrium random variable Q is defined implicitly by its distribution:

$$\Pr\{Q = q\} = \lim_{t \to \infty} \Pr\{Q(t) = q\}, \quad q = 0, 1, \ldots$$

Every customer has its own sojourn time random variable. With the equilibrium sojourn time W we mean the sojourn time of a customer that approaches the system in equilibrium, i.e. after an infinitely long time after the queueing system started operating. The queueing system that we will be studying complies to the following definitions:

Arrival process: Every time mark $t \in \{0, 1, 2, \ldots\}$, the number of customers that arrive is $A(t)$. The distribution of this random variable is independent of t, say $\Pr\{A(t) = k\} = a_k$. Some authors call this process Batch Geometric, referring to the geometric distribution of the inter-arrival time (with mean $((1 - a_0)^{-1})$ of batches of customers (with $\Pr\{\text{batch size} = k\} = a_k/(1 - a_0)$). We choose however, for the name Compound Binomial, which is motivated by its continuous time equivalent being called Compound Poisson process, where batches of customers with a general distribution arrive according to a Poisson point process. In the discrete time situation, batches arrive according to a binomial process (the number of batches arriving in an arbitrary time interval is binomially distributed) with arrival probability $1 - a_0$ on every time mark.

Enqueueing: We confine the derivation in this paper to lossless queueing systems. This means that every arriving customer is allowed to enter the queue, and once in it, will not be lost, which implies that the queueing system must provide infinite waiting room. Customers that arrive at the same time mark are enqueued in random order.

Departures: The system has a single exit, through which at most one customer at each time mark t departs from the queue. Customers leave the queue in the order that they entered it (First In, First Out). It is possible that a customer arriving at t, departs immediately.

This definition does not exclude multiple-priority queues, or multiple class queueing systems in gen-

eral. A single class of a multiple-class system, can be viewed as a system in itself, to which the equations may apply equally well. We also mention that this queueing system is much more restricted than is necessary for the derivation of Little's law, where the only requirement is that no customers are lost in the system. However, the class of systems that the analysis applies to is still very broad.

We define PGF, or z-transform of a discrete random variable X with probability distribution $\Pr\{X = k\} = x_k$ as usual: $X(z) = \sum_{k=0}^{\infty} x_k z^k$. We will make use of its factorial moment generating property:

$$X^{(n)}(1) = \mathrm{E}\{X(X-1)\cdots(X-(n-1))\} \qquad (1)$$

where $X^{(n)}(1)$ denotes the n-th derivative of $X(z)$ evaluated in $z = 1$ and $n = 1, 2, \ldots$

An important PGF in this paper, is that of a sum, say Z, of a random number Y of discrete random variables $X_1, \ldots, X_Y$:

$$Z = \sum_{i=1}^{Y} X_i,$$

where the X_i are i.i.d. random variables with PGF $X(z)$ and where Y is an independent random variable with PGF $Y(z)$. Clearly, if Y would be deterministic, say $Y = k$, then $Z(z) = (X(z))^k$, because of the convolution property (see [6, p. 330]); the PGF of a sum of independent random variables is the product of their PGFs. However, since Y is a random variable we have to uncondition on the various values that Y can assume. This gives:

$$Z(z) = \sum_{k=0}^{\infty} (X(z))^k \cdot \Pr\{Y = k\} = Y(X(z)). \qquad (2)$$

Another prerequisite for the analysis is the PGF of random variable B, that represents the number of customers that arrive in the same batch, i.e. at the same time mark as the tagged customer, but are enqueued behind it. We define $\Pr\{B = k\} = b_k$. The order in which customers of a batch are enqueued is random. The probability that a tagged customer arrives in a batch of i customers is $ia_i/\mathrm{E}\{A\}$. With probability $1/i$ it has exactly k customers behind it, provided that the batch size is larger than k. Therefore, we find with the definition of arrival intensity $\lambda = \mathrm{E}\{A\}$, that (as in [3]):

$$b_k = \sum_{i=k+1}^{\infty} \frac{1}{i} \frac{ia_i}{\mathrm{E}\{A\}} = \frac{1}{\lambda} \sum_{i=k+1}^{\infty} a_i \qquad (3)$$

For the PGF of this random variable we get, using $B(z) = \sum_{k=0}^{\infty} b_k z^k$ and changing the order of summation:

$$B(z) = \frac{1}{\lambda} \sum_{k=0}^{\infty} \frac{1-z^k}{1-z} a_k = \frac{1-A(z)}{\lambda(1-z)}. \qquad (4)$$

The n-th order derivative $B(z)$ in the point $z = 1$ can be found directly by substituting for $A(z)$ its series expansion, which does always exist, since all moments of the random variable A are known.

$$\begin{aligned} B(z) &= \frac{-1}{\lambda(1-z)} \sum_{m=1}^{\infty} \frac{(z-1)^m}{m!} A^{(m)}(1) \\ &= \frac{1}{\lambda} \sum_{n=0}^{\infty} \frac{(z-1)^n}{n!} \frac{A^{(n+1)}(1)}{n+1} \end{aligned} \qquad (5)$$

By inspection we find for the n-th derivative of $B(z)$ in $z = 1$:

$$B^{(n)}(1) = \frac{A^{(n+1)}(1)}{(n+1)\lambda}, \qquad n = 0, 1, \ldots \qquad (6)$$

Another equality that we will be needing in the next section is the discrete time analog of the result that has become known as PASTA (cf. [10]). This acronym stands for 'Poisson Arrivals See Time Averages', meaning that the equilibrium probability that a customer, that arrives according to a (possibly non-stationary, possibly compound) Poisson process, finds a system, with which the customers may interact to be in a particular state, is equal to the fraction of time that the system is in that state. This state does not necessarily have to be the number of customers in the system. This analog, that is due to Halfin (see [2]) is sometimes called BASTA, which may stand for Binomial Arrivals See Time Averages.

3 What you see is not always what you leave

In the study of queueing systems, and as it turns out also in this paper, it is often more convenient to model the queue population immediately after the departure of a customer. The random variable that symbolizes the number of customers in the queue immediately after departure of a customer, will be called U. More colloquially we will refer to this number as 'What you leave'. For the counterpart of this variable, i.e. the number of customers immediately prior to the arrival of a new customer, we adopt the term 'What you

see'. Thanks to the forementioned BASTA property, we do not need to introduce yet another random variable here, since Q has the identical distribution as the queue population that arriving batches see.

For queueing systems at which at most one customer arrives at a time and one or less departs at a time, Kendall (cf. [5]) showed that Q and U have identical distributions. The pleasant consequence of this equality, together with the BASTA result, is once we find U, we also obtain the equilibrium probability distribution of the queue population, that would have been much harder to find directly. In [7] this fact is used to carry out a discrete time queueing analysis.

Kendall's principle may become intuitively clear from a glance at figure 1. There, the number of customers

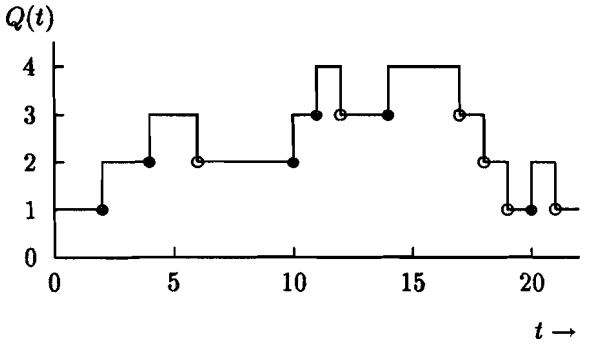


Figure 1: 'What you see' (•) is 'what you leave' (○)

an in exemplary queueing system is plotted as a function of discrete time. The numbers of customers in the queue that arriving batches see, are marked with a solid bullet (•). The number of customers that departing customers leave behind in the queue is designated with an open bullet (○). One can see that, in the long run every • matches with a ○, and hence that Q and U are distributed identically. For a formal proof, we refer to [5] or [6, p. 232, ex. 5.6].

Figure 2 shows that if we permit queue population increments greater than one, but at the same time do not allow more than unit decrements, that customers do not see what they leave anymore. Fortunately we are able to deduct from the same picture that the number of customers that an arriving customer sees, *plus* the number of customers from the same batch that it sees enqueued behind itself does have the same equilibrium distribution as the number of customers that departing customers leave behind, i.e. every ○ corresponds to a □ in the long run. Since the number of customers in a batch, and hence also the number that is enqueued behind a tagged customer is independent of Q, we may use the convolution property

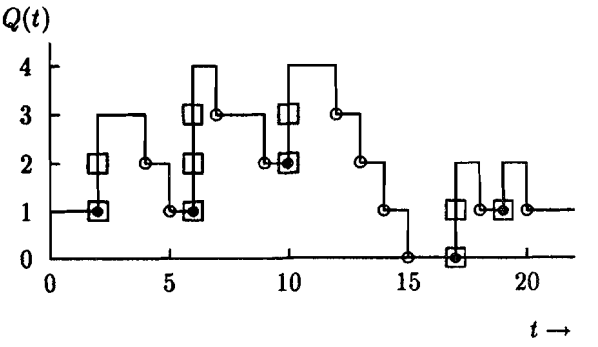


Figure 2: 'What you see' (•) is not 'what you leave' (○) for more than unit population increments. The □ depict 'what you see' *plus* the number of customers enqueued behind 'you'.

again. Hence we find that what you see and what you leave are related through their PGFs as follows:

$$U(z) = Q(z)B(z). \tag{7}$$

We have now all necessary tools to proceed. The preliminaries in section 2 will help to relate the PGF of the sojourn time with $U(z)$. Formula 7 provides the link to go from $U(z)$ to $Q(z)$. From PGFs we can go to moments using equation 1.

4 Population and Delay PGF

Here we want to relate the PGF of the sojourn time $W(z)$ to $U(z)$. Unfortunately this is not possible in a straightforward manner as Kleinrock [6, eqn. 5.98] or Keilson and Servi [4] do for continuous time, because here we have arrivals in batches. To deal with this complication, we have to divide the customers in classes. On entry in the queueing system, a customer is assigned to class k if it was enqueued with k customers of its own batch behind it. Each of these classes has its own sojourn time random variable W_k, queue population Q_k and traffic load λ_k. The λ_k are equal to the probability that more than k customers arrive at a time mark, so:

$$\begin{aligned} \lambda_k &= \mathrm{E}\{\text{number of class-}k\text{ arrivals at } t\} \\ &= \sum_{i=k+1}^{\infty} a_i = \lambda b_k. \end{aligned} \tag{8}$$

Where the number of class-k arrivals can be 0 or 1. Now it has become possible to give an expression for $U(z)$ directly, if we observe that the number of customers left behind in the queue by a tagged customer

of class k is equal to the number of customers that were enqueued behind it initially, which is k, plus the number of customers that arrived during the sojourn time of that customer. The k customers that got enqueued behind the tagged customer in the same batch introduce the degenerated PGF z^k. Using equation 2 we see that the number of customers that arrived during the sojourn time of our tagged class-k customer has PGF $W_k(A(z))$. We use the convolution theorem once more and then uncondition on the customer class, which results in:

$$U(z) = \sum_{k=0}^{\infty} z^k W_k(A(z)) b_k. \qquad (9)$$

5 Waiting time variance

By differentiating equation 9 we hope to arrive at a relationship between the variances of W and Q. Just as a check we differentiate once and evaluate the result in the point $z = 1$. Equation 1 implies particularly that $X'(1) = \mathrm{E}\{X\}$. Thereby we arrive at:

$$\begin{aligned} \mathrm{E}\{U\} &= \sum_{k=0}^{\infty} \mathrm{E}\{W_k\}\lambda b_k + \sum_{k=0}^{\infty} k b_k \\ &= \lambda \mathrm{E}\{W\} + \mathrm{E}\{B\} \qquad (10) \end{aligned}$$

Application of the same procedure to (7) yields:

$$\mathrm{E}\{U\} = \mathrm{E}\{Q\} + \mathrm{E}\{B\}. \qquad (11)$$

Eliminating $\mathrm{E}\{U\}$ from the latter two formulae, gives us $\mathrm{E}\{Q\} = \lambda \cdot \mathrm{E}\{W\}$, in accordance with Little's law.

Now, we differentiate (9) twice and evaluate the result in $z = 1$.

$$\begin{aligned} U''(1) &= B''(1) + \lambda^2 W''(1) + \mathrm{E}\{W\}A''(1) + \\ &\quad 2\lambda \sum_{k=0}^{\infty} \mathrm{E}\{W_k\} k b_k \qquad (12) \end{aligned}$$

Then, equation 7 is dealt with in the same way, so:

$$U''(1) = Q''(1) + 2\mathrm{E}\{Q\}B'(1) + B''(1). \qquad (13)$$

Substituting (6) with $n = 1$ for $B'(1)$ in the second term we obtain using Little's law:

$$U''(1) = Q''(1) + \mathrm{E}\{W\}A''(1) + B''(1). \qquad (14)$$

Equating the right hand sides of (12) and (14), and using Little's law for class-k customers: $\mathrm{E}\{Q_k\} = \lambda_k \mathrm{E}\{W_k\}$ we find:

$$Q''(1) = \lambda^2 W''(1) + 2 \sum_{k=0}^{\infty} k \mathrm{E}\{Q_k\}. \qquad (15)$$

The last term of this equation is not easy to evaluate. As a first approximation we can assume that $\mathrm{E}\{W_k\} \approx \mathrm{E}\{W\}$. As shown in [2], this approximation is exact if the batch size distribution is geometric. This means that $\Pr\{A = k | A > 0\} = a \cdot (1-a)^k, k = 1, 2, \ldots$, for some $a \in [0, 1]$. The current approximation is at its worst when the batch arrival intensity $1 - a_0$ is very low, while at the same time the average batch size $\lambda/(1-a_0)$ is very high, so that the waiting time is highly dependent on the place in the batch.

If we substitute the approximation into equation 15 we have:

$$\begin{aligned} Q''(1) &\approx \lambda^2 W''(1) + 2\mathrm{E}\{Q\}\mathrm{E}\{B\} \\ &= \lambda^2 W''(1) + \mathrm{E}\{W\}A''(1). \qquad (16) \end{aligned}$$

The approximation can be rewritten in terms of variances as follows:

$$\mathrm{Var}\{Q\} \approx \mathrm{Var}\{\lambda W\} + \mathrm{Var}\{A\}\mathrm{E}\{W\}. \qquad (17)$$

In the appendix we pursue an exact evaluation of the term $2\sum_{k=0}^{\infty} k\mathrm{E}\{Q_k\}$. If the result obtained there is substituted into equation 15, and rewritten in terms of variances, we find that the initial approximation (17) is augmented with an extra term:

$$\begin{aligned} \mathrm{Var}\{Q\} &= \mathrm{Var}\{\lambda W\} + \mathrm{Var}\{A\}\mathrm{E}\{W\} + \\ &\quad \rho(B''(1) - 2\mathrm{E}^2\{B\}). \qquad (18) \end{aligned}$$

Where ρ is defined as the queue utilization, or equivalently, as the probability that the queue is not empty. The reader may verify, by going through some tedious calculus, that the correction term vanishes for a geometric batch size distribution.

6 An application

In this section we will study the relationship between load, mean delay and delay variance of a two-priority ATM queueing system, described in [1]. The system, called LDOLL queue, is designed to minimize the mean delay of cells of the so called Low Delay (LD) class as well as the loss probability of the other, Low Loss (LL) class. This is accomplished by giving storage priority to cells of the LL-class and retrieval priority to cells of the LD-class.

More specifically the LDOLL queue is defined as follows. Cells arrive on N synchronized input channels according to a Bernoulli process at a finite buffer with

a capacity of Q cells. The probability that a cell arrives at a time mark is p for each channel. Hence the batch size distribution is binomial:

$$\Pr\{A(t)=k\} = \binom{N}{k} p^k (1-p)^{N-k}$$
$$\Leftrightarrow$$
$$A(z) = (1-p(1-z))^N. \qquad (19)$$

The chance that an arriving customer is of the LD type is r. With probability $1-r$ it is of the LL type. As a result, both customer classes also have binomial batch size distributions.

During a time slot, in between two time marks, retrieval takes place first. Which type of cell to retrieve is prescribed by the LDOLL threshold policy, that was shown (in [1]) to minimize a linear combination of mean LD cell delay and LL cell loss probability. This retrieval policy is described using pseudo code:

```
if n_LL < Δ and n_LD > 0
  retrieve an LD cell
else
  if n_LL > 0
    retrieve an LL cell
  else
    do nothing.
```

Here n_{LD} and n_{LL} is the number of LD cells and LL cells respectively, that are enqueued currently. The threshold $\Delta \in \{1,\ldots,Q\}$ is a parameter with which we can adjust the retrieval priority, so that a trade-off between LD cell delay and LL cell loss probability can be made. An increase of Δ reduces LD cell delay and increases LL cell loss probability. For a decrease of Δ it is just the other way round. For each class, cells are retrieved in the same order in which they were stored.

Cell storage takes place after cell retrieval. The empty buffer space is alloted to the arriving LL cells first. If not all the LL cells find a place, they may push out any LD cell present in the queue, but not cells of their own type. The remaining space is for the LD cells, that do not have push-out rights. LD cells and LL cells that cannot be placed in the buffer are lost.

For this queueing system we did not succeed in finding an explicit delay probability distribution. Yet we want to be able to study the benefit of the LDOLL threshold policies for delay variance, which now has become possible with delay variance formula 18. Both cell classes comply to the definitions in section 2, except for the requirement of the absence of loss. Thus we will use (18) as an approximation. Yet this approximation is very accurate, especially if the cell loss probability does not become too high.

We will calculate the mean delay and delay variance for various loads for three different queueing systems.

1. An LDOLL queue with a queue size of $Q = 40$ cells, $N = 4$ inputs and $\Delta = 20$, for both the LD and the LL cells. Cells of both types arrive at the queue in equal proportions ($r = 0.5$).
2. A discrete time $M^X/D/1/40$ FIFO queue, with the same cell arrival process as for the LDOLL queue. The customer population of this queue has the same statistics as the aggregate number of customers in the LDOLL queue.
3. A discrete time $M^X/D/1$ FIFO queue, with the same cell arrival process as for the LDOLL queue. This system is considered in order to evaluate what the effect of cell losses on the accuracy of the delay variance formula is.

The population mean and variance of the LDOLL queue were obtained from the equilibrium distribution that was calculated by straightforward, repetitive multiplication of the (sparse) probability transition matrix with an initial population probability distribution vector. The multiplication was repeated until the population statistics converged. If only one cell type is fed into the LDOLL queue, the queue reduces to a $M^X/D/1/40$ queue. Therefore this queue could be calculated with the same model and the same technique, by setting r to 1. The population mean and variance of the $M^X/D/1$ queue were obtained by symbolic differentiation and evaluation in $z = 1$ of the PGF of the queue population (refer to [3]):

$$Q(z) = \frac{(1-\rho)\cdot(1-z)}{A(z)-z} A(z), \qquad (20)$$

where $A(z)$ is defined as in (19), and the utilization equals the offered traffic: $\rho = Np$.

The step from population mean and variance to delay mean and variance was made with delay variance formula 18, which for the case of a binomial batch size distribution looks like:

$$\mathrm{Var}\{W\} = \frac{\mathrm{Var}\{Q\} - (1-q)\cdot \mathrm{E}\{Q\}}{N^2 q^2} + \frac{(N+1)\cdot(N-1)}{6N^2} \cdot \Pr\{Q>0\} \qquad (21)$$

where q is the arrival probability per channel, for which we substituted p for the $M^X/D/1/40$ and

$M^X/D/1$ queue, and rp and $(1-r)p$ for the LD and LL cells respectively in the case of the LDOLL queue. $\Pr\{Q>0\}$ for both the LD and the LL population was obtained from the calculated probability distribution vector. For the $M^X/D/1$ queue, $\Pr\{Q>0\}=\rho$.

In figures 3, 4 and 5 the results of the calculation are plotted. The mean and variance axes are logarithmic to make small differences at low loads visible. From figure 3 we see that the mean delay of the finite $M^X/D/1$ queue does not differ substantially from that of the infinite counterpart, if the load ρ varies from 0 to 0.95. Also, we observe that the LDOLL threshold policies favor LD cells with respect to mean delay.

In figure 5 we see that exactly the same may be concluded for delay variance; LD cells have a smaller delay variance than they would have in an identical, but unprioritized queue. This reduction is paid for by the LL cells that suffer a higher delay variance, but have a smaller loss probability in return. Loss probability, which is not a subject of this paper, is discussed in references [8] and [1]. In the latter reference it was argued that a reduction of delay is accompanied by a reduction of delay variance. Figure 4 confirms this statement.

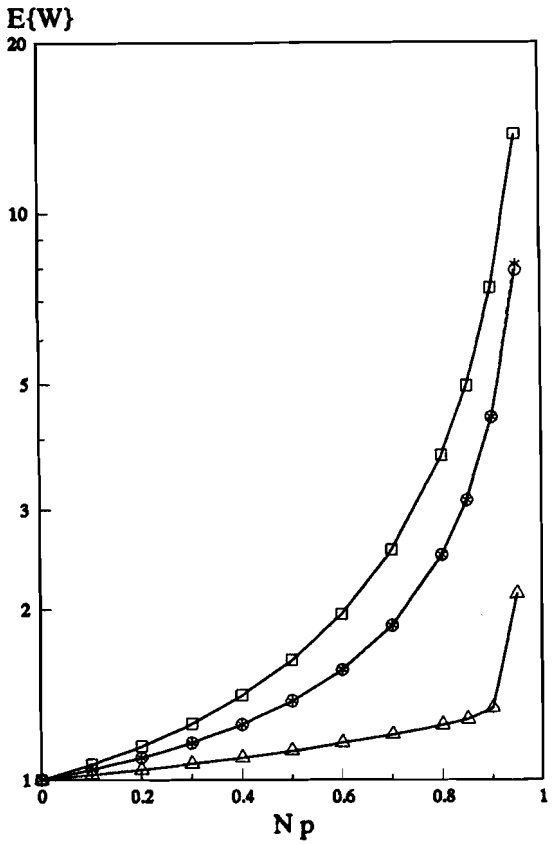


Figure 3: Mean delay versus load for the $M^X/D/1$ queue (- -∗- -), for the $M^X/D/1/40$ queue (—⊙—), for the LD cells of the LDOLL queue (—△—) and for the LL cells of the LDOLL queue (—□—).

Appendix

The queue population consists of an integral number of whole batches and one batch of which one or more customers have already left the queue. This batch we will call the residual batch. In order to assess the class-k population, we observe that, provided that the queue is not empty, the probability distribution of the number of customers in the residual batch is equal to that of B.

It is important to notice that the observation made above is not true for all queueing systems. For instance, if we consider the waiting room of a queue disregarding the server, then the first customer to enter the system after a period in which server was idle, is taken out of the queue immediately to be transferred to the server. This would alter the probability distribution of the residual batch length. In general, if the service time of a customer depends on its position in the batch, the observation would not be valid.

We define C_R as the number of customers in the residual batch. From our observation above, and accounting for a zero residual batch length if the queue is empty:

$$C_R(z) = 1 - \rho(1 - B(z)). \tag{22}$$

Let N be the total number of other batches. The number of customers in these batches is designated with the variables $C_n, n = 1, \ldots, N$. The number of class-k customers in equilibrium is found as:

$$Q_k = \sum_{n=1}^{N} I(C_n > k) + I(C_R > k), \tag{23}$$

where $I(.)$ is the indicator function, that assumes the value 1 if the parenthesized expression is true. Otherwise it takes the value 0. Note that the C_n are also i.i.d. Therefore, taking expectations on both sides yields:

$$\mathrm{E}\{Q_k\} = \mathrm{E}\{N\}\Pr\{C_n > k\} + \Pr\{C_R > k\}. \tag{24}$$

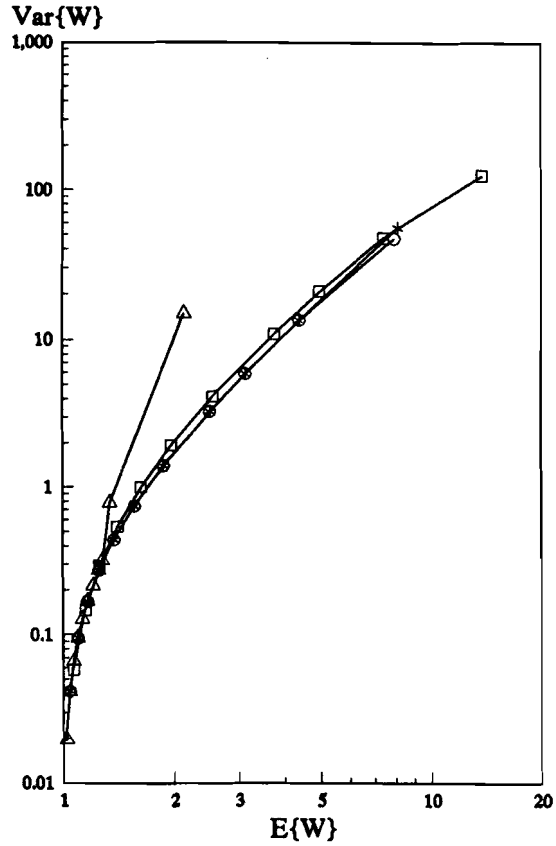


Figure 4: Delay variance versus mean delay, with load as a parameter, for the $M^X/D/1$ queue (- -∗- -), for the $M^X/D/1/40$ queue (—⊙—), for the LD cells of the LDOLL queue (—△—) and for the LL cells of the LDOLL queue (—□—).

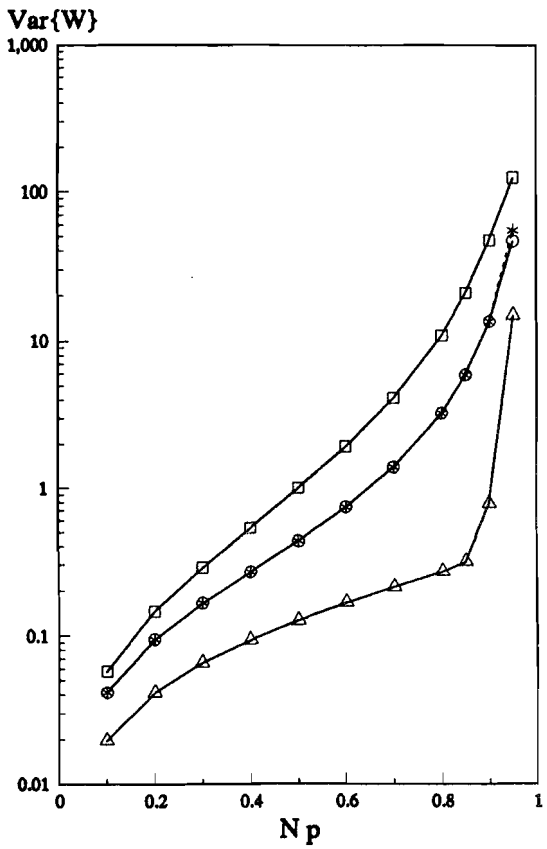


Figure 5: Delay variance versus load for the $M^X/D/1$ queue (- -∗- -), for the $M^X/D/1/40$ queue (—⊙—), for the LD cells of the LDOLL queue (—△—) and for the LL cells of the LDOLL queue (—□—).

For the batch size distribution we have (independent of n) : $\Pr\{C_n > k\} = \lambda_k/(1-a_0)$. The above equation thus can be rewritten as:

$$\mathrm{E}\{Q_k\} = \frac{\mathrm{E}\{N\}}{1-a_0}\lambda_k + \Pr\{C_R > k\}. \qquad (25)$$

Summing over k gives us:

$$\mathrm{E}\{Q\} = \frac{\mathrm{E}\{N\}}{1-a_0}\lambda + \mathrm{E}\{C_R\}. \qquad (26)$$

The latter equation will be used to find $\mathrm{E}\{N\}/(1-a_0)$ later. For that we also have to know $\mathrm{E}\{C_R\}$ which equals $\rho\mathrm{E}\{B\}$, as we see from differentiating (22). With equation 25 this we can evaluate the term we were after in the following way:

$$\begin{aligned} 2\sum_{k=0}^{\infty} k\mathrm{E}\{Q_k\} &= 2\frac{\mathrm{E}\{N\}}{1-a_0}\lambda\sum_{k=0}^{\infty} kb_k + \\ &\quad 2\sum_{k=0}^{\infty} k\Pr\{C_R > k\} \\ &= 2\frac{\mathrm{E}\{N\}}{1-a_0}\lambda\mathrm{E}\{B\} + \\ &\quad \sum_{k=0}^{\infty} k(k-1)\Pr\{C_R = k\} \\ &= 2\frac{\mathrm{E}\{N\}}{1-a_0}\lambda\mathrm{E}\{B\} + C_R''(1). \qquad (27) \end{aligned}$$

This equation is finally rewritten, with the use of

equations 26 and 22 as:

$$2\sum_{k=0}^{\infty} kE\{Q_k\} = 2(E\{Q\} - \rho E\{B\})E\{B\} + \rho B''(1). \tag{28}$$

References

[1] G.A. Awater and F.C. Schoute, "Optimal Queueing Policies for Fast Packet Switching of Mixed Traffic," IEEE J. on Selec. Areas in Commun., vol. SAC–9, no. 3, April 1991, pp. 458–466.

