Superposition of Markovian traffic sources and frame aware buffer acceptance

UNIVERSITEIT ANTWERPEN

Faculteit Wetenschappen

Departement Wiskunde-Informatica

Superposition of Markovian traffic sources and frame aware buffer acceptance.

—

Superpositie van Markov verkeersbronnen en pakketbewuste bufferacceptatie.

Proefschrift voorgelegd tot het behalen van de graad van doctor in de Wetenschappen
aan de Universitaire Instelling Antwerpen
te verdedigen door Kathleen Spaey

Promotor: Prof. Dr. C. Blondia

Antwerpen, 2002

Overview

As its title suggests, this thesis consists of two parts, since it focuses on two separate topics that are related to the performance evaluation of telecommunication network elements: (i) the superposition of Markovian traffic sources, and (ii) frame aware buffer acceptance schemes.

A basic problem in the dimensioning and performance evaluation of telecommunication network elements is the computation of the buffer occupancy and waiting time distribution of a discrete-time single server queue, whose input consists of a superposition of processes modeling traffic streams. An important class of traffic models commonly used in traffic modeling are the Markovian arrival streams, because they allow to capture the burstiness and variability present in network traffic, and because of their analytical tractability, are the Markovian arrival streams. Since most of the time the input to network elements consists of multiple traffic streams, a characterization of the aggregation or superposition of Markovian streams is needed. In theory, this aggregation is exactly described by a new Markov model. A major problem however is the explosion of the state space of this Markov model when the number of input streams takes values that are typical for real life situations.

In the first part of the thesis, a method called *circulant matching* is proposed, which constructs, starting from statistical functions of the exact superposition, a new Markovian arrival stream with a smaller state space to replace the exact superposition. Two statistical functions of the exact input rate process that are known to influence the queueing performance are matched by this new Markov model, namely the autocorrelation sequence and the stationary distribution. The transition matrix of the Markov chain is chosen to be circulant, in order to avoid solving an inverse spectrum problem. Part I of the thesis consists of three chapters. Chapter 1 illustrates the state space explosion problem and introduces some definitions and results. The details of the circulant matching method are presented in Chapter 2. Chapter 3 discusses numerical examples and applications of the method, among which the superposition of MPEG source type models.

In the second part of the thesis frame aware buffer acceptance schemes are considered. When packet or frame based data is transported over an ATM (asynchronous transfer mode) network, these packets are segmented into cells, the small fixed length data units in which ATM by definition transports all data. A buffer acceptance scheme in a network element decides about which cells are allowed to enter its buffer, and which cells have to be dropped. Because the loss of a single cell of a frame leads to a corrupted frame that is in any case discarded at the destination, buffer acceptance schemes that are frame aware, i.e., try to accept or discard all cells of a same frame, thus improve the efficiency. Not only efficiency is an issue, but also the fairness among the effective throughputs of the different connections. So also schemes that preferentially drop frames from connections that use more bandwidth than one would call fair have been defined.

Part II of the thesis consists of four chapters. Chapter 4 gives a more exact definition of a frame. Since most non-real-time packet based data traffic in a network is TCP traffic, also a short introduction on TCP and on the two ATM service categories that are most suited to transport TCP traffic is given. Chapter 5 gives an overview of the most important frame aware buffer acceptance schemes that are proposed in the literature for use with these two service categories. A theoretical model to study the transient performance of one of the schemes that aims at discarding frames in a fair way, namely selective drop, is developed and applied in Chapter 6. This model is then slightly modified in Chapter 7 to study also the performance of fair buffer allocation, another frame aware buffer acceptance scheme that aims at fairness.

Contents

Abbreviations

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACTS	Advanced Communications Technologies and Services
ATM	Asynchronous Transfer Mode
AUU	ATM User-to-User
CAC	Connection Admission Control
CDVT	Cell Delay Variation Tolerance
CLP	Cell Loss Priority
CLR	Cell Loss Ratio
CMPP	Circulant Modulated Poisson Process
C-RED	Cell-based RED
D-BMAP	Discrete-time Batch Markovian Arrival Process
DFBA	Differential Fair Buffer Allocation
D-MAP	Discrete-time Markovian Arrival Process
D-MMPP	Discrete Markov Modulated Poisson Process
EPD	Early Packet Discard
ESPD	Early Selective Packet Discard
FB	Fair Buffering
FBA	Fair Buffer Allocation
F-GCRA	Frame-based Generic Cell Rate Algorithm
FIFO	First In First Out
FS	Fair Share
GCRA	Generic Cell Rate Algorithm
GFR	Guaranteed Frame Rate
GFS	Global FIFO Scheduling
GOP	Group of Pictures
IDC	Index of Dispersion for Counts
IDI	Index of Dispersion for Intervals
IP	Internet Protocol
ITU-T	International Telecommunications Union-Telecommunication Standardization Sector
LAN	Local Area Network
LLC/SNAP	Logical Link Control/Subnetwork Access Protocol

MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MDCR	Minimum Desired Cell Rate
MFS	Maximum Frame Size
MMBP	Markov Modulated Bernouilli Process
MMPP	Markov Modulated Poisson Process
MPEG	Moving Picture Experts Group
4 MSS	Maximum Segment Size
nrt-VBR	non-real-time Variable Bit Rate
PCR	Peak Cell Rate
PLQF	Probabilistic Longest Queue First
PPD	Partial Packet Discard
P-RED	Packet-based RED
PTI	Payload Type Indicator
QoS	Quality of Service
RED	Random Early Detection
RR	Round Robin
RTT	Round Trip Time
SAR	Segmentation and Reassembly
SD	Selective Drop
SDU	Service Data Unit
SMAQ	Statistical Match and Queueing Tool
TBA	Token-based Buffer Allocation
TCP	Transport Control Protocol
TD	Tail Drop
UBR	Unspecified Bit Rate
UPC	Usage Parameter Control
VC	Virtual Channel
WBA	Weighted Buffer Allocation
WFQ	Weighted Fair Queueing

Part I

Circulant matching of the superposition of D-BMAPs

Chapter 1

Introduction

A basic problem in the dimensioning and performance evaluation of telecommunication network elements is the computation of the buffer occupancy and waiting time distribution of a single server queue, whose input consists of a superposition of processes modeling traffic streams. Several main classes of traffic models commonly used in traffic modeling exist, e.g., renewal, Markov based, fluid, autoregressive, self-similar etc. A nice survey of these classes can be found in [51]. In this first part of the thesis, we start from the assumption that a traffic stream is modeled by a D-BMAP (discrete-time batch Markovian arrival process), which is a quite general discrete-time Markov model that includes many well-known source models as special cases [9, 10].

Because the input to network elements most of the time consists of multiple traffic streams, a representation of the aggregation or superposition of traffic streams modeled by D-BMAPs is needed. In theory, this aggregation is exactly described by a new D-BMAP. A major problem however is the explosion of the state space of this new D-BMAP when the number of input streams takes values that are typical for real life situations.

In the first part of the thesis, a method called *circulant matching* is proposed, which constructs another D-BMAP with a smaller state space to replace the exact superposition. This D-BMAP matches two important statistical functions of the exact input rate process, namely the autocorrelation sequence (characterized in the frequency domain by means of the power spectrum) and the stationary cumulative distribution. The transition matrix of this D-BMAP is chosen to be circulant, in order to avoid solving an inverse spectrum problem.

The circulant matching method for D-BMAPs is based on an approach proposed in [46], which is a component of a measurement-based tool developed by San-qi Li et al. for the integration of traffic measurements and queueing analysis. The tool [72, 73], which is called SMAQ (statistical match and queueing tool), has the ambition to model an arbitrary traffic stream from which the statistics are obtained from measurements. The following three components form the basis of the tool: (1) measurement of the power spectrum $P(\omega)$ and the stationary cumulative distribution $F(x)$ of the rate process of a traffic stream using signal processing techniques, (2) construction of a CMPP (circulant modulated Poisson process) which statistically matches $P(\omega)$ and $F(x)$, (3) analysis of queueing problems with the constructed CMPP as input. The nature of the statistics of the traffic stream that should be measured in step (1), and matched in step (2), is studied in the paper [71]. In this paper the influence of first, second, third and fourth order statistics on queueing performance is investigated through the stationary cumulative distribution, power spectrum, bispectrum and trispectrum. The conclusion is that the power spectrum, especially that in the low frequency band, has the most dominant impact. Interesting in this paper is that the vehicle used to explore the nature of queue response to second and higher order input statistics is the MMPP (Markov modulated Poisson process), which is a continuous time Markov model. First it is shown that the eigenstructure of the transition rate matrix of an MMPP captures the input spectral functions, so by tuning the eigenstructure of the MMPP, one can change the input spectral functions of the MMPP. However, finding the spectral functions of an MMPP is easy, but constructing an MMPP from desired spectral functions is difficult, if at all possible, since calculating the eigenvalues of a matrix is easy, but constructing a matrix with a desired set of eigenvalues involves a generally very difficult to solve, so-called inverse spectrum problem. To circumvent this problem, a special class of MMPPs, called circulant modulated Poisson processes (CMPPs), is considered, for which the eigenvalues and eigenvectors of the transition rate matrix are known in closed form. The queue response to input spectral functions as contributed by a single predefined eigenvalue is investigated by constructing a CMPP whose transition rate matrix has that value as eigenvalue. To investigate the effect of spectral functions as contributed by multiple predefined eigenvalues, an independent CMPP is constructed for each eigenvalue, and then the superposition of these CMPPs is considered. Because the dimension of this superposition is the product of the individual dimensions of the multiple CMPPs, this approach is limited by the state space explosion and by the high computation cost of the queueing analysis when this superposition is used as input. In [46], the construction of a single circulant with different predefined eigenvalues is considered. Combining this construction with the observation made in [45] that the power spectrum and the input rate distribution of the superposition of independent MMPPs can be obtained from the power spectra and input rate distributions of the individual MMPPs in the superposition, provides for the fact that the technique used in part (2) of the SMAQ tool could also be used to construct a CMPP which matches the power spectrum and the stationary cumulative distribution of the superposition of MMPPs. And this of course opens perspectives for circumventing the state space explosion problem that occurs when the superposition of multiple independent D-BMAPs is considered. When for a D-BMAP the eigenstructure of its transition matrix also would capture the power spectrum (which it does), a similar technique as that in part (2) of the SMAQ tool could be applied.

The details of the circulant matching method to construct a circulant D-BMAP to replace the superposition of D-BMAPs are presented in Chapter 2. Remark that an important difference with the method of the SMAQ tool is that this tool works in continuous time, while a D-BMAP is a discrete-time model. So the Markov process underlying the CMPP has a circulant transition *rate* matrix, while for a circulant D-BMAP this should be a circulant transition *probability* matrix. Since traffic consists of the arrival of discrete entities (packets, cells etc.) at discrete time instants, it becomes natural however to use discrete-time models such as the D-BMAP. Another difference between the continuous-time MMPP and the discrete-time D-BMAP is that D-BMAPs can generate bulk arrivals, while with MMPPs this is not the case. The motivation in [72] for using a circulant MMPP, and not a more generic process, called versatile Markovian process, which can capture bulk arrivals, is that no matching techniques are available for the construction of such processes, and also that their queueing analysis can become more difficult. The D-BMAP however is the discrete-time version of the versatile Markovian process [9], and efficient algorithms to solve queues with a D-BMAP as input exist. Since much of the traffic streams in networks are highly periodic, periodicity is also often noticed in the transition matrices of D-BMAPs that model these traffic streams. So we added the notion of periodicity to the circulant matching method, such that the circulant D-BMAP that replaces the superposition has the same period as the exact superposition would have. Examples of periodic D-BMAP sources are the MPEG model that is used in Chapter 3, and the description of the traffic profile of a tagged constant bit rate source after it has been jittered by background traffic [11]. Other works studying and capturing periodicities are for example [61, 62].

Chapter 3 discusses numerical examples and applications of the circulant matching method. First a numerical example is worked out in illustration to the theoretical description of the method in Chapter 2. Then applications are considered where the constructed circulant is used as input to a queueing system. Focus is on applications that allow us to validate the obtained results, because either the exact results can also be calculated, or because similar results obtained experimentally are available.

In the remaining part of this chapter, some definitions and results that are used in the following chapters are summarized. Section 1.1 deals with the eigenstructure of finite-state stationary Markov chains. In Section 1.2 the D-BMAP together with some of its properties is introduced, the state space explosion problem associated with the superposition of D-BMAPs is illustrated, and the D-BMAP/D/1/K queue is described.

1.1 An algebraic approach to finite-state stationary Markov chains

The eigenvalues of the transition matrix $\mathbf{P}$ of a discrete-time finite-state stationary Markov chain provide a good deal of information about the periodicity and the number of ergodic classes associated with the Markov chain. The purpose of this section is to provide a list of definitions and results that are used in the following chapter, and since the terminology used in different references about Markov chains is not always uniform, to introduce the terminology used in this thesis. Books on which this section is based are [18, 44, 48].

1.1.1 Classification of states

Given a discrete-time finite-state stationary Markov chain with transition matrix $\mathbf{P}$, a state j is *accessible* from state i if there is a sequence of transitions from i to j that has nonzero probability. The probability of being in state j after the k-th transition, given that the initial state was i, is given by $(\mathbf{P}^k)_{ij}$. Two states i and j *communicate* if they are accessible to each other. Note that each state communicates with itself since $\mathbf{P}^0 = \mathbf{I}$.

Two states are said to belong to the same *ergodic class* if they communicate with each other. If the state space by itself forms an ergodic class (i.e., all states communicate with each other), the Markov chain is called *irreducible.* Otherwise it is called *reducible.* Also the corresponding transition matrix is said to be irreducible or reducible. Each Markov chain has at least one ergodic class, but it is possible that several ergodic classes exist. States that do not belong to any ergodic class are called *transient.* These definitions imply that once an ergodic class is entered, the chain remains within this class for every subsequent transition. Thus, if the chain starts within an ergodic class, it stays within that class. If it starts at a transient state, it will enter an ergodic class after a number of transitions and then remain there.

By relabeling the states of the Markov chain, $\mathbf{P}$ can always be written as

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}^{(1)} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{(2)} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{P}^{(m)} & \mathbf{0} \\ \mathbf{R}^{(1)} & \mathbf{R}^{(2)} & \dots & \mathbf{R}^{(m)} & \mathbf{Q} \end{pmatrix}, \tag{1.1}$$

in which each $\mathbf{P}^{(i)}$ is square, stochastic and irreducible. It represents the transitions within the i-th ergodic class. The matrix $\mathbf{Q}$ corresponds to transitions among the transient states. The matrices $\mathbf{R}^{(i)}$ give the transitions from the transient states into the i-th ergodic class.

Among irreducible Markov chains, two types are distinguished: periodic and aperiodic ones. The period of a Markov chain is concerned with the times at which the chain might return to a state from which it started. If this can only happen at times that are multiples of d, where d is the largest integer with this property, the Markov chain is said to have *period d.* Also the corresponding transition matrix is said to be periodic with period d. An *aperiodic* Markov chain is a Markov chain of period one.

For an irreducible Markov chain of period d, there always exists a relabeling of the states which puts its transition matrix in the form

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{A}^{(0)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^{(1)} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}^{(d-2)} \\ \mathbf{A}^{(d-1)} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{pmatrix}, \tag{1.2}$$

where the diagonal blocks are square, but $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(d-1)}$ probably are not. Two states are said to belong to the same *periodic class* if they both correspond to the same diagonal block.

1.1.2 Stationary distributions

If a Markov chain has only one ergodic class, there exists a unique vector $\boldsymbol{\pi}$ of nonnegative elements summing to one such that $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$. The vector $\boldsymbol{\pi}$ is called the *stationary distribution* of the Markov chain, and its elements π_i equal the long-run proportion of time that the chain is in state i. If i is a transient state, $\pi_i = 0$. Otherwise, $\pi_i > 0$.

If $\lim_{n\to\infty}(\mathbf{P}^n)_{ij} = \pi_j$ for all i, then the stationary distribution $\boldsymbol{\pi}$ is also called the *steady state distribution*. Thus, if $\mathbf{P}$ has a steady state distribution, the probability of being in state j as $n \to \infty$ is a constant independent of the initial state. If $\mathbf{P}$ is aperiodic, there exists a steady state distribution. If $\mathbf{P}$ is periodic, $(\mathbf{P}^n)_{ij}$ does not converge for $n \to \infty$, but appropriate subsequences do: if the chain is periodic with period d, then for each pair i, j of states there is an integer r, $0 \leq r < d$, such that $(\mathbf{P}^n)_{ij} = 0$ unless $n = md + r$, for some nonnegative integer m, and $\lim_{m\to\infty}(\mathbf{P}^{md+r})_{ij} = d\pi_j$.

1.1.3 Eigenstructure of a transition matrix

If $\mathbf{P}$ is the transition matrix of a Markov chain, the composition of its set of eigenvalues is directly related to the periodicity and the number of ergodic classes of the Markov chain:

- If $\mathbf{P}$ is in the form (1.1), then the eigenvalues of $\mathbf{P}$ are the eigenvalues of $\mathbf{P}^{(1)}, \ldots \mathbf{P}^{(m)}$ and $\mathbf{Q}$ put together. None of the eigenvalues of $\mathbf{P}$ has a modulus that is larger than one.

- $\mathbf{P}$ has always 1 as eigenvalue. The multiplicity of this eigenvalue 1 is equal to the number of ergodic classes of the chain.

- If $\mathbf{P}$ is irreducible and periodic with period d, $\mathbf{P}$ has exactly d eigenvalues with modulus 1:

  $$\lambda_0 = 1, \quad \lambda_1 = c, \quad \ldots \quad , \lambda_{d-1} = c^{d-1}, \quad \text{where } c = e^{\frac{2\pi i}{d}}. \tag{1.3}$$

  Left and right eigenvectors $\mathbf{h}_j$ and $\mathbf{g}_j$ corresponding to λ_j, chosen such that $\mathbf{h}_j\mathbf{g}_j = 1$, are given by

  $$\mathbf{h}_j = \begin{pmatrix}\boldsymbol{\pi}_0 & c^{-j}\boldsymbol{\pi}_1 & c^{-2j}\boldsymbol{\pi}_2 & \ldots & c^{-(d-1)j}\boldsymbol{\pi}_{d-1}\end{pmatrix}, \tag{1.4}$$

  where $\boldsymbol{\pi} = \begin{pmatrix}\boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \boldsymbol{\pi}_2 & \ldots & \boldsymbol{\pi}_{d-1}\end{pmatrix}$ is the stationary distribution of $\mathbf{P}$, and

  $$\mathbf{g}_j{}^T = \begin{pmatrix}\mathbf{e}^T & c^j\mathbf{e}^T & c^{2j}\mathbf{e}^T & \ldots & c^{(d-1)j}\mathbf{e}^T\end{pmatrix}. \tag{1.5}$$

The vectors $\boldsymbol{\pi}, \mathbf{h}_j$ and $\mathbf{g}_j$ are partitioned according to the periodic structure of $\mathbf{P}$ (see (1.2)), and $\mathbf{e}$ denotes a column vector of 1's of appropriate length. This property is easily proven by calculating $\mathbf{h}_j\mathbf{P}$ and $\lambda_j\mathbf{h}_j$ (resp. $\mathbf{P}\mathbf{g}_j$ and $\lambda_j\mathbf{g}_j$), while using that $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$ and $\mathbf{P}\mathbf{e} = \mathbf{e}$.

- If $\mathbf{P}$ is irreducible and periodic with period d, the set of its eigenvalues, regarded as a system of points in the complex plane, goes over into itself under a rotation of the plane by the angle $2\pi/d$.

1.2 D-BMAP: discrete-time batch Markovian arrival process

A discrete-time batch Markovian arrival process (D-BMAP) is a quite general traffic model for discrete-time Markov sources [9]. Examples of the use of D-BMAPs as traffic model for realistic sources can be found in e.g., [10, 11, 31]. In [10], a D-BMAP is used as an approximate model for the superposition of video sources. The D-BMAP defined in [11] describes the profile of a tagged ATM connection with renewal interarrival distribution after it shared a multiplexer with background traffic. A method to recursively estimate the parameters of a D-BMAP is proposed in [31] and applied to real LAN traffic. Its simple and transparent notation and the fact that it includes many well-known source models as special cases makes the D-BMAP an attractive model for discrete-time arrival processes.

1.2.1 Definition

Consider a discrete-time stationary Markov chain with transition matrix $\mathbf{D}$, and suppose that at time n this chain is in some state i, $0 \leq i \leq N-1$. At the next time instant $n+1$, a transition to another or possibly the same state is made and a batch arrival may or may not occur. The matrix $\mathbf{D}_0$ governs transitions that correspond to no arrivals, while the matrices $\mathbf{D}_k$, $k \geq 1$, govern transitions that correspond to arrivals of batches of size k. So a D-BMAP is characterized by a sequence of matrices $(\mathbf{D}_k)_{k\geq 0}$, with

$$\mathbf{D} = \sum_{k=0}^{\infty} \mathbf{D}_k. \tag{1.6}$$

In the sequel a D-BMAP is most of the time denoted by the sequence of matrices $(\mathbf{D}_k)_{k\geq 0}$. It is then implicitely assumed that the matrix denoted by the same symbol, but without the subscript k, denotes the transition matrix of the D-BMAP, which is related to the matrices $(\mathbf{D}_k)_{k\geq 0}$ by the expression above.