[2] S. Halfin, "Batch Delays versus Customer Delays," Bell System Tech. J. Vol. 62, No. 7, September 1983, pp. 2011–2015.

[3] M.J. Karol, M.G. Hluchyj and S.P. Morgan, "Input versus Output Queueing on a Space-Division Packet Switch," IEEE Trans. on Commun., Vol. COM–35, No. 12, December 1987, pp. 1347–1356.

[4] J. Keilson, L.D. Servi, "A Distributional Form of Little's Law," Oper. Research Letters 7, 1988, pp. 223–227.

[5] D.G. Kendall, "Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of the Embedded Markov Chain," Ann. Math. Stat. Vol. 24, No. 3, 1953, pp. 338–354.

[6] L. Kleinrock, "Queueing Systems," Vol. 1 : Theory, John Wiley and Sons, New York, 1975.

[7] T. Meisling, "Discrete Time Queueing Theory," Oper. Res., Vol. 6, January–February 1958, pp. 96–105.

[8] F.C. Schoute and E.G. Janssens, "A Comparison of Priority Strategies for ATM Switching," Submitted to International Conference on Integrated Communication Networks and Services, Copenhagen, Denmark, April 20–23, 1993.

[9] W. Whitt, "Two Fundamental Principles of Queueing Theory," Queueing, Performance and Control in ATM, J.W. Cohen and C.D. Pack (eds.), Workshop Proceedings of the Thirteenth International Teletraffic Congress, Copenhagen, Denmark, June 19–26, 1991, pp. 151–156.

[10] R.W. Wolff, "Poisson Arrivals See Time Averages," Oper. Res., Vol. 30, No. 2, March–April 1982, pp. 223–231.

An Approximate Model for the End-to-End Performance in an ATM Network

Bart van Rijnsoever
Eindhoven University of Technology

Abstract

In an ATM network, the network performance must be ensured to each individual virtual connection (VC). We model such a VC by a tandem queueing network of deterministic, single server queues. This model is commonly solved for the end-to-end performance by decomposing it into single queues and assuming Bernoulli cell routing. Here, the tandem queueing network is decomposed into pairs of queues, thus accounting for the dependence between adjacent queues. Furher, cell routing is modeled by an independent Markov chain, allowing some of the true routing process to be retained. Focus is on the basic building block of this method: the two tandem queues model. We show how this model can be solved by the algorithmic procedure for queues of the $M/G/1$-type. We present a numerical example, which shows that both aspects have a considerable effect on the network performance and thus deserve to be incorporated into the model.

1 Introduction

The provider of an Asynchronous Transfer Mode (ATM) Network must ensure the network performance to each individual virtual connection (VC). The network performance is measured in terms of statistics on the loss and delay of packets (cells). This paper outlines an algorithmic procedure to study for an ATM network the relation between end-to-end performance on the one hand and configuration and traffic carried on the other hand.

We customarily model the VC under study by a tandem queueing network of single server queues. Because the cell length and the transmission rate are constant throughout the network, the service processes are deterministic and mutually synchronized. At each queue, the cells on the VC under study interfere with cells on other VCs. This model is commonly solve for the end-to-end performance by assuming that the queues are mutually independent of each other. See, e.g., [1, 2]. In that case, the analysis consists of modeling the traffic streams departing from the queues and then evaluation of the performance in each queue. Another approach is to look for bounds on the performance that can easily be evaluated, see [3].

Decomposition neglects the positive correlation between the waiting times of a cell in the consecutive queues. As a consequence, the probability of large delay or of loss of consecutive cells on the VC is underestimated. Assuming Bernoulli routing of cells, instead of routing according to a cell's VC, may both increase and decrease the burstiness of the traffic stream arriving to the next queue. An increase of the burstiness will cause an increase of the waiting times. So, accuracy and effect depend on the traffic characteristics.

The approach considered in this paper is to decompose the tandem queueing network into pairs of queues, see Fig. 1. Thus, the correlation between the waiting times of a cell in two consecutive queues - which may be expected to be the most important - is taken into account. Cell routing is modeled by an independent Markov chain. We choose this chain such that the true routing process is approximated: the number of randomly routed cells between two deterministically routed cells is statistically (approximately) equal to the number of interfering cells arriving to the previous queue between two arrivals on the VC under study. So, we have assumed that Bernoulli routing is a goud model for the interfering cells. When studying the probability of loss or the delay distribution of

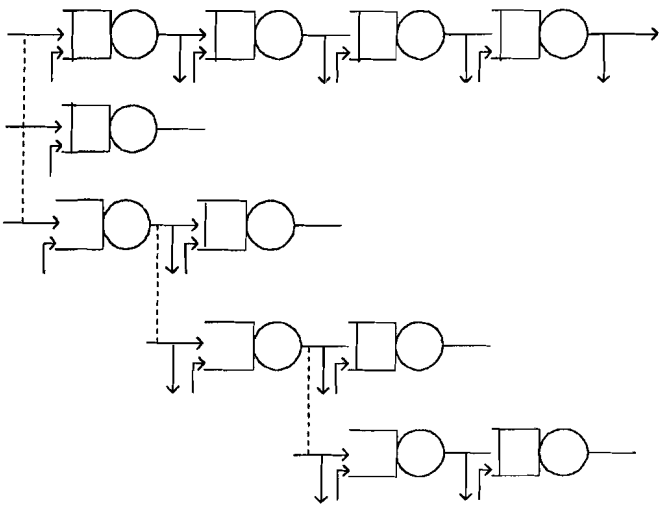


Figure 1: Decomposition of the VC Model

a cell of the VC under study, we require the routing chain to make a transition that corresponds to deterministic routing during the slot this cell is served. The departure process from a queue is modeled by an on-off process.

In the remainder, we will focus on the basic building block of this method: the two tandem queues model. Section 2 shows how it can be solved by the algorithmic procedure for $M/G/1$-type queues presented in [4]. In section 3, we present numerical results on the correlation between the queues and on the routing model. Finally, conclusions are drawn.

2 The two tandem queues model

In this section, we consider a tandem queues model comprising two single server queues. We show how to calculate the steady-state probability distribution and the moments of the busy period of the first queue. The waiting time distribution is obtained by observing a cell that arrives to the system in equilibrium. The moments may be used to model the departure process from the first queue. The servers have equal, deterministic service times and are synchronized. The service time is chosen as unit of time. The buffer spaces of the first and the second queue are infinite and finite, respectively. The service discipline is FCFS. Departures are assumed to occur before arrivals. In a given time slot, the transition in a discrete time Markov chain S^i, $i \in \{1,2\}$, with finite state space determines the probability distribution of the number of external arrivals to the corresponding queue. S^1 describes the arrivals due to both the VC under study and the traffic interfering with it. The routing probability of a cell leaving the first queue depends on the transition in the Markov chain R with finite state space during the same slot.

We describe the state of the system at slot boundaries by the Markov chain (X^1, X^2, S^2, R, S^1) $= \{X_n^1, X_n^2, S_n^2, R_n, S_n^1; n \in N\}$. X_n^1, $X_n^1 \in N$, denotes the number of cells waiting in the first queue. $X_n^2 = 0$ denotes that no or a single cell is waiting in the second queue and $X_n^2 = i, i \in \{1, \ldots, (N_2 - 1)\}$ denotes that $i+1$ cells are waiting. If the states are ordered first according to the state space of X^1 and, given X^1, according to the state space of (X^2, S^2, R, S^1), it follows readily that the transition probabilities matrix is of the $M/G/1$-type, see [4], where X^1 is the level and (X^2, S^2, R, S^1) is the phase process: left transitions of X^1 are by at most one and, except when

$X_n^1 = 0$, $X_{n+1}^1 - X_n^1$ is independent of X_n^1. The transition probabilities matrix looks like this:

$$\begin{pmatrix} B_0 & B_1 & \cdots & \cdots & B_{I_1} & 0 & \cdots & \cdots & \cdots \\ A_0 & A_1 & \cdots & \cdots & A_{I_1} & 0 & \cdots & \cdots & \cdots \\ 0 & A_0 & A_1 & \cdots & \cdots & A_{I_1} & 0 & \cdots & \cdots \\ 0 & 0 & A_0 & A_1 & & & A_{I_1} & 0 & \\ \vdots & & \ddots & \ddots & \ddots & & & \ddots & \ddots \end{pmatrix} \tag{1}$$

I_1 denotes the maximum number of arrivals in a slot to the first queue. B_i and A_i, $0 \leq i \leq I_1$, are substochastic matrices. $B = \sum_{i=0}^{I_1} B_i$ is the transition probabilities matrix of the phase process, given that $X_n^1 = 0$. $A = \sum_{i=0}^{I_1} A_i$ is the transition probabilities matrix of the phase process, given that $X_n^1 \geq 0$. B_i and A_i are matrices of the joint probabilities of phase transitions and i arrivals to the first queue. A_i is given by:

$$A_i = \left[\begin{pmatrix} S_0^2 + S_1^2 & S_2^2 & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & \cdots & 0 \\ S_0^2 & S_1^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & 0 \\ 0 & S_0^2 & S_1^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & & \ddots & 0 \\ \vdots & & & \ddots & S_0^2 & S_1^2 & \cdots & S_{I_2-1}^2 & \tilde{S}_{I_2}^2 \\ \vdots & & & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & S_0^2 & \tilde{S}_1^2 \end{pmatrix} \otimes R_0 + \right.$$

$$\left. \begin{pmatrix} S_0^2 & S_1^2 & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & S_0^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & S_0^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & & & \ddots & 0 \\ \vdots & & & & \ddots & S_0^2 & \cdots & S_{I_2-1}^2 & \tilde{S}_{I_2}^2 \\ \vdots & & & & & \ddots & \ddots & & \vdots \\ \vdots & & & & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & \tilde{S}_0^2 \end{pmatrix} \otimes R_1 \right] \otimes S_i^1$$

B_i is given by:

$$B_i = \begin{pmatrix} S_0^2 + S_1^2 & S_2^2 & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & \cdots & 0 \\ S_0^2 & S_1^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & \cdots & 0 \\ 0 & S_0^2 & S_1^2 & \cdots & \cdots & S_{I_2}^2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & & \ddots & 0 \\ \vdots & & & \ddots & S_0^2 & S_1^2 & \cdots & S_{I_2-1}^2 & \tilde{S}_{I_2}^2 \\ \vdots & & & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & S_0^2 & \tilde{S}_1^2 \end{pmatrix} \otimes I \otimes S_i^1$$

I_2 is the maximum number of external arrivals in a slot to the second queue. S_j^i, $i \in \{1, 2\}$, $j \in \{0, \ldots, I_i\}$, are matrices of the joint probabilities of transitions in S^i and j arrivals to queue i.

$\tilde{S}_i^2 = \sum_{j=i}^{I_2} S_i^2$. R_j, $j \in \{0,1\}$, are matrices of the joint probabilities of transitions in R and the routing decision j. By aggregating the states in wich the second queue is empty and the state in which there is one cell in the second queue, it is avoided that the matrices A_i and B_i are singular.

The steady-state probability distribution for the above system can be obtained by the algorithm for $M/G/1$-type stochastic matrices described in [4]. The structure of the matrices A_i and B_i can be exploited to some advantage in the algorithm. The basic steps of the algorithm are repeated here for continuity. The system state is denoted by (i,j), $i \geq 0$, $0 \leq j \leq M$, where i denotes the level and j the phase. Let x_i denote the steady-state probability vector corresponding to level i, i.e., $x_i = (p(i,0), p(i,1), \ldots, p(i,M))$. Let $G_{jj'}(k)$ denote the probability that starting in state $(i+1,j)$ state (i,j') is the first state reached in level i and that this takes k steps, $k \geq 1$. Defining the matrix $G(k) = [G_{jj'}(k)]$ and the transform matrix $G(z) = \sum_{k=1}^{\infty} z^k G(k)$, it follows readily that G(z) satisfies:

$$G(z) = z \sum_{i=0}^{I_1} A_i G^i(z) \tag{2}$$

We denote $G(1)$ by G and the corresponding steady-state probability vector by g. G can be obtained by an iterative algorithm, see [4, 3.6.4]. g_1 is the vector of the mean times to reach level i from level $i+1$, given the phase of the initial state, i.e., $g_1 = \frac{d}{dz} G(z)\big|_{z=1} e$. Next, let $K_{jj'}(k)$ denote the probability that starting in state $(0,j)$ state $(0,j')$ is the first state reached in level 0 and that this takes k steps, $k \geq 1$. Defining the matrix $K(k) = [K_{jj'}(k)]$ and the transform matrix $K(z) = \sum_{k=1}^{\infty} z^k K(k)$, it follows readily that K(z) satisfies:

$$K(z) = z \sum_{i=0}^{I_1} B_i G^i(z) \tag{3}$$

We denote $K(1)$ by K and the corresponding steady-state probability vector by k. K follows by insertion of G into 3. It is shown in [4, th.3.2.1] that when G is irreducible and the load of the first queue is smaller than one, x_0 is given by:

$$x_0 = (k k_1)^{-1} k,$$

where k_1 is the vector of the mean recurrence times to level 0 given the phase of the initial state, i.e., $k_1 = \frac{d}{dz} K(z)\big|_{z=1} e$. Differentiation of 2 and 3 leads to expressions for g_1 and k_1. Expressions for higher moments can be obtained in the same way, although they get very complex even for low moments. x_i, $i > 0$, is obtained by a recursive algorithm, see [5].

The moments of the busy period can obtained by considering the first queue at slot boundaries during busy periods (including the slot boundaries starting and ending a busy period). By defining the Markov chain in this way, the return time to level 0 equals the length of the busy period plus one slot. The transition probabilities matrix is again of the $M/G/1$-type, see 1. The constituent matrices are given by:

$$\begin{aligned} A_i &= S_i^1 \\ B_0 &= 0 \\ B_i &= (I - S_0^1)^{-1} S_i^1 \end{aligned}$$

Insertion of the expressions for B_i into 3 gives:

$$K(z) = (I - S_0^1)^{-1} (G(z) - zD_0), \tag{4}$$

so the moments of the return time to level 0 can easiliy be obtained from the moments of the time to make a single left transition.

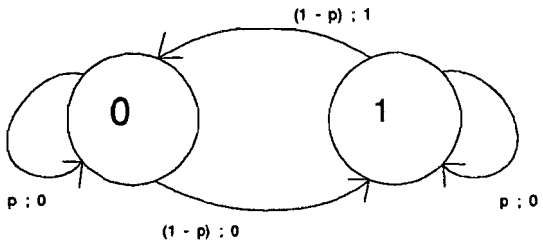


Figure 2: Model for the traffic on the VC under study

3 An Erlang-2 process interfering with batch Bernoulli processes

As a numerical example of the above model, we assume that the traffic on the VC under study is a discrete time Erlang-2 process, see Fig. 2. A cell arrival occurs at transitions from state 1 to state 0. The Erlang-2 process may be considered as a model for a deterministic process of which the intervals between cells have been disturbed randomly. The interfering traffic streams at both queues are modeled by batch Bernoulli processes with binomially distributed batch sizes. The maximum batch size is 7, so that at most 8 cells may simultaneously arrive to any of the queues. The routing Markov chain is also two-state. Transitions from state 1 to state 0 induce the deterministic routing of a cell to queue 2; all other transitions correspond to random routing, with fixed probability. The transition probabilities are chosen such that the first two moments of the number of randomly routed cells between two deterministically routed cells equal the corresponding moments of the number of interfering cells arriving to the previous queue between two arrivals on the VC under study. (It should be noted that this is not possible for all values of the moments). The buffer size of the second queue is 40. In the remainder of this section, we compare routing according to a Markov chain with Bernoulli routing and we assess the correlation of the waiting times of a cell in the consecutive queues. Table 1 shows the set of configurations that has been considered.

Table 2 shows the probability that a cell of the VC under study finds more than a given number of cells waiting upon arrival in queue 2. We compare routing according to a Markov chain with Bernoulli routing. As the Erlang-2 process is less bursty than a Bernoulli process, assuming Bernoulli routing is expected to increase the probability of a given delay. This is confirmed by the figures in the table. The increase is shown to become greater, if probabilities further into the tail of the waiting time distribution are considered. For cell loss probabilities it ranges between a factor 1.1 and 4.2 and is a factor 1.6 for the basic configuration. Further, the increase is greater (1) if the fraction of the load that is due to the VC is greater, (2) if the fraction of the load that is due to crossing interference is greater, and (3) if the load due to interference is smaller.

Next, we assess the correlation of the waiting times of a cell in queue 1 and queue 2, see Table 3. (The routing process has been modeled by a Markov chain.) The table shows the probability that a cell of the VC under study finds more than a given number of cells waiting upon arrival in queue 2, conditioned on its waiting time in queue 1. The waiting times in queue 1 and queue 2 are expected to be possitively correlated. This is confirmed by the figures in the table. The increase of the cell loss probability, given the waiting time in queue 1 is 0 and 40, respectively, ranges between a factor 8.6 and $4.5 \cdot 10^6$ and is a factor $6.7 \cdot 10^2$ for the basic configuration. The correlation is greater (1) if the fraction of the load that is due to crossing interference is smaller and (2) if the load due to interference is smaller.

Table 1: Configurations that have been considered

#	Configuration	Load on VC	Load joining interference	Load crossing interference
0	Basic	0.100	0.300	0.400
1	Increase of fraction of load due to VC	0.200	0.200	0.400
2	Decrease of fraction of load due to VC	0.033	0.367	0.400
3	Increase of fraction of load due to crossing interference	0.100	0.000	0.700
4	Decrease of fraction of load due to crossing interference	0.100	0.600	0.100
5	Decrease of load due to interference	0.100	0.171	0.229
6	Increase of load due to interference	0.100	0.364	0.486

Table 2: Comparison of waiting time and loss in queue 2 for Markovian and Bernoulli routing

#	Markovian			Bernoulli		
	P(wait. $\geq$ 20)	P(wait. $\geq$ 30)	P(loss)	P(wait. $\geq$ 20)	P(wait. $\geq$ 30)	P(loss)
0	3.89e-5	2.90e-7	2.68e-10	4.92e-5	4.00e-7	4.21e-10
1	2.60e-5	1.57e-7	1.17e-10	4.36e-5	3.29e-7	3.24e-10
2				5.16e-5	4.34e-7	4.70e-10
3	3.45e-5	2.00e-7	2.18e-10	7.18e-5	5.79e-7	9.16e-10
4	2.83e-7	3.06e-10	2.07e-14	2.96e-7	3.24e-10	2.24e-14
5	1.45e-13	9.30e-20	1.04e-24	2.55e-13	1.99e-19	1.80e-24
6	7.50e-2	1.77e-2	2.24e-4	7.97e-2	1.93e-2	2.69e-4

Table 3: Waiting time and loss in queue 2 conditioned on the waiting time in queue 1

#	Waiting time in queue 1 = 0		
	P(wait. 2 $\geq$ 20)	P(wait. 2 $\geq$ 30)	P(loss 2)
0	2.34e-5	1.69e-7	1.15e-10
3	3.07e-5	1.78e-7	1.78e-10
4	8.95e-8	9.24e-11	3.89e-15
5	8.22e-14	5.55e-20	9.20e-25
6	4.92e-2	1.02e-2	7.84e-5

#	Waiting time in queue 1 = 20		
	P(wait. 2 $\geq$ 20)	P(wait. 2 $\geq$ 30)	P(loss 2)
0	6.25e-4	8.43e-6	1.18e-8
3	8.62e-5	6.32e-7	8.22e-10
4	3.24e-5	5.51e-8	5.06e-12
5	2.78e-9	2.05e-14	1.76e-20
6	1.18e-1	3.09e-2	4.32e-4

#	Waiting time in queue 1 = 40		
	P(wait. 2 $\geq$ 20)	P(wait. 2 $\geq$ 30)	P(loss 2)
0	1.40e-3	3.65e-5	7.66e-8
3	1.04e-4	9.87e-7	1.54e-9
4	3.50e-4	1.56e-6	2.40e-10
5	1.31e-8	7.53e-13	4.14e-18
6	1.56e-1	4.62e-2	6.77e-4

4 Conclusions

We have presented a procedure to approximately determine the end-to-end performance on a VC in an ATM network. It explicitly takes into account the correlation between the waiting times of a cell in two consecutive queues and models the routing of cells according to their VC. In a numerical example, both aspects have been shown to have a considerable effect on the network performance and thus deserve to be incorporated into the model. Modeling the routing process accurately is important if the fraction of the load that is due to the VC is *high*, if the fraction of the load that is due to crossing interference is high, and if the load due to interference is small. Incorporating the correlation is important if the fraction of the load that is due to crossing interference or the load due to interference itself is small.

Next to obtaining more experience with the two tandem queues model, topics for further study include modeling the departure process from the queues and the actual analysis of the end-to-end performance.

References

[1] N. Shroff et al., Perfomance Analysis of a Virtual Circuit Connection in a high speed ATM WAN using the Best Effort delivery strategy, Infocom '91, paper 12A.1.

[2] I. Stavrakakis, Efficient Modeling of Merging and Splitting Processes in Large Networking Structures, IEEE JSAC, 9, 8, 1336-1347, 1991.

[3] W. Fisher, A Tandem Queueing System with Deterministic Service, ITC-13, 563-569.

[4] M.F. Neuts, Structured Stochastic Matrices of $M/G/1$ type and Their Applications, Marcel Dekker, New York, 1989.

[5] V. Ramaswami, A Stable Recursion for the Steady State Vector in Markov Chains of $M/G/1$ Type, Stoch. Models, 4, 1, 183-188, 1988.

Extended Abstract
Optimization of ATM LDOLL Queueing in Case of Multiple Outlets

Borut Stavrov Geert Awater Frits Schoute

The primary idea behind the ATM (asynchronous transfer mode) is its ability to deal with services with different bandwidth requirements. But the services also differ in performance requirements. Real time services for instance, require small cell delay and delay jitter whereas data transmission requires low cell loss probability.

The ATM cells can be divided in two classes: one which demands low cell loss probability (LL cells) and another with a small cell delay variation requirement (LD cells). In [4] it is shown how this partitioning of the traffic stream in two classes with complementary performance requirements can be exploited for getting lower cell loss probability for LL cells and smaller cell delay variations for LD cells, for an ATM buffer with a single outlet. This is done by giving storage priority to LL cells and retrieval priority to LD cells. For the case of a switching element with only one outlet (in fact a multiplexer) it was shown that the set of optimal policies consists of Q LDOLL (low delay or low loss) threshold policies, which are robust in performance and simple to implement.

This paper explores the possibilities of expanding the ideas of a single server LDOLL queue to the multiple outlet case where the outlets are sharing buffer space. The multiple outlet LDOLL queue can be modeled as a multidimensional discrete time Markov chain. The state space is growing very fast with the increase of Q (buffer size) and particulary with the increase of M (number of outlets), namely

$$\text{number of states} = \begin{pmatrix} Q+M \\ Q \end{pmatrix}.$$

Therefore we are going to look at the rather simple case ($Q = 4, M = 2$) which is still complex enough to have all interesting properties of a multidimensional LDOLL queue. This queue has four classes of customers because there are two types of cells (LL & LD) and two outlets (1 & 2).

In order to define a Markov model of the multidimensional LDOLL queue we have to describe three things :

- Arrival process
- Storage policy
- Retrieval policy

The LDOLL queue operates at CELL level. Assuming that the traffic streams at BURST level are mixed so well that the cell arrival at CELL level doesn't depend on the previous arrivals, the arrival process can be modeled as a Bernoulli process.

With a single outlet case the storage policy is obvious, LL cells should always have storage priority over LD cells. Newly arrived LL cells are also allowed to replace the LD cells in the buffer. In the multiple outlet case it is not so obvious what should be done. For instance can a LL1 cell replace a LD2 cell or may a LL1 cell be replaced with a LL2 cell etc. To answer all this dilemmas we are looking for a storage policy which should be:

- optimal
- general
- order independent

Optimal means that the storage policy should result in lowest possible loss probability for LL cells and smallest average cell delay for LD cells. General means that the policy should be a generalization of the single outlet case and at the same time valid for more complex cases ($M > 2$). The last requirement insures that the state of the queue is independent of the order in which the various types of cells, arriving in a single time slot, are stored. The storage policy which satisfies all this requirements can be

described as follows: It gives storage priority to LL cells over LD cells and it tries to keep the balance between the total number of cells for outlet 1 and for outlet 2. Keeping the balance means that, say, a LL1 cell will overwrite a LL2 cell only if the total number of cells for outlet 1 is less than the total number of cells for outlet 2. The same principle can favor LL1 cells over LL2 cells, LD1 over LD2 or LD2 over LD1. A direct consequence of this storage policy is that Q (the size of the buffer) has to be a multiple of M (number of outlets).

In order to find an optimal retrieval policy we want to optimize over a set of retrieval policies which is as large as possible. Therefore we allow the general retrieval policy to be time, history and state dependent. To find the optimal one from this large set we used Markov decision theory. It is a versatile and powerful tool to analyze sequential decision processes in a Markov chain. It offers procedures that minimize different object functions. The object function that we have chosen is the Expected average cost per unit of time criterion. In Markov decision theory it is proved that the policy which minimizes the expected average cost has to be a deterministic policy. Under any deterministic policy the multidimensional LDOLL queue is irreducible and finite and the object function reduces to a simple linear combination of the state probabilities. We want to find a set of policies which minimize the expected LD cell delay and probability of LL cell loss. To do that we expressed the object function as a linear combination of those two objectives. That can be accomplished with a properly chosen cost function. The set of deterministic policies is still very large (of order 2^{30}). To further reduce this astronomically large set we used the value iteration algorithm which is very suitable for problems with large state spaces.

As a result of the optimization we ended up with a set of 6 optimal policies. We compared the performance of these policies with three different types of threshold policies. The results are presented in the delay-loss plane, which represents the multi-objective space (see figure 1). A policy has an optimal position in this space if no other policy can have a lower mean delay for LD cells (futher west) *and* a lower loss probability for LL cells (further south). A few important conclusions can be drawn from it. First it can be seen from the objective space that the threshold policies are very close in performance to the optimal ones. Second, there is one type of threshold policies which yields an optimal policy for all possible values of the threshold. We call this type 'myopic threshold', because of the property of making decisions based upon the number of only one type of cells. That means the decision of the server 1 depends only on the number of LL1 cells in the buffer. The myopic threshold policies are in fact a straight-forward generalization of the optimal threshold policies for the single server case.

The myopic threshold policies are very attractive from an implementation and a generalization point of view. Because of their simplicity they can operate at very high rates.

It is left for further research to explore the behavior of the threshold policies in larger queues ($Q > 4$). It will be particulary interesting to see the behavior of the myopic threshold policies in queues with more servers ($M > 2$).

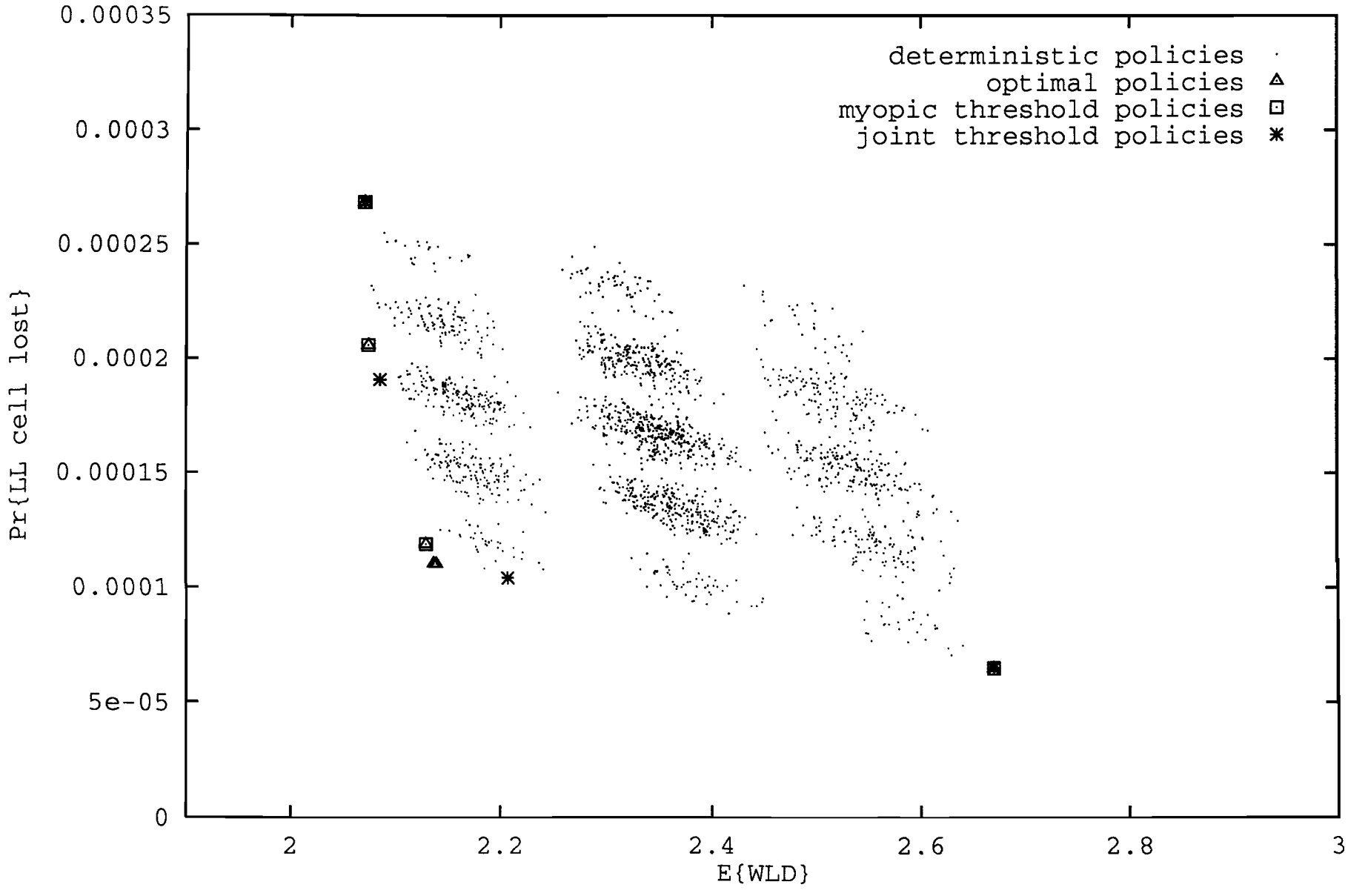
References

[1] C. Derman, "Finite State Markovian Decision Processes," *Mathematics in Science and Engineering*, vol. 67, New York: Academic Press 1970.