If $\boldsymbol{\pi}$ denotes the stationary distribution of $\mathbf{D}$, then the mean arrival rate of the process is given by

$$\lambda = \boldsymbol{\pi} \sum_{k=1}^{\infty} k\mathbf{D}_k \, \mathbf{e}, \tag{1.7}$$

where $\mathbf{e}$ denotes a column vector of 1's.

More details and properties about D-BMAPs can be found in [9]. A special case of the D-BMAP is the D-MAP (discrete-time Markovian arrival process), which is a D-BMAP that is completely characterized by $\mathbf{D}_0$ and $\mathbf{D}_1$, i.e., all arrivals have a batch size of 1. Results concerning D-MAPs are given in [12].

In the sequel, when a D-BMAP is said to be irreducible/reducible or aperiodic/periodic, it is meant that the transition matrix of the underlying Markov chain is irreducible/reducible or aperiodic/periodic. The same applies when mentioning the stationary distribution, the eigenvalues or the eigenvectors of a D-BMAP.

1.2.2 Correlation structure of a D-BMAP

The variability in the arrivals of a traffic stream is an essential characteristic that impacts the buffer occupation when traffic streams are multiplexed. Mathematically, this variability has been characterized by different expressions such as the autocorrelation, the autocovariance, the index of dispersion for counts (IDC), the index of dispersion for intervals (IDI) etc. [76, 39, 94].

In the next chapter, the autocorrelation of the input rate of a D-BMAP is derived. The autocorrelation $R[n]$ is a measure of the rate of change of a stationary stochastic process $(X_k)_k$ [68, p.359]:

$$\forall \varepsilon > 0 : P\left[|X_{k+n} - X_k| \geq \varepsilon\right] \leq \frac{2\left(R[0] - R[n]\right)}{\varepsilon^2}. \tag{1.8}$$

This equation states that if $R[0] - R[n]$ is small, that is $R[n]$ drops slowly, then the probability of a large change of $(X_k)_k$ in n slots is small.

When X_k represents the number of arrivals generated by a D-BMAP at time instant k, then the autocorrelation of $(X_k)_k$ is derived in [9]:

$$\begin{aligned} R[0] &= E\left[X_k^2\right] = \boldsymbol{\pi} \sum_{i=1}^{\infty} i^2 \mathbf{D}_i \, \mathbf{e}, \\ \underset{n>0}{R[n]} &= E\left[X_k X_{k+n}\right] = \boldsymbol{\pi} \left(\sum_{i=1}^{\infty} i\mathbf{D}_i \right) \mathbf{D}^{n-1} \left(\sum_{i=1}^{\infty} i\mathbf{D}_i \right) \mathbf{e}. \end{aligned} \tag{1.9}$$

1.2.3 Superposition of D-BMAPs

Since the input to a network element does not consist of traffic of a single source, but of a multiple of sources, a description of the aggregation of traffic streams modeled by D-BMAPs is needed. Consider M independent D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$, $1 \leq i \leq M$. Their superposition can again be described by a D-BMAP, denoted by $(\mathbf{D}_k)_{k\geq 0}$, where

$$\begin{aligned}
\mathbf{D} &= \bigotimes_{i=1}^{M} \mathbf{D}^{(i)}, \\
\mathbf{D}_0 &= \bigotimes_{i=1}^{M} \mathbf{D}_0^{(i)}, \\
\mathbf{D}_1 &= \mathbf{D}_1^{(1)} \otimes \left(\bigotimes_{i=2}^{M} \mathbf{D}_0^{(i)} \right) + \cdots + \left(\bigotimes_{i=1}^{M-1} \mathbf{D}_0^{(i)} \right) \otimes \mathbf{D}_1^{(M)}, \\
&\vdots
\end{aligned} \tag{1.10}$$

In the following, we refer to this description of the superposition as the 'exact superposition'. The construction of this superposition involves the *Kronecker product* $\otimes$, which is defined as follows: consider a matrix $\mathbf{A} = (a_{ij})$ of dimension $m \times n$ and a matrix $\mathbf{B} = (b_{ij})$ of dimension $r \times s$; the Kronecker product of the two matrices is defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{pmatrix}. \tag{1.11}$$

In [37], numerous properties of this product are given. What is important here is that $\mathbf{A} \otimes \mathbf{B}$ is seen to be a matrix of dimension $mr \times ns$.

1.2.4 The D-BMAP/D/1/K queue

The D-BMAP/D/1/K queue is a single server system with capacity K. The deterministic service time of a customer equals one time unit, and the input to the queue is described by a D-BMAP $(\mathbf{D}_k)_{k\geq 0}$.

When denoting by $L(n)$ the number of customers in the system at time n, and by $J(n)$ the phase of the arrival process at time n, $\{(L(n), J(n)), n \geq 0\}$ is a two dimensional Markov chain. When N is the dimension of the input D-BMAP, the state space of this Markov chain is $\{(l,j) | 0 \leq l \leq K, 0 \leq j \leq N-1\}$, and its transition matrix of size $(K+1)N$ is given by

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{D}_2 & \dots & \mathbf{D}_{K-1} & \sum_{k=K}^{\infty} \mathbf{D}_k \\ \mathbf{D}_0 & \mathbf{D}_1 & \mathbf{D}_2 & \dots & \mathbf{D}_{K-1} & \sum_{k=K}^{\infty} \mathbf{D}_k \\ \mathbf{0} & \mathbf{D}_0 & \mathbf{D}_1 & \dots & \mathbf{D}_{K-2} & \sum_{k=K-1}^{\infty} \mathbf{D}_k \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{D}_0 & \sum_{k=1}^{\infty} \mathbf{D}_k \end{pmatrix}. \tag{1.12}$$

If the stationary distribution of $\mathbf{Q}$ is denoted by $\mathbf{x}$, where $\mathbf{x} = \begin{pmatrix}\mathbf{x}_0 & \dots & \mathbf{x}_K\end{pmatrix}$ with $\mathbf{x}_i = \begin{pmatrix}x_{i,0} & \dots & x_{i,N-1}\end{pmatrix}$, then the elements $x_{i,j}$ of $\mathbf{x}$ represent the stationary joint probability that there are i customers in the system and that the phase of the arrival process is in state j.

The probability that an arriving customer gets lost due to buffer overflow is derived in [10], and is given by

$$P = \frac{\sum_{l=K+1}^{\infty}(l-K)\mathbf{x}_0\mathbf{D}_l\mathbf{e} + \sum_{k=1}^{K}\sum_{l=2}^{\infty}\max\{l-K+k-1,0\}\mathbf{x}_k\mathbf{D}_l\mathbf{e}}{\boldsymbol{\pi}\sum_{k=1}^{\infty}k\mathbf{D}_k\,\mathbf{e}}. \tag{1.13}$$

1.2.5 Motivations for avoiding the exact superposition of D-BMAPs

An exact description of the superposition of M independent traffic streams modeled by D-BMAPs is given in (1.10). Since this description involves the Kronecker product, it has as disadvantage that it leads to a state space explosion: the dimension of the resulting D-BMAP equals the product of the dimensions of all individual D-BMAPs involved in the superposition. This implies that when M takes values that are typical for real life situations, the exact superposition is not usable. Let us illustrate this with an example: consider 10 sources, each modeled by a D-BMAP of four states whose transition matrix contains no zeros. Then the D-BMAP describing the exact superposition has 2^{20} states, which corresponds to 2^{40} real numbers to describe only its transition matrix. To store such a matrix in a program as for example MATLAB, which uses double precision floating points (i.e., 8 bytes per floating point number), 8192 Gigabytes are needed, which clearly is not realistic and motivates the replacement of the exact superposition by another D-BMAP with a smaller state space.

A second motivation to keep the state space of a D-BMAP small, and thus to avoid the exact superposition of D-BMAPs, is that they generally are used as input to a queueing system, such as for example the single server queueing system with capacity K and deterministic service time as described in Section 1.2.4. To compute performance measures like the buffer occupancy and loss probability of such queueing system, the stationary distribution vector $\mathbf{x}$ corresponding to the matrix $\mathbf{Q}$ given in equation (1.12) is needed. Remark that $\mathbf{Q}$ is a square matrix of size $(K+1)N$, where N is the dimension of the input D-BMAP.

Fortunately, there exist efficient algorithms that exploit the structure that is present in the matrix $\mathbf{Q}$ to calculate $\mathbf{x}$ without needing to store the whole matrix $\mathbf{Q}$. Due to its special structure, the matrix $\mathbf{Q}$ belongs to the class of finite M/G/1-type transition matrices. There exists a huge amount of literature concerning the numerical solution (i.e., computing their stationary distribution) of M/G/1-type transition matrices, both for infinite and for finite buffer systems (see [82, 64, 75, 8] for infinite buffer systems, and [9, 98, 66, 56] for finite buffer systems, and the references therein). The algorithm that is used in this thesis to solve the finite D-BMAP/D/1/K queueing system is described in [9], and is based on a result in [38] extended to block partitioned matrices. The algorithm requires to store $O(K)$ blocks of the size of the input D-BMAP, which is the same as most other algorithms. So also here it remains important to keep the state space of the input D-BMAP small.

Chapter 2

Circulant matching of the superposition of D-BMAPs

This chapter describes the circulant matching method in detail. A summary of this chapter was presented in [89]. The purpose of the circulant matching method is to construct a circulant D-BMAP to replace the superposition of independent D-BMAPs. This circulant D-BMAP matches the power spectrum and the stationary cumulative distribution of the input rate process of the exact superposition. In the first three sections of this chapter, the foundation for the description of the circulant matching method in Section 2.4 is laid. In Section 2.1, expressions for the autocorrelation sequence, power spectrum and stationary cumulative distribution of a single D-BMAP are derived. For the autocorrelation sequence and the power spectrum, these expressions are written as a function of the eigenvalues and eigenvectors of the D-BMAP. The circulant D-BMAP is introduced in Section 2.2, and based on the results of Section 2.1, formulas for its autocorrelation sequence, power spectrum and stationary cumulative distribution are obtained. Also the condition for a circulant to be irreducible and some properties about periodic circulants are proven in this section. Section 2.3 gives expressions for the power spectrum and the stationary cumulative distribution of the exact superposition of M independent D-BMAPs. These expressions can be calculated without explicitely constructing the exact superposition. In Section 2.4 the circulant matching method itself is described, while conclusions and some related work are given in Section 2.5.

2.1 Input rate process of a D-BMAP

Consider a N-state irreducible D-BMAP $(\mathbf{D}_k)_{k\geq 0}$. Denote by $\boldsymbol{\pi}$ its stationary distribution, and by $\mathbf{e}$ a column vector of 1's. The input rate process $(\Gamma(k))_k$ of the D-BMAP is then defined as follows: $\Gamma(k) = \Gamma_i$ when the D-BMAP is in state i at the k-th time slot, where

$$\Gamma_i = \sum_{j=0}^{N-1} \left(\sum_{k=1}^{\infty} k\mathbf{D}_k \right)_{ij} = \left(\sum_{k=1}^{\infty} k\mathbf{D}_k \mathbf{e} \right)_i . \tag{2.1}$$

The input rate in a slot is thus a random variable Γ, which takes values $\Gamma_0, \ldots, \Gamma_{N-1}$ with probabilities $\pi_0, \ldots, \pi_{N-1}$, where Γ_i is the expected number of arrivals in a slot when the D-BMAP is in state i. The mean input rate is given by

$$E\left[\Gamma(k)\right] = \sum_{i=0}^{N-1} \pi_i \Gamma_i = \boldsymbol{\pi} \sum_{k=1}^{\infty} k\mathbf{D}_k\, \mathbf{e}. \tag{2.2}$$

Remark that the mean input rate equals the mean arrival rate (cfr. equation (1.7)).

2.1.1 Correlation structure

By definition, the autocorrelation sequence $R[n]$ of the input rate process $(\Gamma(k))_k$ is given by

$$R[n] = E\left[\Gamma(k)\Gamma(k+n)\right]. \tag{2.3}$$

For $n = 0$, this gives using (2.1):

$$\begin{aligned} R[0] = E\left[(\Gamma(k))^2\right] &= \sum_{i=0}^{N-1} (\Gamma_i)^2 \pi_i = \sum_{i=0}^{N-1} \pi_i \left[\left(\sum_{k=1}^{\infty} k\mathbf{D}_k \mathbf{e} \right)_i \right]^2 \\ &= \boldsymbol{\pi} \left[\left(\sum_{k=1}^{\infty} k\mathbf{D}_k \mathbf{e} \right) \odot \left(\sum_{k=1}^{\infty} k\mathbf{D}_k \mathbf{e} \right) \right] = \boldsymbol{\pi}(\boldsymbol{\Gamma} \odot \boldsymbol{\Gamma}), \end{aligned} \tag{2.4}$$

where $\boldsymbol{\Gamma} = \begin{pmatrix} \Gamma_0 & \ldots & \Gamma_{N-1} \end{pmatrix}^T$, and where $\odot$ denotes the element-by-element product of two vectors.

For $n > 0$,

$$\begin{aligned} R[n] &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \Gamma_i \Gamma_j P\left\{ \Gamma(k) = \Gamma_i \text{ and } \Gamma(k+n) = \Gamma_j \right\} \\ &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \sum_{t=0}^{N-1} \left(\sum_{l=1}^{\infty} l\mathbf{D}_l \right)_{it} \sum_{s=0}^{N-1} \left(\sum_{l=1}^{\infty} l\mathbf{D}_l \right)_{js} \pi_i \left(\mathbf{D}^{n-1}\right)_{tj} \\ &= \sum_{j=0}^{N-1} \sum_{t=0}^{N-1} \left(\boldsymbol{\pi} \sum_{l=1}^{\infty} l\mathbf{D}_l \right)_t \left(\mathbf{D}^{n-1}\right)_{tj} \left(\sum_{l=1}^{\infty} l\mathbf{D}_l \mathbf{e} \right)_j \\ &= \boldsymbol{\pi} \left(\sum_{l=1}^{\infty} l\mathbf{D}_l \right) \mathbf{D}^{n-1} \left(\sum_{l=1}^{\infty} l\mathbf{D}_l \right) \mathbf{e}. \end{aligned} \tag{2.5}$$

Since $(\Gamma(k))_k$ is a stationary real valued process, $R[n]$ is even [68, p.359], i.e., for all n, $R[n] = R[-n]$. Thus,

$$\underset{n\neq 0}{R[n]} = \boldsymbol{\pi}\left(\sum_{l=1}^{\infty} l\mathbf{D}_l\right)\mathbf{D}^{|n|-1}\left(\sum_{l=1}^{\infty} l\mathbf{D}_l\right)\mathbf{e}. \tag{2.6}$$

Because also the autocorrelation sequence of $(X_k)_k$, where X_k represents the number of arrivals in slot k, is even, it can be seen from equation (1.9) that for a lag $n \neq 0$, the autocorrelation $R[n]$ of the input rate process equals that of $(X_k)_k$.

Assuming that $\mathbf{D}$ is diagonalizable, which means that corresponding to each eigenvalue which has multiplicity greater than one, as many linearly independent eigenvectors as the multiplicity of that eigenvalue should exist [18, p.368], $\mathbf{D}$ can be written as

$$\mathbf{D} = \sum_{l=0}^{N-1} \lambda_l \mathbf{g}_l \mathbf{h}_l, \tag{2.7}$$

where the λ_l's are the eigenvalues of $\mathbf{D}$, and $\mathbf{g}_l$, resp. $\mathbf{h}_l$, are the corresponding right column, resp. left row, eigenvectors such that $\mathbf{h}_l\mathbf{g}_l = 1$. This gives for the n-th power of $\mathbf{D}$ that [18, p.368]:

$$\mathbf{D}^n = \sum_{l=0}^{N-1} (\lambda_l)^n \mathbf{g}_l \mathbf{h}_l, \tag{2.8}$$

such that

$$\underset{n\neq 0}{R[n]} = \sum_{l=0}^{N-1} (\lambda_l)^{|n|-1}\, \boldsymbol{\pi}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{g}_l\mathbf{h}_l\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{e} = \sum_{l=0}^{N-1} (\lambda_l)^{|n|-1}\psi_l, \tag{2.9}$$

where

$$\psi_l = \boldsymbol{\pi}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{g}_l\mathbf{h}_l\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{e}. \tag{2.10}$$

As can be seen from these formulas, each eigenvalue λ_l of $\mathbf{D}$ contributes a term to $R[n]$. This term is determined by the eigenvalue λ_l and the corresponding ψ_l, which depends on the eigenvectors of $\mathbf{D}$ corresponding to λ_l.

Since $\mathbf{D}$ is an irreducible transition matrix, it has always 1 as a simple eigenvalue. From now on, this eigenvalue is given the index zero: $\lambda_0 = 1$. Remark from (2.2) and (2.10) that ψ_0 is the square of the mean arrival rate of the D-BMAP:

$$\psi_0 = \boldsymbol{\pi}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{e}\boldsymbol{\pi}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{e} = \left(E\left[\Gamma(k)\right]\right)^2. \tag{2.11}$$

For all eigenvalues λ_l, it is true that $|\lambda_l| \leq 1$. All complex eigenvalues appear in conjugate pairs and the conjugate of λ_l is denoted by $\widehat{\lambda}_l$. If λ_l and $\lambda_{l'}$ are conjugate, then the corresponding ψ_l and $\psi_{l'}$ are also conjugate, since $\boldsymbol{\pi} \sum_{k=1}^{\infty} k\mathbf{D}_k$ and $\sum_{k=1}^{\infty} k\mathbf{D}_k\mathbf{e}$ are real vectors and the matrices $\mathbf{g}_l\mathbf{h}_l$ and $\mathbf{g}_{l'}\mathbf{h}_{l'}$ are conjugate. For each eigenvalue, denote $\lambda_l = |\lambda_l|e^{i\omega_l}$ and $\psi_l = |\psi_l|e^{i\theta_l}$. When $\mathbf{D}$ is periodic with period d, it has d distinct eigenvalues with modulus 1: $1, c, \ldots, c^{d-1}$, where $c = e^{\frac{2\pi i}{d}}$. For these eigenvalues, ω_l equals θ_l:

Property 2.1.1. *Consider a transition matrix* $\mathbf{D}$ *which is irreducible and has period d. The* ψ_l *as defined in (2.10) corresponding to the eigenvalue* $\lambda_l = e^{\frac{2\pi i}{d}m}$*,* $m \in \{0, \ldots, d-1\}$ *of* $\mathbf{D}$ *has the same argument as* λ_l*:* $\psi_l = |\psi_l|e^{\frac{2\pi i}{d}m}$*.*

Proof. Consider ψ_l as defined in (2.10) corresponding to eigenvalue $\lambda_l = e^{\frac{2\pi i}{d}m}$ of $\mathbf{D}$, and define $c = e^{\frac{2\pi i}{d}}$. Because all elements of the matrices $\mathbf{D}_k$, $k \geq 0$, are probabilities, and thus positive, and because $\mathbf{D} = \sum_{k=0}^{\infty} \mathbf{D}_k$, according to the periodic structure of $\mathbf{D}$ (cfr. equation (1.2)), $\sum_{k=1}^{\infty} k\mathbf{D}_k$ can be written as

$$\sum_{k=1}^{\infty} k\mathbf{D}_k = \begin{pmatrix} 0 & \sum_{k=1}^{\infty} k\mathbf{D}_k^{(0)} & 0 & \cdots & 0 \\ 0 & 0 & \sum_{k=1}^{\infty} k\mathbf{D}_k^{(1)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sum_{k=1}^{\infty} k\mathbf{D}_k^{(d-2)} \\ \sum_{k=1}^{\infty} k\mathbf{D}_k^{(d-1)} & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

This, combined with (1.4) and (1.5) implies that

$$\boldsymbol{\pi} \sum_{k=1}^{\infty} k\mathbf{D}_k\ \mathbf{g}_l = \sum_{j=0}^{d-1} c^{(j+1)m} \boldsymbol{\pi}_j \sum_{k=1}^{\infty} k\mathbf{D}_k^{(j)}\ \mathbf{e} = c^m \sum_{j=0}^{d-1} c^{jm} \boldsymbol{\pi}_j \sum_{k=1}^{\infty} k\mathbf{D}_k^{(j)}\ \mathbf{e}, \tag{2.12}$$

and

$$\mathbf{h}_l \sum_{k=1}^{\infty} k\mathbf{D}_k\ \mathbf{e} = \sum_{j=0}^{d-1} c^{-jm} \boldsymbol{\pi}_j \sum_{k=1}^{\infty} k\mathbf{D}_k^{(j)}\ \mathbf{e}. \tag{2.13}$$

Because (2.13) is the conjugate of the factor after c^m in (2.12),

$$\psi_l = c^m \underbrace{\left| \mathbf{h}_l \sum_{k=1}^{\infty} k\mathbf{D}_k\ \mathbf{e} \right|^2}_{\in \mathbb{R}^+}, \tag{2.14}$$

from which it is concluded that

$$\psi_l = |\psi_l|e^{\frac{2\pi i}{d}m}. \ \blacksquare \tag{2.15}$$

Define Ω to be the collection of all eigenvalues of $\mathbf{D}$: $\Omega = \{\lambda_0, \ldots, \lambda_{N-1}\}$. By distinguishing between the different types of eigenvalues of $\mathbf{D}$, equation (2.9) can be written as:

$$\begin{aligned} \underset{n\neq 0}{R[n]} &= \psi_0 + (-1)^{|n|-1}\psi_a I_{\{\lambda_a\in\Omega \text{ and } \lambda_a=-1\}} + \sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} (\lambda_l)^{|n|-1}\psi_l \\ &\quad + \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C}\setminus\{1,-1\}) \\ |\lambda_l|=1}} |\psi_l| e^{(|n|-1)i\omega_l} e^{i\omega_l} + \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \Big((\lambda_l)^{|n|-1}\psi_l + (\widehat{\lambda_l})^{|n|-1}\widehat{\psi_l} \Big) \\ &= \psi_0 + (-1)^{|n|-1}\psi_a I_{\{\lambda_a\in\Omega \text{ and } \lambda_a=-1\}} + \sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} (\lambda_l)^{|n|-1}\psi_l \\ &\quad + 2 \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|=1}} |\psi_l| \cos(|n|\omega_l) + 2 \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} |\lambda_l|^{|n|-1} |\psi_l| \cos(|n|\omega_l - \omega_l + \theta_l). \end{aligned} \tag{2.16}$$

2.1.2 Power spectrum

The autocorrelation sequence of a stochastic process in the time domain is equivalently characterized in the frequency domain by its *power spectrum*, which is defined as the discrete-time Fourier transform of the autocorrelation sequence [68, p.409]:

$$P(\omega) = \sum_{n=-\infty}^{+\infty} R[n] e^{-in\omega}. \tag{2.17}$$

Note that only frequencies (expressed in rad/sec) in the range $-\pi < \omega \leq \pi$ need to be considered, since $P(\omega)$ is periodic in ω with period 2π. The following inversion formula [68, p.409] allows to recover $R[n]$ from $P(\omega)$:

$$R[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) e^{in\omega} d\omega. \tag{2.18}$$

For the input rate process $(\Gamma(k))_k$, $R[n]$ is an even sequence, such that (2.17) reduces to

$$P(\omega) = \sum_{n=-\infty}^{\infty} R[n] \cos(n\omega) = R[0] + 2\sum_{n=1}^{\infty} R[n] \cos(n\omega), \quad \text{with } -\pi < \omega \leq \pi. \tag{2.19}$$

From this expression, it is seen that $P(\omega) = P(-\omega)$, which shows that the knowledge of $P(\omega)$ for $0 \leq \omega \leq \pi$ is sufficient. By using expression (2.16) for $R[n]$, a formula for $P(\omega)$ is obtained from which the contribution of each eigenvalue of $\mathbf{D}$ to $P(\omega)$ can easily be read. A number of results used in the calculation of this formula is presented first.

Definition 2.1.1 (Dirac delta function). *The Dirac delta function is a 'function' that obeys*

$$\begin{aligned}
&(a) \quad \delta(\omega-\omega_0)=0 \quad \textit{when} \quad \omega \neq \omega_0,\\
&(b) \quad \int\limits_{-\infty}^{\infty} \delta(\omega-\omega_0)d\omega = 1.
\end{aligned} \tag{2.20}$$

Interpretation: the Dirac delta function can be considered as the limit of a function with a width decreasing to zero while its amplitude becomes infinite. However, the product of both (the area under the function) remains constant. Remark that the definition given above is only a 'loose' description of the Dirac delta function. A mathematically correct discussion of Dirac delta functions should use the notion of a *distribution*, a linear functional on a function space.

Property 2.1.2 (Properties of the Dirac delta function).

$$\begin{aligned}
&(a) \quad \textit{Scale property: } \delta(a(\omega-\omega_0)) = \frac{1}{|a|}\delta(\omega-\omega_0).\\
&(b) \quad \textit{Product with a function that is continuous at } \omega=\omega_0\textit{:}\\
&\qquad p(\omega)\delta(\omega-\omega_0)=p(\omega_0)\delta(\omega-\omega_0).\\
&(c) \quad \sum_{n=-\infty}^{\infty} e^{-in\omega} = 2\pi \sum_{n=-\infty}^{\infty} \delta(\omega-2\pi n).
\end{aligned}$$

More details about these properties can be found in [87, p.59, 95, 242]. Remark that when $-\pi < \omega \leq \pi$ and $0 < \omega_l < \pi$, then $\sum_{n=-\infty}^{\infty}\delta(\omega-2\pi n)=\delta(\omega)$, $\sum_{n=-\infty}^{\infty}\delta(2\omega-2\pi n)=\frac{1}{2}\delta(\omega)+\frac{1}{2}\delta(\omega-\pi)$, $\sum_{n=-\infty}^{\infty}\delta(\omega-\omega_l-2\pi n)=\delta(\omega-\omega_l)$ and $\sum_{n=-\infty}^{\infty}\delta(\omega+\omega_l-2\pi n)=\delta(\omega+\omega_l)$.

Using these intermediate results, $P(\omega)$ can be calculated by plugging equation (2.16) in equation (2.19). For clarity, this calculation is split up in parts corresponding to all possible types of eigenvalues λ_l of $\mathbf{D}$.

Type 1. $\lambda_l = \lambda_0 = 1$

$$\begin{aligned}
&2\psi_0\sum_{n=1}^{\infty}\cos(n\omega) = -\psi_0 + \psi_0\sum_{n=-\infty}^{\infty} e^{-in\omega} = -\psi_0 + 2\pi\psi_0\sum_{n=-\infty}^{\infty}\delta(\omega-2\pi n)\\
&= -\psi_0 + 2\pi\psi_0\delta(\omega).
\end{aligned} \tag{2.21}$$

Type 2. $\lambda_l = -1$

$$\begin{aligned}
&2\psi_a \sum_{n=1}^{\infty}(-1)^{n-1}\cos(n\omega) = -2\psi_a\left(\sum_{n=1}^{\infty}\cos(2n\omega) - \sum_{n=0}^{\infty}\cos\left((2n+1)\omega\right)\right)\\
&= -\psi_a\left(-1 + \sum_{n=-\infty}^{\infty} e^{-2in\omega} - \sum_{n=-\infty}^{\infty} e^{-i(2n+1)\omega}\right)\\
&= \psi_a - \pi\psi_a\left(\delta(\omega) + \delta(\omega-\pi)\right) + \pi\psi_a e^{-i\omega}\left(\delta(\omega) + \delta(\omega-\pi)\right)\\
&= \psi_a - 2\pi\psi_a\delta(\omega-\pi).
\end{aligned} \tag{2.22}$$

Type 3. $\lambda_l \in (\Omega \cap \mathbb{R} \setminus \{0, 1, -1\})$

$$\begin{aligned}
&2\sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \psi_l \sum_{n=1}^{\infty}(\lambda_l)^{n-1}\cos(n\omega) = 2\sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \frac{\psi_l}{\lambda_l}\,\mathbb{R}e\left\{\sum_{n=1}^{\infty}\left(\lambda_l e^{i\omega}\right)^n\right\}\\
&= 2\sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \frac{\psi_l}{\lambda_l}\,\mathbb{R}e\left\{\frac{1}{1-\lambda_l e^{i\omega}} - 1\right\}\\
&= 2\sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \frac{\psi_l}{\lambda_l}\,\frac{\lambda_l\cos(\omega) - (\lambda_l)^2}{1-2\lambda_l\cos\omega + (\lambda_l)^2}\\
&= 2\sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \psi_l\frac{\cos\omega - \lambda_l}{1-2\lambda_l\cos\omega + (\lambda_l)^2}.
\end{aligned} \tag{2.23}$$

Type 4. $\lambda_l \in (\Omega \cap \mathbb{C})\,, \mathbb{I}m(\lambda_l) > 0$ and $|\lambda_l| = 1$

$$\begin{aligned}
&4\sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C})\\ \mathbb{I}m(\lambda_l)>0\\ |\lambda_l|=1}} |\psi_l| \sum_{n=1}^{\infty}\cos(n\omega)\cos(n\omega_l)\\
&= 2\sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C})\\ \mathbb{I}m(\lambda_l)>0\\ |\lambda_l|=1}} |\psi_l| \sum_{n=1}^{\infty}\left(\cos\left(n(\omega-\omega_l)\right) + \cos\left(n(\omega+\omega_l)\right)\right)\\
&= \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C})\\ \mathbb{I}m(\lambda_l)>0\\ |\lambda_l|=1}} |\psi_l| \left(-2 + \sum_{n=-\infty}^{\infty} e^{-in(\omega-\omega_l)} + \sum_{n=-\infty}^{\infty} e^{-in(\omega+\omega_l)}\right)\\
&= \sum_{\substack{\lambda_l\in(\Omega\cap\mathbb{C})\\ \mathbb{I}m(\lambda_l)>0\\ |\lambda_l|=1}} |\psi_l| \left(-2 + 2\pi\delta(\omega-\omega_l) + 2\pi\delta(\omega+\omega_l)\right).
\end{aligned} \tag{2.24}$$

Type 5. $\lambda_l \in (\Omega \cap \mathbb{C}), \mathbb{I}m(\lambda_l) > 0$ and $|\lambda_l| < 1$

$$\begin{aligned}
&4 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \sum_{n=1}^{\infty} |\lambda_l|^{n-1} |\psi_l| \cos\left(n\omega_l - \omega_l + \theta_l\right) \cos(n\omega) \\
&= 4 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \sum_{n=1}^{\infty} \mathbb{R}e\left\{(\lambda_l)^{n-1}\psi_l\right\} \mathbb{R}e\left\{e^{in\omega}\right\} \\
&= 2 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \left(\mathbb{R}e\left\{ \frac{\psi_l}{\lambda_l} \sum_{n=1}^{\infty} (\lambda_l e^{i\omega})^n \right\} + \mathbb{R}e\left\{ \frac{\psi_l}{\lambda_l} \sum_{n=1}^{\infty} (\lambda_l e^{-i\omega})^n \right\} \right) \\
&= 2 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \left(\mathbb{R}e\left\{ \frac{\psi_l}{\lambda_l} \left(\frac{1}{1-\lambda_l e^{i\omega}} - 1 \right) \right\} + \mathbb{R}e\left\{ \frac{\psi_l}{\lambda_l} \left(\frac{1}{1-\lambda_l e^{-i\omega}} - 1 \right) \right\} \right) \\
&= 4 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \mathbb{R}e\left\{ \psi_l \frac{\cos\omega - \lambda_l}{1 - 2\lambda_l \cos\omega + (\lambda_l)^2} \right\}.
\end{aligned} \tag{2.25}$$

Summarizing these results leads to

$$\begin{aligned}
P(\omega) = {} & R[0] - \psi_0 + 2\pi\psi_0\delta(\omega) \\
& + \left(\psi_a - 2\pi\psi_a\delta(\omega-\pi)\right) I_{\{\lambda_a\in\Omega \text{ and } \lambda_a=-1\}} \\
& + 2 \sum_{\lambda_l\in(\Omega\cap\mathbb{R}\setminus\{0,1,-1\})} \psi_l \frac{\cos\omega - \lambda_l}{1 - 2\lambda_l\cos\omega + (\lambda_l)^2} \\
& + \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|=1}} |\psi_l| \left(-2 + 2\pi\delta(\omega-\omega_l) + 2\pi\delta(\omega+\omega_l)\right) \\
& + 4 \sum_{\substack{\lambda_l \in (\Omega\cap\mathbb{C}) \\ \mathbb{I}m(\lambda_l)>0 \\ |\lambda_l|<1}} \mathbb{R}e\left\{ \psi_l \frac{\cos\omega - \lambda_l}{1 - 2\lambda_l\cos\omega + (\lambda_l)^2} \right\}, \qquad \text{with } -\pi < \omega \leq \pi.
\end{aligned} \tag{2.26}$$

This formula shows that each eigenvalue λ_l of $\mathbf{D}$ contributes to $P(\omega)$ with a term determined by that eigenvalue and the corresponding ψ_l, and that the discrete part in the power spectrum is caused by the eigenvalues with modulus 1.

2.1.3 Stationary cumulative distribution

The *stationary cumulative distribution* $F(x)$ of Γ, the input rate in a slot, is defined as

$$F(x) = P\{\Gamma \leq x\}. \tag{2.27}$$

Since $P\{\Gamma = \Gamma_i\} = \pi_i$, $F(x)$ is completely determined by $\boldsymbol{\pi}$, the stationary distribution of the D-BMAP, and by the input rate vector $\mathbf{\Gamma} = \begin{pmatrix}\Gamma_0 & \ldots & \Gamma_{N-1}\end{pmatrix}^T = \sum_{k=0}^{\infty} k\mathbf{D}_k\,\mathbf{e}$:

$$F(x) = \sum_{\Gamma_i \leq x} \pi_i. \tag{2.28}$$

2.2 Circulant D-BMAP

In this section the circulant D-BMAP is introduced, which is a D-BMAP with as transition matrix a circulant matrix. An attractive property of a circulant matrix is that a closed formula for its eigenvalues exists, which depends on its elements and its dimension. Also the eigenvectors can be written down explicitely, since they depend only on the dimension of the circulant. This, together with a special choice for the matrices $\mathbf{Q}_k$ of the circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$, allows to simplify the expression for the ψ_l's defined in the previous section, and to identify the coupling between components of the rate vector of the circulant D-BMAP and its power spectrum. We also present in this section an easy-to-check condition for a circulant to be irreducible.

2.2.1 Definition

A N-state *circulant D-BMAP* $(\mathbf{Q}_k)_{k\geq 0}$, with $\mathbf{Q} = \sum_{k=0}^{\infty} \mathbf{Q}_k$, is a D-BMAP with as transition matrix $\mathbf{Q}$ a circulant stochastic matrix:

$$\mathbf{Q} = \begin{pmatrix} a_0 & a_1 & \ldots & a_{N-1} \\ a_{N-1} & a_0 & \ldots & a_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \ldots & a_0 \end{pmatrix}. \tag{2.29}$$

The matrices $\mathbf{Q}_k$ will be chosen such that they depend on $\mathbf{a}$, the first row of $\mathbf{Q}$, and on a vector $\boldsymbol{\gamma}$, such that

$$\sum_{k=1}^{\infty} k\mathbf{Q}_k = \mathrm{diag}(\boldsymbol{\gamma})\mathbf{Q}, \tag{2.30}$$

where $\mathrm{diag}(\boldsymbol{\gamma})$ is a diagonal matrix with the elements of $\boldsymbol{\gamma}$ on the main diagonal. The reason for choosing the matrices $\mathbf{Q}_k$ in this way is that then the input rate vector $\mathbf{\Gamma}_c$, whose elements are defined by (2.1), equals $\boldsymbol{\gamma}$. A choice for the $\mathbf{Q}_k$'s could be such that the number of arrivals that are generated while making a transition from state i follows a Poisson distribution with mean γ_i:

$$\mathbf{Q}_k = \begin{pmatrix} a_0 \frac{(\gamma_0)^k e^{-\gamma_0}}{k!} & a_1 \frac{(\gamma_0)^k e^{-\gamma_0}}{k!} & \dots & a_{N-1} \frac{(\gamma_0)^k e^{-\gamma_0}}{k!} \\ a_{N-1} \frac{(\gamma_1)^k e^{-\gamma_1}}{k!} & a_0 \frac{(\gamma_1)^k e^{-\gamma_1}}{k!} & \dots & a_{N-2} \frac{(\gamma_1)^k e^{-\gamma_1}}{k!} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 \frac{(\gamma_{N-1})^k e^{-\gamma_{N-1}}}{k!} & a_2 \frac{(\gamma_{N-1})^k e^{-\gamma_{N-1}}}{k!} & \dots & a_0 \frac{(\gamma_{N-1})^k e^{-\gamma_{N-1}}}{k!} \end{pmatrix}, \quad \forall k \in \mathbb{N}. \tag{2.31}$$

Remark that this is only a possible choice for the $\mathbf{Q}_k$'s. Everything in this chapter remains valid for another choice of the $\mathbf{Q}_k$'s, as long as equation (2.30) stays fulfilled. Further on, symbols introduced before for ordinary D-BMAPs, and used for a circulant D-BMAP, are given a 'c' as subindex.

2.2.2 Autocorrelation and power spectrum

From the previous section it is known that the autocorrelation sequence and the power spectrum of the input rate process of a D-BMAP are completely characterized by the eigenvalues λ_l of its transition matrix, the corresponding ψ_l's and $R[0]$. For a circulant D-BMAP, these values can be written as expressions which depend on the vectors $\mathbf{a}$ and $\boldsymbol{\gamma}$.

First of all, the l-th ($l \in \{0, \dots, N-1\}$) eigenvalue $(\lambda_c)_l$ of a circulant $\mathbf{Q}$ is given by [77, p.169]

$$(\lambda_c)_l = a_0 + a_1 c^l + a_2 c^{2l} \dots + a_{N-1} c^{(N-1)l}, \qquad \text{where } c = e^{\frac{2\pi i}{N}}. \tag{2.32}$$

Remark that this implies that $\widehat{(\lambda_c)_l} = (\lambda_c)_{(N-l) \bmod N}$, which means that all real eigenvalues of $\mathbf{Q}$ occur in pairs, except for $(\lambda_c)_0$ and for $(\lambda_c)_{\frac{N}{2}}$ if N is even. Notice that $(\lambda_c)_0 = 1$. Further, if $\mathbf{Q}$ is irreducible and periodic with an even period, $(\lambda_c)_{\frac{N}{2}} = -1$, since -1 needs to be a simple eigenvalue of $\mathbf{Q}$. The eigenvectors $(\mathbf{g}_c)_l$ and $(\mathbf{h}_c)_l$ which correspond to $(\lambda_c)_l$, and which are chosen such that $(\mathbf{h}_c)_l(\mathbf{g}_c)_l = 1$, are given by

$$\begin{aligned} (\mathbf{g}_c)_l &= \begin{pmatrix} 1 & c^l & c^{2l} & \dots & c^{(N-1)l} \end{pmatrix}^T, \\ (\mathbf{h}_c)_l &= \frac{1}{N} \begin{pmatrix} 1 & c^{-l} & c^{-2l} & \dots & c^{-(N-1)l} \end{pmatrix}. \end{aligned} \tag{2.33}$$

The stationary distribution $\boldsymbol{\pi}_c$ of $\mathbf{Q}$, which is the normalized left eigenvector corresponding to eigenvalue 1, is then given by

$$\boldsymbol{\pi}_c = \begin{pmatrix} \frac{1}{N} & \frac{1}{N} & \dots & \frac{1}{N} \end{pmatrix}, \tag{2.34}$$

and is thus independent of the elements of $\boldsymbol{Q}$.

Enough information is now available to derive an expression for $(\psi_c)_l$:

$$\begin{aligned}
(\psi_c)_l &= \boldsymbol{\pi}_c \left(\sum_{k=1}^{\infty} k\mathbf{Q}_k\right) (\mathbf{g}_c)_l (\mathbf{h}_c)_l \left(\sum_{k=1}^{\infty} k\mathbf{Q}_k\right) \mathbf{e} \\
&= \boldsymbol{\pi}_c \operatorname{diag}(\boldsymbol{\gamma})\mathbf{Q}(\mathbf{g}_c)_l(\mathbf{h}_c)_l \operatorname{diag}(\boldsymbol{\gamma})\mathbf{Qe} \\
&= \boldsymbol{\pi}_c \operatorname{diag}(\boldsymbol{\gamma})(\lambda_c)_l(\mathbf{g}_c)_l(\mathbf{h}_c)_l\boldsymbol{\gamma} \\
&= \frac{1}{N^2}\left(\sum_{k=0}^{N-1} \gamma_k c^{kl}\right)\left(\sum_{k=0}^{N-1} \gamma_k c^{-kl}\right)(\lambda_c)_l,
\end{aligned} \tag{2.35}$$

by using definition (2.10) in the first step, equation (2.30) in the second step, the fact that $(\mathbf{g}_c)_l$ is an eigenvector of $\mathbf{Q}$ corresponding to $(\lambda_c)_l$ in the third step, and (2.33) and (2.34) in the last step. Remark that in this expression

$$\frac{1}{N^2}\left(\sum_{k=0}^{N-1} \gamma_k c^{kl}\right)\left(\sum_{k=0}^{N-1} \gamma_k c^{-kl}\right) = \left|\frac{1}{N}\sum_{k=0}^{N-1} \gamma_k c^{kl}\right|^2 \in \mathbb{R}^+, \tag{2.36}$$

which implies that $(\psi_c)_l$ is a positive real multiple of $(\lambda_c)_l$:

$$(\psi_c)_l = \chi_l(\lambda_c)_l, \qquad \text{with } \chi_l = \left|\frac{1}{N}\sum_{k=0}^{N-1} \gamma_k c^{kl}\right|^2 \in \mathbb{R}^+. \tag{2.37}$$

Since $\widehat{(\lambda_c)_l} = (\lambda_c)_{(N-l) \bmod N}$, χ_l equals $\chi_{(N-l) \bmod N}$.