[2] H.C. Tijms, "Stochastic Modeling and Analysis," *Wiley Series in Probability and Mathematical Statistics*, John Wiley & Sons 1986.

[3] M. de Prycker, "Asynchronous Transfer Mode, solution for Broadband ISDN," *Ellis Horwood Series in Computer Communications and Networking*, Ellis Horwood Ltd., 1991.

[4] G.A. Awater, F.C. Schoute, "Optimal Queueing Policies for Fast Packet Switching of Mixed Traffic," *IEEE J. Select. Areas Commun.*, vol. 9, no. 3, pp. 458–466, April 1991.


deterministic policies
optimal policies
myopic threshold policies
joint threshold policies
Pr{LL cell lost}
E{WLD}
0.00035
0.0003
0.00025
0.0002
0.00015
0.0001
5e-05
0
2
2.2
2.4
2.6
2.8
3

Queueing analysis of an ATM switch with correlated routing.

by Herwig BRUNEEL and Sabine WITTEVRONGEL
Laboratory for Communications Engineering
University of Ghent
Sint–Pietersnieuwstraat 41
B–9000 Gent, Belgium.

Abstract.

In this paper we consider an ATM switching element with output queueing. One separate infinite–capacity output buffer is used for each possible destination (output). The cell arrival processes on the inlets of the switching element are of a bursty nature. Specifically, geometrically distributed on/off–periods are assumed to describe this burstiness. Cells are routed from the inputs to the outputs of the switching element in a uniform but correlated manner, i.e., all cells belonging to the same on–period are routed to the same destination, but all destinations are equiprobable.

The queueing performance of the switch is analyzed here by a combination of analytic techniques and approximations. Close analytic upper bounds are obtained for such measures as the means and the tail distributions of the buffer contents and the cell delay in a tagged output buffer. The formulas are easy to evaluate and the results are useful in practice, for instance, to calculate the cell loss ratio or the delay jitter in an output queue. The influence of the degree of burstiness in the cell arrival processes is investigated. Also, the results are compared with the case of uncorrelated routing.

The first author wishes to thank the Belgian National Fund for Scientific Research (N.F.W.O) for support of this research.

1. Introduction.

Although discrete–time queueing analysis has been applied (occasionally) since over twenty years in the performance evaluation of various types of slotted communication systems (see e.g. [1–7]), the conception of ATM–based multiservice networks seems to have caused an increased interest in discrete–time models since the end of the 1980's. In particular, discrete–time approaches have been adopted on various occasions for the queueing analysis of (ATM) statistical multiplexers and (ATM) switching elements. In both these applications, a finite number of random traffic sources deliver data units ("cells") to a common buffer, from which cells are transmitted at the rate of one per slot as long as the buffer is nonempty. In case of a multiplexer, all cells have the same (common) destination and the cell sources are directly connected to the common buffer via the inlets of the multiplexer. In case of switching elements, however, a routing mechanism (between the sources and various destinations) determines the actual arrival process of cells in the buffer (corresponding to one of many destinations). A queueing model for a multiplexer thus essentially implies a statistical description of the traffic sources that generate the cells to be transmitted. Studying the queueing performance of a switching element, however, also requires an (additional) statistical description of the applied routing mechanism, and is therefore, in general, more complicated.

Performance studies of multiplexers and switches can be categorized according to the nature of the sources, i.e., the cell arrival processes on the inlets, and, in the case of switching elements, according to the nature of the routing mechanism. The simplest models assume uncorrelated arrival streams on the inlets; examples include [1, 3, 4, 8] for multiplexers and [9–15] for switching elements (with independent routing mechanism). However, in view of the rather complicated traffic patterns which may occur in multiservice networks, several researchers have also concerned themselves with more general, i.e., "nonindependent" or "bursty" arrival models, during the last several years. Although other types of characterizations have been used as well, source models of the on/off–type have been particularly popular, both in the continuous–time (see e.g. [16]) and the discrete–time domain [6, 17–24]. Concentrating on the discrete–time studies, we note that [6, 17–21] deal with multiplexers, whereas [22–24] have to do with switching elements. In [6, 17–20] both the on–periods and the off–periods of the traffic sources are modeled as geometrically distributed random variables. A somewhat deviating source model is considered in [21], where the on–periods are modeled as geometrically distributed

multiples of a given fixed number of slots. The same source models are also found in switching element analyses : geometric on/off–periods in [22–23] and geometric multiples of a fixed interval for the on–periods in [24].

As to the statistical description of the routing process, most studies where the arrival processes on the inlets of the switch are modeled as uncorrelated, such as [9–15], have also assumed uncorrelated routing mechanisms (no correlation between the destinations of consecutive cell arrivals on any given inlet). Among the studies with on/off–type input processes, independent uniform routing of cells was assumed in [22], whereas in [23–24] various types of correlation in the routing process were also considered. However, to the best of the authors' knowledge, a full queueing analysis has never been reported for sources with geometric on/off–periods in case of "fully correlated routing", i.e., in case that all the cells arriving during the same on–period are destined for the same output, a situation which is rather likely to occur in practice. The purpose of the present paper is exactly to provide this kind of analysis. In order to do so, we have adapted an analytic method, developed in [24], to tackle the problem, which is basically a combination of a generating–functions approach with bounding (approximation) techniques. As a result, approximate analytic formulas are obtained for various quantities of interest. Specifically, the analysis yields close upper bounds for such measures as the means and the tail probabilities of the buffer contents and the cell delay in a tagged output queue of the switching element.

The outline of the paper is as follows. In section 2, we describe the switching element under study and state the main elements of the mathematical model. In section 3, we establish a fundamental functional equation in terms of a trivariate generating function which characterizes the queueing behavior of the tagged output buffer. Section 4 concentrates on the steady–state cell arrival process in the tagged output buffer and introduces bounds for some remaining unknowns of the analysis. These approximations are then used in sections 5 and 6 to derive close upper bounds for the mean and the tail probabilities of the buffer occupancy. In section 7, corresponding bounds are derived for the cell delay. Section 8 is concerned with verifying the accuracy of the analytic approximations. Finally, in section 9, we present a number of numerical examples and investigate the influence of the burstiness of the sources and the correlation in the routing process on the performance of the switch.

2. Mathematical model.

In this paper we consider a symmetric switching element for ATM cells, with N inlets and N outlets. Cells enter the switch via one of the inlets and are then routed to one of the outlets (according to their destination address) where they are temporarily buffered in a designated output queue (output buffer) to await the transmission of earlier cell arrivals with the same destination. Synchronous transmission is used on both the input links and the output links of the switch, i.e., time is divided into fixed–length slots and at most one ATM cell is transmitted during each slot on any of these links. Note that due to the slotted transmission mode on the outlets of the switch, a cell can never leave the output buffer before the end of the slot right after its arrival slot.

We assume that cell arrivals on the inlets of the switching element are generated by N independent sources with identical statistical characteristics. The sources are of a bursty nature, which will be described by means of an on/off–type of model. Specifically, we assume that each source stochastically alternates between an active state ("on") and a passive state ("off"). When active, a source generates exactly one ATM cell per slot; when passive, a source does not generate any ATM cells at all. The lengths of the active and passive periods of a source are modeled here as independent, geometrically distributed random variables, with parameters α and β respectively, i.e.,

$$\text{Prob[active period = n slots]} = (1-\alpha)\alpha^{n-1} \ , \ n \geq 1 \ ;$$
$$\text{Prob[passive period = n slots]} = (1-\beta)\beta^{n-1} \ , \ n \geq 1 \ .$$

This implies that the sources are correlated in a first–order Markovian way : the probability that any given source is active or passive in any given slot is fully determined by the state of this source in the previous slot. Specifically, if p(Y|X) denotes the probability that a source is currently in state Y, given that it was in state X in the preceding slot, we have

$$p(\text{active}|\text{active}) = \alpha \ ; \qquad p(\text{passive}|\text{active}) = 1-\alpha \ ;$$
$$p(\text{passive}|\text{active}) = 1-\beta \ ; \qquad p(\text{passive}|\text{passive}) = \beta \ .$$

Note that the classical Bernoulli arrival model is obtained here as a special case, i.e., for $\alpha + \beta = 1$.

The average load of one source is defined as the fraction of time (slots) this source spends in the active state, and is thus given by

$$p = \frac{E[\text{active period}]}{E[\text{active period}] + E[\text{passive period}]} = \frac{\frac{1}{1-\alpha}}{\frac{1}{1-\alpha} + \frac{1}{1-\beta}} ,$$

where E[...] denotes the expected value of the expression between square brackets. This implies that, in general, the mean lengths of active and passive periods can be expressed as

$$E[\text{active period}] = \frac{1}{1-\alpha} = \frac{K}{1-p}$$

and

$$E[\text{passive period}] = \frac{1}{1-\beta} = \frac{K}{p} ,$$

for some value of the real quantity K. It is clear that the statistical properties of a source can be fully characterized by the parameters p and K (instead of α and β) : the load p is a measure for the ratio of the active and passive periods, whereas the constant K (in the sequel referred to as the "burstiness factor") is representative for the absolute lengths of these periods. High values of K are indicative of a high degree of correlation in the cell arrival process. A classical (uncorrelated) Bernoulli arrival process (with load p) corresponds to K=1.

As mentioned before, we assume in this paper that the destination addresses of consecutive cell arrivals on a given inlet are not independent. Specifically, we assume that all the ATM cells generated by a given source during the same active period belong together and have exactly the same destination. Cells generated during different active periods (of a given source), however, are routed entirely independently. In addition, all the destination addresses are equiprobable (in the long run). Stated otherwise, we are considering in this paper uniform and independent routing of bursts (i.e., active periods) rather than (individual) cells.

We observe that the (eventual) cell arrival process in a selected ("tagged") output queue is determined by the interaction between source characteristics and routing mechanism. Specifically, in each slot, each of the inlets of the switching

element, from the point of view of the tagged output queue, may be either "active" (if it delivers a cell to the tagged output queue) or "blocked" (if it does not). An inlet is active (state A) if it receives a cell from the corresponding source and this cell is routed to the tagged output. An inlet may be blocked (B) for several reasons : either it does not receive any cells from the associated source (state B1), or it does receive a cell from the source, but this cell is not routed to the tagged output (state B2). We therefore conclude that, from the point of view of the tagged output queue, each inlet can be characterized by a three–state Markov chain with states A, B1 and B2, and transition probabilities as indicated in Fig. 1.

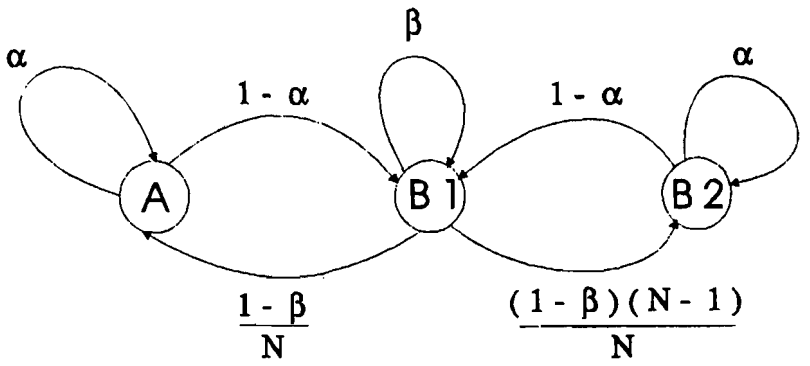


Fig. 1 Markov chain model of an inlet.

3. Fundamental relationships and functional equation.

Let us define the random variables e_k and v_k as the number of cell arrivals in the tagged output queue and the non–tagged output queues respectively, i.e., the number of inlets in state A and B2 respectively, during slot k. Then, in view of Fig. 1, e_k and v_k can be derived from e_{k-1} and v_{k-1} as follows :

$$e_k = \sum_{i=1}^{e_{k-1}} c_i + \sum_{i=1}^{N-e_{k-1}-v_{k-1}} d_i \ , \qquad (1)$$

and

$$v_k = \sum_{i=1}^{v_{k-1}} c'_i + \sum_{i=1}^{N-e_{k-1}-v_{k-1}} f_i \ . \qquad (2)$$

Here the c_i's and the c'_i's are i.i.d. random variables with probability generating function (pgf)

$$C(z) \triangleq E[z^{c_i}] = E[z^{c'_i}] = 1 - \alpha + \alpha z \ , \qquad (3)$$

whereas the pairs (d_i,f_i) are i.i.d. with joint pgf

$$Q(x,y) \triangleq E[x^{d_i} y^{f_i}] = \beta + \frac{1-\beta}{N} x + \frac{(1-\beta)(N-1)}{N} y \ . \qquad (4)$$

Moreover, the c_i's and the c'_i's and the pairs (d_i,f_i) are mutually independent (because they refer to different inlets of the switching element).

Let s_k be the random variable representing the number of cells stored in the tagged output queue at the beginning of slot k, i.e., just after slot (k–1). Then the evolution of the buffer occupancy is described by the following equation,

$$s_{k+1} = (s_k - 1)^+ + e_k \ , \qquad (5)$$

where $(.)^+$ denotes max(0,.). Equations (1), (2) and (5) imply that the set $\{(e_{k-1}, v_{k-1}, s_k)\}$ is a Markov chain. Let us define the three–dimensional joint pgf of (e_{k-1}, v_{k-1}, s_k) as

$$P_k(x,y,z) \triangleq E[x^{e_{k-1}} y^{v_{k-1}} z^{s_k}] \ .$$

From (5) it then follows that

$$P_{k+1}(x,y,z) = E[x^{e_k} y^{v_k} z^{s_{k+1}}] = E\left[(xz)^{e_k} y^{v_k} z^{(s_k-1)^+}\right] \ .$$

Next, using (1) and (2) and averaging over the distributions of the c_i's, the c'_i's and the (d_i,f_i)'s yields

$$P_{k+1}(x,y,z) = [Q(xz,y)]^N \, E\left[\left[\frac{C(xz)}{Q(xz,y)}\right]^{e_{k-1}} \left[\frac{C(y)}{Q(xz,y)}\right]^{v_{k-1}} z^{(s_k-1)^+}\right] ,$$

where the expectation is over the joint distribution of (e_{k-1}, v_{k-1}, s_k). As $s_k=0$ implies that $e_{k-1}=0$, it follows that

$$P_{k+1}(x,y,z) = [Q(xz,y)]^N \left\{\frac{1}{z} P_k\left[\frac{C(xz)}{Q(xz,y)}, \frac{C(y)}{Q(xz,y)}, z\right] + \left[1-\frac{1}{z}\right] \text{Prob}[s_k=0] \sum_{j=0}^{N} \text{Prob}[v_{k-1}=j \mid s_k=0] \left[\frac{C(y)}{Q(xz,y)}\right]^j\right\} .$$

We now assume that the average number of cell arrivals in the tagged output queue is strictly less than 1, so that the queueing system can reach a steady state. In this case

$$\lim_{k\to\infty} P_{k+1}(x,y,z) = \lim_{k\to\infty} P_k(x,y,z) = P(x,y,z) ,$$

where $P(x,y,z)$ is the steady–state joint pgf. Therefore, $P(x,y,z)$ must satisfy the following functional equation :

$$P(x,y,z) = [Q(xz,y)]^N \left\{\frac{1}{z} P\left[\frac{C(xz)}{Q(xz,y)}, \frac{C(y)}{Q(xz,y)}, z\right] + \left[1-\frac{1}{z}\right] p_0 \, U\left[\frac{C(y)}{Q(xz,y)}\right]\right\} , \tag{6}$$

where p_0 is the steady–state probability of an empty buffer. The U–function is defined as

$$U(x) \triangleq \lim_{k\to\infty} \sum_{j=0}^{N} \text{Prob}[v_{k-1}=j \mid s_k=0] \, x^j . \tag{7}$$

Unfortunately, we are not able to derive from (6) an explicit expression for $P(x,y,z)$ or not even for $S(z) = P(1,1,z)$, which is the steady–state pgf of the buffer contents. In order to get more information about the behavior of the tagged output queue, we now consider only those values of x,y and z for which the arguments of the

P–function on both sides of (6) are equal to each other, i.e., $x = \frac{C(xz)}{Q(xz,y)}$ and $y = \frac{C(y)}{Q(xz,y)}$. Using (3) and (4) it is easy to see that this is equivalent to

$$x = \frac{(1-\alpha)y}{1 - \alpha + \alpha y(1-z)} \tag{8}$$

and

$$y = \frac{1 - \alpha + \alpha y}{\beta + \frac{1-\beta}{N}\,[(N-1)y + xz]} \quad . \tag{9}$$

From (8) and (9), x and y can be solved in terms of z. It is easily verified that for a given value of z, y satisfies a third–order equation. Hence, there are three sets of solutions. Here we only select a set of solutions which has the additional property that x=y=1 for z=1, which is denoted by $\chi(z)$ and $\xi(z)$. Choosing $x=\chi(z)$ and $y=\xi(z)$ in (6) yields a linear equation for the function $P(\chi(z),\xi(z),z)$, which has the following solution :

$$P(\chi(z),\xi(z),z) = \frac{(z-1)\ p_0\ \varphi_s(z)\ G(z)}{z - G(z)} \quad , \tag{10}$$

where

$$G(z) \triangleq [Q(\chi(z)z,\xi(z))]^N = \left[\beta + \frac{1-\beta}{N}\,[(N-1)\xi(z) + \chi(z)z]\right]^N \tag{11}$$

and

$$\varphi_s(z) \triangleq U(\xi(z)) \quad . \tag{12}$$

From these definitions, we have G(1)=1 and $\varphi_s(1)=1$. The unknown parameter p_0 can be determined from the normalization equation $P(\chi(z),\xi(z),z)\big|_{z=1} = 1$. By using de l'Hospital's rule, we find $p_0 = 1 - G'(1)$. However, in equation (10), $\varphi_s(z)$ is unknown. Nevertheless, it is possible to derive upper bounds for the mean and the tail distribution of the buffer occupancy, as we will describe in the following.

4. Unconditional and conditional arrival process in the steady state.

Let s denote the number of cells stored in the tagged output queue just after a slot in the steady state, while e and v represent the number of arriving cells in the tagged output queue or non–tagged output queues respectively, during that slot. The two–dimensional joint pgf of the steady–state random variables e and v is defined as $N(x,y) \triangleq E[x^e\, y^v]$. As e and v can be considered as the (steady–state) numbers of inlets (of the switching element) in state A or B2 respectively, N(x,y) can be expressed as

$$N(x,y) = \Big[\, p(B1) + p(A)\, x + p(B2)\, y \,\Big]^N \quad ,$$

where p(A), p(B1) and p(B2) denote the steady–state probabilities of finding an inlet in state A, B1 or B2 respectively. These probabilities can easily be found by solving the balance equations for the Markov chain in Fig. 1. As a result we obtain

$$N(x,y) = \Big[\, 1 - p + \frac{p}{N}\, x + \frac{(N-1)p}{N}\, y \,\Big]^N \quad , \tag{13}$$

which is also intuitively clear. Equation (13) describes the arrival process in the steady state. The marginal pgf E(x) of e can be obtained by setting y=1 in (13). The marginal pgf V(y) of y can be derived by setting x=1 in (13).

Next, we consider the unknown conditional probability mass function Prob[v=j|s=0], appearing in (7). Since s=0 also implies that e=0, it is reasonable to think that the difference between the arrival processes to the non–tagged output queues observed when s=0 or when e=0 respectively, is very small, i.e.,

$$\text{Prob}[v=j\,|\,s=0] \cong \text{Prob}[v=j\,|\,e=0] \quad . \tag{14}$$

Furthermore, as s=0 implies that few cells were sent to the tagged output queue during several previous slots, due to the correlated routing, we have

$$\text{Prob}[v=j\,|\,s=0] > \text{Prob}[v=j\,|\,e=0] \quad , \quad \text{for large } j \, . \tag{15}$$

Using equation (14) in (7), with equations (12) and (13), then yields

$$\varphi_s(z) \cong \varphi_e(z) \triangleq \sum_{j=0}^{N} \text{Prob}[v=j|e=0]\,[\xi(z)]^j = \frac{N(0,\xi(z))}{N(0,1)}$$

$$= \frac{\left[1 - p + \frac{(N-1)p}{N}\,\xi(z)\right]^N}{\left[1 - \frac{p}{N}\right]^N} \quad . \tag{16}$$

On the other hand, it is clear that, under the condition that s=0, for large j, the probability that j cells are sent to the non–tagged output queues is larger than in the unconditional case. That is,

$$\text{Prob}[v=j|s=0] > \text{Prob}[v=j] \quad , \quad \text{for large } j\,. \tag{17}$$

Substituting Prob[v=j|s=0] in (7) by Prob[v=j], we obtain from (12)

$$\varphi_s(z) \cong \varphi(z) \triangleq \sum_{j=0}^{N} \text{Prob}[v=j]\,[\xi(z)]^j = N(1,\xi(z))$$

$$= \left[1 - \frac{(N-1)p}{N} + \frac{(N-1)p}{N}\,\xi(z)\right]^N \quad . \tag{18}$$

5. Upper bounds of the mean buffer occupancy in the steady state.

As mentioned before, the pgf S(z) of s can be expressed as S(z) = P(1,1,z). In order to obtain the mean buffer occupancy $\bar{s}$, we evaluate the first derivative of (10) with respect to z at z=1. This leads to

$$\left.\frac{dP}{dz}\right|_{z=1} = \varphi_s'(1) + G'(1) + \frac{G''(1)}{2\,[1 - G'(1)]} \quad , \tag{19}$$

where

$$\frac{dP}{dz} = \frac{\partial P}{\partial \chi}\frac{d\chi}{dz} + \frac{\partial P}{\partial \xi}\frac{d\xi}{dz} + \frac{\partial P}{\partial z} \quad .$$

Since $\chi(1) = \xi(1) = 1$, we have $\left.\frac{\partial P}{\partial z}\right|_{z=1} = S'(1) = \bar{s}$, $\left.\frac{\partial P}{\partial \chi}\right|_{z=1} = E'(1)$ en $\left.\frac{\partial P}{\partial \xi}\right|_{z=1} = V'(1)$. So equation (19) yields

$$\bar{s} = \varphi_s'(1) + G'(1) + \frac{G''(1)}{2\,[1 - G'(1)]} - E'(1)\,\chi'(1) - V'(1)\,\xi'(1) \quad . \tag{20}$$

Here $G'(1)$ and $G''(1)$ can be expressed in terms of $\chi'(1)$, $\xi'(1)$, $\chi''(1)$ and $\xi''(1)$ by using (11), whereas these derivatives of $\chi(z)$ and $\xi(z)$ can be derived from equations (8) and (9). We note in particular that $G'(1) = p$, so that $p_0 = 1-p$, which is expected. Hence, the only unknown term left in (20) is $\varphi_s'(1)$, where $\varphi_s(z)$ is defined in (12). Although we are not able to obtain $\varphi_s(z)$, two upper bounds of $\bar{s}$ can be derived as follows.

From the definition of the U–function in (7), $\left.\frac{dU}{d\xi}\right|_{z=1} = \sum_{j=0}^{N} j\ \text{Prob}[v=j|s=0]$ is the average number of cell arrivals during a slot to the non–tagged output queues when s=0. The inequality in (15) shows that $\left.\frac{dU}{d\xi}\right|_{z=1}$ is larger than $\sum_{j=0}^{N} j\ \text{Prob}[v=j|e=0]$. Since $\varphi_s(z) = U(\xi(z))$ and $\xi'(1) = -\frac{1}{1-\alpha}\frac{p}{N} < 0$, we have $\varphi_s'(1) < \varphi_e'(1)$. Using this inequality in (20) leads to the following upper bound for the mean buffer occupancy :

$$\bar{s} < \bar{s}_{u(1)} \triangleq p + \frac{(N-1)p^2}{2N(1-p)}\left[1 + 2\,\frac{K+p-1}{1-p} - \frac{2pK}{N-p}\right] \quad . \tag{21}$$

Next, from (17), $\left.\frac{dU}{d\xi}\right|_{z=1}$ is also larger than $V'(1)$, the mean value of v. Since $\xi'(1) < 0$, this implies that $\varphi_s'(1) < \varphi'(1)$. Based on this inequality, a second upper bound of $\bar{s}$ can be obtained, i.e.,

$$\bar{s} < \bar{s}_{u(2)} \triangleq p + \frac{(N-1)p^2}{2N(1-p)}\left[1 + 2\,\frac{K+p-1}{1-p}\right] \quad . \tag{22}$$

Comparing (21) with (22), we see that $\bar{s}_{u(2)}$ is always larger than $\bar{s}_{u(1)}$.

6. Upper–bound tail distribution of the buffer occupancy.

It has been observed that, for a wide range of discrete–time queueing systems, the tail distribution of the buffer occupancy has a geometric form, i.e., for some threshold T,

$$\text{Prob}[s=n] \cong B\,\gamma^n , \qquad n > T . \tag{23}$$

Numerical results have revealed that this is also true for the queueing system under study. In this section we will present an analytical approach to obtain the geometric decay rate γ and upper bounds for the coefficient B. In this way, we can derive an upper–bound tail distribution of the queue length.

The geometric decay rate γ.

From the inversion formula for z–transforms it follows that Prob[s=n] can be expressed as a weighted sum of negative powers of the poles of S(z). Since the modulus of all these poles is larger than one, it is obvious that for large n, Prob[s=n] is dominated by the contribution of the pole with the smallest modulus. Let us denote this dominating pole by z_0. To ensure that the tail distribution is nonnegative anywhere, z_0 must necessarily be real and positive. Furthermore, we assume that z_0 has multiplicity one. The obtained results prove that this assumption is correct. Therefore, with respect to the asymptotic behavior of the buffer occupancy, S(z) can be approximated as

$$S(z) \cong \frac{\theta}{z-z_0} , \tag{24}$$

where θ is the residue of S(z) in the point $z=z_0$. Taking the inverse z–transform of (24) thus gives rise to the following asymptotic result :

$$\text{Prob}[s=n] \cong -\frac{\theta}{z_0}\left[\frac{1}{z_0}\right]^n , \qquad n > T . \tag{25}$$

Comparing (25) to (23), we have

$$\gamma = \frac{1}{z_0} \tag{26}$$

and

$$B = -\frac{\theta}{z_0} \ . \tag{27}$$

As in [21, 22, 24], it can be argued that z_0 is also the pole with the smallest modulus of $P(\chi(z),\xi(z),z)$. Hence, in view of (10) and (11), z_0 is a real root of

$$\left[\beta + \frac{1-\beta}{N}\left[(N-1)\xi(z) + \chi(z)z\right]\right]^N - z = 0 \tag{28}$$

and can be obtained numerically by using, for instance, the Newton–Raphson algorithm, where in each step $\chi(z)$ and $\xi(z)$ are calculated from (8) and (9). Selecting the correct solution of (8) and (9) is one of the problems here. It can be solved on the basis of the observation (not proved here) that $0 < \xi(z) < 1$ for all real $z > 1$. Finally, we note from equations (8), (9) and (28) that the unknown conditional probability mass function Prob[v=j|s=0] has no influence on z_0, which means that γ can be calculated exactly.

Upper bounds for the coefficient B of the geometric form.