For $R_c[0]$, using (2.4) and (2.8), it is derived that

$$\begin{aligned}
R_c[0] &= \boldsymbol{\pi}_c(\boldsymbol{\Gamma}_c \odot \boldsymbol{\Gamma}_c) = \boldsymbol{\pi}_c(\boldsymbol{\gamma} \odot \boldsymbol{\gamma}) = \boldsymbol{\pi}_c \operatorname{diag}(\boldsymbol{\gamma})\mathbf{I}\boldsymbol{\gamma} \\
&= \boldsymbol{\pi}_c \operatorname{diag}(\boldsymbol{\gamma}) \left(\sum_{(\lambda_c)_l \in \Omega_c \setminus \{0\}} (\mathbf{g}_c)_l(\mathbf{h}_c)_l\right) \boldsymbol{\gamma} \\
&= \sum_{(\lambda_c)_l \in \Omega_c \setminus \{0\}} \frac{1}{N^2}\left(\sum_{k=0}^{N-1} \gamma_k c^{kl}\right)\left(\sum_{k=0}^{N-1} \gamma_k c^{-kl}\right) \\
&= \sum_{(\lambda_c)_l \in \Omega_c \setminus \{0\}} \frac{(\psi_c)_l}{(\lambda_c)_l} = \sum_{(\lambda_c)_l \in \Omega_c \setminus \{0\}} \chi_l.
\end{aligned} \tag{2.38}$$

Also $R_c[n]$, $n \neq 0$, resp. $P_c(\omega)$, can be written in function of the χ_l's and the eigenvalues

$(\lambda_c)_l$ of $\mathbf{Q}$, using (2.16), resp. (2.26), and the results just now derived:

$$\begin{aligned}\underset{n\neq 0}{R_c[n]} &= \chi_0 + (-1)^{|n|}\chi_{N/2} I_{\{N \text{ is even and } (\lambda_c)_{N/2}=-1\}} \\ &\quad + \sum_{(\lambda_c)_l \in (\Omega_c \cap \mathbb{R}\setminus\{0,1,-1\})} (\lambda_c)_l^{|n|}\chi_l + 2 \sum_{\substack{(\lambda_c)_l \in (\Omega_c \cap \mathbb{C}) \\ \mathbb{I}m((\lambda_c)_l)>0 \\ |(\lambda_c)_l|=1}} \chi_l \cos\left(|n|(\omega_c)_l\right) \\ &\quad + 2 \sum_{\substack{(\lambda_c)_l \in (\Omega_c \cap \mathbb{C}) \\ \mathbb{I}m((\lambda_c)_l)>0 \\ |(\lambda_c)_l|<1}} |(\lambda_c)_l|^{|n|}|\chi_l| \cos\left((|n|-1)(\omega_c)_l + (\theta_c)_l\right),\end{aligned} \tag{2.39}$$

and

$$\begin{aligned}P_c(\omega) &= 2\pi\chi_0\delta(\omega) + 2\pi\chi_{N/2}\delta(\omega-\pi) I_{\{N \text{ is even and } (\lambda_c)_{N/2}=-1\}} \\ &\quad + 2 \sum_{(\lambda_c)_l \in (\Omega_c \cap \mathbb{R}\setminus\{0,1,-1\})} \chi_l \left(\frac{(\lambda_c)_l \cos\omega - (\lambda_c)_l^2}{1 - 2(\lambda_c)_l \cos\omega + (\lambda_c)_l^2} + \frac{1}{2} \right) \\ &\quad + \sum_{\substack{(\lambda_c)_l \in (\Omega_c \cap \mathbb{C}) \\ \mathbb{I}m((\lambda_c)_l)>0 \\ |(\lambda_c)_l|=1}} \chi_l \left(2\pi\delta(\omega - (\omega_c)_l) + 2\pi\delta(\omega + (\omega_c)_l)\right) \\ &\quad + 4 \sum_{\substack{(\lambda_c)_l \in (\Omega_c \cap \mathbb{C}) \\ \mathbb{I}m((\lambda_c)_l)>0 \\ |(\lambda_c)_l|<1}} \chi_l \left(\mathbb{R}e\left\{ \frac{(\lambda_c)_l \cos\omega - (\lambda_c)_l^2}{1 - 2(\lambda_c)_l \cos\omega + (\lambda_c)_l^2} \right\} + \frac{1}{2} \right), \\ &\quad \text{with } -\pi < \omega \leq \pi.\end{aligned} \tag{2.40}$$

2.2.3 Stationary cumulative distribution

Because the stationary distribution $\boldsymbol{\pi}_c$ of a circulant D-BMAP is independent of the elements of its transition matrix (cfr. equation (2.34)), the stationary cumulative distribution $F_c(x)$ of Γ_c, the input rate in a slot, depends only on the input rate vector $\boldsymbol{\Gamma}_c$, which equals $\boldsymbol{\gamma}$:

$$F_c(x) = \sum_{\gamma_i \leq x} (\boldsymbol{\pi}_c)_i = \frac{1}{N} \sum_{\gamma_i \leq x} 1. \tag{2.41}$$

Define β_l, $l = 0, \ldots, N-1$, as

$$\beta_l = \frac{1}{N} \sum_{k=0}^{N-1} \gamma_k c^{lk}, \qquad \text{where } c = e^{\frac{2\pi i}{N}}. \tag{2.42}$$

To recover γ_k from the β_l's, the following inversion formula can be used, which is the discrete Fourier transform of the sequence $\beta_0, \ldots, \beta_{N-1}$:

$$\gamma_t = \sum_{m=0}^{N-1} \beta_m c^{-tm}. \tag{2.43}$$

For each β_l, denote $\beta_l = |\beta_l| e^{i\alpha_l}$. From (2.37) and (2.42) it is then concluded that

$$|\beta_l| = \sqrt{\chi_l}. \tag{2.44}$$

Hence, equation (2.43) leads to

$$\begin{aligned}\gamma_t &= \beta_0 + \sum_{m=1}^{N-1} \sqrt{\chi_m} e^{i\alpha_m} c^{-tm} = \boldsymbol{\pi}_c \boldsymbol{\gamma} + \sum_{m=1}^{N-1} \sqrt{\chi_m} e^{i(\alpha_m - \frac{2\pi}{N} tm)} \\ &= \boldsymbol{\pi}_c \boldsymbol{\gamma} + \sum_{m=1}^{N-1} \sqrt{\chi_m} \cos(\alpha_m - \frac{2\pi}{N} tm) + i \sum_{m=1}^{N-1} \sqrt{\chi_m} \sin(\alpha_m - \frac{2\pi}{N} tm).\end{aligned} \tag{2.45}$$

Because $\widehat{\beta_m} = \beta_{N-m}$ (see equation (2.42)), $\chi_{N-m} = \chi_m$ and $\alpha_{N-m} = -\alpha_m$. This implies that (2.45) reduces to

$$\begin{aligned}\gamma_t &= \boldsymbol{\pi}_c \boldsymbol{\gamma} + 2 \sum_{m=1}^{p} \sqrt{\chi_m} \cos(\alpha_m - \frac{2\pi}{N} tm), \quad \text{for } N \text{ odd: } N = 2p + 1, \\ \gamma_t &= \boldsymbol{\pi}_c \boldsymbol{\gamma} + 2 \sum_{m=1}^{p-1} \sqrt{\chi_m} \cos(\alpha_m - \frac{2\pi}{N} tm) + \sqrt{\chi_p} \cos(\alpha_p - \pi t), \quad \text{for } N \text{ even: } N = 2p.\end{aligned} \tag{2.46}$$

2.2.4 Irreducible and periodic circulants

In this subsection, a few properties concerning the irreducibility and periodicity of a circulant stochastic matrix $\mathbf{Q}$ are given. They are used later on in Section 2.4. The first property gives a necessary and sufficient condition for $\mathbf{Q}$ to be irreducible.

Property 2.2.1. *Consider a N-dimensional circulant $\mathbf{Q}$ with $\mathbf{a}$ as first row and define $I = \{i | i \neq 0$ and $a_i \neq 0\}$. Then $\mathbf{Q}$ is irreducible if and only if* $\operatorname{lcm}\left(\frac{\operatorname{lcm}(i,N)}{i}\right)_{i \in I} = N$.

Proof. Define $t = \operatorname{lcm}\left(\frac{\operatorname{lcm}(i,N)}{i}\right)_{i \in I}$.

Necessary condition. Remark first that t is always less than or equal to N, because $\forall a, b \in \mathbb{N}_0 : \operatorname{lcm}(a,b) \gcd(a,b) = ab$ [96, p.35], which implies that

$$\forall i \in I : \frac{\operatorname{lcm}(i,N)}{i} = \frac{N}{\gcd(i,N)}.$$

So t is the least common multiple of numbers that are all divisors of N.

Suppose now that $t < N$. From the definition of t, it is known that for all $i \in I$, it is a multiple of $\operatorname{lcm}(i,N)$, which implies that it is also a multiple of N. For the t-th eigenvalue of $\mathbf{Q}$ (cfr. equation (2.32)), this gives:

$$(\lambda_c)_t = \sum_{k=0}^{N-1} a_k c^{kt} = a_0 + \sum_{k\in I} a_k c^{kt} = a_0 + \sum_{k\in I} a_k = 1.$$

But also the eigenvalue $(\lambda_c)_0$ of $\mathbf{Q}$ equals 1, which means that the multiplicity of eigenvalue 1 is at least 2 (remark that by definition of t, $t \neq 0$). Because this is in contradiction with the irreducibility of $\mathbf{Q}$, the supposition was wrong, which means that t equals N.

Sufficient condition. Distinguish 2 cases:

1. $\gcd(\{i|i \in I\} \cup \{N\}) = 1$. Denote the elements of I by $i_1, \ldots, i_K$. From [96, p.54], it is known that

   $$\exists\, z_1, \ldots, z_{K+1} \in \mathbb{Z} : \sum_{j=1}^{K} i_j\, z_j + z_{K+1} N = 1.$$

   For all the z_i's, consider a $n_i \in \mathbb{N}$ such that $z_i + n_i N \geq 0$. Then

   $$\left(\sum_{j=1}^{K} i_j\,(z_j + n_j N)\right) \bmod N = 1.$$

   This means that the following sequence of transitions from state 0 to state 1 exists:

   $$0 \to i_1 \to (2i_1) \bmod N \to \cdots \to (i_1(z_1 + n_1 N)) \bmod N \to$$
   $$(i_1(z_1+n_1N) + i_2) \bmod N \to \cdots \to (i_1(z_1+n_1N) + i_2(z_2+n_2N)) \bmod N$$
   $$\to \cdots \to \left(\sum_{j=1}^{K} i_j\,(z_j + n_j N)\right) \bmod N = 1.$$

   But then also transitions $0 \to 1 \to 2 \to \cdots \to N-1 \to 0$ are possible, which means that all states of $\mathbf{Q}$ communicate with each other.

2. $\gcd(\{i|i \in I\} \cup \{N\}) > 1$. By contraposition, it is proven that this case cannot occur. Suppose it can, and denote $x = \gcd(\{i|i \in I\} \cup \{N\})$. Then N can be written as $N = bx$, where $b < N$ since $x > 1$. By [96, p.28] it is known that

   $$\forall i \in I, \exists\, m_i \in \mathbb{N}_0 : \gcd(i,N) = m_i x.$$

   Then

   $$\forall i \in I : \frac{\operatorname{lcm}(i,N)}{i} = \frac{N}{\gcd(i,N)} = \frac{N}{m_i x} = \frac{b}{m_i} \in \mathbb{N}.$$

   So for all i, b is a multiple of $\operatorname{lcm}(i,N)/i$, through which $t \leq b < N$. But this is in contradiction with $t = N$, so the supposition was wrong. ■

Remark that property 2.2.1 is different from the necessary and sufficient condition for the irreducibility of a circulant stochastic matrix as stated in [79, p.385, problem 21], which says that *"a Markov chain with a stochastic circulant matrix is irreducible if and only if* $a_0 \neq 1$*"*. This condition is however not correct, since for example the circulant with $\mathbf{a} = \begin{pmatrix} 0.2 & 0 & 0.2 & 0 & 0.6 & 0 \end{pmatrix}$ as first row is reducible, because it is impossible to reach an even numbered state starting from an odd numbered state.

Property 2.2.2. *Consider a N-dimensional irreducible circulant $\mathbf{Q}$ with period $d > 1$. Then N is a multiple of d and each periodic class of $\mathbf{Q}$ contains N/d states.*

Proof. Consider the states $u_1, \ldots, u_n$ of periodic class 1 and a state j of periodic class 0. Then, by the definition of the irreducibility and the periodicity of a Markov chain, $\forall\, u_l \in \{u_1, \ldots, u_n\}$, $\exists m_l \in \mathbb{N}$ for which the state u_l is accessible from state j in $m_l d + 1$ steps. Because $\mathbf{Q}$ is a circulant, this means that from an arbitrary state q of an arbitrarily chosen periodic class p, exactly n other states $v_1, \ldots, v_n$ are accessible in a number of steps which is a multiple of d plus 1. Namely, the state $v_i = (q + u_i - j) \bmod N$ is reachable from q in $m_i d + 1$ steps. Thus, the periodic class $(p + 1) \bmod d$ contains exactly n states. Because the periodic class p was arbitrarily chosen among all periodic classes, this means that all the periodic classes of $\mathbf{Q}$ contain n states, and thus $N = nd$. ■

Property 2.2.3. *Consider an irreducible circulant $\mathbf{Q}$ with period $d > 1$ and dimension $N = kd$. The periodic class to which a state q belongs consists of the following states: $q, (q + d) \bmod N, \ldots, (q + (k-1)d) \bmod N$.*

Proof. From property 2.2.2 it is known that each periodic class contains k states. So in the case that $k = 1$, this property is trivially true. Consider $k \geq 2$. Suppose that state 0 belongs to periodic class i ($i \in \{0, \ldots, d-1\}$), and consider two different states m and t of periodic class $j = (i + 1) \bmod d$, such that there are no states u and v, $u \neq v$, in class j for which $(u - v) \bmod N < (t - m) \bmod N$. Remark that since class j contains k states, with $k \geq 2$, it is always possible to find two such states m and t. Then there exists an $l_1 \in \mathbb{N}$ such that state m is accessible from state 0 in $l_1 d + 1$ steps. But this implies that state t is accessible from state $(t - m) \bmod N$ in $l_1 d + 1$ steps, because $\mathbf{Q}$ is a circulant. This then means that the states 0 and $(t - m) \bmod N$ belong to the same periodic class i, which on its turn implies that there exists an $l_2 \in \mathbb{N}$ such that state m is accessible from state $(t - m) \bmod N$ in $l_2 d + 1$ steps. Because $\mathbf{Q}$ is a circulant, this implies that state t is accessible from state $(2t - 2m) \bmod N$ in $l_2 d + 1$ steps, such that also state $(2t - 2m) \bmod N$ belongs to periodic class i. By continuing this reasoning, it is concluded that all the states of the form $(lt - lm) \bmod N$, $l \in \mathbb{N}$, belong to periodic class i. Consequently, all states of the form $(lt - (l-1)m) \bmod N$, $l \in \mathbb{N}$, belong to periodic class j, since they are accessible in $l_1 d + 1$ steps from state $(lt - lm) \bmod N$.

It is shown now that $((t - m) \bmod N)$ divides N. Suppose it does not, and define $x = \left\lfloor \frac{N}{(t-m) \bmod N} \right\rfloor$. Then

1. $x\left((t-m) \bmod N\right) < N$, and because also $(t-m) \bmod N < N$, it follows that $(x+1)\left((t-m) \bmod N\right) < 2N$,

2. $(x+1)\left((t-m) \bmod N\right) > N$,

3. $((x+1)(t-m)) \bmod N = ((x+1)\left((t-m) \bmod N\right)) \bmod N < (t-m) \bmod N$.

This means that there exist two states u and v in class j, $u = ((x+2)t-(x+1)m) \bmod N$ and $v = t$,

- which are different: otherwise $(u-v) \bmod N = 0$, i.e., $((x+1)t-(x+1)m) \bmod N = ((x+1)\left((t-m) \bmod N\right)) \bmod N = 0$, which means that $(x+1)\left((t-m) \bmod N\right)$ should be a multiple of N, which is not the case by item 1 and 2 up here, and

- for which $(u-v) \bmod N = ((x+1)(t-m)) \bmod N < (t-m) \bmod N$ by item 3.

Since this is in contradiction with the way t and m were chosen, the supposition made is wrong, and thus $((t-m) \bmod N)$ divides N. Denote $y = N/\left((t-m) \bmod N\right)$, $y \in \mathbb{N}$. Then all states of the form $(lt-lm) \bmod N$, $l \in \mathbb{N}$, equal one of the following y different states: $0, (t-m) \bmod N, 2\left((t-m) \bmod N\right), \ldots, (y-1)\left((t-m) \bmod N\right)$.

If it is supposed that $y > k$, then class i contains more than k states, which contradicts property 2.2.2. If it is supposed that $y < k$, then a state p of class i which is not of the form $(lt-lm) \bmod N$, $l \in \mathbb{N}$, also needs to exist. But then always one of the states of class i of the form $(lt-lm) \bmod N$ exists for which $(lt-lm-p) \bmod N < (t-m) \bmod N$. This implies that there are two different states u and v in class j, namely $u = (lt-(l-1)m) \bmod N$ and $v = (p+m) \bmod N$, for which $(u-v) \bmod N = (lt-lm-p) \bmod N < (t-m) \bmod N$, which contradicts the way t and m were chosen. So $y = k$, which means by the definition of y that $(t-m) \bmod N = d$, and thus periodic class i contains the k states $0, d, 2d, \ldots, (k-1)d$.

In an analogous way it is proven that the periodic class of an arbitrary state q contains the states $q, (q+d) \bmod N, \ldots, (q+(k-1)d) \bmod N$. ■

Corollary 2.2.4. *Consider an irreducible circulant* $\mathbf{Q}$ *with* $\mathbf{a}$ *as first row, which has period* $d > 1$ *and dimension* $N = kd$. *If* $a_l \neq 0$, *where* $l \in \{1, \ldots, N-1\}$, *then* $a_m = 0$ *if* $\nexists p \in \{0, \ldots, k-1\}$ *for which* $m = (l+pd) \bmod N$.

Proof. Since $a_l = Q_{0,l}$, and $a_m = Q_{0,m}$, it is immediately clear by writing $\mathbf{Q}$ in the form of equation (1.2), that a_m needs to be zero if state m does not belong to the periodic class of state l. So by property 2.2.3, if there is no p in $\{0, \ldots, k-1\}$ such that $m = (l+pd) \bmod N$, then $a_m = 0$. ■

Remark that in the formulation of corollary 2.2.4, $l \in \{1, \ldots, N-1\}$, since if $a_0 \neq 0$, then $\mathbf{Q}$ is aperiodic.

2.3 Superposition of M independent D-BMAPs

Consider M independent D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$, $1 \leq i \leq M$. With each of these D-BMAPs an input rate process $(\Gamma^{(i)}(k))_k$, as defined in Section 2.1, corresponds. The superposition of the M D-BMAPs is again a D-BMAP (cfr. Section 1.2.3), denoted by $(\mathbf{D}_k)_{k\geq 0}$. Denote by $(\Gamma(k))_k$ the corresponding input rate process. The autocorrelation sequence $R[n]$ and the power spectrum $P(\omega)$ of this process, and the stationary cumulative distribution $F(x)$ of Γ, the input rate of the superposition in a slot, could be obtained as explained in Section 2.1. But then it is necessary to explicitly construct the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$, which becomes practically unrealizable if M is large, or if the dimensions of the individual D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$ are large, because of the state space explosion. It is however also possible to calculate $R[n]$, $P(\omega)$ and $F(x)$ from the autocorrelation sequences $R^i[n]$, the power spectra $P^{(i)}(\omega)$ and the stationary cumulative distributions $F^{(i)}(x)$, $1 \leq i \leq M$, of the individual D-BMAPs in the superposition, as is illustrated in this section.