When the number of cells stored in the tagged output queue just after a given slot is sufficiently large (>> N), we may think that the number of cell arrivals during this slot (which is not larger than N) has nearly no impact on the total buffer occupancy. That is, if n is sufficiently large (n > T), we may assume that the conditional probabilities Prob[e=i, v=j|s=n] are nearly independent of n, and approach to some limiting values for $n \to \infty$, denoted by $\omega(i,j)$, with corresponding joint pgf $\Omega(x,y)$, i.e.,

$$\text{Prob}[e=i,\ v=j|s=n] \cong \omega(i,j)\ , \qquad n > T\ . \tag{29}$$

Now, let $\pi(i,j|k,\ell)$ denote the one–step transition probability that there are i cell arrivals in the tagged output queue and j cell arrivals in the non–tagged output queues in the current slot, given that there were k and ℓ cell arrivals respectively in

the previous slot. Then, we have (for large n)

$$\text{Prob}[e=i,\ v=j|s=n] = \frac{\text{Prob}[e=i,\ v=j,\ s=n]}{\text{Prob}[s=n]}$$

$$= \frac{\sum_{k=0}^{N}\sum_{\ell=0}^{N-k} \pi(i,j|k,\ell)\ \text{Prob}[e=k,\ v=\ell,\ s=n+1-i]}{\text{Prob}[s=n]} \ .$$

Taking the limit for $n \to \infty$, and using equations (23), (26) and (29), the above equation leads to

$$z_0\,\omega(i,j) = z_0^{\,i} \sum_{k=0}^{N}\sum_{\ell=0}^{N-k} \pi(i,j|k,\ell)\ \omega(k,\ell) \ , \qquad (30)$$

$$0 \le i \le N\ ,\ 0 \le j \le N-i\ .$$

From (30), the following equation for the pgf $\Omega(x,y)$ can then be derived :

$$z_0\,\Omega(x,y) = [Q(xz_0,y)]^N\ \Omega\left[\frac{C(xz_0)}{Q(xz_0,y)}\ ,\ \frac{C(y)}{Q(xz_0,y)}\right] \ . \qquad (31)$$

As can be expected intuitively, it is possible to show that the solution $\Omega(x,y)$ of (31) has the same form of expression as $N(x,y)$. Specifically, $\Omega(x,y)$ can be expressed as

$$\Omega(x,y) = \left[1 - \sigma_1^* - \sigma_2^* + \sigma_1^*\,x + \sigma_2^*\,y\right]^N \ , \qquad (32)$$

where σ_1^* and σ_2^* are the (conditional) probabilities of finding an inlet in state A or B2 respectively, when the number of cells in the tagged output queue is extremely large. Using equations (31) and (32), σ_1^* and σ_2^* can be derived explicitly as

$$\sigma_1^* = \frac{(1-\beta)\ z_0\ (z_0^{\,1/N} - \alpha)}{N(z_0^{\,1/N} + 1 - \alpha - \beta)(z_0^{\,1/N} - \alpha z_0) + (1-\beta)(z_0-1)z_0^{\,1/N}}$$

and

$$\sigma_2^* = \frac{(1-\beta)\ (N-1)\ (z_0^{1/N} - \alpha z_0)}{N(z_0^{1/N} + 1 - \alpha - \beta)(z_0^{1/N} - \alpha z_0) + (1-\beta)(z_0-1)z_0^{1/N}} \ .$$

So the joint pgf $\Omega(x,y)$ can be obtained analytically.

Using equation (29), the joint pgf $P(x,y,z)$ can now be approximately expressed as

$$P(x,y,z) \cong \sum_{i=0}^{N} \sum_{j=0}^{N-i} \sum_{n=i}^{T} \text{Prob}[e=i,\ v=j,\ s=n]\ x^i\ y^j\ z^n$$

$$+\ \Omega(x,y) \left[S(z) - \sum_{n=0}^{T} \text{Prob}[s=n]\ z^n \right] \ . \qquad (33)$$

For $x=\chi(z)$ and $y=\xi(z)$, we know that z_0 is a pole of both the P–function and S(z). Since T is finite, multiplying by $(z-z_0)$ on both sides of equation (33) and taking the $z \to z_0$ limit, yields

$$\theta = \frac{\eta}{\Omega(\chi(z_0),\ \xi(z_0))} \ , \qquad (34)$$

where θ is the residue of S(z) at $z=z_0$, as defined in (24). The quantity η can be obtained from equation (10) with de l'Hospital's rule as

$$\eta = \lim_{z \to z_0} (z-z_0)\ P(\chi(z),\xi(z),z) = \frac{(z_0-1)\ (1-p)\ \varphi_s(z_0)\ G(z_0)}{1-G'(z_0)} \ . \qquad (35)$$

Finally, from (27), (34) and (35), we find that the coefficient B is given by :

$$B = \frac{(z_0-1)\ (1-p)\ \varphi_s(z_0)}{[G'(z_0)-1]\ \Omega(\chi(z_0),\xi(z_0))} \ . \qquad (36)$$

In equation (36), the quantity $\varphi_s(z_0)$ is unknown. However, as mentioned before, it can be shown that $0 < \xi(z) < 1$ for all real $z > 1$. In particular, this is also

true for $z=z_0$. Therefore, from the inequalities in (15) and (17), and the definitions of $\varphi_e(z)$ and $\varphi(z)$ in (16) and (18), it follows that $\varphi_s(z_0)$ is upper bounded by both $\varphi_e(z_0)$ and $\varphi(z_0)$. Using these two values as approximations for $\varphi_s(z_0)$ in (36) therefore yields two upper bounds for the coefficient B, denoted as $B_{u(1)}$ and $B_{u(2)}$ (corresponding to $\varphi_e(z_0)$ and $\varphi(z_0)$ respectively).

7. The cell delay.

We define the delay of a cell as the number of slots between the end of its arrival slot and the end of the slot during which the cell is transmitted and thus leaves the output queue. In [26], the following relationship was established between the pgf of the system contents S(z) and the pgf of the cell delay D(z), for discrete–time single–server queueing systems with general, possibly correlated, arrivals :

$$D(z) = \frac{S(z) - S(0)}{1 - S(0)} = \frac{S(z) - p_0}{1 - p_0} \quad . \tag{37}$$

Using this equation, the moments of the cell delay can be derived in terms of the moments of the system contents. For instance, for the mean cell delay $\bar{d}$, we have

$$\bar{d} = D'(1) = \frac{S'(1)}{1 - p_0} = \frac{\bar{s}}{p} \quad , \tag{38}$$

in agreement with Little's theorem. Using the upper bounds obtained for $\bar{s}$, we then get corresponding upper bounds for $\bar{d}$. Also, with (37), the tail distribution of the cell delay can be found from the tail distribution of the system contents. We have :

$$\text{Prob}[d=n] \cong -\frac{\theta}{z_0}\left[\frac{1}{z_0}\right]^n \frac{1}{p} = \frac{B}{p}\gamma^n \quad , \qquad n > T. \tag{39}$$

That is, the tail distribution of the delay also has a geometric form, with the same decay rate as the tail distribution of the system contents.

8. Numerical versus analytical results.

In order to check the accuracy of the analytical results derived in the previous sections, we have also analyzed the tagged output queue under the assumption of a finite, but "large" waiting room, using the same type of three–dimensional state description as in the analytical approach. In this case, however, a numerical solution of the resulting balance equations, rather than a solution in terms of generating functions, was performed. As the dimension of the set of balance equations grows rapidly with the number of inlets/outlets N of the switch and the (finite) size of the tagged output queue, this numerical approach was only practicable for small switch sizes, low to intermediate traffic loads, and relatively small values for the burstiness factor. (These restrictions do not exist for the analytical method.) Some results are presented below for N=4 and various values of p and K.

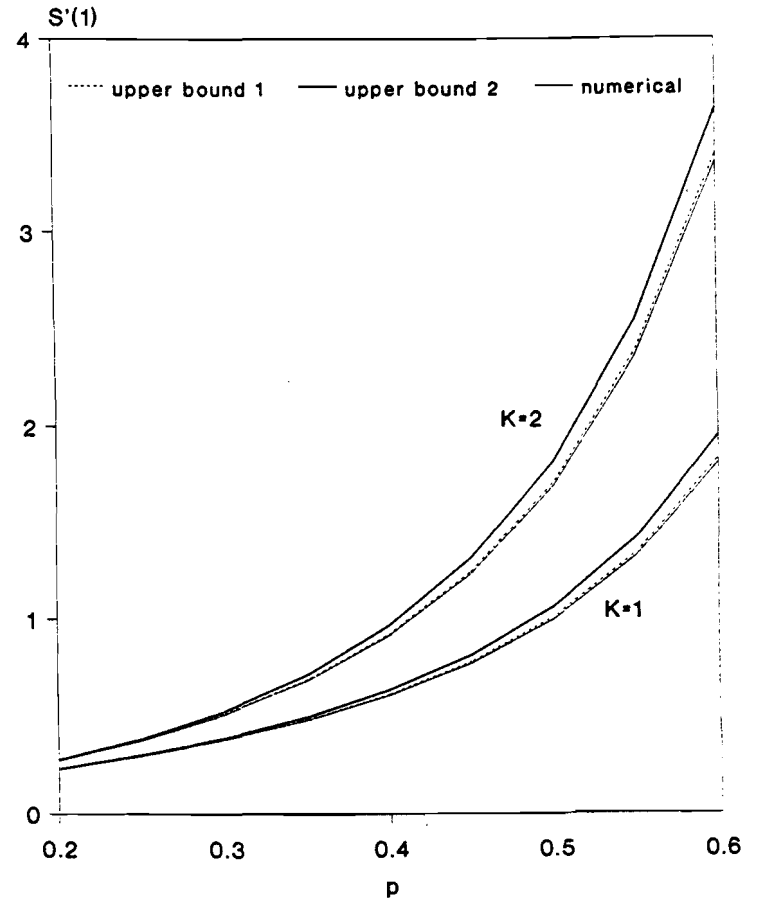


Fig. 2 Mean buffer contents versus load p : upper bounds and numerical results, for N=4 and K=1, 2.

In Fig. 2, we have plotted the analytical upper bounds $\bar{s}_{u(1)}$ and $\bar{s}_{u(2)}$ for the mean buffer contents, as well as numerical results, versus the load p, for K=1 and K=2. As one can see, $\bar{s}_{u(1)}$ is very close to the numerical results, which could be expected, based on equation (14). It is also clear that the difference between $\bar{s}_{u(2)}$ and the numerical results becomes larger as p and K increase, whereas $\bar{s}_{u(1)}$ is also accurate for large p and K. Similar conclusions can be drawn for the mean cell delay $\bar{d}$, as is illustrated in Fig. 3.

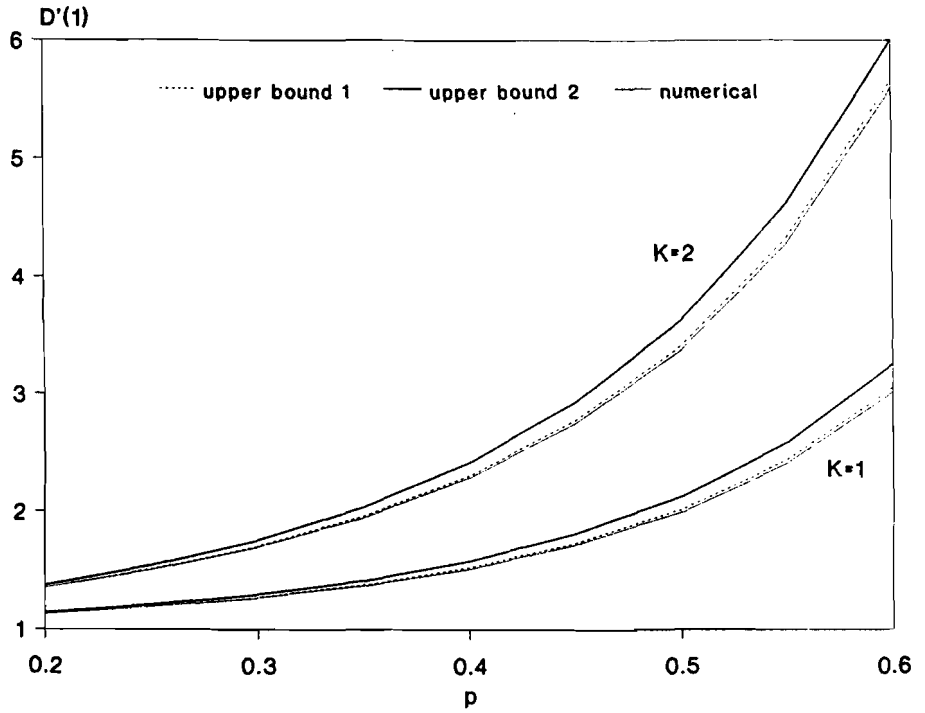


Fig. 3 Mean cell delay versus load p : upper bounds and numerical results, for N=4 and K=1, 2.

In Fig. 4, we compare the upper–bound tail distribution of the queue length given by $B_{u(1)}\gamma^n$ with numerical results, for K=2 and various values of the load p. The figure illustrates that the approximate method described above gives a very tight upper bound for the tail of the distribution. Moreover, the numerical results confirm that the analytical approach to obtain z_0 and γ is correct. In Table 1, we compare the derived upper bounds for the coefficient B, with numerical results. As expected, $B_{u(1)}$ is smaller than $B_{u(2)}$. Furthermore, $B_{u(1)}$ is very close to the numerical results.

Table 1 Coefficient B of the geometric form : upper bounds and numerical results, for N=4.

p	K	$B_{u(1)}$	$B_{u(2)}$	numerical
0.2	1	0.36950	0.37472	0.36884
0.4	1	0.24030	0.24991	0.23841
0.6	1	0.13393	0.14162	0.13246
0.4	2	0.05076	0.05232	0.05052
0.6	2	0.04853	0.05093	0.04814

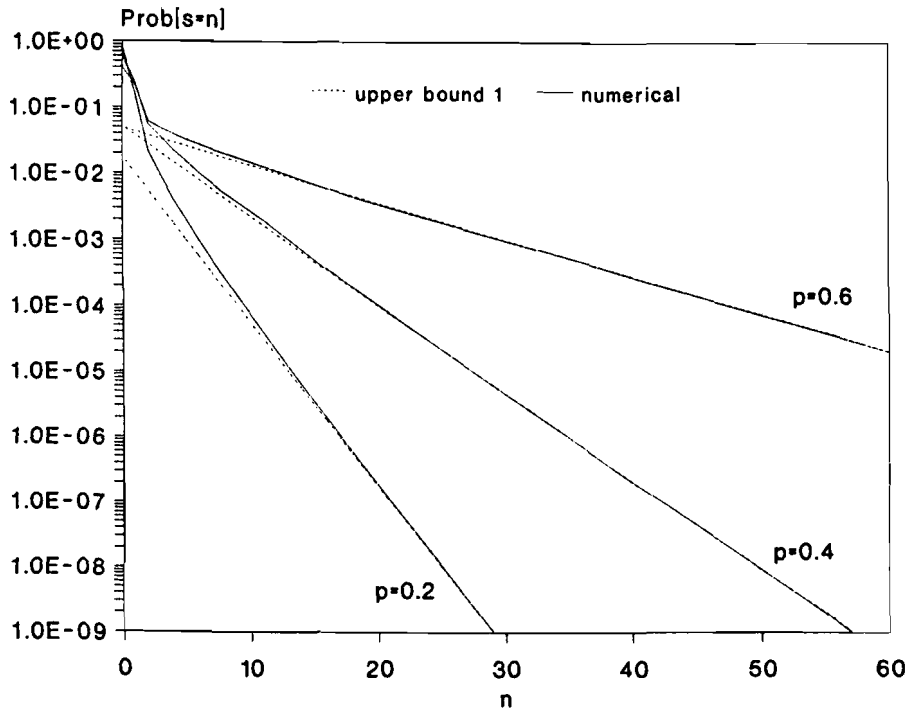


Fig. 4 Prob[s=n] versus n : upper bound 1 and numerical results, for N=4, K=2 and p=0.2, 0.4, 0.6 .

In our analytical approach, discussed in the previous sections, we have assumed an unlimited storage capacity for the (tagged) output queue. In practice, however, buffers are always of finite size, and a fraction of the arriving cells will be lost. It is important to be able to predict this so–called cell loss ratio (CLR) for a switch with a given configuration, and a prescribed buffer size S for the output queues. One way to deal with this problem is to solve the balance equations for the system by numerical means, and, from this, calculate the exact CLR–values.

However, as mentioned before, this becomes extremely time–consuming (and error prone) for high values of N and S. In order to overcome this difficulty, we have devised a heuristic approach to predict the cell loss ratio for a finite buffer from the tail distribution of the buffer contents in an unlimited–capacity queue. Specifically, our heuristic formula is based on the observation that the overflow probability in a continuous–time M/M/1/S queueing system, for high values of S, is nearly equal to the product of the probability of having more than S customers in an M/M/1 system, and a "correction factor" equal to $\frac{1-p}{p}$, where p is the load. Using the same formula in the current discrete–time context, we thus approximate the cell loss ratio for an output buffer of size S as follows :

$$CLR \cong Prob[s > S] \frac{1-p}{p} , \tag{40}$$

where, in view of (23),

$$Prob[s > S] = \frac{B}{1-\gamma} \gamma^{S+1} . \tag{41}$$

In Table 2, we compare the actual cell loss ratio (obtained by numerically solving a set of balance equations) with the heuristic in (40)–(41), for various values of p, K and S, using either $B_{u(1)}$ or $B_{u(2)}$ as an upper bound for B in equation (41). The results in Table 2 show that our heuristic approach leads to estimates of the CLR which, in general, are somewhat higher than the actual CLR, and, for realistic values of the load (p=0.8), are even quite close. We therefore believe that this approach can be very useful in practice for buffer dimensioning purposes.

Table 2 Cell loss ratio and Prob[s>S] $\frac{1-p}{p}$ for N=4.

p	K	S	CLR	Prob[s>S] $\frac{1-p}{p}$ upper bound 1	upper bound 2
0.2	1	8	1.216008E–7	1.587516E–7	1.609946E–7
0.4	1	15	1.248172E–6	1.846438E–6	1.920328E–6
0.6	1	30	2.335455E–5	3.416390E–5	3.612570E–5
0.8	1	20	4.259122E–2	4.614403E–2	4.833701E–2
0.2	2	15	8.780028E–6	1.609771E–5	1.624636E–5
0.4	2	40	4.700978E–7	8.483441E–7	8.744366E–7
0.6	2	30	3.050019E–3	4.771488E–3	5.007526E–3
0.8	2	30	6.405418E–2	6.552858E–2	6.845530E–2

9. Discussion.

Having demonstrated in the previous section the validity of the analytic techniques developed in this paper, we now present some further results, obtained by applying these techniques in the range of parameters where a numerical approach is unpractical. Fig. 5 shows the mean buffer occupancy in an output queue versus the load, for a switch with 16 inlets and outlets, for various values of the burstiness factor K of the sources. From the position of the curves in this figure, it is clear that, for a given value of the mean load p, the burstiness of the sources has a tremendous impact on the mean number of cells in the output queue. Specifically, we note that the congestion in an output queue may be seriously underestimated if a Bernoulli arrival process is used as an approximation for bursty traffic, since Bernoulli arrivals correspond to K=1, while bursty sources will typically give rise to much higher values of K. Similar conclusions can be drawn from Fig. 6, where we have plotted the cumulative tail probabilities Prob[s > n] of the output buffer occupancy versus n, at a given load p=0.6 , for different burstiness factors K.

In Figs. 7 and 8, we compare the correlated routing mechanism investigated in this paper, to the case of uncorrelated routing, which was studied in [22], for a switching element with 16 inlets and outlets. Specifically, Fig. 7 shows the mean output–queue contents versus the load, and Fig. 8 shows the tail probabilities Prob[s > n] versus n, at a load p=0.8 . Several conclusions can be drawn from Figs. 7 and 8. First, it can be observed that the required buffer space in an output queue is always much higher in the case of correlated routing than in the case of uncorrelated routing, regardless of the burstiness factor K of the sources, although the performance deteriorates more if the sources are more bursty. Second, the influence of the burstiness factor K on the level of congestion in an output queue is much more pronounced in case of correlated routing than for uncorrelated routing. This can be intuitively understood by the observation that independent destination addresses from cell to cell more or less "destroy" most of the burstiness of the arrival stream between the inlets of the switching element and the entrance of an output queue, while a correlated routing mechanism in some sense simply passes the burstiness of the sources to the entrance of the output queues. Third, routing correlation has a more substantial impact on the queueing behavior of a switch than input correlation (i.e., burstiness of the sources), although, in general, we may conclude that both types of correlation amplify eachother's effect.

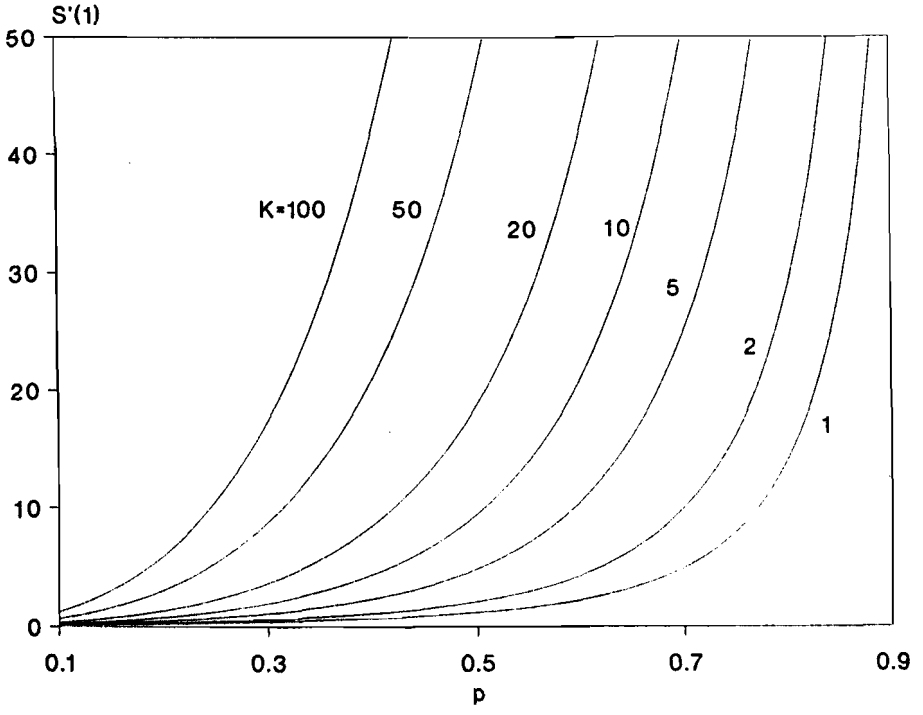


Fig. 5 Upper bound 1 of the mean buffer contents versus load p, for N=16 and K=1, 2, 5, 10, 20, 50, 100.

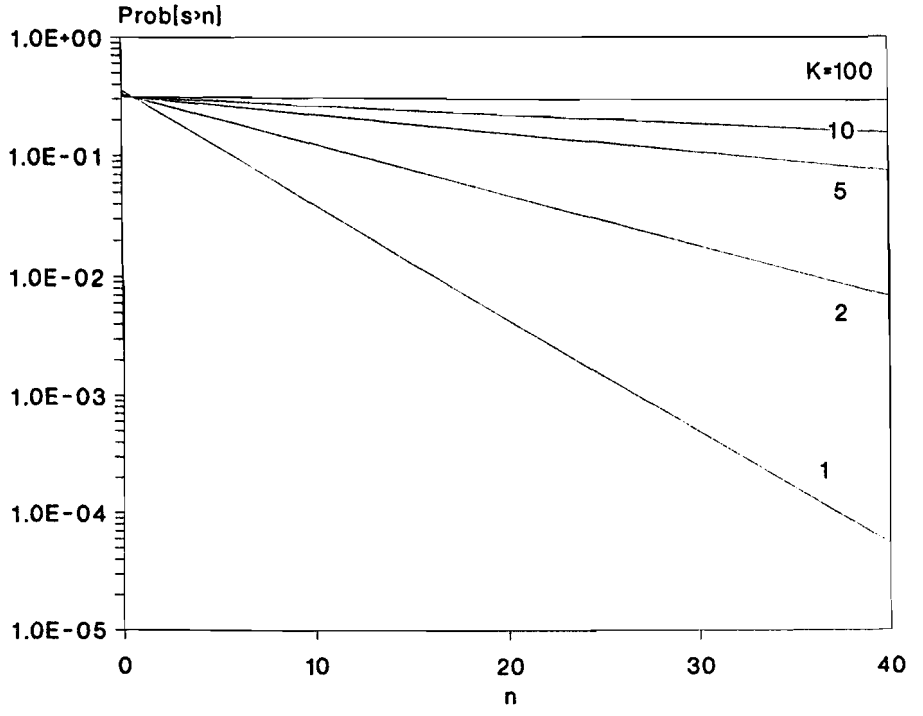


Fig. 6 Upper bound 1 of Prob[s>n] versus n, for N=16, p=0.6 and K=1, 2, 5, 10, 100.

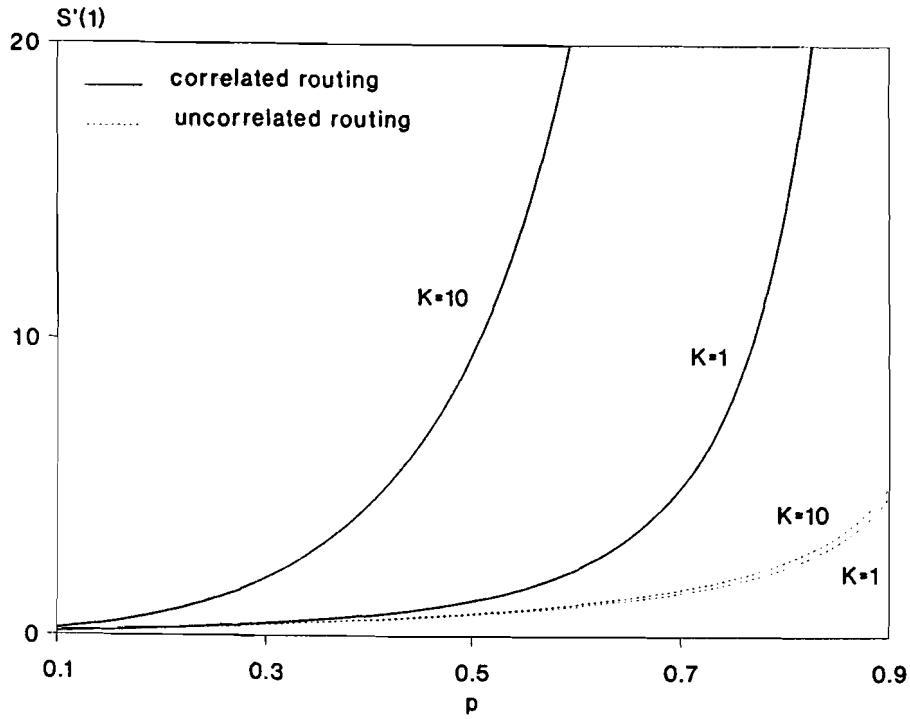


Fig. 7 Mean buffer contents versus load p for correlated routing (upper bound 1) and uncorrelated routing (exact), for N=16 and K=1, 10.

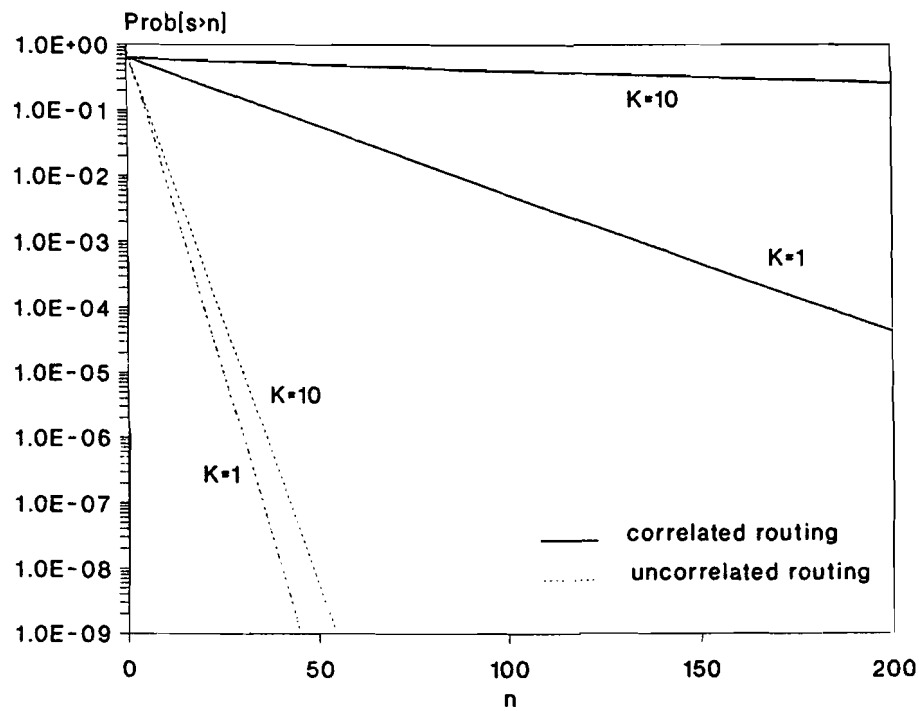


Fig. 8 Prob[s>n] versus n for correlated routing (upper bound 1) and uncorrelated routing (exact), for N=16, p=0.8 and K=1, 10.

10. Conclusions.

In analyzing the performance of an ATM switching element with bursty sources and correlated routing, we have derived explicit upper bounds for the means and tail distributions of the system contents of a tagged output queue and the delay of a cell. These upper bounds were obtained under the assumption of an infinite–capacity queue, by combining a generating–functions approach with approximation techniques. Comparison with numerical results shows that the obtained upper bounds are very tight, and can even be used to predict cell loss ratios in finite–capacity buffers. We have also observed that the queueing performance of the switching element deteriorates as the burstiness of the sources and/or the amount of correlation in the routing process get higher.

References.

[1] W.W. Chu and A.G. Konheim, On the analysis and modeling of a class of computer communication systems, *IEEE Trans. Commun.*, vol. COM–20 (1972), pp. 645–660.

[2] H. Kobayashi and A.G. Konheim, Queueing models for computer communications system analysis, *IEEE Trans. Commun.*, vol. COM–25 (1977), pp. 2–29.

[3] D. Towsley and J.K. Wolf, On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission schemes, *IEEE Trans. Commun.*, vol. COM–27 (1979), pp. 693–702.

[4] J.F. Hayes, Modeling and analysis of computer communications networks, *Plenum Press, New York, 1984.*

[5] H. Bruneel, Message delay in TDMA channels with contiguous output, *IEEE Trans. Commun.*, vol. COM–34 (1986), pp. 681–684.

[6] A.M. Viterbi, Approximate analysis of time–synchronous packet networks, *IEEE J. Sel. Ar. Commun.*, vol. SAC–4 (1986), pp. 879–890.

[7] H. Bruneel, On discrete buffers in a two–state environment, *IEEE Trans. Commun.*, vol. COM–35 (1987), pp. 32–38.

[8] H. Bruneel, On statistical multiplexers with randomly changing input characteristics, *Comput. & Ops. Res.*, vol. 13 (1986), pp. 481–487.

[9] M.J. Karol, M.G. Hluchyj and S.P. Morgan, Input versus output queueing on a space–division packet switch, *IEEE Trans. Commun.*, vol. COM–35 (1987), pp. 1347–1356.

[10] A. Gravey, J.–R. Louvion and P. Boyer, On the Geo/D/1 and Geo/D/1/n queues, *Perf. Eval.*, vol. 11 (1990), pp. 117–125.

[11] J.A. Schormans, E.M. Scharf and J.M. Pitts, Analysis of telecommunications switch model (Geo/D/1) with time priorities, *Electron. Lett.*, vol. 26 (1990), pp. 325–326.

[12] E. Desmet and G.H. Petit, Performance analysis of the discrete time multiserver queueing system Geo(N)/D/c/K, *RACE Workshop, Munich, 3–4 July 1990.*

[13] H. Bruneel, B. Steyaert, E. Desmet and G.H. Petit, An analytic technique for the derivation of the delay performance of ATM switches with multiserver output queues, *Int. Journal of Dig. and Anal. Commun. Syst.*, vol. 5 (1992), to be published.

[14] E. Desmet, B. Steyaert, H. Bruneel and G.H. Petit, Tail distributions of queue length and delay in discrete–time multiserver queueing models, applicable in ATM networks, *Proceedings ITC 13, Copenhagen, June 1991,* vol. *Queueing, Performance and Control in ATM,* pp. 1–6.

[15] H. Bruneel and B. Steyaert, Buffer requirements for ATM switches with multiserver output queues, *Electron. Lett.*, vol. 27 (1991), pp. 671–672.

[16] D. Anick, D. Mitra and M.M. Sondhi, Stochastic theory of a data–handling system with multiple sources, *Bell Syst. Tech. J.*, vol. 61 (1982), pp. 1871–1894.

[17] N. Janakiraman, B. Pagurek and J.E. Neilson, Multiplexing low–speed buffered data terminals, *IEEE Trans. Commun.*, vol. COM–28 (1980), pp. 1838–1843.

[18] H. Bruneel, Queueing behavior of statistical multiplexers with correlated inputs, *IEEE Trans. Comm.*, vol. COM–36 (1988), pp. 1339–1341.

[19] M. Hirano and N. Watanabe, Characteristics of a cell multiplexer for bursty ATM traffic, *Proceedings ICC '89, Boston, June 1989*, pp. 399–403.

[20] B. Steyaert and H. Bruneel, An effective algorithm to calculate the distribution of the buffer contents and the packet delay in a multiplexer with bursty sources, *Proceedings GLOBECOM '91, Phoenix, December 1991*, pp. 471–475.

[21] Y. Xiong and H. Bruneel, Performance of statistical multiplexers with finite number of inputs and train arrivals, *Proceedings INFOCOM '92, Firenze, May 1992*, pp. 2036–2044.

[22] B. Steyaert and H. Bruneel, Analytical study of the buffer contents in an ATM switch with Markovian arrivals, *Proceedings ISTN–92, Beijing, September 1992*, pp. 9–12.

[23] F. Bonomi, S. Montagna and R. Paglino, Busy period analysis for an ATM switching element output line, *Proceedings INFOCOM '92, Firenze, May 1992*, pp. 544–551.

[24] Y. Xiong and H. Bruneel, Approximate analytic performance study of an ATM switching element with train arrivals, *Proceedings ICC '92, Chicago, June 1992*, pp. 1614–1620.

[25] L. Kleinrock, Queueing Systems, Part I : Theory, *Wiley, New York, 1975.*

[26] Y. Xiong and H. Bruneel, Buffer contents and delay for statistical multiplexers with fixed–length packet–train arrivals, *Perf. Eval.* (1993).

ON ATM TRAFFIC CONTROL

CURRENT STATUS / OPEN ISSUES
UPC THROUGHPUT / REACTION TIME CHARACTERISTICS

G.H. Petit

ALCATEL-BELL Mfg. Co.
Research Center, Traffic Technology Dept.
Fr. Wellesplein 1
B-2018 Antwerp, BELGIUM

CONTENTS

▼ CURRENT STATUS (CCITT/ATM FORUM) - PART I

- Traffic Parameters & Descriptors
- Generic Cell Rate Algorithm
- Peak Cell Rate Definition
- Sustainable Cell Rate Definition
- Connection Traffic Descriptors
- User/Network Contract
- QoS Class & Cell Loss Priority
- UPC/NPC and CAC Open Issues

▼ UPC PERFORMANCE CHARACTERISTICS FOR NON-COMPLIANT CBR SOURCES - PART II

- Study Case (Virtual Scheduling Algorithm)
- UPC Throughput Characteristics
- UPC Reaction Time Characteristics

ALCATEL
BELL TELEPHONE

TRAFFIC PARAMETERS AND DESCRIPTORS

▼ **TRAFFIC PARAMETER:** Specification of a particular traffic aspect

- QUALITATIVE i.e. source type (e.g. telephony)
- QUANTITATIVE (e.g. Peak Cell Rate)

REQUIREMENTS:
- User/terminal understable for possible auto-compliance testing
- Participate in CAC schemes for meeting network performance requirements
- Enforceable by UPC/NPC mechanisms

▼ **ATM TRAFFIC DESCRIPTOR:** Generic list of ATM traffic parameters

▼ **SOURCE TRAFFIC DESCRIPTOR:** Subset of traffic parameters of a particular source selected from the ATM traffic descriptor

ALCATEL
BELL TELEPHONE

VPC/VCC ATM CELL TRAFFIC SPECIFICATION

▼ STATISTICAL APPROACH

- Conventional in teletraffic theory
- Ideal for addressing CAC issues
- No real-time compliance testing i.e. UPC/NPC possible

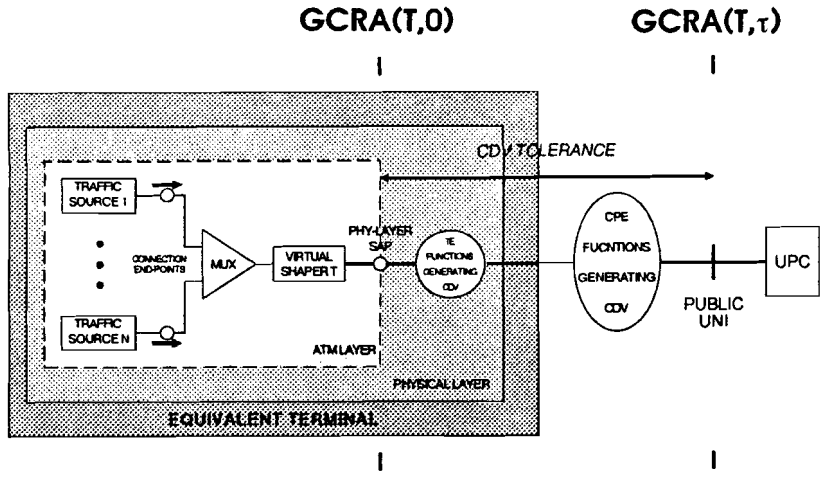
▼ OPERATIONAL APPROACH

- Focus on real-time conforming/non-conforming cells
- Use of worst-case traffic patterns as input for CAC
- Rule based parameter definition and compliance testing at physical interfaces (private/public UNI) with

GENERIC CELL RATE ALGORITHM (GCRA)

PEAK CELL RATE (PCR)

▼ PCR REFERENCE MODEL



▼ PCR DEFINITION (CBR + VBR SOURCES)

- **LOCATION:** PHY-SAP in an equivalent terminal representing the VPC/VCC
- **BASIC EVENT:** Request to send an ATM_PDU in the equivalent terminal
- **DEFINITION:** The PCR (R_p) of an ATM VPC/VCC = 1/T
  T: minimum Inter Arrival Time (IAT) between two basic events
  = VPC/VCC peak emission interval

i.e. A RULE BASED DEFINITION USING GCRA (T,τ)

ALCATEL
BELL TELEPHONE

CELL DELAY VARIATION (CDV) TOLERANCE

▼ ORIGIN OF CDV

- Access to slotted transfer medium
- Terminal multiplexing e.g. OAM cell insertion
- Cell multiplexing within Customer Premises Equipment (CPE) before UNI

▼ CONSEQUENCES OF CDV ON PEAK EMISSION INTERVAL

- Cell dispersion: IAT > T
- Cell clumping: IAT < T

▼ CDV TOLERANCE τ

- Distortion measure for cell clumping effect of a VPC/VCC at a physical inferface

▼ GCRA(T,τ)

- Conformance testing of VPC/VCC ATM peak cell rate at UNI

IMPACT OF CDV

CELL CLUMPING

- As τ increases ==> minimum spacing between cells decreases
- Maximum number of compliant back-to-back cells

$$B = 1 + [\tau / (T-1)]^{-} \quad \text{with}$$

$[X]^{-}$: largest integer smaller than X
T and τ expressed in cell time units

- Large τ values drastically impact the allocation of network resources

τ	B	N_{max} (# conn.)	Load (Erlang)
25	1	60	0.80
50	1	60	0.80
75	2	60	0.80
150	3	42	0.56
225	4	33	0.44
300	5	27	0.36
375	6	22	0.29

Impact of the CDV tolerance on the link load efficiency for the example of 2 Mbps CBR sources (T = 75).

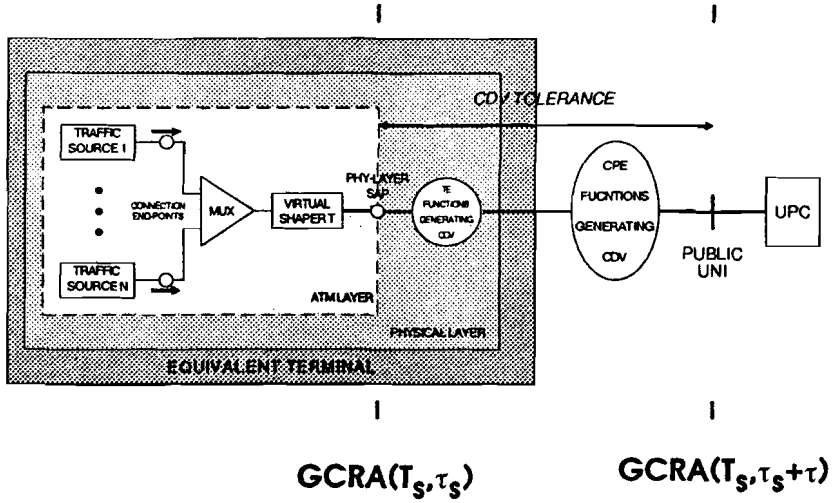
CELL DISPERSION

- Large dispersion impacts dejittering buffer sizes and end-to-end delays for circuit emulation services

ALCATEL
BELL TELEPHONE

SUSTAINABLE CELL RATE & BURST TOLERANCE

▼ SCR REFERENCE MODEL



▼ SCR DEFINITION (VBR SOURCES)

- **LOCATION:** PHY-SAP in an equivalent terminal representing the VPC/VCC
- **BASIC EVENT:** Request to send an ATM_PDU in the equivalent terminal
- **DEFINITION:** The SCR (R_S) and the burst tolerance τ_S of an ATM VPC/VCC is defined by the GCRA(T_S,τ_S) based on the arrivals of basic events where $T_S = 1/R_S$

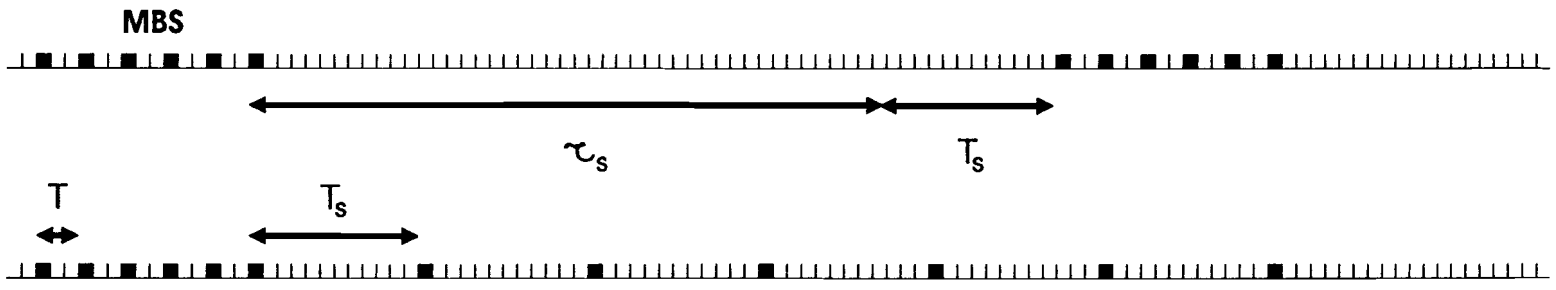
ALCATEL BELL TELEPHONE

INTERPRETATION OF SCR AND BURST TOLERANCE

- THE SCR IS NOT THE MEAN RATE OF A STOCHASTIC CELL GENERATION PROCESS

- GCRA(T_S,τ_S) DETERMINES THE MAXIMUM BURST SIZE (MBS) THAT MAY BE TRANSMITTED AT PEAK CELL RATE (1/T)

  $MBS = 1 + [\ \tau_S / (T_S - T)]^-$ with $[X]^-$: largest integer smaller than X



- MAXIMUM NUMBER M OF CELLS WITHIN ANY SLIDING WINDOW OF DURATION τ_S:

  $M = MBS + 1$

ALCATEL
BELL TELEPHONE

CONNECTION TRAFFIC DESCRIPTORS

PURPOSE

- Unabiguous specification (explicitly or implicitly) of the traffic characteristics of an ATM VPC/VCC for conformance testing at the UNI (private/public)

DEFINITION

- Source traffic descriptor e.g. PCR [MANDATORY]
  SCR and τ_s [OPTIONAL]
- CDV tolerance τ
- GCRA configuration rule i.e. interconnection pattern of cell rate monitoring algorithms

ALCATEL
BELL TELEPHONE

USER-NETWORK TRAFFIC CONTRACT

▼ PURPOSE

- Efficient operation of traffic control functions namely Usage Parameter Control (UPC) and Connection Admission Control (CAC)

▼ DEFINITION

- Connection traffic descriptor
- Requested Quality of Service (QoS) class
- Compliant connection definition
  (e.g. a certain amount of cells may be non-compliant)
  - for complinat VPC/VCCs, the requested QoS will be supported
  - for non-compliant VPC/VCCs, the network need NOT to respect the negotiated QoS

ALCATEL
BELL TELEPHONE

QoS CLASS & CELL LOSS PRIORITY (CLP)

- ATM VPC/VCC NEGOTIATES A SINGLE QoS CLASS
- QoS CLASS
  - Cell transfer delay (irrespective of CLP)
  - Cell delay variation sensitivity (irrespective of CLP)
  - Cell loss ratio for CLP=0 substream
  - Cell loss ratio for CLP=1 substream
- PEAK CELL RATE (PCR) AND CLP
  - Parameter for CLP=0 substream
  - Parameter for aggregate CLP=0+1 cell flow } single CDV tolerance τ
  - Parameters for OAM substream (optional T_{OAM} and τ_{OAM})
- SUSTAINABLE CELL RATE (SCR) AND CLP
  - Parameters for CLP=0 and/or CLP=1 and/or CLP=0+1?
    (SCR^0, $\tau_s{}^0$, SCR^1, $\tau_s{}^1$, SCR^{0+1}, $\tau_s{}^{0+1}$)

ALCATEL
BELL TELEPHONE

EXHAUSTIVE USER-NETWORK TRAFFIC CONTRACT

- PCR^{0}
- PCR^{0+1}
- PCR^{OAM}
- $SCR^{0}, \tau_s{}^{0}$
- $SCR^{1}, \tau_s{}^{1}$
- $SCR^{0+1}, \tau_s{}^{0+1}$

SOURCE TRAFFIC DESCRIPTOR AT PHY-SAP

CONNECTION TRAFFIC DESCRIPTOR AT UNI

- User data CDV tolerance τ
- OAM flow CDV tolerance τ
- GCRA configuration rule

- Cell transfer delay
- Cell delay variation sensitivity
- Cell loss ratio CLP=0
- Cell loss ratio CLP=1

REQUESTED QoS CLASS

- Compliant connection definition
- Experimental traffic parameters for proprietary use e.g. within private ATM networks

USER-NETWORK TRAFFIC CONTRACT

ALCATEL
BELL TELEPHONE

UPC/NPC OPEN ISSUES

▼ GCRA = REFERENCE ALGORITHM FOR CHECKING COMPLIANCE OF THE PCR (SCR) WITH RESPECT TO THE CONTRACTED TOLERANCE τ (τ_s) ON A PER CELL BASIS

- UPC/NPC algorithm not (yet?) standardized
- UPC/NPC transparancy w.r.t. GCRA
- Definition of a compliant connection
- Separate OAM cell flow enforcement
- Cell rate granularity and coding
  - T (T_s) versus $R_p = 1/T$ ($R_s = 1/T_s$)
- Upperbounds and granularity of tolerances τ (τ_s)
- UPC/NPC and CLP
  - discard versus tagging options
  - update of state variables (CLP=0 versus CLP=0+1)

CAC OPEN ISSUES (cont.)

▼ MULTIPLEXING SCHEME/PROCEDURE REQUIREMENTS

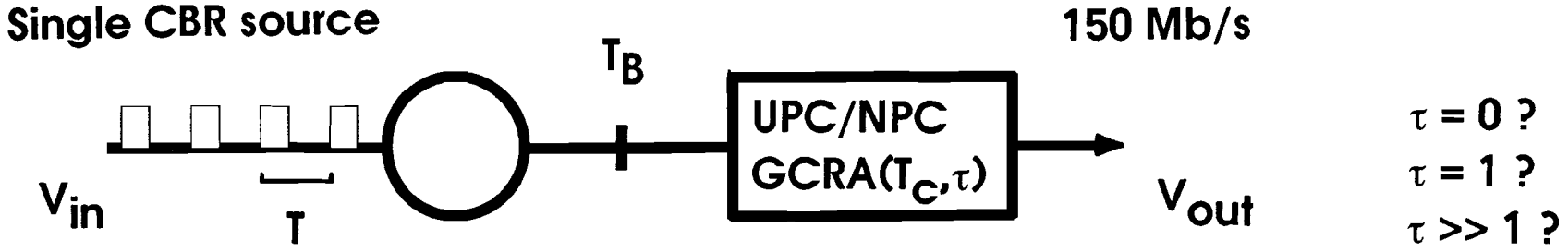
- Unambiguous definition of the CLP usage
  - QoS indicator <==> Discard Eligibility indicator
    (TRAFFIC CONTROL) (CONGESTION CONTROL)
- CDV tolerance limit as a function of PCR value for secure peak bandwidth allocation
- Statistical multiplexing using SCR and burst tolerance
  - network utilization = f(ATM network queue sizes,
    linkrates,
    peak cell rate (PCR),
    CDV tolerance τ,
    sustainable cell rate (SCR),
    burst tolerance τ_s)
- Need for fast resource management procedures?
- Need for shaping/spacing functions?
- Unambiguous definition and assessment of ATM-layer statistical network performance measures

STUDY CASE

▼ PURPOSE:

STUDY OF THE THROUGHPUT AND RESPONSIVENESS CHARACTERISTICS OF A UPC FUNCTION, BASED UPON THE GENERIC CELL RATE ALGORITHM (GCRA), FOR NON-COMPLIANT ATM CONNECTIONS SUPPORTING CONSTANT BIT RATE (CBR) SERVICES AS A FUNCTION OF THE CDV TOLERANCE τ

▼ SCHEMATIC REPRESENTATION:



Provisional definition of compliant CBR source:

$T >= T_C$: COMPLIANT
$T < Tc$: NON-COMPLIANT

○ Transfer to Slotted Medium

GENERIC CELL RATE ALGORITHM

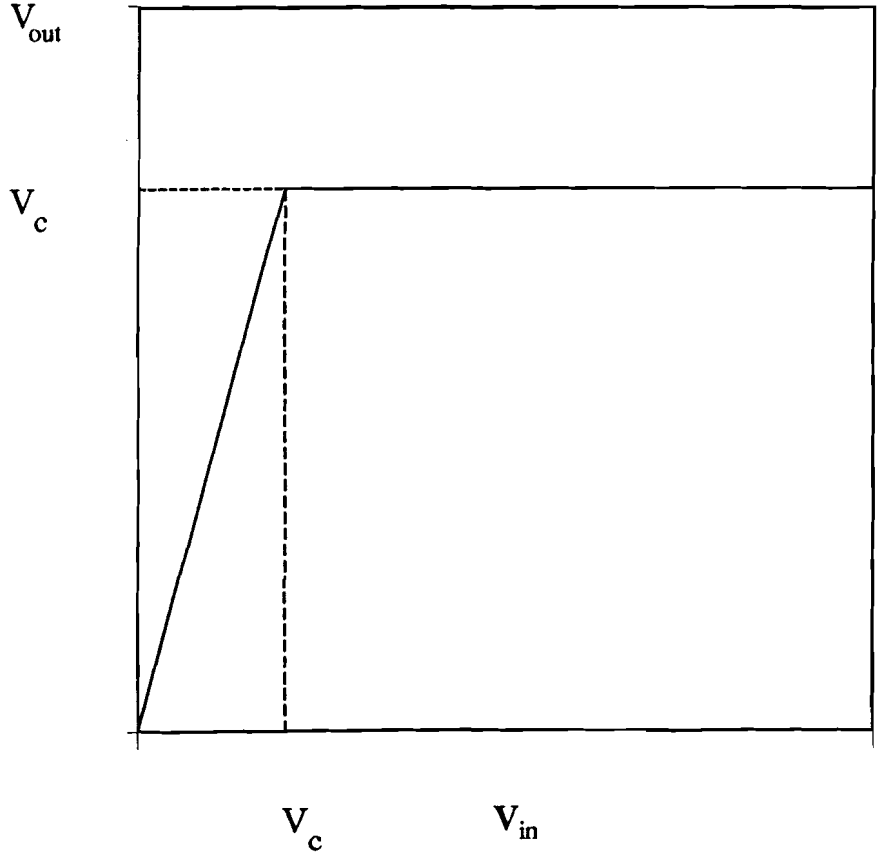
ALGORITHM

```
IF (PredictedAT <= ActualAT)
    THEN
        BEGIN
        PredictedAT := ActualAT + Tc;
        ReturnStatus(Go);
        END
    ELSE
        IF (PrdecitedAT > ActualAT + τ)
            THEN ReturnStatus(NoGo)
            ELSE
                BEGIN
                PredictedAT := PredictedAT + Tc;
                ReturnStatus(Go);
                END;
```

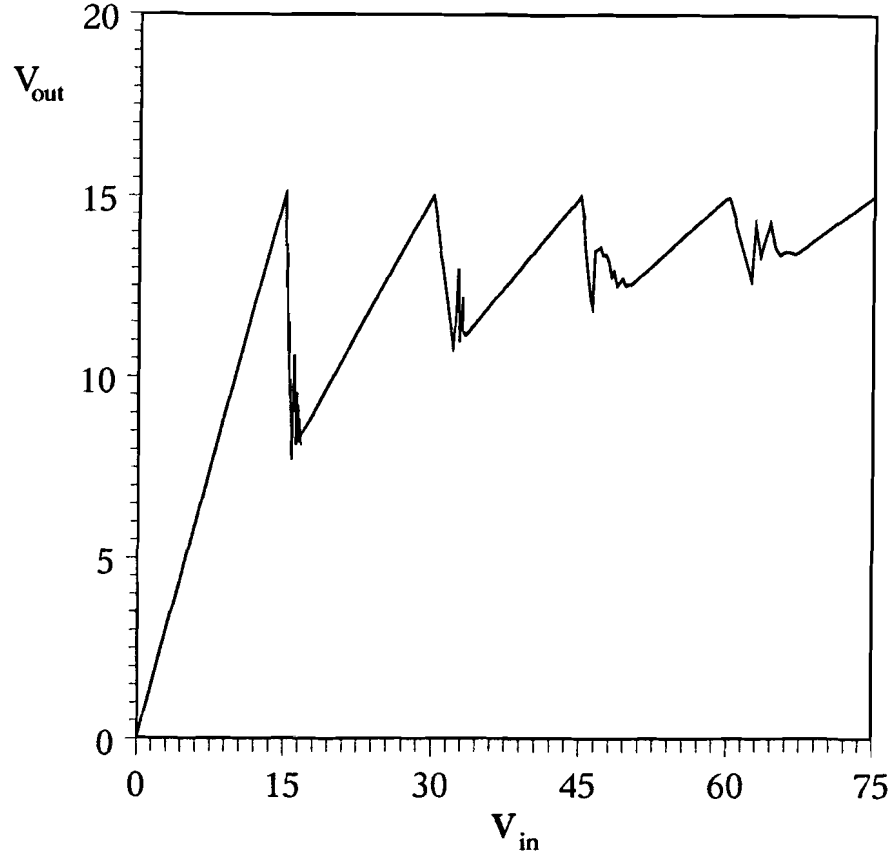
ALCATEL
BELL TELEPHONE

UPC THROUGHPUT CHARACTERISTICS

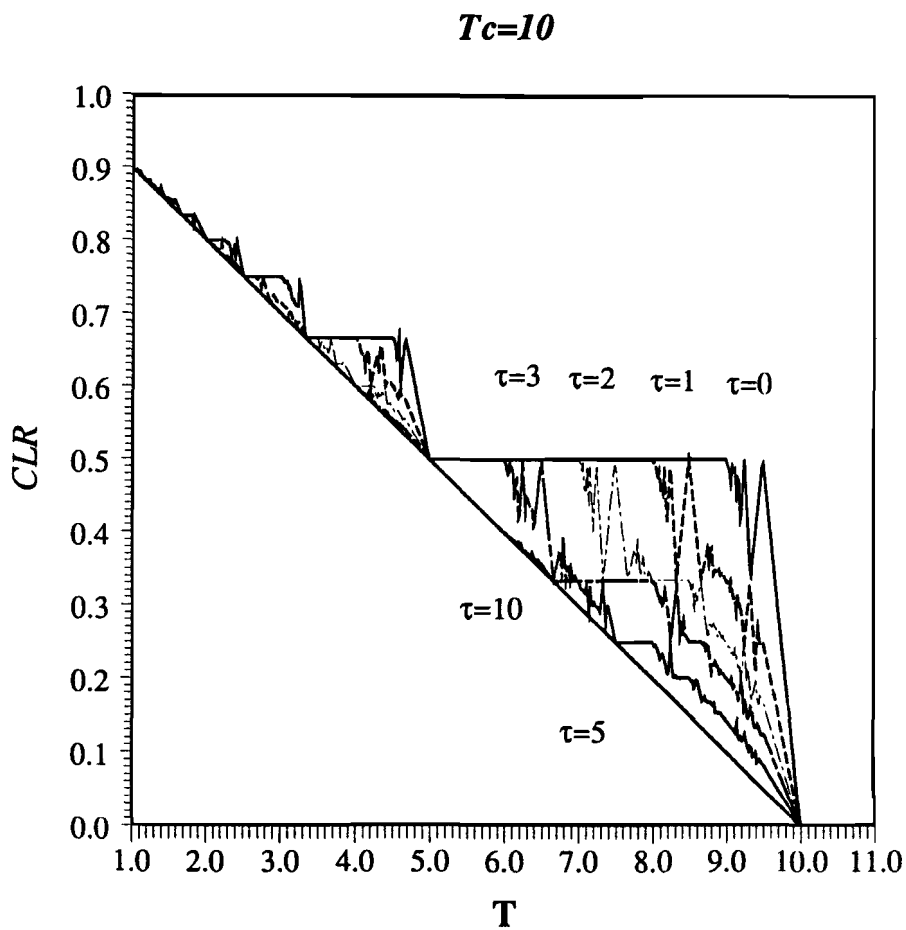
Ideal UPC behaviour

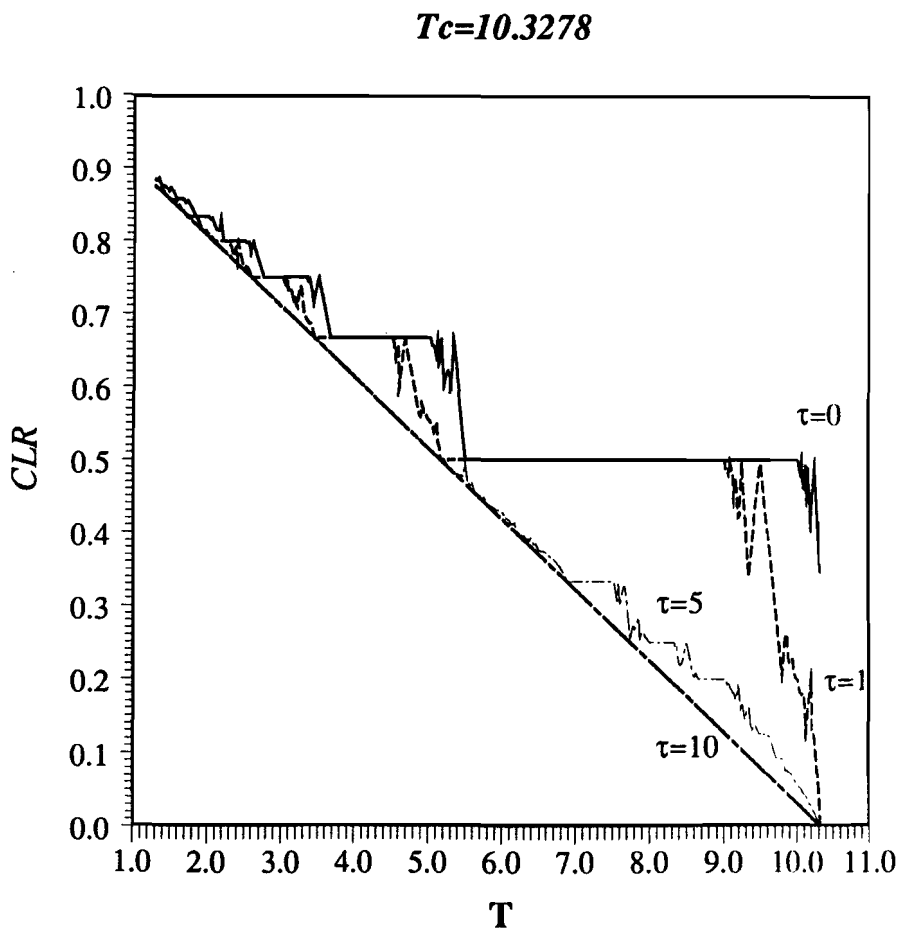


Vc = 15Mbit/sec



UPC THROUGHPUT CHARACTERISTICS (cont.)