2.3.1 Power spectrum of the superposition

The input rate process $(\Gamma(k))_k$ is the aggregation of the M independent input rate processes $(\Gamma^{(i)}(k))_k$:

$$\Gamma(k) = \sum_{i=1}^{M} \Gamma^{(i)}(k). \tag{2.47}$$

Using this relation in equation (2.3) gives an expression for $R[n]$, the autocorrelation sequence of the input rate process $(\Gamma(k))_k$, in function of $R^{(1)}[n], \ldots, R^{(M)}[n]$, where $R^{(i)}[n]$, $1 \leq i \leq M$, is the autocorrelation sequence of $(\Gamma^{(i)}(k))_k$:

$$\begin{aligned}
R[n] &= E\left[\left(\sum_{i=1}^{M}\Gamma^{(i)}(k)\right)\left(\sum_{i=1}^{M}\Gamma^{(i)}(k+n)\right)\right] \\
&= E\left[\sum_{i=1}^{M}\Gamma^{(i)}(k)\Gamma^{(i)}(k+n) + \sum_{i=1}^{M}\sum_{\substack{j=1\\ j\neq i}}^{M}\Gamma^{(i)}(k)\Gamma^{(j)}(k+n)\right] \\
&= \sum_{i=1}^{M} E\left[\Gamma^{(i)}(k)\Gamma^{(i)}(k+n)\right] + \sum_{i=1}^{M}\sum_{\substack{j=1\\ j\neq i}}^{M} E\left[\Gamma^{(i)}(k)\Gamma^{(j)}(k+n)\right] \\
&= \sum_{i=1}^{M} R^{(i)}[n] + \sum_{i=1}^{M}\sum_{\substack{j=1\\ j\neq i}}^{M} E\left[\Gamma^{(i)}(k)\right] E\left[\Gamma^{(j)}(k+n)\right]
\end{aligned} \tag{2.48}$$

$$= \sum_{i=1}^{M} R^{(i)}[n] + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j\neq i}}^{M} E\left[\Gamma^{(i)}(k)\right] E\left[\Gamma^{(j)}(k)\right],$$

where in the last step but one the independence of the processes $(\Gamma^{(i)}(k))_k$ is used, and in the last step their stationariness. Because of equation (2.11), the autocorrelation sequence $R[n]$ can also be written as

$$R[n] = \sum_{i=1}^{M} R^{(i)}[n] + 2\sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}}. \tag{2.49}$$

When plugging this result in equation (2.19), the power spectrum $P(\omega)$ of the input rate process $(\Gamma(k))_k$ is obtained in function of the $P^{(i)}(\omega)$'s, where $P^{(i)}(\omega)$, $1 \leq i \leq M$, is the power spectrum of $(\Gamma^{(i)}(k))_k$:

$$\begin{aligned}
P(\omega) &= \sum_{i=1}^{M} R^{(i)}[0] + 2\sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}} \\
&\quad + 2\sum_{n=1}^{\infty} \cos(n\omega) \left(\sum_{i=1}^{M} R^{(i)}[n] + 2\sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}} \right) \\
&= \sum_{i=1}^{M} P^{(i)}(\omega) + 2\sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}} \left(1 + 2\sum_{n=1}^{\infty} \cos(n\omega) \right) \\
&= \sum_{i=1}^{M} P^{(i)}(\omega) + 2\sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}} \sum_{n=-\infty}^{+\infty} e^{-in\omega} \\
&= \sum_{i=1}^{M} P^{(i)}(\omega) + 4\pi\delta(\omega) \sum_{i=1}^{M} \sum_{j=i+1}^{M} \sqrt{\psi_0^{(i)}}\sqrt{\psi_0^{(j)}} \quad \text{(use property 2.1.2).}
\end{aligned} \tag{2.50}$$

From equation (2.26) it is known that each eigenvalue $\lambda_l^{(i)}$ of $\mathbf{D}^{(i)}$, $1 \leq i \leq M$, contributes to $P^{(i)}(\omega)$ with a term determined by that eigenvalue and the corresponding $\psi_l^{(i)}$. So from the formula above it is concluded that all eigenvalues of the individual D-BMAPs in the superposition contribute to $P(\omega)$. But when applying equation (2.26) to the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$ which describes the superposition, one might wonder if this conclusion is correct, since the transition matrix $\mathbf{D}$ of the superposition has more eigenvalues than only these of the individual D-BMAPs. The property below shows that the conclusion made is indeed correct, since the contribution of these eigenvalues is zero. But remark first that if $\{\mu_i\}$ and $\{\mathbf{x}_i\}$ are the eigenvalues and the corresponding eigenvectors of a matrix $\mathbf{A}$, and $\{\nu_j\}$ and $\{\mathbf{y}_j\}$ are the eigenvalues and the corresponding eigenvectors of a matrix $\mathbf{B}$, then $\mathbf{A}\otimes\mathbf{B}$ has as eigenvalues $\{\mu_i\nu_j\}$ with corresponding eigenvectors $\{\mathbf{x}_i \otimes \mathbf{y}_j\}$ (see [37, p.27]).

Property 2.3.1. *Consider the D-BMAP* $(\mathbf{D}_k)_{k\geq 0}$ *which is the superposition of* M *independent D-BMAPs* $(\mathbf{D}_k^{(i)})_{k\geq 0}$*, and one of its eigenvalues* $\lambda = \lambda^{(1)}. \ldots .\lambda^{(M)}$*, where* $\lambda^{(i)}$*,*

$i = 1, \ldots, M$, is an eigenvalue of $\mathbf{D}^{(i)}$, such that at least two of the values $\lambda^{(i)}$ are different from one. Denote the right column and left row eigenvector corresponding to λ by $\mathbf{g}$ and $\mathbf{h}$. Then $\psi = \boldsymbol{\pi}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{gh}\left(\sum_{k=1}^{\infty} k\mathbf{D}_k\right)\mathbf{e} = 0$.

Proof. At least two of the values $\lambda^{(i)}$ in $\lambda = \lambda^{(1)}. \ldots .\lambda^{(M)}$, say $\lambda^{(k)}$ and $\lambda^{(l)}$, $k \neq l$, are different from one. Denote the right eigenvectors corresponding to the eigenvalues $\lambda^{(i)}$ by $\mathbf{g}^{(i)}$. Then $\boldsymbol{\pi}^{(k)}\mathbf{g}^{(k)} = 0$ and $\boldsymbol{\pi}^{(l)}\mathbf{g}^{(l)} = 0$, because for $m = k, l$, $\boldsymbol{\pi}^{(m)}\mathbf{D}^{(m)}\mathbf{g}^{(m)} = \boldsymbol{\pi}^{(m)}\mathbf{g}^{(m)}$, $\boldsymbol{\pi}^{(m)}\mathbf{D}^{(m)}\mathbf{g}^{(m)} = \lambda^{(m)}\boldsymbol{\pi}^{(m)}\mathbf{g}^{(m)}$ and $\lambda^{(m)} \neq 1$.

By using some elementary properties of the Kronecker product [37, chapter 2], it is proven that

$$\begin{aligned}\sum_{k=1}^{\infty} k\mathbf{D}_k = \left(\sum_{k=1}^{\infty} k\mathbf{D}_k^{(1)}\right) \otimes \left(\bigotimes_{i=2}^{M} \mathbf{D}^{(i)}\right) + \mathbf{D}^{(1)} \otimes \left(\sum_{k=1}^{\infty} k\mathbf{D}_k^{(2)}\right) \otimes \left(\bigotimes_{i=3}^{M} \mathbf{D}^{(i)}\right) + \ldots \\ + \left(\bigotimes_{i=1}^{M-2} \mathbf{D}^{(i)}\right) \otimes \left(\sum_{k=1}^{\infty} k\mathbf{D}_k^{(M-1)}\right) \otimes \mathbf{D}^{(M)} + \left(\bigotimes_{i=1}^{M-1} \mathbf{D}^{(i)}\right) \otimes \left(\sum_{k=1}^{\infty} k\mathbf{D}_k^{(M)}\right). \quad (2.51)\end{aligned}$$

Because $\boldsymbol{\pi} = \bigotimes_{i=1}^{M} \boldsymbol{\pi}^{(i)}$ and $\mathbf{g} = \bigotimes_{i=1}^{M} \mathbf{g}^{(i)}$,

$$\begin{aligned}\boldsymbol{\pi}\sum_{k=1}^{\infty} k\mathbf{D}_k = \left(\boldsymbol{\pi}^{(1)}\sum_{k=1}^{\infty} k\mathbf{D}_k^{(1)}\right) \otimes \left(\bigotimes_{i=2}^{M} \boldsymbol{\pi}^{(i)}\right) + \ldots \\ + \left(\bigotimes_{i=1}^{M-1} \boldsymbol{\pi}^{(i)}\right) \otimes \left(\boldsymbol{\pi}^{(M)}\sum_{k=1}^{\infty} k\mathbf{D}_k^{(M)}\right), \quad (2.52)\end{aligned}$$

where the property that $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ is used [37, p.24]. Then

$$\begin{aligned}\boldsymbol{\pi}\sum_{k=1}^{\infty} k\mathbf{D}_k\mathbf{g} = \left(\boldsymbol{\pi}^{(1)}\sum_{k=1}^{\infty} k\mathbf{D}_k^{(1)}\mathbf{g}^{(1)}\right) \otimes \left(\bigotimes_{i=2}^{M} \boldsymbol{\pi}^{(i)}\mathbf{g}^{(i)}\right) + \ldots \\ + \left(\bigotimes_{i=1}^{M-1} \boldsymbol{\pi}^{(i)}\mathbf{g}^{(i)}\right) \otimes \left(\boldsymbol{\pi}^{(M)}\sum_{k=1}^{\infty} k\mathbf{D}_k^{(M)}\mathbf{g}^{(M)}\right), \quad (2.53)\end{aligned}$$

and because $\boldsymbol{\pi}^{(k)}\mathbf{g}^{(k)} = 0 = \boldsymbol{\pi}^{(l)}\mathbf{g}^{(l)}$, each of the terms in this sum is zero, such that $\boldsymbol{\pi}\sum_{k=1}^{\infty} k\mathbf{D}_k\mathbf{g} = 0$, which implies that $\psi = 0$. ■

2.3.2 Stationary cumulative distribution of the superposition

The stationary cumulative distribution $F(x)$ of Γ, the input rate of the superposition in a slot, is given by (see equation (2.28)):

$$F(x) = \sum_{\Gamma_i \leq x} \pi_i, \quad (2.54)$$

where $\boldsymbol{\pi}$ is the stationary distribution of the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$ describing the superposition, and Γ_i is the i-th element of the input rate vector $\boldsymbol{\Gamma} = \sum_{k=0}^{\infty} k\mathbf{D}_k\mathbf{e}$ of the superposition.

For $\boldsymbol{\pi}$, which is the left eigenvector which sums to one corresponding to eigenvalue 1 of the matrix $\mathbf{D}$, it holds that

$$\boldsymbol{\pi} = \bigotimes_{i=1}^{M} \boldsymbol{\pi}^{(i)}, \tag{2.55}$$

where $\boldsymbol{\pi}^{(i)}$ is the stationary distribution of the D-BMAP $(\mathbf{D}_k^{(i)})_{k\geq 0}$. Remark that the sum of the elements of $\boldsymbol{\pi}$ is one, since if the elements of a vector $\mathbf{a}$ sum to one, and the elements of a vector $\mathbf{b}$ sum to one, then also the elements of the vector $\mathbf{a} \otimes \mathbf{b}$ sum to one.

The input rate vector $\boldsymbol{\Gamma}$ of the superposition is obtained from the input rate vectors $\boldsymbol{\Gamma}^{(i)}$ using the expression

$$\boldsymbol{\Gamma} = \bigoplus_{i=1}^{M} \boldsymbol{\Gamma}^{(i)}. \tag{2.56}$$

This expression uses the Kronecker sum, which is defined analogously as the Kronecker product (cfr. Section 1.2.3), but the operation used now is the addition.

To derive equation (2.56), consider two independent D-BMAPs $(\mathbf{D}_k^{(1)})_{k\geq 0}$ and $(\mathbf{D}_k^{(2)})_{k\geq 0}$, with transition matrices $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ respectively. From equation (1.10) it is known that their superposition is again a D-BMAP $(\tilde{\mathbf{D}}_k)_{k\geq 0}$, with $\tilde{\mathbf{D}}_k = \sum_{l=0}^{k} \mathbf{D}_l^{(1)} \otimes \mathbf{D}_{k-l}^{(2)}$. Denote the corresponding input rate vector by $\tilde{\boldsymbol{\Gamma}}$. By using the definition of $\tilde{\boldsymbol{\Gamma}}$, and the fact that the Kronecker product is distributive with respect to the addition (see [37, p.23]), it is obtained that

$$\begin{aligned}
\tilde{\boldsymbol{\Gamma}} &= \sum_{k=0}^{\infty} k \sum_{l=0}^{k} \left(\mathbf{D}_l^{(1)} \otimes \mathbf{D}_{k-l}^{(2)}\right)\mathbf{e} = \sum_{l=0}^{\infty}\sum_{k=l}^{\infty} k\left(\mathbf{D}_l^{(1)} \otimes \mathbf{D}_{k-l}^{(2)}\right)\mathbf{e} \\
&= \sum_{l=0}^{\infty}\left[\mathbf{D}_l^{(1)} \otimes \left(\sum_{k=l}^{\infty} k\mathbf{D}_{k-l}^{(2)}\right)\right]\mathbf{e} = \sum_{l=0}^{\infty}\left[\mathbf{D}_l^{(1)} \otimes \left(\sum_{k=0}^{\infty}(k+l)\mathbf{D}_k^{(2)}\right)\right]\mathbf{e} \\
&= \sum_{l=0}^{\infty}\left[\mathbf{D}_l^{(1)} \otimes \left(\sum_{k=0}^{\infty} k\mathbf{D}_k^{(2)}\right)\right]\mathbf{e} + \sum_{l=0}^{\infty}\left[l\mathbf{D}_l^{(1)} \otimes \left(\sum_{k=0}^{\infty}\mathbf{D}_k^{(2)}\right)\right]\mathbf{e} \\
&= \left[\left(\sum_{l=0}^{\infty}\mathbf{D}_l^{(1)}\right) \otimes \left(\sum_{k=0}^{\infty} k\mathbf{D}_k^{(2)}\right)\right]\mathbf{e} + \left[\left(\sum_{l=0}^{\infty} l\mathbf{D}_l^{(1)}\right) \otimes \mathbf{D}^{(2)}\right]\mathbf{e} \\
&= (\mathbf{D}^{(1)}\mathbf{e}) \otimes \left(\sum_{k=0}^{\infty} k\mathbf{D}_k^{(2)}\mathbf{e}\right) + \left(\sum_{l=0}^{\infty} l\mathbf{D}_l^{(1)}\mathbf{e}\right) \otimes (\mathbf{D}^{(2)}\mathbf{e}) \\
&= (\mathbf{e} \otimes \boldsymbol{\Gamma}^{(2)}) + (\boldsymbol{\Gamma}^{(1)} \otimes \mathbf{e}) = \boldsymbol{\Gamma}^{(1)} \oplus \boldsymbol{\Gamma}^{(2)},
\end{aligned} \tag{2.57}$$

where in the third last step the following is used: $\mathbf{A}\mathbf{z} \otimes \mathbf{B}\mathbf{w} = (\mathbf{A} \otimes \mathbf{B})(\mathbf{z} \otimes \mathbf{w})$ (see [37, p.22]). Before this step, the $\mathbf{e}$'s used are vectors with as length the dimension of $\mathbf{D}^{(1)} \otimes \mathbf{D}^{(2)}$, while afterwards it are vectors with as length the dimension of $\mathbf{D}^{(1)}$, resp. $\mathbf{D}^{(2)}$. Since the dimension of the vectors $\mathbf{e}$ is clear from the context, no effort is done to provide this information in the notation of the vector. By applying the reasoning above $M-1$ times, it is obtained that for the superposition of the M D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$, $1 \leq i \leq M$, the input rate vector $\mathbf{\Gamma}$ is given by equation (2.56).

Thus, by using equations (2.55) and (2.56), the stationary cumulative distribution $F(x)$ of Γ can now be calculated from equation (2.54), using only information of the individual D-BMAPs constituting the superposition.

2.4 Circulant matching procedure

In this section, the procedure to match the superposition of M independent D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$, $1 \leq i \leq M$, by a circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ is presented. The circulant D-BMAP is constructed such that $P_c(\omega)$ matches $P(\omega)$ and $F_c(x)$ matches $F(x)$, where $P_c(\omega)$ and $F_c(x)$ denote the power spectrum and stationary cumulative distribution of the input rate process of the circulant D-BMAP, while $P(\omega)$ and $F(x)$ denote the power spectrum and stationary cumulative distribution of the input rate process of the superposition. As is known from Section 2.2, a circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ is defined by a vector $\mathbf{a}$, the first row of its transition matrix $\mathbf{Q}$, and by a vector $\boldsymbol{\gamma}$, which is chosen such that it equals the input rate vector $\mathbf{\Gamma}_c$. The construction of $(\mathbf{Q}_k)_{k\geq 0}$ consists of two steps:

1. the construction of $\mathbf{a}$ and the fixing of the χ_l's such that $P_c(\omega)$ matches $P(\omega)$,

2. the construction of $\boldsymbol{\gamma}$ such that $F_c(x)$ matches $F(x)$.

It is clear that since both $P_c(\omega)$ and $\boldsymbol{\gamma}$ depend on the χ_l's (see equations (2.40) and (2.46)), these two steps cannot be performed completely uncoupled from each other. The χ_l's fixed in the first step need to be taken into account in the second step.

2.4.1 Matching the power spectrum

From equation (2.26) it is known that the power spectrum of a D-BMAP $(\mathbf{D}_k^{(i)})_{k\geq 0}$ is completely determined by $R^{(i)}[0]$ and a contribution of each of its eigenvalues. The contribution of an eigenvalue $\lambda_l^{(i)}$ depends on that eigenvalue and on the corresponding $\psi_l^{(i)}$. Because of equation (2.50), this means that the power spectrum of the superposition of the M D-BMAPs $(\mathbf{D}_k^{(i)})_{k\geq 0}$, $1 \leq i \leq M$, is completely known by the $R^{(i)}[0]$'s and by all eigenvalues of the D-BMAPs and their contributions to their respective power spectra. Thus, if a D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ could be constructed with as eigenvalues of $\mathbf{Q}$ the same eigenvalues that contribute to the power spectrum $P(\omega)$ of the superposition, and if the ψ_l's of this D-BMAP are tuned right, this new D-BMAP would be a D-BMAP with the same power spectrum as the superposition. Remark however that constructing a matrix with a desired set of eigenvalues is difficult, if at all possible, and involves a so-called inverse spectrum problem [77, chapter 7]. To circumvent this problem, the D-BMAP that is constructed is a circulant D-BMAP, for which closed formulas exist to describe its eigenvalues, such that the inverse spectrum problem reduces to an easier to solve index search problem.

Construction of a

The first task to tackle is thus the construction of a circulant stochastic matrix $\mathbf{Q}$ which has as eigenvalues all values from a predefined set. Because the eigenvalues of a circulant N-dimensional matrix are obtained (see equation (2.32)) by

$$\boldsymbol{\lambda}_c = \mathbf{aF}, \tag{2.58}$$

where $\mathbf{F}_{jk} = c^{jk}$, $0 \leq j,k \leq N-1$, $c = e^{\frac{2\pi i}{N}}$, and $\boldsymbol{\lambda}_c = \begin{pmatrix}(\lambda_c)_0 & (\lambda_c)_1 & \dots & (\lambda_c)_{N-1}\end{pmatrix}$, it is possible to obtain $\mathbf{a}$ from $\boldsymbol{\lambda}_c$ by

$$\mathbf{a} = \boldsymbol{\lambda}_c \mathbf{F}^{-1}, \quad \text{where } (\mathbf{F}^{-1})_{jk} = \frac{1}{N} c^{-jk}. \tag{2.59}$$

This relation is however not useful for constructing a circulant *stochastic* matrix, because nothing guarantees that the elements of $\mathbf{a}$ will be positive real numbers which add up to one. There also not necessarily exists a stochastic circulant which has only the values of the predefined set as eigenvalues. So the approach is to search for a circulant which has the envisaged values as eigenvalues, but very likely also some extra ones. To eliminate the contribution of those last ones to $P_c(\omega)$, the corresponding χ_l's are chosen zero.

Denote all the different predefined eigenvalues in a vector $\boldsymbol{\lambda}_P = \begin{pmatrix}(\lambda_P)_0 & \dots & (\lambda_P)_{D-1}\end{pmatrix}$. The objective is then to find a vector $\mathbf{a} = \begin{pmatrix}a_0 & \dots & a_{N-1}\end{pmatrix}$ such that $\forall i \in \{0, \dots, N-1\}$: $a_i \in \mathbb{R}^+$, $\mathbf{ae} = 1$ and $\forall \lambda \in \boldsymbol{\lambda}_P : \lambda \in \boldsymbol{\lambda}_c$. The vector $\mathbf{a}$ is sought through the adjustment of $(N, \mathbf{i})$, where N represents the length of $\mathbf{a}$, and thus also of $\boldsymbol{\lambda}_c$, and $\mathbf{i} = \begin{pmatrix}i_0 & \dots & i_{D-1}\end{pmatrix}$ represents the position in $\boldsymbol{\lambda}_c$ of the D predefined eigenvalues. Eigenvalue 1 is always an element of $\boldsymbol{\lambda}_P$. Choose $(\lambda_P)_0 = 1$. Then i_0 equals 0. For each selected $(N, \mathbf{i})$, a linear programming scheme will be drawn up to find a solution $\mathbf{a}$. If no solution exists, the eigenvalue indices $\mathbf{i}$ are adaptively changed and the dimension N is gradually expanded, until a solution $\mathbf{a}$ is found.