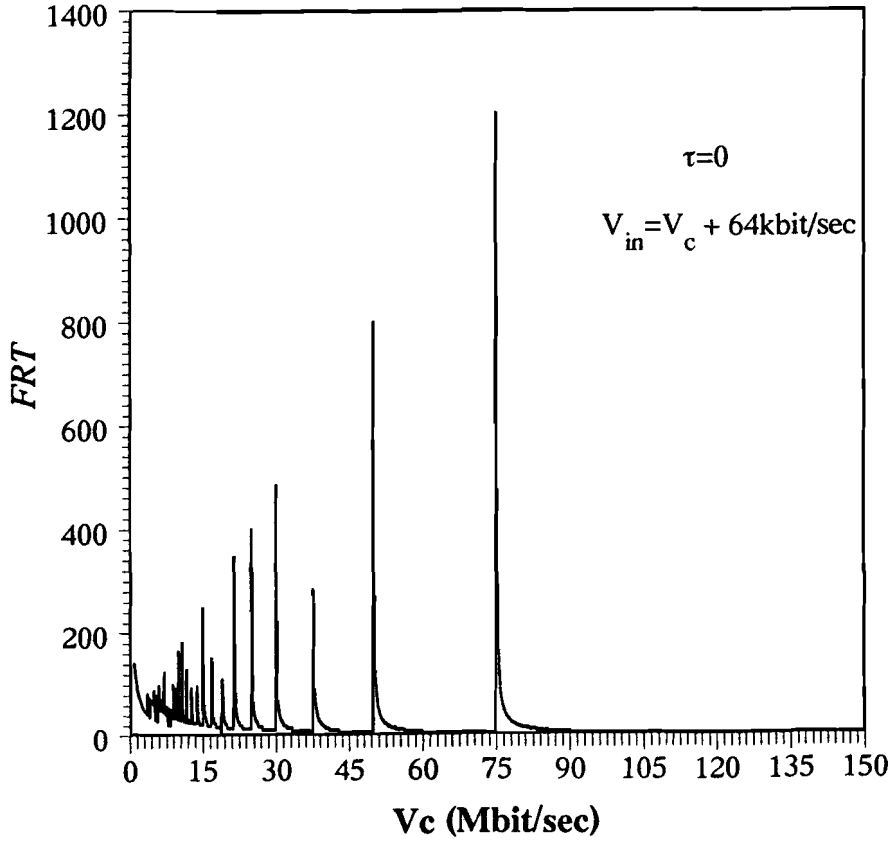




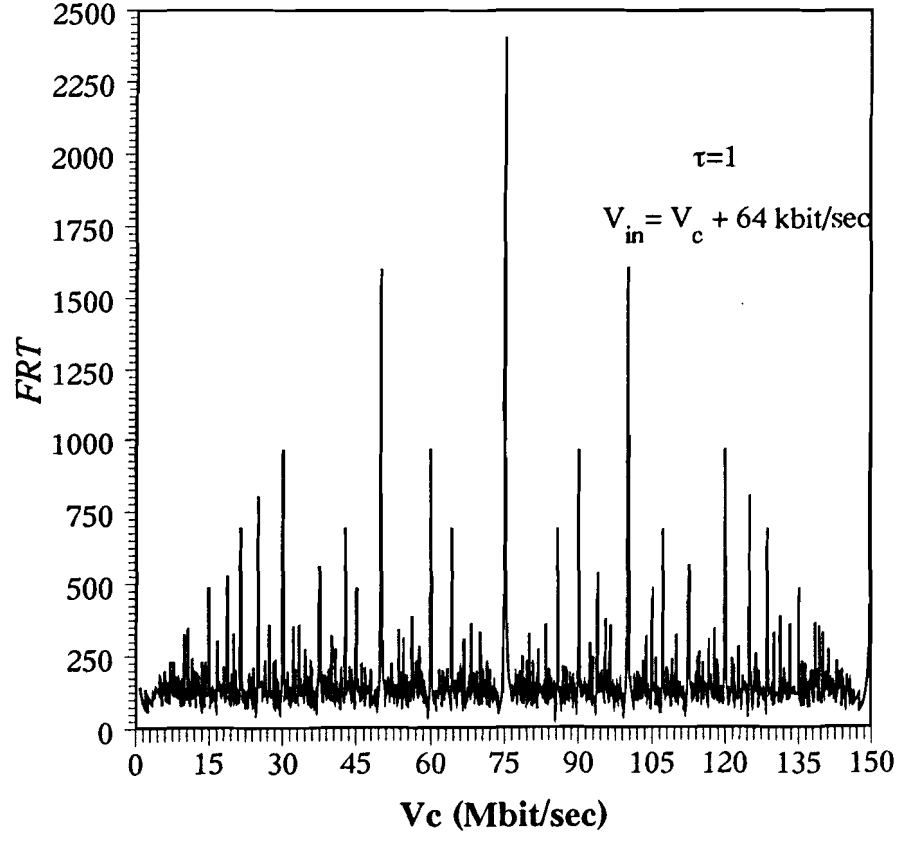
ALCATEL
BELL TELEPHONE

UPC RESPONSIVENESS CHARACTERISTICS

First Reaction Time (FRT) in cell time units



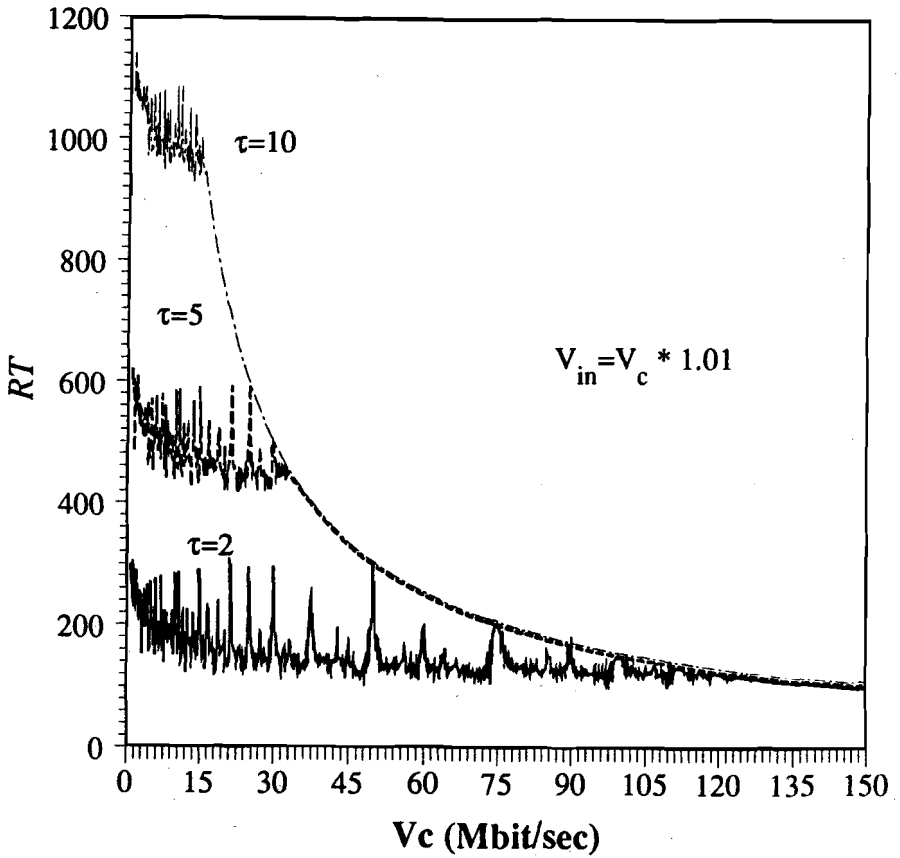
First Reaction Time (FRT) in cell time units



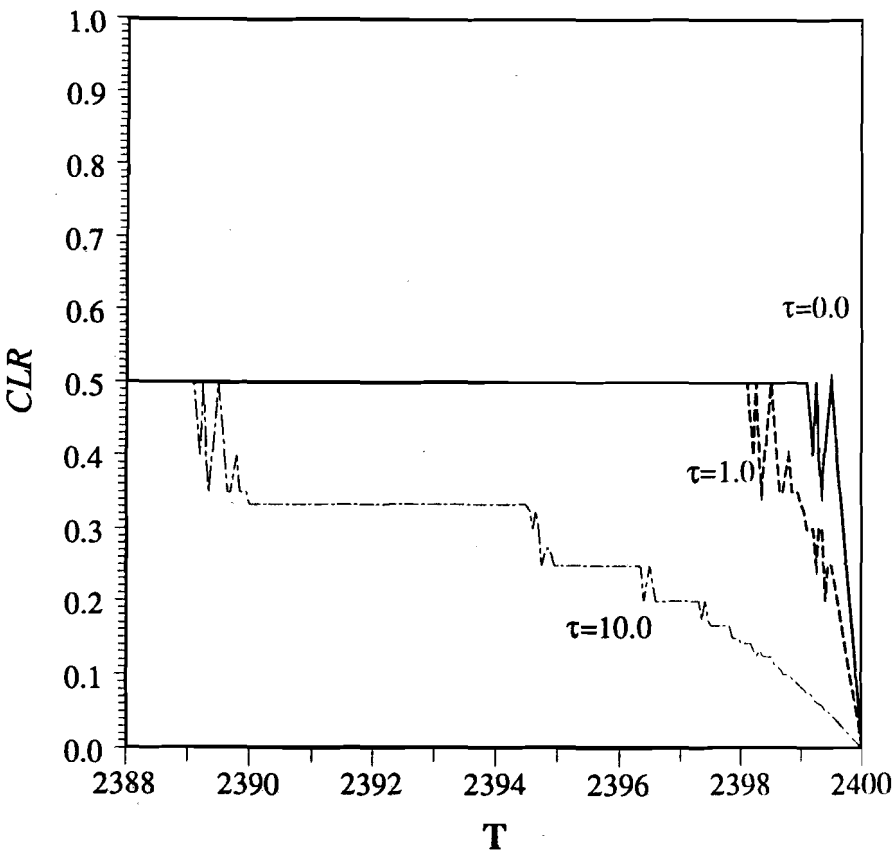
ALCATEL BELL TELEPHONE

UPC RESPONSIVENESS CHARACTERISTICS (cont.)

Average Reaction Time (RT) in cell time units



CBR of 64 kbit/sec (Tc=2400)



ALCATEL
BELL TELEPHONE

CONCLUSIONS

- IDEAL UPC INPUT-OUTPUT BEHAVIOUR CAN BE ATTAINED BY CHOOSING A SUFFICIENTLY HIGH τ VALUE (τ = T ?) TO ACHIEVE THROUGHPUT FAIRNESS
- HOWEVER A HIGH CDV TOLERANCE VALUE τ LEADS TO A LARGER REACTION TIME TO DETECT CONTRACT VIOLATIONS
- ON THE CONTRARY, SMALL τ VALUES MIGHT LEAD TO A SIGNIFICANT LOSS OF CELLS WHEN THE CONTRACTED BIT RATE FOR CBR SOURCES IS MARGINALLY EXCEEDED (THROUGHPUT UNFAIRNESS)
- IN ABSENCE OF SPACERS IN THE ATM NETWORK, MUCH MORE NETWORK RESOURCES ARE REQUIRED TO CATER FOR LARGE τ VALUES (CAC HAS TO BE BASED ON WORST CASE ON-OFF PATTERNS, PASSING TRANSPARANTLY THE UPC)

MORE IN DEPTH STUDIES ARE REQUIRED TO UNDERSTAND THE RELATIONSHIPS BETWEEN THE CONNECTION TRAFFIC DESCRIPTORS AT THE UNI AND THEIR IMPACT ON THE UPC THROUGHPUT AND RESPONSIVENESS CHARACTERISTICS IN REALISTIC TRAFFIC CASES.

The Safety Margin in the Leaky Bucket Policing Function

M.J.G. Dirksen[1]

Abstract

To guarantee the network performance objectives in ATM networks policing of the individual cell streams is necessary. Due to the asynchronous multiplexing technique, cell streams may experience a variable delay (jitter) before they arrive at the policing function. Despite the jitter, cells originating from a compliant source must pass the policing function. Therefore, the policing function must set its parameters in such way that a safety margin is provided. This paper presents an analytical approach to dimension the parameters of the leaky bucket policing algorithm such that it never discards cells from a cell stream suffering from jitter. In addition it is investigated how the safety margin is influenced if it is allowed that the policing function discards cells from a cell stream experiencing an extreme amount of jitter.

1 Introduction

It has been recognized that in high speed transmission networks traffic control must be based on preventive congestion control rather than reactive congestion control to guarantee the network performance objectives. In particular in ATM networks the *Connection Acceptance Control* (CAC) and the *Usage Parameter Control* (UPC) functions take care of a guaranteed network performance [1]. The task of the CAC is to establish whether sufficient network resources are available to allow a new connection. The CAC only accepts new connections as long as the network performance objectives are still met. At call set up the user has to declare a number of parameters (e.g. to characterize the connection and to indicate the required Quality of Service, etc.) upon which the CAC bases its decision. If the connection is allowed, these parameters are fixed in a *traffic contract*. Once a connection is admitted, the UPC, generally known as the policing function, monitors the offered traffic of each individual connection passing the User Network Interface to the network. The policing function takes action (e.g. discards cells) if the connection does not comply to the traffic parameters embodied in the traffic contract. This way the Quality of Service of all existing connections can be guaranteed. The CAC should take into account the worst case traffic passing through the policing function to avoid impairments to other connections.

According to CCITT [1] the *Peak Cell Rate* is a mandatory traffic parameter which must be declared at call set up. The peak cell rate determines the maximum rate at which the source is allowed to submit ATM cells to the network. However, if several ATM cell streams are multiplexed, cells of one connection may be delayed while cells of another are sent first at the output of the multiplexer. As a consequence the individual ATM cells may experience beside a fixed delay (e.g. propagation delay), a variable delay component, known as jitter. This variable delay is more or less random and depends among others on the traffic mix and the multiplexing protocol. Due to this *Cell Delay Variation* (CDV) cells may arrive at the policing function temporary spaced more closely to each other than transmitted by the source. The policing function momentarily observes a cell rate higher than the agreed peak cell rate and may decide to discard these cells. However, the policing function should not discard cells in an ATM connection if the source conforms to the traffic contract negotiated at connection establishment. Therefore, the policing function must allow a *safety margin* between the peak cell rate and the actual policed cell rate, to accommodate for the effects of the maximum CDV. The larger the safety margin, the worse the traffic the policing function must allow to the network. Since the CAC must be based on the worst case traffic that can pass the policing function, it is evident to keep the safety margin as small as possible. If the CDV is not bounded at a point where the policing function is performed, it is not possible to design a suitable policing function mechanism. Therefore, CCITT states that the cell delay tolerance should be declared by the user at call set up to properly dimension the policing function parameters.

[1]The author is with PTT Research, Dr. Neher Laboratories; St. Paulusstraat 4; P.O. Box 421; 2260 AK Leidschendam; The Netherlands. E-mail: M.J.G.Dirksen@research.ptt.nl

This paper presents for a specific policing algorithm, the leaky bucket, how to determine the size of the policing parameters to ensure that a cell stream suffering from jitter is policed correctly. In this paper it is assumed that the source transmits at a constant peak cell rate (CBR source) and the declared peak cell rate can be expressed as a fraction x/y of the link rate (both x and y are integers). Based on two traffic parameters, the peak cell rate and the maximum CDV, the required values of the policing parameters are derived such that cells from a compliant source are never discarded. However, it may happen that the probability that this worst case CDV occurs is extremely small, much smaller than the cell loss probability that due to buffer overflow or transmission errors. In that case the network operator may decide to reduce the safety margin such that the probability the policing function discards cells, suffering from extreme jitter, is in the same order of magnitude as the probability of e.g. buffer overflow. If the safety margin is reduced, less network resources (like link and buffer capacity) have to be reserved and the CAC may allow more connections resulting in a better network efficiency.

2 The leaky bucket algorithm

In the literature several algorithms for the policing function have been proposed such as the moving window, the jumping window, the leaky bucket and a number of variants [2,3]. They differ with respect to the reaction time to parameter violations, the allowed worst case traffic and the implementation complexity. In this paper the *leaky bucket* algorithm [4,5,6] is used because of its good performance together with a low implementation cost.

The leaky bucket controls the amount of traffic entering the network through three parameters: the splash value S, the leak value L and the bucket limit B_{lim}. The values of these parameters are determined and fixed at call set up and fix the amount of traffic that is able to pass the policing function. In addition to these parameters there is a variable B_{lev}, representing the level in the leaky bucket. Each time a cell is passed to the network, the value of the bucket level B_{lev} is increased with the splash value S. Meanwhile, the bucket level is periodically decremented with the leak value L. If a cell arrives at the policing function, the bucket level is compared with the bucket limit. If the bucket level is smaller or equal to the bucket limit, the cell is passed to the network and the bucket level is increased with the splash value S. If upon a cell arrival the bucket level is larger than the bucket limit, the cell is discarded. The leaky bucket does not queue the cells, either the cell is passed to the network or the cell is discarded.

The leak to splash ratio L/S controls the allowed sustained cell rate that is passed to the network. The ratio B_{lim}/S determines the amount of clustering of cells. The required safety margin to cope with jitter can either be obtained by increasing the leak to splash ratio L/S, thus allowing a higher sustained cell rate, or by increasing B_{lim}/S resulting in admitting a more 'bursty' cell stream to the network.

If the cell inter arrival time does not match an integer multiple of slot times, the bucket limit may have to be set to a value larger than zero even if the cell stream is not multiplexed with other cell streams. Figure 1 shows

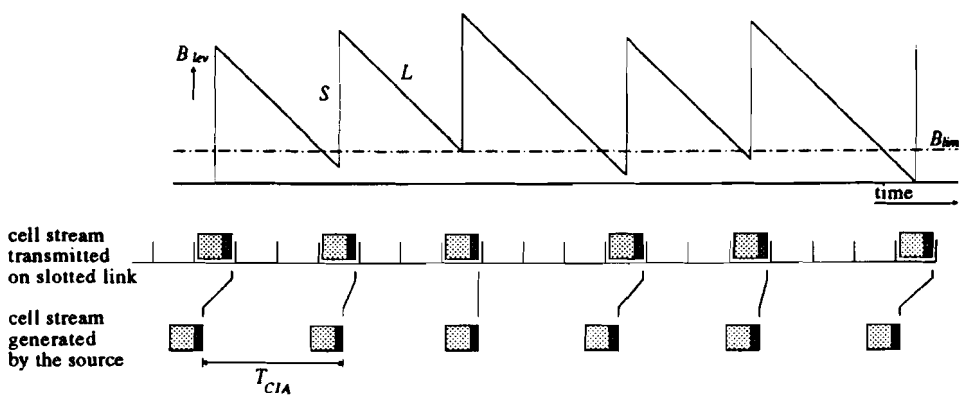


Figure 1: *Even for a constant cell rate source, the bucket limit may be set to a value larger than zero due to the slotted structure of ATM.*

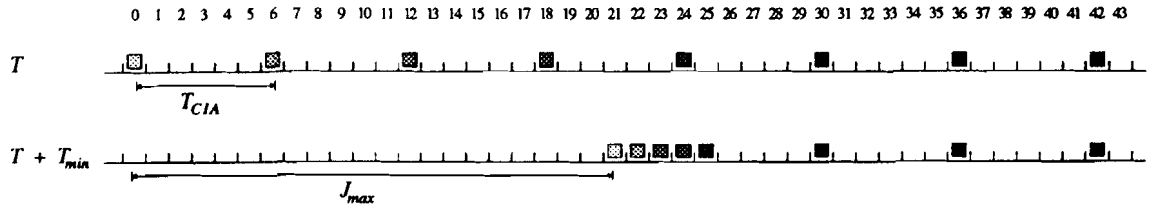


***Figure 2:** Worst case situation occurs if a cell endures a maximum jitter while other cells experience a cell delay as small as possible.*

a cell stream with a cell inter arrival time 3.4 (=17/5). Since the source has to wait until the beginning of a new slot, the cell inter arrival time on the link alternates between 3 and 4 such that the long term average cell inter arrival time equals 3.4. If the leak to splash ratio is chosen equal to the peak cell rate $L/S = 5/17$, it appears that the bucket limit must be set to $B_{lim}/S = 4/17$. Although not formally proven, empirically it has been found that for constant cell rates equal to a fraction of x/y of the link rate (x and y both integers) the bucket limit becomes

$$B_{lim} = \frac{L}{S} - \frac{\gcd(L,S)}{S} \qquad (1)$$

where gcd(L,S) means the 'greatest common divisor' of the terms L and S. The formula holds only if $L/S = x/y$.

3 Dimensioning the leaky bucket parameters

3.1 The worst case cell stream

The worst case situation occurs if a cell encounters a maximum CDV, while the next cells of the same connection encounter a delay as minimal as possible. Figure 2 shows the worst case situation for a CBR cell stream with a cell inter arrival time $T_{CIA} = 6$ and a maximum jitter of $J_{max} = 21$. The upper line corresponds to the cell stream as transmitted by the source starting at a moment T. The lower line corresponds to the cell stream arriving at the policing function a time $T + T_{min}$ later. T_{min} is the minimal delay for a cell to travel from the source to the policing function. The cell transmitted at moment T by the source suffers from the maximum jitter, and thus arrives at moment $T + J_{max}$ at the policing function. Subsequent cells of the connection arrive at the policing function experiencing the minimal possible delay. In case T_{CIA} is larger than J_{max} this means subsequent cells do not experience any queueing delay. If T_{CIA} is smaller than or equal to J_{max} then a number of cells arrive in consecutive slots as in figure 2. In general it appears that in the worst case situation the cell stream arriving at the policing function consists of a 'bursty' phase and a 'stationary' phase. Figure 3 shows an example of a worst case cell stream arriving at the policing function with $T_{CIA} = 16$ suffering a maximum jitter $J_{max} = 100$.

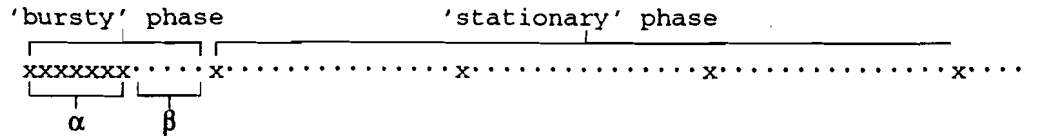


***Figure 3:** 'Bursty' and 'stationary' phase of a worst case cell stream ($T_{CIA} = 16$, $J_{max} = 100$).*

During the 'bursty' phase there are α cells in consecutive slots, followed by β empty slots. Then, cells arrive spaced T_{CIA} slots apart. Before these observations can be used to derive the required bucket limit, the parameters α and β have to be expressed in the 'known' parameters T_{CIA} and J_{max}.

Let t_1 be the number of time slots since T and t_2 the number of time slots since $T + T_{min}$. Cells are generated by the source at instances

$$t_1(\alpha) = \lfloor (\alpha - 1) \times T_{CIA} \rfloor$$

where $t_1(\alpha)$ denotes the time slot that contains the α^{th} cell and $\lfloor ... \rfloor$ represents the integer part of the expression.

A similar equation can be obtained for the cell stream arriving at the policing function. As long as the cell stream is in the 'bursty' phase, cells arrive at instances

$$t_2(\alpha) = J_{max} + (\alpha - 1)$$

where $t_2(\alpha)$ represents the time slot number where cell α arrives at the policing function. As long as $t_1(\alpha)$ is smaller than or equal to $t_2(\alpha)$, then cell α belongs to the 'bursty' phase. From this, and the two equations above α can be obtained:

$$t_1(\alpha) \leq t_2(\alpha)$$

$$\lfloor (\alpha - 1) \times T_{CIA} \rfloor \leq J_{max} + (\alpha - 1)$$

$$\alpha \leq \left\lfloor \frac{J_{max}}{T_{CIA} - 1} \right\rfloor + 1$$

The number of cells in the bursty phase corresponds to the nearest integer lower than or equal to the right hand expression:

$$\alpha = \left\lfloor \frac{J_{max}}{T_{CIA} - 1} \right\rfloor + 1$$

The number of empty slots before the 'stationary' phase begins is the difference between the time slot number of the first cell of the stationary phase and the first time slot after the last cell of the bursty phase.

$$\beta = \alpha \times T_{CIA} - (J_{max} + \alpha)$$

$$\beta = \alpha \times (T_{CIA} - 1) - J_{max}$$

3.2 The leak rate

The leak rate is determined by the amount L that is decremented from the bucket level per slot time. The leak to splash ratio L/S determines the sustained cell rate that is allowed to the network. The lower bound for this value corresponds to the peak cell rate $1/T_{CIA}$ negotiated at call set up. If the leak to splash ratio is chosen equal to $1/T_{CIA}$, then any clustering of cells due to jitter has to be absorbed by the bucket limit.

If a lower bucket limit is desired, L/S must be chosen higher so that after a splash a larger amount leaks away. A special case is if the allowed sustained cell rate is chosen equal to the cell rate in the bursty phase. In the bursty phase α cells must be allowed in $\alpha+\beta$ time slots. The corresponding leak to splash ratio becomes:

$$\frac{L}{S} = \frac{\alpha}{\alpha + \beta}$$

If L/S is chosen equal to the cell rate in the bursty phase, the bucket just becomes empty if the first cell of the stationary phase arrives.