For each choice $(N, \mathbf{i})$, the $D + N$ conditions that have to be fulfilled when searching for a solution $\mathbf{a}$ are:

$$\begin{cases} \sum_{j=0}^{N-1} a_j = 1 & \\ \sum_{j=0}^{N-1} a_j c^{lj} = (\lambda_P)_k, & l = i_k, k = 1, \dots, D-1, \\ a_l \geq 0 & l = 0, \dots, N-1. \end{cases} \tag{2.60}$$

Define

$$\mathbf{x} = \begin{pmatrix} a_0 & \dots & a_{N-1} \end{pmatrix}^T, \tag{2.61}$$

$$\mathbf{b} = \begin{pmatrix} 1 & \mathbb{R}e\{(\lambda_P)_1\} & \mathbb{I}m\{(\lambda_P)_1\} & \dots & \mathbb{R}e\{(\lambda_P)_{D-1}\} & \mathbb{I}m\{(\lambda_P)_{D-1}\} \end{pmatrix}^T, \tag{2.62}$$

$$\text{and } \mathbf{A} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & C_{i_1,1} & C_{i_1,2} & \dots & C_{i_1,N-1} \\ 0 & S_{i_1,1} & S_{i_1,2} & \dots & S_{i_1,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & C_{i_{D-1},1} & C_{i_{D-1},2} & \dots & C_{i_{D-1},N-1} \\ 0 & S_{i_{D-1},1} & S_{i_{D-1},2} & \dots & S_{i_{D-1},N-1} \end{pmatrix}, \tag{2.63}$$

where $C_{l,j}$ and $S_{l,j}$ are defined as $C_{l,j} = \cos\frac{2\pi lj}{N}$ and $S_{l,j} = \sin\frac{2\pi lj}{N}$.
Then the conditions in (2.60) can be rewritten as

$$\mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}, \tag{2.64}$$

which are exactly the constraints as they appear in the standard form of a linear program (LP) (cfr. [95, p.2]):

$$\begin{array}{ll} \text{minimize} & z = \mathbf{cx} \\ \text{subject to} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array} \tag{2.65}$$

where $\mathbf{c}$ is a cost vector. This means that the standard algorithm to solve a LP, namely the revised simplex algorithm [95, p.5], can be used here.

The revised simplex algorithm consists of two phases. Phase I is used to find a feasible solution to $\mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, or to determine that no feasible solution exists. Phase II uses the solution generated in phase I to start with and solves the minimization part of the LP. Thus, for our problem, phase I is sufficient to decide if there exists a solution $\mathbf{a}$ for a given $(N, \mathbf{i})$, and to find one if there exists one. If no solution exists, the vector $\mathbf{i}$ is adaptively changed. If no feasible solution is obtained after a finite number of adaptations, N is expanded and a new cycle of index adaptations is started. Obviously, when a feasible solution exists for a certain N, the computation time to find it depends on the index adaptation scheme. The size of the solution set expands rapidly with N: in theory, $D-1$ indices have to be given a different value between 1 and $N-1$, which means that there are $\frac{(N-1)!}{(N-D)!}$ possible ways to do this. However, for some index vectors $\mathbf{i}$, it is known in advance that no solution exists, so these do not need to be considered. The same is true for some values of N. Further, it is possible to eliminate some of the values from the predefined set of eigenvalues, and as such cut down the number of conditions of the LP problem, while still enforcing that these conditions are fulfilled in the final solution. This is done by eliminating in advance the index vectors $\mathbf{i}$ for which these conditions would not be fulfilled.

The benefit of this procedure is that the dimension of the LP problems which need to be solved becomes smaller, and that fewer possibilities for $\mathbf{i}$ need to be considered.

The following observations allow some reduction in the number of choices $(N, \mathbf{i})$ and in the size of $\boldsymbol{\lambda}_P$:

- For each nonreal value λ in $\boldsymbol{\lambda}_P$, also its complex conjugate $\hat{\lambda}$ will be present in $\boldsymbol{\lambda}_P$, since the values in $\boldsymbol{\lambda}_P$ are obtained as eigenvalues of stochastic matrices. There is however no need to take both λ and $\hat{\lambda}$ into account for the construction of $\mathbf{a}$, since it holds for a circulant that $\widehat{(\lambda_c)_l} = (\lambda_c)_{(N-l) \bmod N}$: if $\lambda \in \boldsymbol{\lambda}_P$ appears in $\boldsymbol{\lambda}_c$ at position l, then $\hat{\lambda}$ automatically appears in $\boldsymbol{\lambda}_c$ at position $(N-l) \bmod N$. Thus, all values λ with $\mathbb{I}m(\lambda) < 0$ can be eliminated from λ_P. Then all index vectors $\mathbf{i}$ containing the values l and $(N-l) \bmod N$ should not be considered anymore, since no feasible solution consists for them. Further, for all real values λ in $\boldsymbol{\lambda}_P$, only index values smaller than or equal to $N/2$ need to be considered, because when $(\lambda_c)_t$ with $t > (N/2)$ equals λ, then also $(\lambda_c)_{N-t}$ equals λ.
- When $\mathbf{Q}$ needs to have period d, it is known from the properties in Section 2.2.4 that
  - N needs to be a multiple of d, and
  - only N/d values in $\mathbf{a}$ are free to take values different from 0, such that the number of columns of $\mathbf{A}$ in equation (2.63) becomes N/d instead of N.

  We choose one of these values to be a_1, implying (see corollary 2.2.4) that the other values are $a_{1+d}, a_{1+2d}, \ldots, a_{1+(k-1)d}$, where $k = N/d$. This choice has the following advantages:

  - When $a_1 \neq 0$, then the resulting circulant $\mathbf{Q}$ is irreducible (see property 2.2.1).
  - When $(\lambda_c)_l = \lambda$, then $(\lambda_c)_{(l+mk) \bmod N} = c^{mk}\lambda$, where $m \in \{0, \ldots, d-1\}$ and $c = e^{\frac{2\pi i}{N}}$.

  As such, all values with argument not in the segment $[0, \frac{2\pi}{d}[$ can be eliminated from $\boldsymbol{\lambda}_P$, because all these values are the result of a rotation over an angle in $\{\frac{2\pi}{d}, 2\frac{2\pi}{d}, \ldots, (d-1)\frac{2\pi}{d}\}$ of a value with argument in the segment $[0, \frac{2\pi}{d}[$ (see Section 1.1.3). So if a λ which belongs to the resulting $\boldsymbol{\lambda}_P$ appears in $\boldsymbol{\lambda}_c$ on position l, then the values $c^k\lambda, c^{2k}\lambda, \ldots, c^{(d-1)k}\lambda$ appear automatically in $\boldsymbol{\lambda}_c$ at positions $(l+k) \bmod N, (l+2k) \bmod N, \ldots, (l+(d-1)k) \bmod N$. Then for all index vectors $\mathbf{i}$ containing a value l and a value $(l+mk) \bmod N$, where $m \in \{1, \ldots, d-1\}$, no feasible solution exists, so they do not need to be considered anymore. When λ with argument in $]0, \frac{2\pi}{d}[\setminus\{\frac{\pi}{d}\}$ belongs to $\boldsymbol{\lambda}_P$, then also $\lambda^* = \widehat{\lambda c^{(d-1)k}}$ has its argument in $]0, \frac{2\pi}{d}[\setminus\{\frac{\pi}{d}\}$ and belongs to $\boldsymbol{\lambda}_P$. For each couple (λ, λ^*), one of these values can also be removed from $\boldsymbol{\lambda}_P$, since when λ appears in $\boldsymbol{\lambda}_c$ on position l, then λ^* appears in $\boldsymbol{\lambda}_c$ on position $(N-l+k) \bmod N$. Thus also for all index vectors $\mathbf{i}$ containing a value l and a value $(N-l-mk) \bmod N$, where $m \in \{1, \ldots, d-1\}$, no feasible solution exists, such that these index vectors can also be ignored.

- Consider the vector $\boldsymbol{\lambda}_P$ containing the predefined eigenvalues, but now with all values as described above removed, i.e., now $\boldsymbol{\lambda}_P$ contains only values λ with $\mathbb{I}m(\lambda) \geq 0$, $\arg(\lambda) \in [0, \frac{2\pi}{d}[$ and only one value of each couple (λ, λ^*), with $\lambda^* = \widehat{\lambda c^{(d-1)k}}$ is present in $\boldsymbol{\lambda}_P$. When the number of values in $\boldsymbol{\lambda}_P$ is D and d is the period $\mathbf{Q}$ should have, then the minimal dimension $N_{\min}$ the circulant $\mathbf{Q}$ should have is given by $d(2D-1)$ when $\boldsymbol{\lambda}_P$ contains no real values different from 1 or values with as argument π/d. Otherwise $N_{\min} = 2d(D-1)$. This is because position 0 in $\boldsymbol{\lambda}_c$ is taken by eigenvalue 1, and the positions $k, 2k, \ldots, (d-1)k$ are then taken by the values $c^k, c^{2k}, \ldots, c^{(d-1)k}$ (k = N/d, where N is the dimension of the circulant). When k is even, one of the positions in $\{k/2, 3k/2, \ldots, (2d-1)k/2\}$ can be taken by a real value or a value with argument π/d of $\boldsymbol{\lambda}_P$. All the other positions in this set are then taken by the rotations of the value over the angles $2\pi m/d$, where $m = 1, \ldots, d-1$. All the other values λ in $\boldsymbol{\lambda}_P$ take a free position l, by which also the positions $(l + mk) \bmod N$, $m = 1, \ldots, d-1$ and $(N - l - mk) \bmod N$, $m = 0, \ldots, d-1$, are taken by the rotations of λ over the angles $2\pi m/d$, $m = 1, \ldots, d-1$, and by their complex conjugates. So for all values N smaller than $N_{\min}$, no feasible solution exists. Remark that the same values for $N_{\min}$ are obtained when $d = 1$, i.e., when $\mathbf{Q}$ should be aperiodic.

- Consider the set of points in the complex plane bounded by the N-sided polygon, $N \geq 2$, inscribed in the unit circle and with one of its vertices at $(0, 1)$, and denote it by P_N. When not all values in $\boldsymbol{\lambda}_P$ belong to P_N, then no circulant of dimension N with all values in $\boldsymbol{\lambda}_P$ as eigenvalues exists. This is a consequence of the property that says that a complex number is an eigenvalue of a stochastic circulant of dimension N if and only if it belongs to P_N [77, corollary 1.3, p.169].

Although the observations made above substantially reduce the number of choices $(N, \mathbf{i})$ that have to be investigated and the size of $\boldsymbol{\lambda}_P$, especially for periodic circulants, the size of the solution set still expands rapidly with N and with the number of eigenvalues in the predefined set $\boldsymbol{\lambda}_P$. As a consequence, the circulant matching method is only useful when all D-BMAPs in the superposition are identical, or can be divided into a limited group of identical ones, since then many of their eigenvalues are identical. Because contributions of identical eigenvalues to the power spectrum can be added up, they only need to appear once as eigenvalue of the circulant.

Fixing the χ_l's

Consider Ω_c, the collection of all eigenvalues of the N-dimensional circulant $\mathbf{Q}$ whose construction was described above. By construction, a portion Ω_c^P of Ω_c contains the eigenvalues of the matrices $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(M)}$. It are these eigenvalues which contribute to the power spectrum $P(\omega)$ of the superposition. The aim is now to determine the contribution these eigenvalues should have to $P_c(\omega)$, such that $P_c(\omega)$ matches $P(\omega)$. From equations (2.40) and (2.37) it is known that the contribution of each eigenvalue $(\lambda_c)_l \in \Omega_c$ is determined by that eigenvalue and by a corresponding χ_l, which needs to be a positive real number.

Further, $\forall l \in \{0, \ldots, N-1\}$, χ_l should equal $\chi_{(N-l) \bmod N}$. To avoid that the eigenvalues of $\mathbf{Q}$ in $\Omega_c \setminus \Omega_c^P$, i.e., the eigenvalues which do not contribute to $P(\omega)$, do make a contribution to $P_c(\omega)$, they are chosen equal to zero:

$$\forall l \in \{0, \ldots, N-1\} \text{ for which } (\lambda_c)_l \in (\Omega_c \setminus \Omega_c^P) : \chi_l = 0. \tag{2.66}$$

Remark that this is not in contradiction with the requirement that χ_l should equal $\chi_{(N-l) \bmod N}$, since if $(\lambda_c)_l \in (\Omega_c \setminus \Omega_c^P)$, then $(\lambda_c)_{(N-l) \bmod N} = \widehat{(\lambda_c)}_l$, and as such also $(\lambda_c)_{(N-l) \bmod N} \in (\Omega_c \setminus \Omega_c^P)$, because each value in Ω_c^P is an eigenvalue of a stochastic matrix.

For eigenvalue $(\lambda_c)_0 = 1$, choose

$$\chi_0 = \left(\sum_{i=1}^{M} \sqrt{\psi_0^{(i)}} \right)^2 . \tag{2.67}$$

For the other eigenvalues $(\lambda_c)_l \in \Omega_c^P$ with $|(\lambda_c)_l| = 1$, choose

$$\chi_l = \sum_{i=1}^{M} |\psi_{i_l}^{(i)}|, \tag{2.68}$$

where $\psi_{i_l}^{(i)}$ is the 'ψ-value' of the i-th D-BMAP $(\mathbf{D}_k^{(i)})_{k\geq 0}$ which corresponds to eigenvalue $\lambda_{i_l}^{(i)} = (\lambda_c)_l$. When not all $\mathbf{D}^{(i)}$'s are identical, it is possible that $(\lambda_c)_l$ is not an eigenvalue of a certain $\mathbf{D}^{(i)}$. In that case assume that $\psi_{i_l}^{(i)} = 0$ in (2.68). When choosing the χ_l's corresponding to eigenvalues with modulus 1 as defined above, the discrete parts of the power spectrum of the superposition $P(\omega)$ and of the power spectrum of the circulant $P_c(\omega)$ are exactly matched (see equations (2.40), (2.50) and (2.26)).

To match the continuous parts of the power spectra $P_c(\omega)$ and $P(\omega)$, define first the following functions of ω:

$$\begin{aligned} X_c(\omega) = 2 & \sum_{\substack{(\lambda_c)_l \in (\Omega_c^P \cap \mathbb{R} \setminus \{0,1,-1\}) \\ l \leq N/2}} \left(1 + I_{\{l<N/2\}}\right) \chi_l \left(\frac{(\lambda_c)_l \cos\omega - (\lambda_c)_l^2}{1 - 2(\lambda_c)_l \cos\omega + (\lambda_c)_l^2} + \frac{1}{2} \right) \\ & + 4 \sum_{\substack{(\lambda_c)_l \in (\Omega_c^P \cap \mathbb{C}) \\ \mathbb{I}m((\lambda_c)_l) > 0 \\ |(\lambda_c)_l| < 1}} \chi_l \left(\mathbb{R}e \left\{ \frac{(\lambda_c)_l \cos\omega - (\lambda_c)_l^2}{1 - 2(\lambda_c)_l \cos\omega + (\lambda_c)_l^2} \right\} + \frac{1}{2} \right), \end{aligned} \tag{2.69}$$

and

$$X(\omega) = \sum_{i=1}^{M} X^{(i)}(\omega), \tag{2.70}$$

where

$$
\begin{aligned}
X^{(i)}(\omega) = {} & R^{(i)}[0] - \psi_0^{(i)} + \psi_a^{(i)} I_{\{\lambda_a^{(i)} \in \Omega^{(i)} \text{ and } \lambda_a^{(i)}=-1\}} - 2 \sum_{\substack{\lambda_l^{(i)} \in (\Omega^{(i)} \cap \mathbb{C}) \\ \mathbb{I}m(\lambda_l^{(i)})>0 \\ |\lambda_l^{(i)}|=1}} |\psi_l^{(i)}| \\
& + 2 \sum_{\lambda_l^{(i)} \in (\Omega^{(i)} \cap \mathbb{R} \setminus \{0,1,-1\})} \psi_l^{(i)} \frac{\cos\omega - \lambda_l^{(i)}}{1 - 2\lambda_l^{(i)} \cos\omega + (\lambda_l^{(i)})^2} \\
& + 4 \sum_{\substack{\lambda_l^{(i)} \in (\Omega^{(i)} \cap \mathbb{C}) \\ \mathbb{I}m(\lambda_l^{(i)})>0 \\ |\lambda_l^{(i)}|<1}} \mathbb{R}e\left\{ \psi_l^{(i)} \frac{\cos\omega - \lambda_l^{(i)}}{1 - 2\lambda_l^{(i)} \cos\omega + (\lambda_l^{(i)})^2} \right\}.
\end{aligned}
\tag{2.71}
$$

Remark the factors $1 + I_{\{l<N/2\}}$ in the definition of $X_c(\omega)$. Their presence is due to the fact that if $(\lambda_c)_l \in \Omega_c^P \cap \mathbb{R} \setminus \{0, 1, -1\}$, with $l \leq N/2$, then if $l \neq N/2$, also $(\lambda_c)_{N-l} \in \Omega_c^P \cap \mathbb{R} \setminus \{0, 1, -1\}$. Thus in the continuous part of the power spectrum of a circulant a term for χ_l and a term for χ_{N-l} with identical coefficients occur. Since it is required that χ_l equals χ_{N-l}, only the unknown value χ_l is considered in $X_c(\omega)$. Later on, after a value has been established for χ_l, χ_{N-l} is set equal to χ_l.

Choose S different values $\omega_1, \ldots, \omega_S \in]-\pi, \pi]$ and let p be the number of χ_l's which appear in $X_c(\omega)$. Further, define the matrix $\mathbf{E}$, in which $\mathbf{E}_{ij}$ $(1 \leq i \leq S, 1 \leq j \leq p)$ is the coefficient of the j-th χ_l in $X_c(\omega_i)$, and the column vector $\mathbf{f}$ whose i-th $(1 \leq i \leq S)$ component equals $X(\omega_i)$. By solving the nonnegative least square problem (NNLS)

$$
\begin{array}{ll}
\text{minimize} & ||\mathbf{Ex} - \mathbf{f}|| \\
\text{subject to} & \mathbf{x} \geq \mathbf{0},
\end{array}
\tag{2.72}
$$

using the nonnegative least square algorithm [65, p.161], a vector $\mathbf{x}$ with p components is found, in which the i-th component gives the value that is assigned to the i-th χ_l in $X_c(\omega)$.

For all χ_l's for which no value is fixed yet, a value is already assigned to $\chi_{(N-l) \bmod N}$, and since χ_l should equal $\chi_{(N-l) \bmod N}$, set $\chi_l = \chi_{(N-l) \bmod N}$.

2.4.2 Matching the stationary cumulative distribution

In this section, the vector $\boldsymbol{\gamma}$ will be constructed such that $F_c(x)$, the stationary cumulative distribution of the input rate process of the circulant, matches the stationary cumulative distribution $F(x)$ of the input rate process of the superposition. From equations (2.34) and (2.41) it is known that since $\boldsymbol{\gamma}$ is an equal probability vector, $F_c(x)$ is a cumulative distribution which jumps by $1/N$ at each component γ_i of the vector $\boldsymbol{\gamma}$. So in order for $F(x)$ to be matched by $F_c(x)$, it is needed to rediscretize $F(x)$ by partitioning its range into N equal probability rates. Denote the partitioned range, sorted in ascending order, by $\boldsymbol{\gamma}' = \begin{pmatrix}\gamma'_0 & \dots & \gamma'_{N-1}\end{pmatrix}$. Also sort the elements in $\boldsymbol{\gamma}$ in ascending order and denote the sorted $\boldsymbol{\gamma}$ by $\boldsymbol{\gamma}_s = \begin{pmatrix}\gamma_{s_0} & \dots & \gamma_{s_{N-1}}\end{pmatrix}$.

From equation (2.46) it is known that the components of $\boldsymbol{\gamma}$ (and thus also of $\boldsymbol{\gamma}_s$) depend on N, the χ_i's and the α_i's. Remark that in equation (2.46),

$$\boldsymbol{\pi}_c\boldsymbol{\gamma} = \boldsymbol{\pi}_c\boldsymbol{\Gamma}_c = \boldsymbol{\pi}_c\sum_{k=1}^{\infty} k\mathbf{Q}_k\mathbf{e} = \sqrt{(\psi_c)_0} = \sqrt{\chi_0}. \tag{2.73}$$

Since N and the χ_i's were already fixed before when constructing the circulant matrix $\mathbf{Q}$ and matching the power spectrum, the components of $\boldsymbol{\gamma}$ can only be tuned via the components of $\boldsymbol{\alpha} = \begin{pmatrix}\alpha_1 & \dots & \alpha_{\lfloor N/2\rfloor}\end{pmatrix}$, and more in particular via those components α_m of $\boldsymbol{\alpha}$ for which $(\lambda_c)_m \in \Omega_c^P$, because if $(\lambda_c)_m \notin \Omega_c^P$, then χ_m is chosen zero, and the term with α_m disappears from equation (2.46). The distribution matching can then be formulated as a minimization problem:

$$\begin{array}{ll} \underset{\alpha_1,\dots,\alpha_{\lfloor N/2\rfloor}}{\text{minimize}} & \dfrac{1}{N}\displaystyle\sum_{k=0}^{N-1} ||\gamma'_k - \gamma_{s_k}|| \\ \text{subject to} & \gamma_{s_k} \geq 0,\ \forall k \in \{0,\dots,N-1\}, \end{array} \tag{2.74}$$

which is solved by a direct search method which does not need gradients or other derivative information because the objective function is not differentiable.

2.5 Conclusions and related work

In this chapter we described the circulant matching method, of which the purpose is to replace the superposition of independent D-BMAPs by a circulant D-BMAP, which matches the power spectrum and the stationary cumulative distribution of the input rate process of the exact superposition. The reason why a replacement of the exact superposition is needed, is that this exact superposition suffers from a state space explosion, which makes that it becomes most of the time impossible to construct this superposition, not to mention using it as input to a queueing system. The circulant matching method for D-BMAPs is based on a component of a measurement-based tool developed by San-qi Li et al. [46] that constructs a circulant modulated Poisson process to model a traffic stream. An important difference with the method of San-qi Li is that he works in continuous time, while a D-BMAP is a discrete-time model. So to replace the superposition of D-BMAPs by a new circulant D-BMAP, we had to adapt the method for discrete time. Simultaneously, the method was extended such that the periodicity which is present in the transition matrix of D-BMAPs that model periodic traffic streams, and which is thus also noticed in their superposition, is preserved.

Although the circulant matching method allows to solve some realistic queueing problems (see for example Section 3.3 of the next chapter), it certainly is not generally usable. A first problem is in the construction of the circulant transition matrix, and more in particular in the number of possible choices of $(N, \mathbf{i})$ that have to investigated. When the predefined set of eigenvalues the circulant should have becomes large (say more than 10, after the reductions we proposed), it might take a long time before a circulant with these values as eigenvalues is found. So as mentioned already before, the circulant matching method is only useful when all D-BMAPs in the superposition are identical, or can be divided into a limited group of identical ones, since then many of their eigenvalues are identical. A positive point on the other hand is that the same circulant transition matrix can be used when considering a superposition of another number of the same D-BMAPs. The difference will then be in the rate vector $\boldsymbol{\gamma}$ associated with the circulant D-BMAP, not in its transition matrix. A second possible problem is in the construction of the rate vector $\boldsymbol{\gamma}$ when a large part of the probability mass of the rate distribution of the exact superposition is situated at the value zero, or very close to it, as can occur when considering the superposition of on/off sources. In that case, it happens that no solution for the minimization problem formulated in (2.74) exists for which all constraints are fulfilled, i.e., for which all components of the rate vector $\boldsymbol{\gamma}$ are positive. An example of this problem is given in Section 3.2 of the next chapter.

Of course the circulant matching method is not the only method which tries to circumvent the state space explosion problem that occurs with the superposition of Markovian sources. However, not too much literature is found about it, certainly not for discrete-time sources, although these source models have received increasing interest with the introduction of packet-based transport protocols. An aggregation technique for the superposition of N identical sources is proposed in [23]. The technique is based on grouping together states of the exact superposition that are equivalent from both the total rate generated when the source is in that state and their future evolution in the system at each transition step. The similarity of two states in terms of their evolution is estimated by calculating the distance between these two states, considered as N-tuples, when their elements are ordered in lexicographic order. In [43] a method is proposed which provides an approximate so-called 'discrete MMPP (D-MMPP)' for the superposition of two independent D-MMPPs. Based on the observation that the multiplexed process has many states for which the rates generated in that state are very close together, another D-MMPP with a much smaller set of states is constructed, whose associated rates are spread out to cover the original range of states. Remark however that in this method only first order statistics of the exact superposition are matched. Second order statistics such as for example the autocorrelation are ignored.

Most other related articles start from traffic traces and design a parameter fitting method for continuous-time Markov modulated Poisson processes (MMPPs). One example is of course the method behind the SMAQ tool [46], on which we based the circulant matching method, and that was already discussed in Chapter 1. Another more recent example is given in [86], where a technique is proposed to construct an MMPP with $2M$ states that matches the autocovariance tail and the marginal distribution of the process that counts the number of arrivals in sampling intervals. First an MMPP with 2 states is constructed that matches the decay of the autocovariance tail. Then an MMPP with M states is constructed to match the distribution function. The final MMPP with $2M$ states is obtained by superimposing these two MMPPs. Focus is in [86] on the modeling of traffic traces exhibiting long range dependence. Although Markov models are not intrinsically long range dependent, the MMPP is used to capture the tail of the autocovariance up to the so-called correlation horizon, which is related to the maximum buffer size. Because the procedure matches two statistical functions that can also be calculated from statistical functions of the individual sources in a superposition, without explicitely constructing the superposition, the method might also be used to circumvent the state space explosion problem.

Chapter 3

Numerical examples and applications

In this chapter numerical examples and applications of the circulant matching method are given. In Section 3.1 we first illustrate the rather theoretical description in the previous chapter of the different steps of the method, by commenting upon a numerical example. In Section 3.2 the circulant matching method is applied to the superposition of M identical two dimensional Markovian sources. For such sources it is possible to calculate exact queueing results, by using an exact description of the superposition as input to the queueing system. This makes a comparison between the results obtained with the constructed circulant as input to the queueing system and the exact results possible. Moreover, a special type of two dimensional Markovian sources, i.e., the on/off sources, allows us to demonstrate when the circulant matching method does not perform well, or not at all. Using a Markovian MPEG model, the multiplexing of MPEG video sources is considered in Section 3.3. In this section the circulant matching method is applied to the superposition of a mix of two types of these sources. We presented this application also in [89]. Based on loss results obtained by using the circulant as input for a finite queueing system, CAC boundaries for the mix of these two types of sources are obtained. These boundaries are then compared to CAC boundaries that are obtained experimentally. More details about the CAC experiments performed can be found in [1, 2]. Section 3.4 concludes this chapter.

3.1 An illustrative example of the circulant matching method

The objective of this section is to illustrate the theoretical description in the previous chapter of the different steps of the circulant matching method by an example. A circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ is constructed which matches the superposition of $M = 50$ identical D-BMAPs $(\mathbf{D}_k)_{k\geq 0}$. Remark that no concrete application is hidden behind the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$ that is used, it is purely chosen to be small enough to write down and large enough to be periodic and have different types of eigenvalues, such that the different aspects of the method can be illustrated by it.

Consider the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$ with as transition matrix

$$\mathbf{D} = \sum_{k=0}^{\infty} k\mathbf{D}_k = \begin{pmatrix} \mathbf{0} & \mathbf{A}^{(0)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \tag{3.1}$$

where

$$\mathbf{A}^{(0)} = \begin{pmatrix} 0.3 & 0.5 & 0.1 & 0.1 & 0 \\ 0.1 & 0 & 0.2 & 0.4 & 0.3 \\ 0.1 & 0.1 & 0.1 & 0.2 & 0.5 \\ 0 & 0 & 0.5 & 0.3 & 0.2 \\ 0.2 & 0.2 & 0 & 0.3 & 0.3 \\ 0.1 & 0.2 & 0.1 & 0.5 & 0.1 \end{pmatrix}, \quad \mathbf{A}^{(1)} = \begin{pmatrix} 0.2 & 0.1 & 0.4 & 0.3 & 0 \\ 0.5 & 0.2 & 0.1 & 0 & 0.2 \\ 0.3 & 0.5 & 0 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.6 \\ 0.2 & 0.5 & 0.1 & 0 & 0.2 \end{pmatrix},$$

$$\text{and} \quad \mathbf{A}^{(2)} = \begin{pmatrix} 0.3 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.3 & 0.3 & 0.1 & 0 \\ 0.2 & 0 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0.3 & 0.3 & 0.1 & 0 & 0.1 \\ 0.4 & 0.2 & 0 & 0.1 & 0 & 0.3 \end{pmatrix}. \tag{3.2}$$

First remark that $\mathbf{D}$ is irreducible and periodic with period $d = 3$. Its stationary distribution is given by

$$\begin{aligned} \boldsymbol{\pi} = (&0.0833 \quad 0.0614 \quad 0.0570 \quad 0.0600 \quad 0.0257 \quad 0.0459 \quad 0.0466 \quad 0.0617 \\ &0.0609 \quad 0.0930 \quad 0.0712 \quad 0.0820 \quad 0.0924 \quad 0.0412 \quad 0.0294 \quad 0.0884). \end{aligned} \tag{3.3}$$

The matrices $\mathbf{D}_k$, $k \geq 0$, are not explicitely written out here, but they are such that

$$\begin{aligned} \boldsymbol{\Gamma} = \sum_{k=1}^{\infty} k\mathbf{D}_k\mathbf{e} = (&1.2 \quad 1 \quad 1 \quad 1.3 \quad 0.8 \quad 1 \quad 1.6 \quad 1.3 \quad 1.4 \quad 1 \quad 1.3 \\ &2.1 \quad 1.9 \quad 2.2 \quad 1.9 \quad 1.8)^T. \end{aligned} \tag{3.4}$$

Then

$$E\left[\Gamma(k)\right] = \boldsymbol{\pi}\boldsymbol{\Gamma} = \boldsymbol{\pi}\sum_{k=1}^{\infty} k\mathbf{D}_k\mathbf{e} = \sqrt{\psi_0} = 1.4416. \tag{3.5}$$

The position of the 16 eigenvalues of the matrix $\mathbf{D}$ in the complex plane is shown in Figure 3.1. Because under a rotation of the plane by $2\pi/3$ this set of eigenvalues needs to go over into itself, and because the dimension of $\mathbf{D}$ is not a multiple of three, its period,

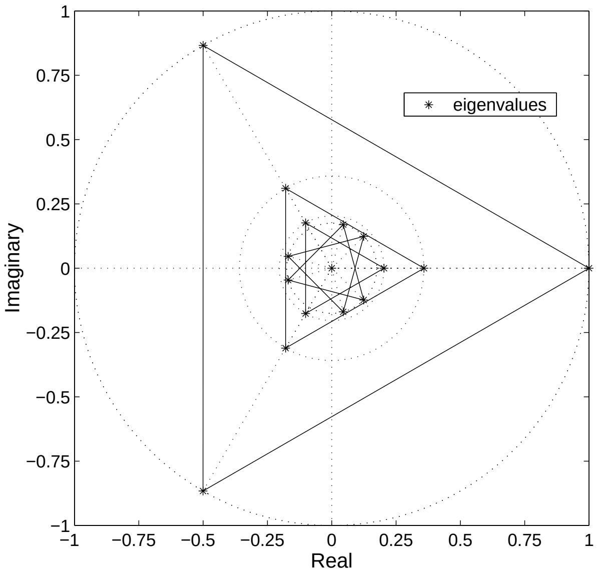

Figure 3.1: Position in the complex plane of the eigenvalues of the transition matrix **D**. This set of eigenvalues goes over into itself under a rotation of the plane by $2\pi/3$.

one of the eigenvalues needs to be zero. The eigenvalues of **D** that are different from zero contribute to the power spectrum $P(\omega)$ of the D-BMAP $(\mathbf{D}_k)_{k\geq 0}$, each by a term determined by that eigenvalue and a corresponding ψ_l. The eigenvalues different from zero are:

$$\begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{pmatrix} = \begin{pmatrix}1\\ 0.3586\\ 0.2035\\ 0.1244+0.1236i\\ 0.0448+0.1696i\end{pmatrix}, \quad \begin{pmatrix}\lambda_5\\ \lambda_6\\ \lambda_7\\ \lambda_8\\ \lambda_9\end{pmatrix} = \begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{pmatrix} c = \begin{pmatrix}-0.5+0.8660i\\ -0.1793+0.3106i\\ -0.1018+0.1762i\\ -0.1692+0.0460i\\ -0.1692-0.0460i\end{pmatrix},$$

$$\text{and} \quad \begin{pmatrix}\lambda_{10}\\ \lambda_{11}\\ \lambda_{12}\\ \lambda_{13}\\ \lambda_{14}\end{pmatrix} = \begin{pmatrix}\lambda_0\\ \lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{pmatrix} c^2 = \begin{pmatrix}-0.5-0.8660i\\ -0.1793-0.3106i\\ -0.1018-0.1762i\\ 0.0448-0.1696i\\ 0.1244-0.1236i\end{pmatrix}, \tag{3.6}$$

where $c = e^{\frac{2\pi i}{3}}$ and the eigenvalues are ordered in such a way that all λ_i, $0 \leq i \leq 4$, have

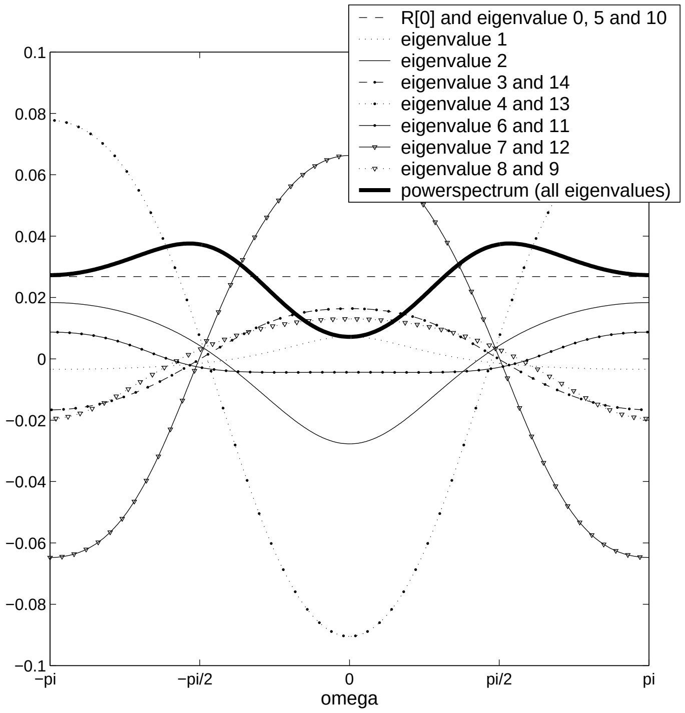

Figure 3.2: Contribution of the different eigenvalues of the transition matrix $\mathbf{D}$ to the continuous part of the power spectrum.

their argument in $[0, \frac{2\pi}{3}[$. The corresponding ψ_l's have the following values:

$$\begin{aligned}(\psi_0 \quad \ldots \quad \psi_{14}) = (&2.0783 \quad 0.0023 \quad -0.0110 \quad 0.0043+0.0043i \\ &-0.0215+0.0043i \quad -0.0350+0.0607i \quad -0.0017-0.0010i \quad 0.0171-0.0102i \\ &0.0039+0.0027i \quad 0.0039-0.0027i \quad -0.0350-0.0607i \quad -0.0017+0.0010i \\ &0.0171+0.0102i \quad -0.0215-0.0043i \quad 0.0043-0.0043i)\,. \end{aligned} \tag{3.7}$$

Notice that the ψ_l's corresponding to conjugate eigenvalues are also conjugate, and that the ψ_l's corresponding to the eigenvalues with modulus one have the same argument as their corresponding eigenvalue.

Figure 3.2 shows the contribution of the different eigenvalues of $\mathbf{D}$ to the continuous part of the power spectrum, and also the continuous part of the power spectrum itself, which is the sum of all contributions. Remark that as in equation (2.26), the contribution of a non-real eigenvalue and its conjugate are taken together. Also all contributions which are constant (i.e., not dependent on ω) are taken together. The eigenvalues λ_0, λ_5 and λ_{10} contribute also to the discrete part of the power spectrum.

The idea is now to look for a circulant $\mathbf{Q}$ of period $d = 3$ which has among its eigenvalues all eigenvalues of $\mathbf{D}$ (except 0, since 0 does not contribute to the power spectrum). From

Section 2.4.1 it is known that it suffices to look for a circulant, with $\begin{pmatrix} a_0 & \dots & a_{N-1} \end{pmatrix}$ as first row, where N is a multiple of $d = 3$ and where only $a_1, a_4, a_7, \dots, a_{N-2}$ are free to take values different from zero, that has all predefined values of a vector $\boldsymbol{\lambda}_P$ as eigenvalues. Because from all eigenvalues of $\mathbf{D}$ with argument in $[0, \frac{2\pi}{3}[$ the value $0.1244 + 0.1236i$ equals the complex conjugate of $(0.0448 + 0.1696i)e^{\frac{4\pi i}{3}}$, after performing all reductions of the size of $\boldsymbol{\lambda}_P$ as proposed in Section 2.4.1, $\boldsymbol{\lambda}_P$ contains the elements

$$\boldsymbol{\lambda}_P = \begin{pmatrix} 1 & 0.3586 & 0.2035 & 0.0448 + 0.1696i \end{pmatrix}. \tag{3.8}$$

From Section 2.4.1 it is also known that the minimal dimension for a circulant with these predefined values as eigenvalues is 18. There indeed exists a circulant of dimension 18 which has the values in $\boldsymbol{\lambda}_P$ as eigenvalues, i.e., the circulant with first row $\begin{pmatrix} a_0 & \dots & a_{17} \end{pmatrix}$, where $a_1 = 0.3033$, $a_4 = 0.1965$, $a_7 = 0.0078$, $a_{10} = 0.1465$, $a_{13} = 0.0871$, $a_{16} = 0.2588$, and all other a_i's equal to zero. From property 2.2.1 it is known that the circulant is irreducible, and from property 2.2.3 it is seen that the circulant has period $d = 3$. Denote $c = e^{\frac{2\pi i}{18}}$. The eigenvalues of the circulant are then given by

$$(\lambda_c)_l = a_1 c^l + a_4 c^{4l} + a_7 c^{7l} + a_{10} c^{10l} + a_{13} c^{13l} + a_{16} c^{16l}, \tag{3.9}$$

where $0 \le l \le 17$. By construction,

$$(\lambda_c)_{(l+mk) \bmod 18} = c^{mk} (\lambda_c)_l, \tag{3.10}$$

where $k = 18/d = 6$ and $m \in \{0, \dots, d-1\}$. Thus, the eigenvalues of $\mathbf{Q}$ are:

$$\begin{pmatrix} (\lambda_c)_0 \\ (\lambda_c)_1 \\ (\lambda_c)_2 \\ (\lambda_c)_3 \\ (\lambda_c)_4 \\ (\lambda_c)_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 0.3586 \\ 0.1244 + 0.1236i \\ -0.1018 - 0.1762i \\ 0.0448 + 0.1696i \\ -0.1793 + 0.3106i \end{pmatrix}, \quad \begin{pmatrix} (\lambda_c)_6 \\ (\lambda_c)_7 \\ (\lambda_c)_8 \\ (\lambda_c)_9 \\ (\lambda_c)_{10} \\ (\lambda_c)_{11} \end{pmatrix} = \begin{pmatrix} (\lambda_c)_0 \\ (\lambda_c)_1 \\ (\lambda_c)_2 \\ (\lambda_c)_3 \\ (\lambda_c)_4 \\ (\lambda_c)_5 \end{pmatrix} c^6 = \begin{pmatrix} -0.5 + 0.8660i \\ -0.1793 + 0.3106i \\ -0.1692 + 0.0460i \\ 0.2035 \\ -0.1692 - 0.0460i \\ -0.1793 - 0.3106i \end{pmatrix},$$

$$\text{and} \quad \begin{pmatrix} (\lambda_c)_{12} \\ (\lambda_c)_{13} \\ (\lambda_c)_{14} \\ (\lambda_c)_{15} \\ (\lambda_c)_{16} \\ (\lambda_c)_{17} \end{pmatrix} = \begin{pmatrix} (\lambda_c)_0 \\ (\lambda_c)_1 \\ (\lambda_c)_2 \\ (\lambda_c)_3 \\ (\lambda_c)_4 \\ (\lambda_c)_5 \end{pmatrix} c^{12} = \begin{pmatrix} -0.5 - 0.8660i \\ -0.1793 - 0.3106i \\ 0.0448 - 0.1696i \\ -0.1018 + 0.1762i \\ 0.1244 - 0.1236i \\ 0.3586 \end{pmatrix}. \tag{3.11}$$

As can be seen, the circulant has all values of $\boldsymbol{\lambda}_P$ as eigenvalues, but also all other eigenvalues of $\mathbf{D}$ (except 0). The complex conjugate of a value $(\lambda_c)_j$ is found as $(\lambda_c)_{(18-j) \bmod 18}$. Remark that since a circulant exists with the minimal dimension possible to have the values of $\boldsymbol{\lambda}_P$ as eigenvalues, the circulant has no other eigenvalues than values which are also eigenvalues of $\mathbf{D}$. When no circulant of dimension 18 would exist which has the values of $\boldsymbol{\lambda}_P$ as eigenvalues, or when we would search for a circulant of a higher dimension than the minimal one, $\mathbf{Q}$ would have also other eigenvalues than these of $\mathbf{D}$. Remark that until now, the information about the number $M = 50$ of D-BMAPs $(\mathbf{D}_k)_{k\geq 0}$ in the superposition was never used. So the same circulant transition matrix can be used in the matching of the superposition of another number of D-BMAPs $(\mathbf{D}_k)_{k\geq 0}$.