Figure	L/S	B_{lev1}	B_{lev2}	B_{lim}
a	7/12	2.5	0	2.5
b	1/6	5	5	5
c	1/16	5.625	6.25	6.25

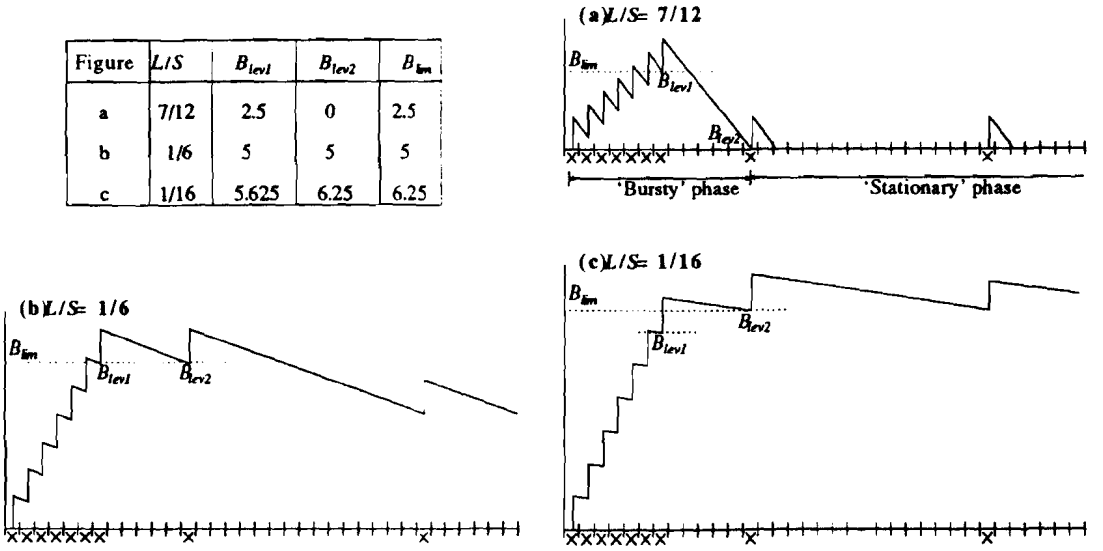


Figure 4: Bucket level for various L/S values (T_{CIA} = 16, J_{max} = 100)
a) L/S = 7/12 (bit rate in bursty phase)
b) L/S = 1/6 (Blev1 = Blev1)
c) L/S = 1/16 (bit rate in stationary phase)

3.3 The bucket limit

Once the leak and the splash have been chosen, the corresponding bucket limit can be calculated. To determine B_{lim} the bucket level has to be observed at two moments. The first moment is when α cells have arrived in consecutive slots. The second bucket level of interest is when the first cell of the stationary phase arrives. The highest bucket level of these two moments corresponds to the bucket limit B_{lim}. Which bucket level is higher depends on the chosen leak to splash value, see Figure 4.

The first bucket level is determined after α time slots when α-1 splashes have been added to the bucket while α-1 times an amount L has leaked away. The bucket level is increased to

$$B_{lev1} = (\alpha - 1)\times S - (\alpha - 1)\times L = (\alpha - 1)\times(S - L) \qquad (11)$$

The second bucket level under investigation is β slots later when the first cell of the 'stationary' phase arrives. At that moment one splash has been added to the bucket while the bucket leaked (β+1) time slots a value L:

$$B_{lev2} = B_{lev1} + S - (\beta + 1)\times L \qquad (12)$$

The bucket limit B_{lim} is the maximum of the two bucket levels B_{lev1} and B_{lev2}. Which bucket level must be chosen depends on the value of S - (β+1)L. The two bucket levels B_{lev1} and B_{lev2} are equal if this expression is zero. This corresponds to a leak to splash ratio of

$$\frac{L}{S} = \frac{1}{\beta + 1}$$

If a higher leak to splash ratio is chosen B_{lev1} is larger than B_{lev2} else B_{lev2} is larger than B_{lev1}. The bucket limit can be derived from (11) and (12) and written in one formula becomes

$$B_{lim} = (\alpha - 1)\times(S - L) + \{S - (\beta + 1)\times L\}^{>0} \qquad (14)$$

The term enclosed by $\{...\}^{>0}$ is only taken into account when its value is larger than zero. If α and β are substituted in Equation (14), we obtain:

$$B_{lim} = \left\lfloor \frac{J_{max}}{T_{CIA}-1} \right\rfloor \times (S-L) + \left\{ S - \left(\left\lfloor \frac{J_{max}}{T_{CIA}-1} \right\rfloor \times (T_{CIA}-1) + (T_{CIA}-J_{max}) \right) \times L \right\}^{>0} \quad (15)$$

It should be noted that this formula holds only if T_{CIA} is an integer value. However, the formula can easily be extended to hold for any $T_{CIA} = y/x$. Though it is not formally proven it appears that for a fractional T_{CIA} the bucket limit must be set to the sum of equation (15) and equation (1). Moreover it should be noted that this only holds if $L/S = x/y$, while equation (15) holds van any L/S. Finally it can be concluded that equation (15) also holds for fractional T_{CIA} if

$$\frac{L}{S} \geq \frac{1}{\lfloor T_{CIA} \rfloor} = \frac{1}{\lfloor y/x \rfloor}$$

4 Reducing the safety margin

In the previous paragraph a formula has been derived to dimension the leaky bucket parameters such that never a cell is discarded due to jitter. However it may be that the worst case situation is likely to happen only with a very small probability. If the probability that the worst case situation occurs is much lower than the agreed cell loss probability, it may be advantageous for the network operator to reduce the safety margin. As a result the policing function may discard cells of a cell stream suffering from extreme jitter, but as long as the probability that the policing function discards cells of a compliant source is in the same order of magnitude as the cell loss probability, the Quality of Service objectives are still met. Moreover, if the safety margin is reduced less network resources have to be claimed and can be used for other connections, thus increasing the network efficiency.

4.1 The simulation model

In order to investigate the size of the required safety margin to keep the cell discard probability below a predefined fixed value simulations have been performed for various traffic situations. The simulation model is shown in figure 5. A constant bit rate (CBR) source is multiplexed with some interfering background traffic before it arrives at the policing function. In the FIFO multiplexer the CBR source is subject to jitter due to interference with the background traffic. As background traffic a number of other CBR sources is chosen, each with a different cell rate. In addition, as a worst case approximation for this type of background traffic a bulk traffic generator has been used that generates cells according to a poisson distribution.

The following parameters characterize the simulation:

- The load of the CBR source under investigation ρ_{CBR}.
- The total load in the multiplexer ρ_{tot}.
- The number of interfering CBR sources N. (not relevant for the poisson case.)
- The leak value L and the bucket limit B_{lim}. (the splash value S is fixed to 1.)

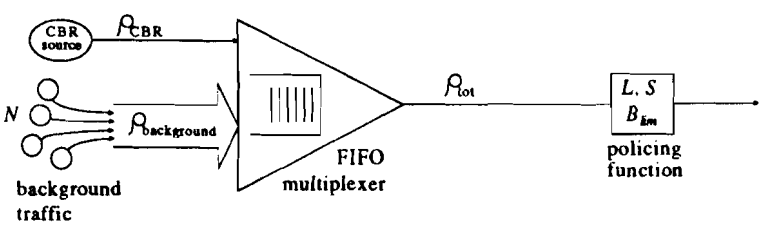


Figure 5: Simulation model with parameters.

4.2 Effects of simulation parameters on the safety margin

It appears that the higher the total load ρ_{tot}, the higher the bucket limit must be chosen to obtain the same cell discard probability. This may be evident, since a higher ρ_{tot} means more interference resulting in a larger fluctuations in the buffer occupancy.

A more interesting aspect is if the load of the CBR source ρ_{CBR} is changed while maintaining the total load at a constant value. For a low CBR load there is a large amount of background traffic. In this case the CBR cell stream experiences more cell delay variation, but since the cell stream has a large cell inter arrival time, the effect on the cell stream may not be large. On the other hand, for a CBR source with a high cell rate there is only a small amount of interfering traffic, but if a cell experiences extra delay, it may affect the cell stream considerably. Figure 6 shows the average experience queueing delay normalized to the cell inter arrival time of the tagged CBR source. It appears there is a load where the jitter has a maximum effect on the CBR cell stream. This effect can be found again in figure 7 where the required bucket limit is shown to obtain a cell discard probability for several ρ_{CBR} at a constant ρ_{tot}.

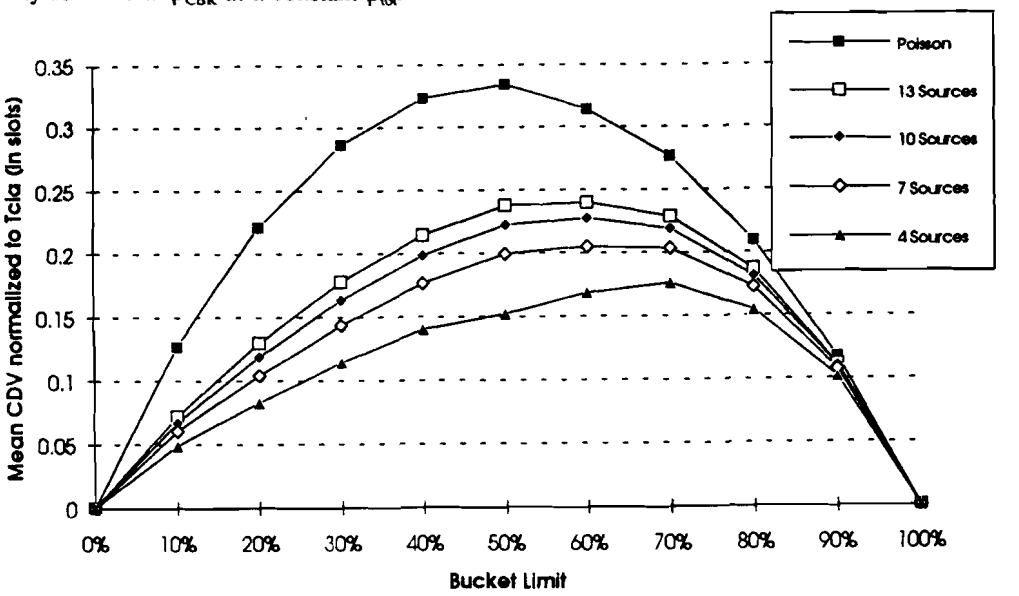


Figure 6: Average experienced queueing delay normalized to T_{CLA} for various ρ_{CBR} at a constant ρ_{tot}.

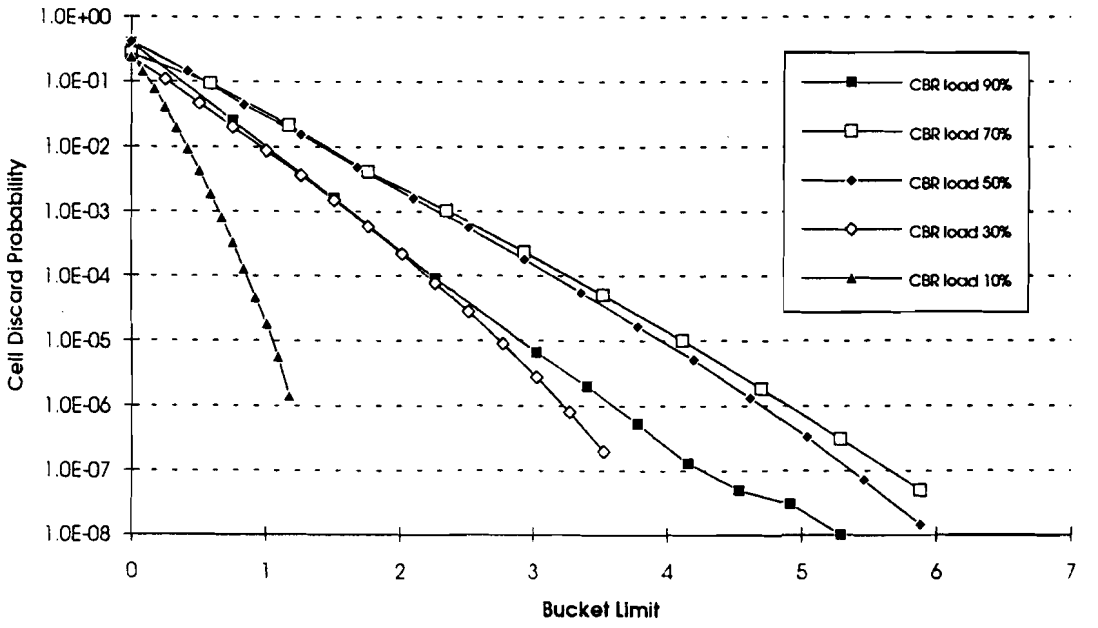


Figure 7: Cell discard probability as function of B_{lim} for several values of ρ_{CBR} (ρ_{tot} = 0.8, N = 100).

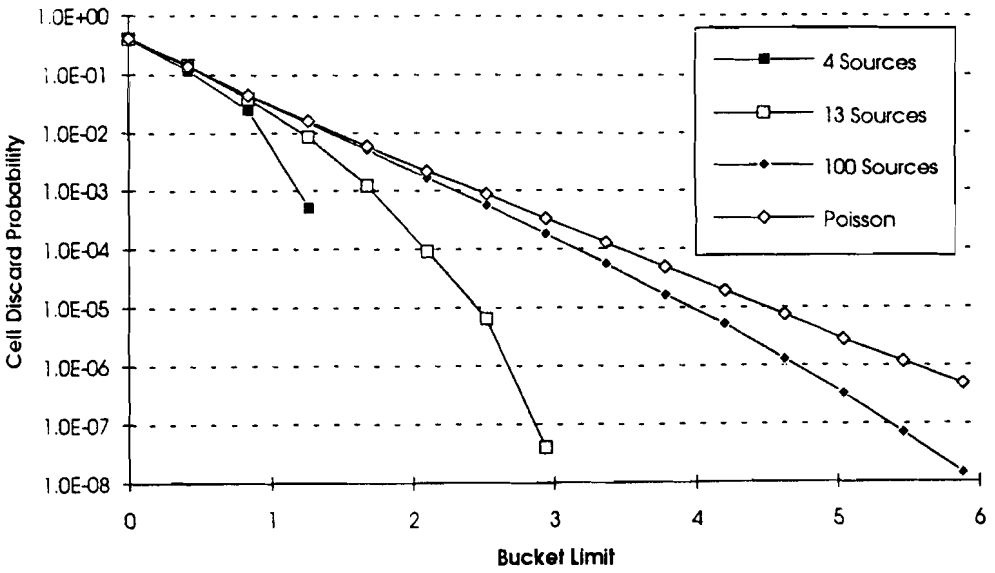


Figure 8: Cell discard probability as function of B_{lim} for various number of background sources (ρ_{tot} = 0.8, ρ_{CBR} = 50% of ρ_{tot}).

Another point of interest is when the number of background sources is varied. The number of background sources puts an upper limit to the number of cells arriving simultaneously at the multiplexer. The higher the number of background sources, the more jitter may be experienced. This can be seen in figure 8 where the influence of the number of background sources is shown on the required bucket limit to obtain a fixed cell discard probability.

Finally it has been investigated how the bucket limit is affected if the leak to splash ratio is changed. Simulations have been performed for *L/S* settings such that the policing function allowed a sustained cell rate ranging from 0% to 10% higher than the negotiated peak cell rate. It appeared that the bucket limit was not much affected in this range. Since a higher *L/S* setting directly translates into allocating more bandwidth, it is recommended to keep *L/S* as close as possible to the negotiated peak cell rate.

4.3 Comparing the results with the 'always safe' scenario

In this paragraph the bucket limit in the 'always safe' scenario (worst case dimensioning) is compared with the required bucket limit if a small cell discard probability is allowed. Table 1 shows for various loads of the CBR source the bucket limits in the 'always safe' scenario and the required bucket limit if a cell discard probability lower than 10^{-7} is allowed. It appears that in the 'always safe' case the required bucket limit grows for increasing ρ_{CBR}, irrespective of the amount of background traffic. Only the number of background sources determines the maximum jitter. From the simulations it is observed that the decreasing amount of background traffic has an influence on the required bucket limit, although the number of background sources does not change. This is because the 'always safe' scenario only takes into account that *N* cells may arrive simultaneously at the multiplexer, but not that for a decreasing amount of background traffic the *probability* that *N* cells arrive simultaneously becomes smaller.

Table 2 presents the required bucket limits if the number of background sources is varied. The bucket limit for the 'always safe' case as well as for a cell discard probability lower than 10^{-7} is shown. As expected the more background sources are present the more the maximum CDV can be and the higher the bucket limit must be set. This applies to both the 'always safe' dimensioning and the 10^{-7} quantile dimensioning. However, it appears that for increasing number of background sources the required bucket limits diverge from each other. This happens because the 'always safe' scenario only takes the number of sources into account and not the probability that *N* cells arrive simultaneously. If the background load is kept constant then for increasing number of background sources the probability that all sources transmit a cell at the same time becomes smaller.

ρ_{CBR} (in % of ρ_{tot})	B_{lim} (Worst case scenario)	B_{lim} (10^{-7} quantile)
10	8.04	> 1.2
30	24.16	3.8
50	40.2	5.5
70	56.48	5.9
90	72.68	4.5

Table 1: Bucket limits for various ρ_{CBR} (ρ_{tot} = 0.8, N = 100).

No. of Sources	B_{lim} (Worst case scenario)	B_{lim} (10^{-7} quantile)
4	1.8	> 1.3
13	5.4	2.9
100	40.2	5.5
Poisson	∞	6.7

Table 2: Bucket limits for various number of background sources (ρ_{tot} = 0.8, ρ_{CBR} is 0.4).

5 Conclusions

Inherent to the asynchronous way cells are multiplexed in ATM, individual cells may experience a variable delay component in addition to a fixed delay. This variable cell delay is known as jitter. The amount of jitter experienced depends on the load in the system, the traffic mix and the used multiplexing strategy. If cell streams are multiplexed before they arrive at the policing function, the policing function must cope with this jitter to prevent that cells from a conforming source are discarded. This paper presented a formula to derive the parameters of the leaky bucket policing function such that no cell is discarded due to jitter. The formula is only based upon the peak cell rate of the source and the maximum jitter a cell may encounter. The formula holds for the following domains:

- T_{CIA} is an integer value and L and S may adopt any positive value.
- The formula holds if T_{CIA} has a fractional value and L/S is chosen equal to or larger than $1/\lfloor T_{CIA} \rfloor$. In this case the effects of the slotted structure of ATM are not visible any more.
- If T_{CIA} is a fractional number and L/S is chosen equal to the negotiated peak cell rate $1/T_{CIA}$ then the formula is extended to account for the effects of the slotted structure of ATM. Although this case has not been formally proven, strong intuitive arguments indicate the correctness of the formula.

The only situation which is not covered is if T_{CIA} is a fractional number and $1/\lfloor T_{CIA} \rfloor < L/S < 1/T_{CIA}$.

It appears that the required bucket limit may become rather large, especially if the peak cell rate is high and/or the maximum jitter is large. If the probability that the 'worst case' situation occurs is much lower than the probability that a cell is lost due to other causes, the network operator may choose to lower the bucket limit until the cell discard probability of the policing function of a compliant cell stream is in the same order of magnitude as the objected cell loss probability. If this strategy of reducing the safety margin is pursued, less network resources have to be allocated for this connection and the resulting spare capacity may be used for other connections.

Therefore simulations have been performed to compare the parameter dimensioning in the 'always safe' scenario (worst case dimensioning) with a quantile dimensioning where a small cell discard probability is permitted. Two important conclusions can be drawn:

- The first one is if the CBR load is increased while the total load and the number of background sources is kept at a constant value. It appears that in the 'always safe' case the bucket limit increases, while from the simulation results it is observed that there is a maximum in the bucket limit curve: for

high CBR loads the required bucket limit decreases. This can be explained as follows: For a high CBR load there is less interfering traffic thus the probability that the 'worst case' situation occurs becomes smaller. The 'always safe' scenario is based on the worst case, while the quantile dimensioning also takes the probability into account.

• The second conclusion can be drawn if the number of background sources is increased, while both the CBR load and the total load remains constant. In this case the bucket limit in both the 'always safe' as in the quantile dimensioning case increases. However, the required bucket limit in the 'always safe' case grows almost linearly with the number of background sources, while the bucket limit in the quantile dimensioning seems to approach an asymptote. This is because 'always safe' case takes only the worst case into account irrespective of the probability that this worst case situation occurs.

Which strategy should be chosen (either the 'always safe' strategy or the quantile dimensioning strategy) is left to the network operator. The 'always safe' dimensioning is robust and guarantees that every cell of a compliant source is passed to the network independent of the amount of jitter experienced. But for high peak cell rates and/or a large maximum jitter the safety margin becomes rather large. The quantile dimensioning strategy polices the cell streams more tightly, thus enabling the reduction of claimed network resources, while the Quality of Service objectives are still met. However, the required safety margin to keep the cell discard probability below a desired value depends on the total load in the system, the share of the load of the source and the number of background sources. To maintain the same cell discard probability the policing parameters should be adapted for each new connection set up or release.

Literature

[1] CCITT Rec. I.371: *"Traffic Control and Congestion Control in B-ISDN"*, Geneva, June 1992

[2] E.P. Rathgeb: *"Modelling and Performance Comparison of Policing Mechanisms for ATM Networks"*, IEEE Journ. on Sel. Areas in Comm., April 1991, Vol. 9, No. 3, pp. 325-334

[3] L. Dittman: *"Flow Enforcement Algorithms for ATM Networks"*, IEEE Journ. on Sel. Areas in Comm., April 1991, Vol. 9, No. 3, pp. 343-350

[4] G. Niestegge: *"The Leaky Bucket Policing Method in the ATM Network"*, Int. Journ. of Digital and Analog Comm. Systems, April 1990, Vol. 3, pp. 187-197

[5] M. Butto et.al.: *"Effectiveness of the Leaky Bucket Policing Mechanism in ATM Networks"*, IEEE Journal on Sel. Areas in Comm., April 1991, Vol. 9, No. 3, pp. 335-342

[6] E.P. Rathgeb and T.H. Theimer: *"The Policing Function in ATM Networks"*, ISS'90, Stockholm, May 1990,Vol. V, Session A.8, Paper 4, pp. 127-130

Performance Analysis of a MAC Protocol for a Broadband Network Access Facility [1]

C. Blondia*, O. Casals, J. Garcia****

** University of Nijmegen*
Computer Science Department
Toernooiveld, 1
Nl-6525 Nijmegen, The Netherlands
tel : + 31 80 652590
fax : + 31 80 553450
e-mail : chrisb@cs.kun.nl

*** Universitat Politecnica de Catalunya*
Computer Architecture Department
c/ Sor Eulalia de Anzizu, D4
E-08034 Barcelona, Spain
tel : + 34 3 4016985
fax : + 34 3 4017055
e-mail : olga@ac.upc.es

Abstract

This paper describes a MAC protocol for a broadband network access facility, using a passive optical network (PON) with a tree structure. This protocol will be implemented in the demonstrator of the RACE project R2024 (Broadband Access Facilities). Access to the shared medium is controlled by means of a request/permit mechanism. The bandwidth allocation algorithm approximates a global FIFO strategy and enforces the peak bitrate of each Network Termination. In order to guarantee a limited reaction time, the notion of Request Block is introduced. Using an appropriate queueing model, the reaction speed of the protocol on changing traffic situations, is evaluated.

[1]This work was supported in part by the Commission of the European Communities, under project RACE R2024 (Research and Development on Advanced Communications in Europe) on Broadband Access Facilities.

1 Introduction

Big bussiness customers need and can afford a dedicated fiber access to the B-ISDN. For small bussiness and residential customers on the other hand, sharing the access resources may be much more efficient and cost-effective. Recent progress in the area of optical transmission technology (optical fibers, passive optical splitters and combiners) make a Passive Optical Network (PON) a good candidate for such an access medium. In particular when the advantages of a PON are combined with the statistical multiplexing capabilities of the Asynchronous Transfer Mode (ATM), the outcome may be a highly efficient, reliable and cost-effective broadband access facility.

We let N Network Terminations (NT) share the access medium by means of a PON with a tree structure (see Figure 1).
The passive tree is characterized by the existence of points where one fiber is connected to many by means of an optical splitter. A single fibre coming from the Line Termination (LT) can be connected to many termination points. Major cost savings can be achieved because of the resource sharing between users. Both the exchange end and the segments of the external plant (cabling, splicing, installation), are common to a number of users, so the network is economic for existing telephony and data services. Tree topologies strongly fit the geographical constraints of the local loop and the already trenching and ducting.

The access to the shared medium is arbitrated by the Medium Access Control (MAC) protocol. A tree structure lends itself very well to a protocol with centralized control. It is defined by

- The way the central control is informed about the state of the NTs (e.g. the number of cells that are waiting in the buffer of the NT)
- The way the NTs are informed when a cell can be sent (i.e. permission to access the medium)
- The way the bandwidth is distributed among the NTs.

The MAC protocol treated in this paper uses a request/permit mechanism. Each NT declares its required bandwidth by sending *requests* to the master of the protocol located in the Line Termination (LT) (at the root of the tree). Each NT is allowed

to send two types of requests. The first type of request is coupled to upstream ATM cells. These requests contain the number of cells waiting for transmission in the NT. Using such mechanism, an NT can only reveal its bandwidth needs when it is allowed to send an ATM cell. This could lead to a slow reaction to changing traffic situations. Therefore, the protocol is provided with a second type of request. The so called *Request Blocks*, contain requests originating from several NTs, not coupled to upstream ATM cells. They are issued during idle periods and allow the protocol to react fast.
The MAC protocol allocates the available bandwidth to the NTs according to a well-defined algorithm (see later), based on the received information from the requests. The NTs are informed about this obtained bandwidth by means of *permits*. Such a permit authorizes the NT to send a cell. The MAC protocol proposed in this paper is cell-based, meaning that an issued permit refers to the transmission of a single cell.

The MAC protocol determines the information flow at the entrance of the B-ISDN, and therefore has a major impact on the overal performance of the system. In particular, the shape of the traffic entering the network, a major issue in ATM networks, is highly influenced by the MAC protocol. The proposed permit distribution algorithm takes this into account in two different ways :
(i) the permit distribution algorithm approximates a global FIFO strategy in order to minimize cell delay variation and to make the mechanism fair,
(ii) cells originating from an NT are spaced (see also [3] and [7]). The peak bit rate an NT is allowed to produce is defined to be the sum of the peak bit rates of each of the connections (VCI/VPI) carried by this NT, as agreed upon at call set up. The MAC protocol enforces a minimum distance (equal to the inverse of the peak bit rate) between consecutive cells of an NT.

The MAC protocol should be suitable for all possible B-ISDN services, in particular also for constant bit rate (CBR) services. For these services, stringent requirements concerning cell delay and delay jitter have to be guaranteed. In many MAC protocols (e.g. DQDB), special measures are taken for CBR services, such as pre-arbitrated access. Hence, an important question is whether for the proposed protocol such a special treatment for CBR sources is necessary (e.g. by introducing priorities [2]). The bundle-spacer used by the permit distribution algorithm can introduce some jitter due to the fact that what is spaced is the sum of the connections and not each connection individually. The cell delay variation introduced on

an individual connection is studied in [1].

The protocol as presented in this paper will be implemented in the demonstrator system of the Broadband Access Facilities project of the European research programme RACE (project R2024).

Section 2 of this paper describes the MAC protocol in detail. In Section 3 we evaluate the reaction speed of the protocol. The results for the reaction speed are based on a busy period analysis of the global FIFO queue. Finally, in Section 5 conclusions are drawn.

2 The MAC Protocol

Consider N Network Terminations (NT) sharing the access medium by means of a Passive Optical Network (PON) with a tree structure (see Figure 1). In this section, we describe the Medium Access Control (MAC) protocol, used to arbitrate the access to the shared medium. The main function of the MAC protocol is to avoid collisions of traffic originating from the different NTs connected to the PON. Moreover it should aim at

- *Efficiency* : the overhead introduced by the MAC protocol should be low.
- *Performance* : the delay and delay variance introduced by the MAC protocol should be kept within certain bounds defined by the Quality of Service requirements, in particular for CBR traffic.
- *Fairness* : one NT should not be subject to more access delay than another.

Each NT advertises its bandwidth requirements through *requests* which are sent to the master of the protocol located in the Line Termination (LT). These requests contain information about the state of the queue in the NT (e.g. number of cells waiting for transmission). Using these requests, together with parameters agreed at call set up, the MAC protocol allocates dynamically the available upstream bandwidth to the various NTs, by means of a well defined *bandwidth allocation algorithm.* The NT's are then informed about the allocated bandwidth by means of *permits.* When an NT receives a permit, it is allowed to send one cell upstream. In what follows we give a detailed description of the various components of the protocol.

2.1 Requests

Each NT may send requests in two different ways :

- *Coupled with upstream cells.* An upstream cell originating from NT_i is preceded by a MAC information field (request) containing the NT buffer length.
- *Request Blocks* (RB) : When requests coupled with upstream cells are used, an NT can only reveal its state to the LT when it has received a permit to send an upstream cell. Clearly, this may lead to slow reaction on changing traffic situations (as a typical example consider an on/off source which switches from off to on; as a request can only be issued when an upstream cell is sent, the NT may never be able to declare its required bandwidth). To overcome this drawback, Request Blocks are introduced. A request block is built with the requests of a number of consecutive NTs. An RB can take the place of an ATM cell (i.e. ATM cell (424 bits) + physical layer preamble (17 bits) + MAC information field (7 bits)). Hence, a request block may have a length of 448 bits. This determines the number of NTs that can advertise their bandwidth requirement (i.e. queue length) per RB : each NT needs 32 bits (physical layer overhead (17 bits) + MAC information field (7 bits) + CRC (4 bits) + spare bits (4 bits)). Hence, per Request Block, 14 NTs can send their status to the LT. For implementation reasons, it is assumed that only 9 NTs can send their status per RB.

Figure 2 shows the upstream information structure, both for ordinary cells and request blocks.

2.2 Permits

The output of the Bandwidth Allocation Algorithm (see next section) consists of *permits.* We distinguish two kinds of permits.

- *Permits for ATM cells.* When according to the allocation algorithm, an NT is allowed to send a cell, then the LT issues a permit containing the address of this NT and adds it to a downstream ATM cell (downstream traffic is broadcasted, and hence there need not to be a coupling between this downstream cell and the permit). An additional bit, namely the permit class bit CL, is added to indicate that the permit addresses an ATM cell (CL=1).

- *Permits for Request Blocks.* When during a time slot, no permit is generated for an ATM cell, the LT issues a permit for a Request Block (permit class bit CL=0). In this case, the permit field contains the address of the NT (e.g. NT_i) that is the first to send a request in the request block. The next NT to send a request in the RB is NT_{i+1} up to NT_{i+8}.

Hence, the LT issues permits for RB whenever it has no permits for ATM cells to send, and therefore, the idle periods of the upstream traffic (i.e. no ATM cells are sent) are used for transmitting RBs. In that way, the spared upstream bandwidth capacity is used to increase the reaction speed of the protocol on changing traffic situations. Figure 3 shows the downstream information structure, both for Permits for ATM cells and Permits for Request Blocks.

2.3 The Bandwidth Allocation Algorithm

We describe how the bandwidth is allocated among the NTs, or equivalently, how the permits are distributed (therefore, the algorithm is also called the Permit Distribution Algorithm). The main characteristics of this algorithm are

(i) Enforce the peak bit rate per NT by spacing the cells

(ii) Approximate a global FIFO discipline (over all NTs) for fairness reasons and to minimize the delay variance

(iii) Use the available buffer capacity in the NTs to store the cells that wait for transmission (distributed buffering).

The peak bit rate of each NT is calculated as the sum of the peak bit rates of each of its connections (VCI/VPI). The MAC protocol counts the number of new cell arrivals (deduced from the request) and assigns the necessary permits to the NT. These permits are then put into a FIFO queue together with permits for the other NTs, with the additional constraint that a minimal distance between two consecutive permits for the same NT is enforced. This minimal distance is determined by the inverse of the peak bit rate.
In this way, the actual queueing takes place in the NT, while the central control of the PON maintains a *permit FIFO queue*, by which the transmission instants and the order in which the different NTs are emptied, are governed. Now we show how this distribution algorithm can be efficiently implemented.

Consider N NTs, $NT_1, ..., NT_N$, each provided with a buffer. Let t_i be the inverse of the sum of the peak bit rates of each of the connections carried by NT_i. The value of t_i will determine the minimal number of slots allowed between two consecutive cells originating from NT_i. NT_i is said to be *ready to send a cell*, whenever a cell is present in the buffer of NT_i and the number of time slots since the last transmission is at least t_i.
The central controller of the PON maintains a global permit FIFO queue and two counters for each NT, the function of which is explained in what follows.

(i) The *global permit FIFO queue* contains permits for NTs corresponding to cells which are ready to be sent. A permit for NT_i is put in the global permit FIFO queue as soon as this NT has a cell ready to be sent.

(ii) Two counters per NT are maintained, a *countdown counter* and a *request counter.* For connection i, $i = 1, ..., N$, we have that

- the countdown counter $CNTDOWN\_CNTR(i)$ is given the value t_i, at the moment the permit for NT_i is put in the virtual FIFO queue. Its value is decreased by 1, at each time slot of the outgoing line. Counting down stops as soon as $CNTDOWN\_CNTR(i) \leq 0$
- the request counter $REC\_CNTR(i)$ is increased each time a request comes in by the number of arrivals since the last request. It is decreased by 1 whenever a permit for NT_i is put in the global permit FIFO queue.

Let us now describe how the protocol itself works. As soon as a permit for NT_i is put in the global permit FIFO queue, t_i is added to the countdown counter $CNTDOWN\_CNTR(i)$ and the countdown process starts (each time slot $CNTDOWN\_CNTR(i)$ is decreased by 1). When the two conditions

$$CNTDOWN\_CNTR(i) \leq 0 \quad \text{and} \quad REQ\_CNTR(i) > 0$$

are both satisfied,then
(1) a permit for NT_i is put into the global permit FIFO queue,
(2) $REQ\_CNTR(i) := REQ\_CNTR(i) - 1$.
When this permit comes at the head of the global FIFO queue, it is sent to NT_i and the first cell in the buffer of NT_i is transmitted.
If at the end of the countdown process (i.e. as soon as $CNTDOWN\_CNTR(i) \leq 0$), $REQ\_CNTR(i) = 0$, then the countdown process goes sleeping, and when the next

request for this NT arrives, besides increasing $REQ\_CNTR(i)$, the countdown process of NT_i will be waked up again. This makes the two above conditions true, so that a permit is put immediately in the global FIFO queue. Remark that letting the countdown counter become negative and adding the real value t_i to it , is done in order to achieve higher granularity of the peak bit rates than when the countdown counter and t_i would be integers.

Figure 4 shows an example to illustrate the operation of the permit distribution algorithm. In this example,

- The LT issues a permit for NT_1.
- A request, originating from NT_2, is sent to the LT, resulting in an increase of $REQ\_CNTR(2)$.
- The conditions $REQ\_CNTR(3) \neq 0$ and $CNTDOWN\_CNTR(3) \leq 0$ are satisfied, hence a permit for NT_3 will be put in the global FIFO queue.
- A permit for NT_N has been put in the global FIFO queue, and $REQ\_CNTR(N)$ is decreased by 1 and $CNTDOWN\_CNTR(N)$ is incresed by t_N.

2.4 Robustness of the Protocol

In order to operate correctly, the NTs must declare their bandwidth requirement (using requests coupled with upstream cells or using Request Blocks), through the number of newly arrived cells since the last request. This value is added to the REQ_CNTR. However, the loss of a request may lead to the situation that cells remain in the NT buffer forever. In order to avoid this situation, the request contains the queue length of the NT. The MAC controller must be able to compute the number of new arrivals from this queue length. In order to do so, it must maintain the number of permits that have been generated for which the upstream cell has not been received yet, for one of the following reasons :

- The permit is still waiting in the global FIFO to be transmitted
- The permit is underway between the LT and NT
- The cell is underway between the NT and the LT.

¿From this number and the new queue length, the LT can compute the number of newly arrived cells since the last request.

3 Evaluation of the Reaction Speed

3.1 The Reaction Time

The Request Blocks have been introduced to guarantee fast reaction on changing traffic situations. By issueing permits for RBs during idle periods of the global FIFO queue, no bandwidth is wasted, but an NT has to wait for an idle period during which a permit for an RB is generated which contains his address. The aim of this section is to evaluate this waiting time (i.e. the reaction time). Apart from being a measure for access delay and delay jitter, the reaction time is also a measure for the required buffer capacity in the NT. Indeed, consider an on/off source which switches from the off to the on state. Between this instant and the arrival of the first permit for an RB, the source generates cells which have to be stored in the NT buffer. After having issued a request in an RB, the NT has to wait a complete round trip delay (without taking the possible waiting time in the FIFO into account) before the first cell of the burst can be actually transmitted. In what follows we assume a maximum NT-LT distance of 10 km, resulting in a round-trip delay of 100 μs.
The time an NT has to wait before the first permit for an RB, which it is allowed to use to issue a request, arrives,consists of a sequence of busy and idle periods of the global FIFO. Indeed, assume that the first cell arrives while the FIFO is busy. At the end of that busy period, a permit for an RB, adressing 9 NTs, will be issued. If our tagged NT belongs to this sequence of NTs, it may issue a request. Otherwhise, it has to wait one (or possibly more) busy period/idle period times before it can issue a request, depending on its address and on the length of the idle periods of the global FIFO. In a worst case, when each idle period consists of exactly one slot and the first permit addresses the group of NTs next to the group of the tagged NT, the tagged NT has to wait 8 slots (i.e. 8 idle periods) and 8 busy periods, before it can issue a request.
In what follows, we introduce a queueing model which enables to evaluate the reaction time.

3.2 A queueing model for the reaction time

Tag an NT which generates traffic according to an on/off source model. Assume the background traffic (i.e. traffic originating from the other NTs) to be CBR traffic (i.e. deterministic streams) having the same period (i.e. the interarrival time of cells of each source is the same) and random phases. The aim of this study is to identify

worst case situations and determine the reaction time under these conditions. In a forthcoming study we treat this problem where the background traffic is assumed to consist of VBR sources.

Consider a single server queue with a deterministic server. We let the service time be the length of one time slot (i.e. the time needed to transmit 448 bits). Hence, for a 600 Mbit/s link, the service time equals $0.75\mu s$. The input process consists of the superposition of N independent identical deterministic arrival streams having a common period, say D slots. This queueing system is known in literature as the N*D/D/1 system (see [6] for a continuous-time analysis and [4] for a discrete-time analysis).

When the load $\rho = N/D < 1$, then a busy period of this system is smaller than D. Let $p_B(l)$ denote the probability that the lenght of a busy period is l time slots, $0 < l < D$, Virtamo [6] has proved the following closed form formulas

$$p_B(l) = C_{l-1}^{N-1} \frac{D-N}{D-N+1} \frac{l^{l-2}(D-l)^{N-l-1}}{D^{N-2}}.$$

Denoting $P_I(l)$ the probability that the length of an idle period is more than l slots, then

$$P_I(l) = (1 - \frac{l}{D})^{N-2}(1 - \frac{l}{D-N+1}).$$

In what follows we use the previous formulas to evaluate

- the impact of the bit rate of the background traffic on the reaction speed (in order to identify worst case conditions), given the bit rate of the tagged source and the total load ρ;
- the impact of the bit rate of the tagged source on the required buffer capacity in the NT, given the bit rate of the background traffic and the total load ρ.
- The required buffer capacities in the NT to cope with this reaction time and the round-trip delay.

Since we are interested in buffer capacities and cell delay variations, we shall always compute the 10^{-9}-quantile of the distributions.

3.3 Impact of the bit rate of the background traffic

Consider the N*D/D/1 system described above for a 600 Mbit/s link. Assume a constant load ρ, and let the bit rate of the sources, i.e. the period D, vary. For

$D = D_x$, we compute the number of sources $N = N_x$, such that the load equals $\rho = 0.8$. The result for the length of the busy period is used to compute its 10^{-9}-quantile for different values of D_x. The following table summarizes the results. The 10^{-9}-quantile of the length of the busy period is given in slots.

bit rate	D	N	10^{-9}-quantile
64 kbit/s	9375	7500	572
560 kbit/s	1071	857	386
1 Mbit/s	600	480	298
2 Mbit/s	300	240	195
10 Mbit/s	60	48	49
60 Mbit/s	10	8	9

From this table we remark that the lower the bit rate, the longer the busy period becomes. If we assume 64 kbit/s sources, the resulting process is periodic with period 9375 time slots. During 7500 of these time slots, the server is busy, while during the remaining 1875 slots it is idle. If on the other hand we assume 10 Mbit/s sources, then during 48 slots of the 60 slots period, the server is busy. Hence, clearly the first case generates longer busy periods. In fact, the arrival process tends to a Poisson process, leading to long busy periods.

3.4 Impact of the bit rate of the tagged source

Tag an NT which at time t_0 switches from an off state to an on state. Cells arrive periodically with arrival rate A bit/s, or equivalently, with a period of D_t slots. Assume that the background traffic is modeled by N deterministic arrival streams with common period D such that

$$\rho = \frac{1}{D_t} + \frac{N}{D}.$$

We consider two cases for the background traffic :

(i) 2 Mbit/s sources : $D = 300$ slots

(ii) 64 kbit/s sources : $D = 9375$ slots.

In view of the previous section, (ii) clearly represents a worst case, which is improbable to occur. Using the formulas for the busy period, we can compute the number

of cell arrivals of the tagged source in the NT during a busy period of the global FIFO. We consider several bit rates of the tagged source. The 10^{-9}-quantile of the length of the busy period is expressed both in slots and in number of arrivals of the tagged source in the NT.

(i) Case 1 : background sources of 2 Mbit/s.

bit rate tagged source	number of background sources	busy period in slots	busy period in arrivals
150 Mbit/s	165	74	20
100 Mbit/s	190	102	18
50 Mbit/s	215	142	13
15 Mbit/s	232	176	5
10 Mbit/s	235	183	3
5 Mbit/s	238	190	2
1 Mbit/s	240	194	0.3

(ii) Case 2 : background sources of 64kbit/s.

bit rate tagged source	number of background sources	busy period in slots	busy period in arrivals
150 Mbit/s	5156	102	27
100 Mbit/s	5938	162	29
50 Mbit/s	6719	283	25
15 Mbit/s	7266	453	12
10 Mbit/s	7344	490	9
5 Mbit/s	7469	525	5
1 Mbit/s	7484	552	1

The length of the busy period is longer when the bitrate of the tagged source is low since the number of background sources in this case is high (total load is constant $\rho = 0.8$). This is not the case when the duration of a busy period is expressed in number of cell arrivals from the tagged source.

3.5 Required buffer capacity in the NTs

For the evaluation of the reaction time we consider worst case conditions. First, the tagged NT has to wait for 9 permits for an RB before it can transmit a request in an RB (this event happens with probability 1/9). Secondly, according to the results given in Section 3.2, the probability that an idle period of one slot occurs is given by

$$p_I(1) = 1 - (1 - \frac{1}{D})^{N-2}(1 - \frac{1}{D-N+1}).$$

Hence, for both the cases where the background traffic consists of a superposition of 2 Mbit/s sources or 64 kbit/s sources, the probability of having 8 consecutive idle periods of one slot (assuming that these events are independent) is not negligable. Therefore, under the assumption that the consecutive busy period/idle period pairs are independent, we let the reaction time be the duration of a sequence of 8 busy period/idle period (of one slot) pairs. The next tables give the reaction time (in number of arrivals of the tagged source) and the needed buffer capacity of the tagged NT when besides the reaction time, also a complete round-trip delay of 100 μs is taken into account.

(i) Case 1 : background sources of 2 Mbit/s.

bit rate tagged source	number of background sources	reaction time in arrivals	required buffers in cells
150 Mbit/s	165	159	194
100 Mbit/s	190	145	167
50 Mbit/s	215	101	112
15 Mbit/s	232	38	42
10 Mbit/s	235	26	29
5 Mbit/s	238	14	15
1 Mbit/s	240	3	4

(ii) Case 2 : background sources of 64kbit/s.

bit rate tagged source	number of background sources	reaction time in arrivals	required buffers in cells
150 Mbit/s	5156	218	253
100 Mbit/s	5938	230	252
50 Mbit/s	6719	200	212
15 Mbit/s	7266	96	99
10 Mbit/s	7344	69	72
5 Mbit/s	7469	37	38
1 Mbit/s	7484	8	8

Remark that the independence assumption of consecutive busy period/idle period pairs leads to a conservative estimation. Indeed, the probability of having a long second busy period decreases with increasing length of the first busy period (since the number of deterministic streams is constant).

4 Conclusions

We have described a MAC protocol for a broadband access facility using a passive optical network with a tree structure. The protocol uses a request/permit mechanism to control the access to the shared medium. The available bandwidth is allocated by means of a strategy which approximates a global FIFO queue and such that the peak bit rate is enforced. In order to guarantee a limited reaction time on changing traffic situations, Request Blocks have been introduced. They allow quick NT queue status information transfer to the LT. They are scheduled during idle periods of the global FIFO queue. A busy period analysis permits to give a first evaluation of this reaction time. Clearly more detailed models and analysis are needed.

References

[1] C. Blondia, O. Casals and J. Garcia, *Performance Evaluation of a MAC Protocol for an ATM Oriented Passive Optical Network*, Paper to be presented at "Modelling and Analysis of Telecommunication Systems" Conference, Nashville, February 1993

[2] J. García and O. Casals, *Approximate Analysis of Statistical Multiplexing of Variable Bit Rate and Periodic Sources*, Paper presented at IMSIBAC-4, San Sebastian, August 1992

[3] F. Guillemin, P. Boyer and L. Romoeuf, *The spacer-controller : architecture and first assessments*, Proc. IFIP Workshop on Broadband Communications, Estoril, Portugal, 1992

[4] F. Hübner, *Discrete-time analysis of the output process of a finite capacity ATM multiplexer with periodic input*, paper submitted to INFOCOM '93

[5] J. Roberts, J. Virtamo, *The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer*, IEEE Trans. Communications, vol.39, No. 2, Feb. 1991

[6] J. Virtamo, *Idle and busy period distribution of an infinite capacity N*D/D/1 queue*, paper submitted to Globecom '92

[7] E. Wallmeier and T. Worster, *A Cell Spacing and Policing Device for Multiple Virtual Connections on one ATM Pipe*, Proc. RACE R1022 Workshop on ATM Network Planning and Evolution, London, 1991

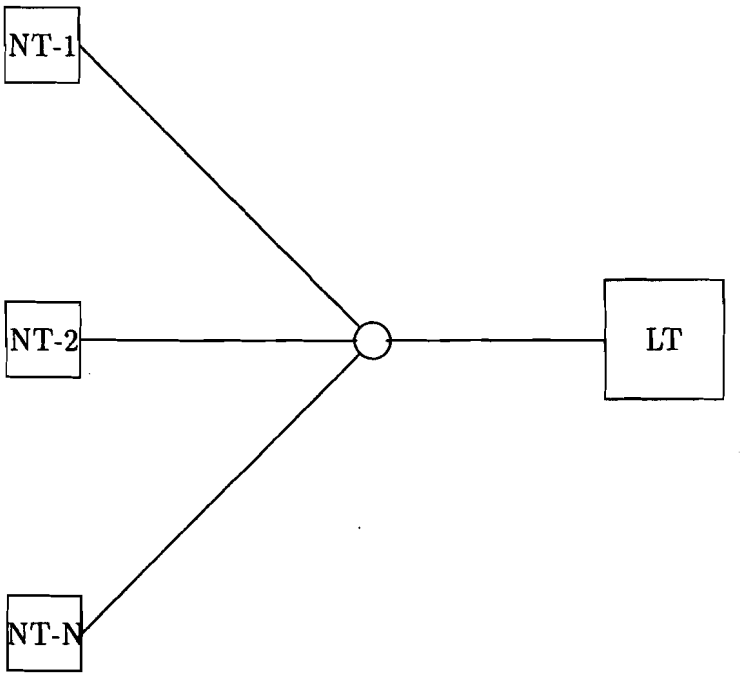


Figure 1: PON with a Tree Structure

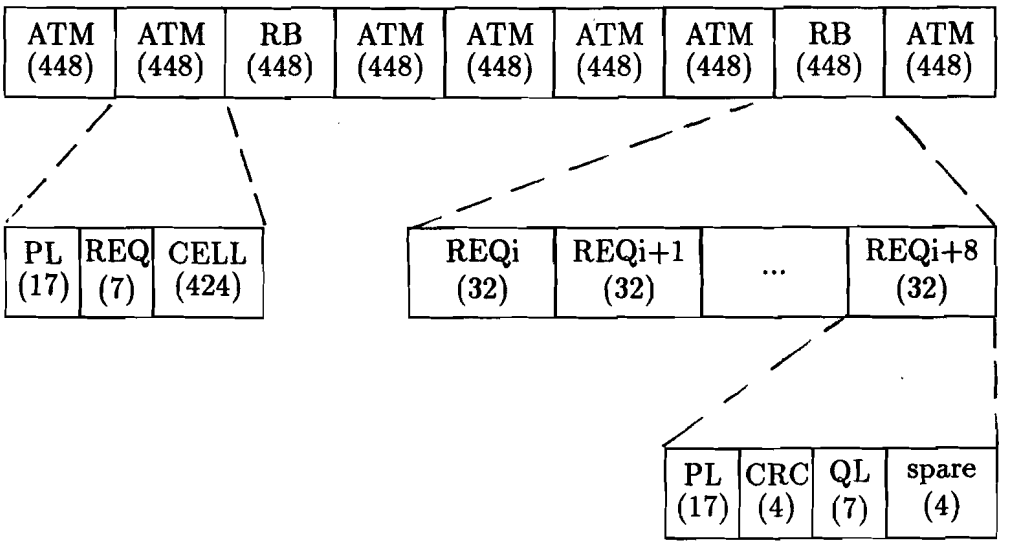


Figure 2: Upstream Information Structure

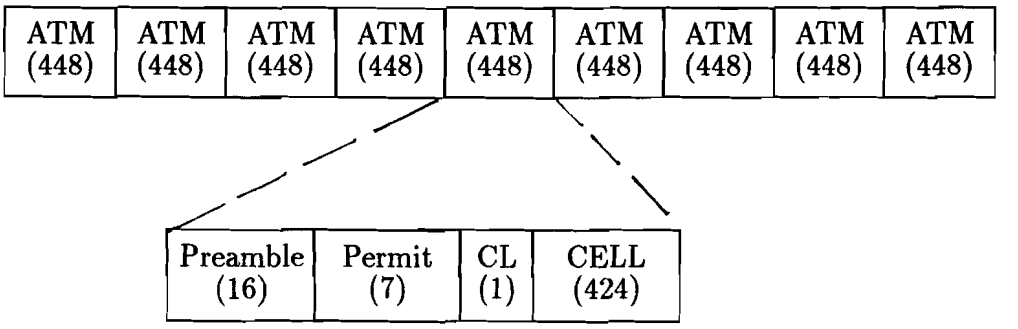


Figure 3: Downstream Information Structure

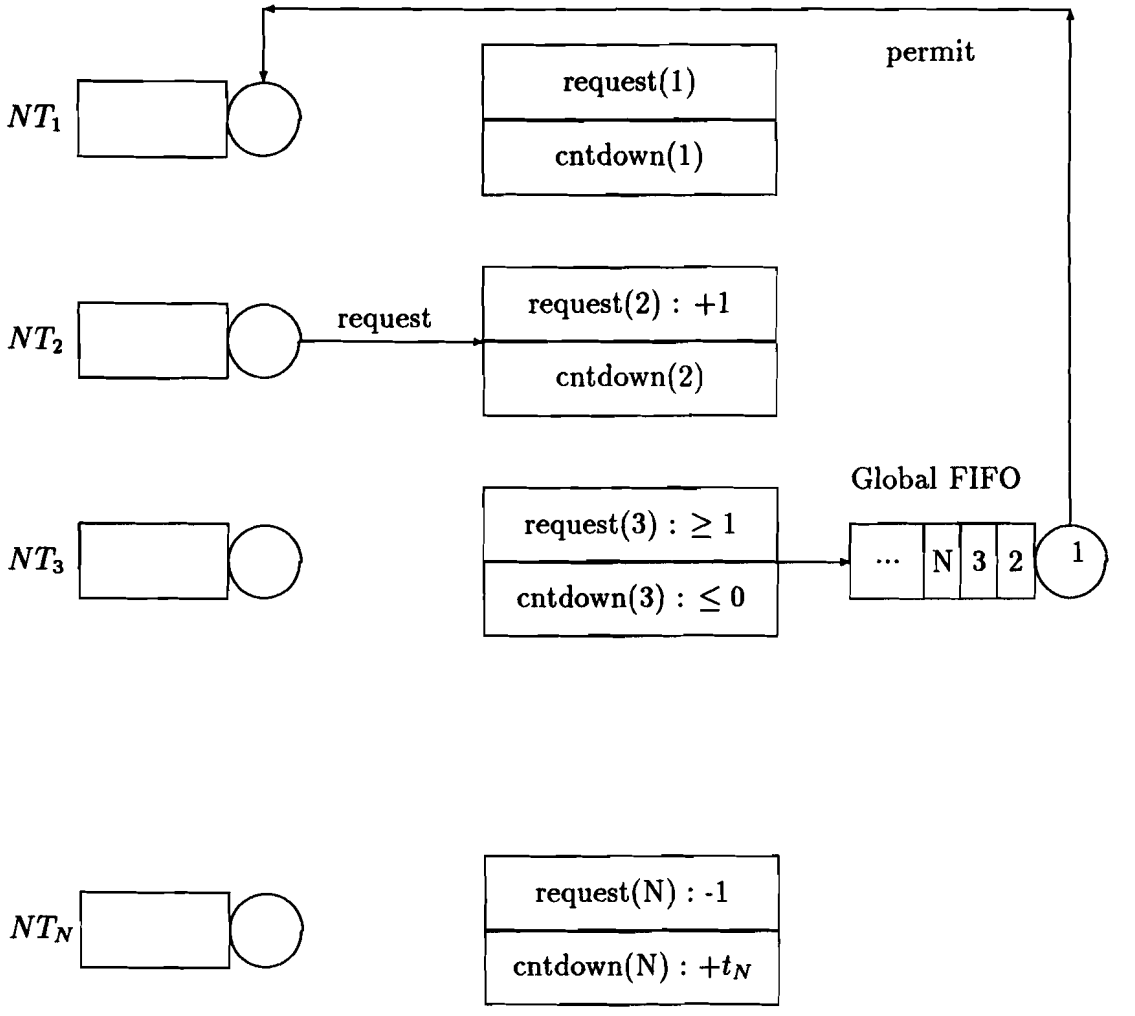


Figure 4: Operation of the MAC Protocol

Deelnemerslijst:

Ir. G.A. Awater
Technische Universiteit Delft

Dr. J.L. van den Berg
PTT Research

Dr. J.P.C. Blanc
Katholieke Universiteit Brabant

Dr. C. Blondia
Katholieke Universiteit Nijmegen

Ir. J.C.A. Boekhorst
AT&T Network Systems NL

Prof. dr. ir. O.J. Boxma
Centrum voor Wiskunde en Informatica

Ir. G.R. Braaf
PTT Research

Dr. D. Brandt
PTT Research

Ir. J.H.M. van Breemen
AT&T Network Systems NL

Prof. dr. H.L. Bruneel
Rijksuniversiteit Gent

Ir. C.P.L. Cames van Batenburg
PTT Research

Prof. dr. ir. J.W. Cohen
Centrum voor Wiskunde en Informatica

Hr. E. Diks
Technische Universiteit Eindhoven

Dr. ir. E.A. van Doorn
Universiteit Twente

Prof. dr. N.M. van Dijk
Universiteit van Amsterdam

Ir. M.J.G. Dirksen
PTT Research

Prof. ir. F. van den Dool
PTT Research

Drs. M.J.A. van Eenige
Technische Universiteit Eindhoven

Ir. G. Heijenk
Technische Universiteit Twente

Hr. P.W.J. ter Horst
Technische Universiteit Eindhoven

Drs. W.B. van den Hout
Katholieke Universiteit Brabant

Mw. E. Janssens
Technische Universiteit Delft

Dr. P. Kallenberg
AT&T Network Systems Int.

Ir. J.A. Kroeze
Katholieke Universiteit Nijmegen

Ir. K. Laevens
Rijksuniversiteit Gent

Hr. R.D. van der Mei
Katholieke Universiteit Brabant

Ir. A van Moorsel
Universiteit Twente

Prof. dr. ir. I.G.M.M. Niemegeers
Universiteit Twente

Ir. F.J.M. Panken
Katholieke Universiteit Nijmegen

Hr. G.H. Petit
Alcatel Bell Telephone

Dr. J.A.C. Resing
Technische Universiteit Eindhoven

Dr. A. Ridder
Vrije Universiteit

Ir. B.J. van Rijnsoever
Technische Universiteit Eindhoven

Ir. A.H. Roosma
PTT Research

Prof. dr. F.C. Schoute
Philips Communication Systems

Dr. ir. E. Smeitink
PTT Research

Ir. T.A. Smit
PTT Research

Hr. B. Stavrov
Technische Universiteit Delft

Ir. B. Steyaert
Rijksuniversiteit Gent

Prof. dr. H.C. Tijms
Vrije Universiteit

Dr. ir. R.J.F. de Vries
PTT Research

Dr. ir. J. van der Wal
Technische Universiteit Eindhoven

Ir. J.C. van der Wal
PTT Research

Ir. M.J.M. van Weert
Technische Universiteit Eindhoven

Mw. S. Wittevrongel
Rijksuniversiteit Gent

Hr. Yijun Xiong
Rijksuniversiteit Gent

Hr. He Yue
Rijksuniversiteit Gent