The next step is to fix the χ_i's corresponding to the eigenvalues $(\lambda_c)_i$ of the circulant, in such a way that the power spectrum of the circulant matches the power spectrum of the superposition of 50 D-BMAPs $(\mathbf{D}_k)_{k\geq 0}$. First the discrete part of the power spectrum is matched. For the superposition, this discrete part is given by (see equations (2.26) and (2.50))

$$5000\pi\psi_0\delta(\omega) + 100\pi|\psi_5|\left(\delta(\omega - 2\pi/3) + \delta(\omega + 2\pi/3)\right), \tag{3.12}$$

while that of the circulant is given by (see equation (2.40))

$$2\pi\chi_0\delta(\omega) + 2\pi\chi_6\left(\delta(\omega - 2\pi/3) + 2\pi\delta(\omega + 2\pi/3)\right). \tag{3.13}$$

So when choosing $\chi_0 = 2500\psi_0 = 5.1957 \times 10^3$ and $\chi_6 = 50|\psi_5| = 3.5025$, the discrete parts of both power spectra match exactly. By definition, also χ_{12} is now fixed: $\chi_{12} = \chi_6 = 3.5025$. In case that the circulant would have also other eigenvalues than these in $\boldsymbol{\lambda}_P$, their corresponding χ-values would be set to zero. To match the continuous parts of both power spectra, the nonnegative least square algorithm is used (cfr. Section 2.4.1). Combining the output of this algorithm with the χ_i's already fixed results in the following values for the χ_i's:

$$\begin{aligned}(\chi_0 \quad \ldots \quad \chi_{17}) = \big(&5.1957 \times 10^3 \quad 0 \quad 0 \quad 0 \quad 0.0163 \quad 0.3292 \quad 3.5025 \quad 0.3292 \\ &0 \quad 0 \quad 0 \quad 0.3292 \quad 3.5025 \quad 0.3292 \quad 0.0163 \quad 0 \quad 0 \quad 0\big). \end{aligned} \tag{3.14}$$

Remark that χ_i's corresponding to conjugate eigenvalues are forced to be equal, i.e., $\chi_j = \chi_{(18-j) \bmod 18}$.

The power spectra are now matched. From the resulting circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$, the transition matrix $\mathbf{Q}$ is already known. The vector $\boldsymbol{\gamma}$, which should equal the input rate vector $\boldsymbol{\Gamma}_c$ of the circulant D-BMAP, and which is needed to completely describe it, is still missing and is obtained now by matching the stationary cumulative distribution of the input rate process of the circulant with that of the superposition. During this matching, it should be taken into account that the components γ_t of $\boldsymbol{\gamma}$ depend on the χ_i's, which are already fixed:

$$\gamma_t = \sqrt{\chi_0} + 2\sum_{m=1}^{8}\sqrt{\chi_m}\cos(\alpha_m - \frac{2\pi}{18}tm) + \sqrt{\chi_9}\cos(\alpha_9 - \pi t), \tag{3.15}$$

where $\sqrt{\chi_0} = \sqrt{(\psi_c)_0} = \boldsymbol{\pi}_c\boldsymbol{\Gamma}_c = \boldsymbol{\pi}_c\boldsymbol{\gamma}$, and $\sqrt{\chi_0} = 50\sqrt{\psi_0}$, so the mean input rate of the circulant is also already fixed, and equals the mean input rate of the superposition. Remark that in equation (3.15), only the α_m's are still free variables (more in particular, α_4, α_5, α_6 and α_7, since for other m's the χ_m is zero).

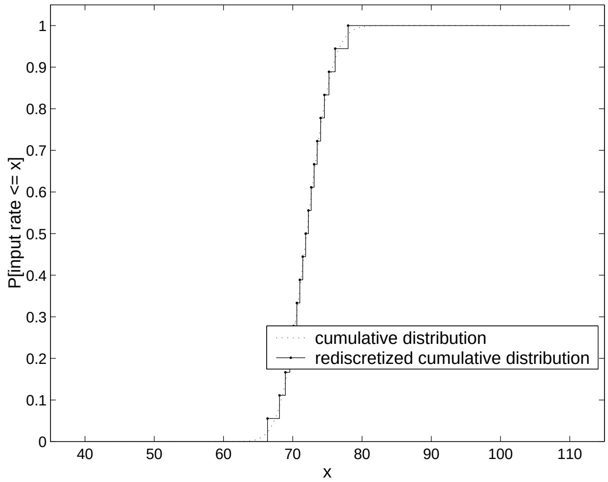

Figure 3.3: Stationary cumulative distribution of the input rate process of the superposition, together with its rediscretized version. Rediscretization is done such that all steps have a height of 1/18.

Because $\boldsymbol{\pi}_c = \big(1/18 \quad \dots \quad 1/18\big)$, $\boldsymbol{\gamma}$ is an equal probability vector. So $F(x)$, which is the stationary cumulative distribution of the superposition, should first be rediscretized such that its range is partitioned into N equal probability rates. $F(x)$ is obtained as

$$F(x) = \sum_{\Gamma_i^* \leq x} \pi_i^*, \qquad \text{where } \boldsymbol{\pi}^* = \bigotimes_{i=1}^{50} \boldsymbol{\pi}, \qquad \text{and } \boldsymbol{\Gamma}^* = \bigoplus_{i=1}^{50} \boldsymbol{\Gamma}, \tag{3.16}$$

and is shown in Figure 3.3, together with its rediscretized equivalent. Remark that in the figure $F(x)$ is plot as a continuous function for the sake of clearness, but of course it is also a discrete staircase function, although one with many small steps. The partitioned range of $\boldsymbol{\Gamma}^*$, sorted in ascending order, is

$$\begin{aligned}(\gamma_0' \quad \dots \quad \gamma_{17}') = (&66.3486 \quad 68.0841 \quad 68.9373 \quad 69.5737 \quad 70.1103 \quad 70.5886 \\ &71.0252 \quad 71.4442 \quad 71.8532 \quad 72.2575 \quad 72.6674 \quad 73.0899 \quad 73.5358 \quad 74.0240 \\ &74.5741 \quad 75.2361 \quad 76.1289 \quad 77.9869)\,. \end{aligned} \tag{3.17}$$

When solving the minimization problem formulated in equation (2.74), the following values are found to minimize the goal function: $\alpha_4 = -0.2822$, $\alpha_5 = -2.8825$, $\alpha_6 = -10.9887$, and $\alpha_7 = -0.7469$. The resulting rate vector $\boldsymbol{\gamma}$ is then

$$\begin{aligned}\boldsymbol{\gamma} = (\gamma_0 \quad \dots \quad \gamma_{17}) = (&72.0336 \quad 74.0654 \quad 70.6549 \quad 71.4404 \quad 74.0062 \quad 70.7135 \\ &71.5844 \quad 74.8095 \quad 68.6273 \quad 72.5679 \quad 76.5514 \quad 66.5414 \quad 72.5486 \quad 77.1326 \\ &67.2766 \quad 72.1583 \quad 75.4499 \quad 69.3037)\,, \end{aligned} \tag{3.18}$$

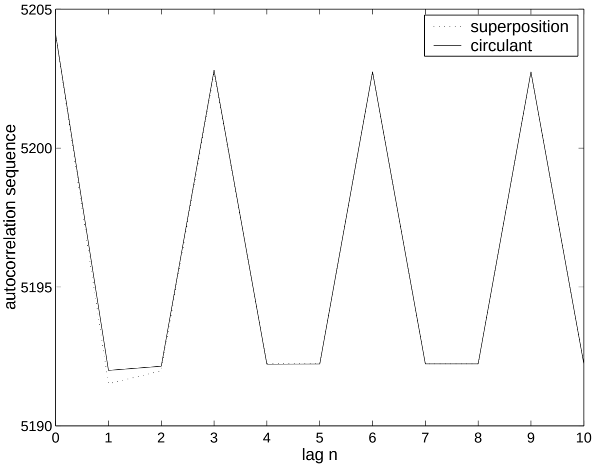

Figure 3.4: Autocorrelation of the input rate process of the superposition and of the circulant. The lag shown is limited to ten, since for a larger lag the difference is not perceptible anymore. The largest absolute difference between both sequences is 0.48, which gives a relative difference of 9.22×10^{-5}.

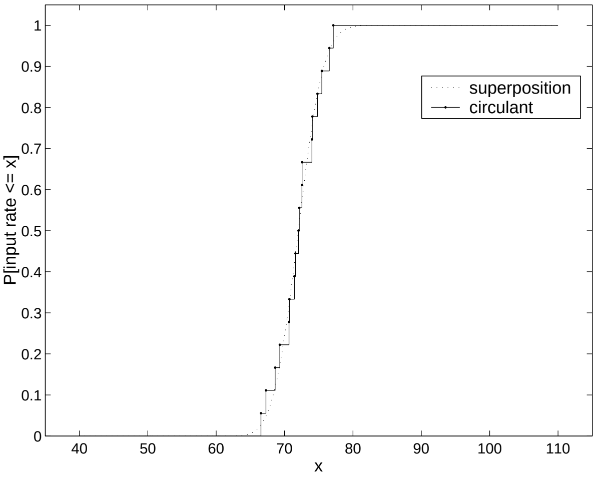

Figure 3.5: Stationary cumulative distribution of the input rate process of the superpostition and of the circulant. All steps in the distribtution of the circulant have a height of 1/18.

which finalizes the circulant matching process. The result is a circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$, where the matrices $\mathbf{Q}_k$ are constructed from $\mathbf{a} = \begin{pmatrix} a_0 & \dots & a_{17} \end{pmatrix}$ and $\boldsymbol{\gamma}$ as in equation (2.31). This circulant D-BMAP, which has by the matching a similar autocorrelation sequence and stationary cumulative distribution as the input rate process of the superposition of 50 D-BMAPs $(\mathbf{D}_k)_{k\geq 0}$ (see Figures 3.4 and 3.5), can be used as a tractable replacement of this superposition.

3.2 Superposition of two dimensional MMBP sources

In this section the circulant matching method is applied to the superposition of M identical two dimensional Markov modulated Bernouilli arrival processes (MMBP). A first motivation for this is validation: because such a source has only two states, there exists an exact method without state space explosion to describe the superposition of M identical MMBP sources. So queueing results with the exact superposition as input traffic on one hand, and the circulant that approximates the superposition on the other hand, can be obtained and compared. A second motivation is that a Markovian on/off source is a special case of a two dimensional MMBP, and with on/off sources a situation for which the circulant matching method does not work well, or does not work at all, can be illustrated.

3.2.1 Markov modulated Bernouilli sources

Consider a discrete-time Markov chain with transition matrix $\mathbf{D}$. When this chain is in state i, an arrival is generated according to a Bernouilli distribution with parameter p_i, i.e., with probability p_i an arrival occurs, and with probability $1 - p_i$ no arrival occurs. Hence, this arrival process is similar to a Bernouilli arrival process, but the arrival probability is modulated by the state of a discrete-time Markov chain. Such an arrival process is called a Markov modulated Bernouilli process (MMBP).

In this section only two dimensional MMBPs are considered. Denote by α the probability that the source makes a transition from the first state to the second state, and by β the probability that it makes a transition from the second to the first state. A D-MAP description of this source is then

$$\mathbf{D}_0 = \begin{pmatrix} (1-\alpha)(1-p_1) & \alpha(1-p_1) \\ \beta(1-p_2) & (1-\beta)(1-p_2) \end{pmatrix}, \quad \mathbf{D}_1 = \begin{pmatrix} (1-\alpha)p_1 & \alpha p_1 \\ \beta p_2 & (1-\beta)p_2 \end{pmatrix}, \tag{3.19}$$

and $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$. The stationary distribution of this source is given by $\boldsymbol{\pi} = \big(\beta/(\alpha+\beta) \quad \alpha/(\alpha+\beta)\big)$, and its mean arrival rate by $\lambda = \boldsymbol{\pi}\mathbf{D}_1\mathbf{e} = (\beta p_1 + \alpha p_2)/(\alpha+\beta)$. Remark that the durations that the source stays in a state are geometrically distributed, with mean $1/\alpha$ for the first state, and $1/\beta$ for the second state.

When one of the parameters p_1 or p_2 equals zero, the MMBP source is an on/off source. Suppose that $p_2 = 0$. When the source is then in the second state, it is 'off' or 'silent', i.e., no arrivals are generated. When the source is in the first state, it is 'on' or 'active'.

3.2.2 Superposition of two dimensional MMBP sources

As for every D-BMAP, the exact superposition of M identical two dimensional MMBP sources that are described by the D-MAP $(\mathbf{D}_0, \mathbf{D}_1)$, is given by the D-BMAP with 2^M states obtained from the matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ as described in Section 1.2.3. However, because each source has only two states, also the D-BMAP $(\mathbf{S}_k)_{0\leq k\leq M}$ with $M+1$ states, in which a state i, $0 \leq i \leq M$, corresponds to the fact that i sources are in the first state (and thus $M - i$ sources are in the second state) can describe the traffic generating process of the superposition. The elements $\mathbf{S}_{i,j}$ of the transition matrix $\mathbf{S}$ describe the probability of making a transition from a situation in which i sources are in the first state, to a situation in which j sources are in the first state. When the number of sources that stay in the first state is denoted by l, which then implies that $i-l$ sources make a transition from the first to the second state, while $j - l$ of the sources that are in the second state transit to the first state, and thus $M - i - j + l$ of the sources stay in the second state, $\mathbf{S}_{i,j}$ is obtained as follows:

$$\mathbf{S}_{i,j} = \sum_{l=\max\{0,i+j-M\}}^{\min\{i,j\}} \binom{i}{l}(1-\alpha)^l\alpha^{i-l}\binom{M-i}{j-l}\beta^{j-l}(1-\beta)^{M-i-j+l}. \tag{3.20}$$

Type	α	β	p_1	p_2	mean sojourn time in state 1	mean sojourn time in state 2	mean arrival rate λ
A	1/25	1/50	1/30	1/40	25	50	1/36
B	4/25	1/50	1/8	1/64	25/4	50	1/36
C	1/75	1/150	1/30	1/40	75	150	1/36
D	2/25	1/100	1/8	1/64	25/2	100	1/36

Table 3.1: Parameters and characteristics of the MMBP sources used in Section 3.2.4.

When a source is in the first state, it generates an arrival with probability p_1, while when it is in the second state an arrival is generated with probability p_2. So when t sources are in the first state, then $m \in \{0, \ldots, t\}$ arrivals from sources that are in the first state occur with probability $\binom{t}{m} p_1^m (1-p_1)^{t-m}$, while $n \in \{0, \ldots, M-t\}$ arrivals from sources that are in the second state occur with probability $\binom{M-t}{n} p_2^n (1-p_2)^{M-t-n}$, such that

$$(\mathbf{S}_k)_{i,j} = \sum_{l=\max\{0,k-M+i\}}^{\min\{i,k\}} \binom{i}{l} p_1^l (1-p_1)^{i-l} \binom{M-i}{k-l} p_2^{k-l} (1-p_2)^{M-i-k+l} \, \mathbf{S}_{i,j}. \qquad (3.21)$$

3.2.3 Circulant matching of two dimensional MMBP sources

Because a transition matrix has 1 as eigenvalue, and because the sum of the eigenvalues of a matrix is equal to the sum of the diagonal entries of that matrix, the two eigenvalues of the transition matrix of a two dimensional MMBP source are 1 and $1-\alpha-\beta$. For the same reasons the two dimensional circulant with $\big(1-(\alpha+\beta)/2 \quad (\alpha+\beta)/2\big)$ as first row has these values as eigenvalues. So there always exists a two dimensional circulant with the same eigenvalues as a two dimensional MMBP. However, using a two dimensional circulant to replace the superposition of such sources would imply that the stationary cumulative distribution of the input rate of the superposition should be rediscretized using two values of probability 0.5, which obviously would result in a very bad description of the cumulative distribution. So a circulant of a higher dimension should be taken. In the remainder of this section, circulants of size 25 are used.

3.2.4 Numerical examples

Consider the superposition of 30 identical MMBP sources with parameters and characteristics as mentioned in Table 3.1. Figures 3.6 and 3.7 compare the distributions of the system lengths when the D-BMAP/D/1/K queues are considered with as input D-BMAP either

- the D-BMAP $(\mathbf{S}_k)_{0 \le k \le M}$ as defined by equation (3.21), which gives an exact description of the traffic generated by the superposition, or

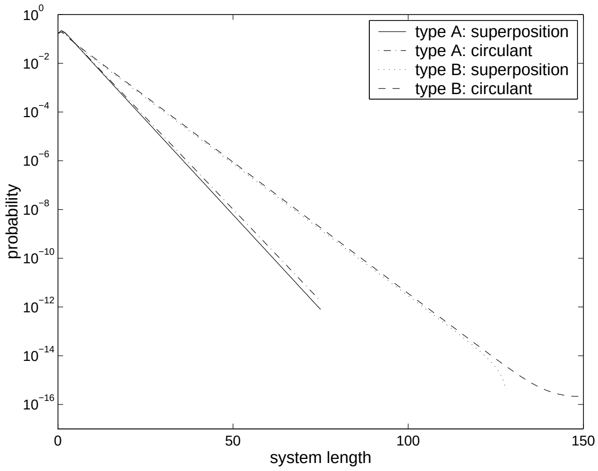

Figure 3.6: System length distribution for the exact superposition and for the circulant match of 30 type A or type B sources.

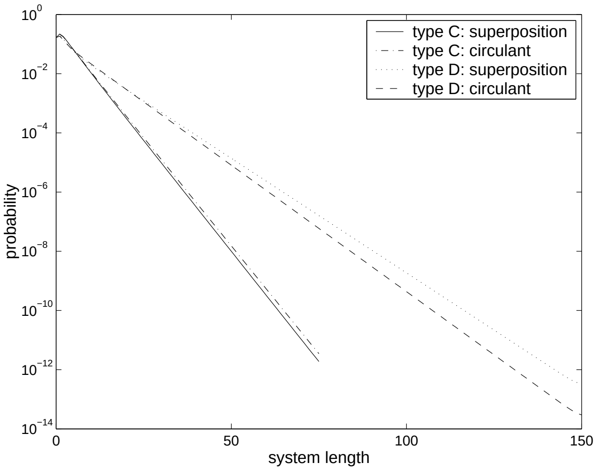

Figure 3.7: System length distribution for the exact superposition and for the circulant match of 30 type C or type D sources.

- the circulant D-BMAP constructed by the circulant matching method which approximates the superposition.

For the scenarios with sources of type A or C, a system capacity $K = 75$ is used, while for the other scenarios $K = 150$ is used. Although all types of sources were chosen to have the same mean arrival rate $\lambda = 1/36$, such that 30 sources generate in all cases a load of 83%, the traffic generated by sources of type B and D is more bursty, since when they are in the first state, these sources generate arrivals at a considerable higher average rate than the type A or type C sources ever do. So for scenarios with type B or D sources, a larger queue is needed to accommodate these bursts.

As can be seen from the Figures 3.6 and 3.7, the system length distributions obtained with the circulants as input match the system length distributions obtained when the exact superpositions are used as input rather well. The figures also illustrate something else. First remark that the input rate distribution of a source of type A is the same as that of a source of type C, and ditto for sources of type B and D. In particular: 1/30 with probability 1/3 and 1/40 with probability 2/3 for sources A and C, and 1/8 with probability 1/9 and 1/64 with probability 8/9 for sources B and D. For sources of type A and type C the system length distributions are almost the same, but for sources of type B and D there is a considerable difference (remark that Figures 3.6 and 3.7 have a different range on the Y-axis): with sources of type D the probability that the system length takes a certain value is larger for most values. Although sources of type B and D generate traffic at the same rates when they are in the first or second state, sources of type D stay for longer periods in the same state, thus also in the first state where traffic is generated at a higher rate, such that the traffic of source D is more bursty. So sources B and D are nice illustrations of the fact that when a matching method would only take first order characteristics of the

Type	α	β	p_1	mean on duration	mean off duration	mean arrival rate λ
1	1/25	1/50	1/12	25	50	1/36
2	1/25	1/200	1/4	25	200	1/36
3	1/25	1/650	3/4	25	650	1/36
4	1/75	1/150	1/12	75	150	1/36
5	1/75	1/600	1/4	75	600	1/36
6	1/75	1/1950	3/4	75	1950	1/36

Table 3.2: Parameters and characteristics of the on/off sources used in Section 3.2.4.

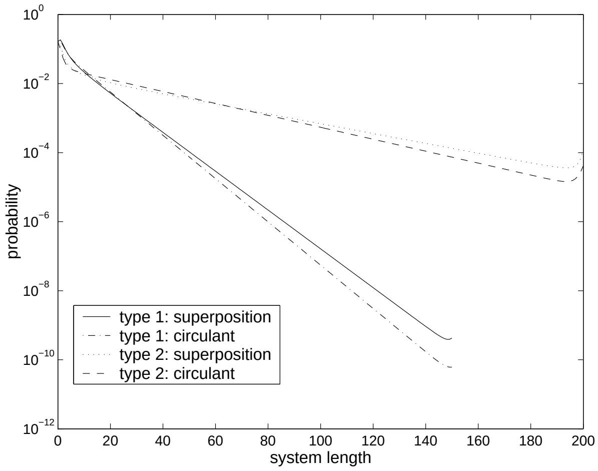

Figure 3.8: System length distribution for the exact superposition and for the circulant match of 30 type 1 or type 2 on/off sources.

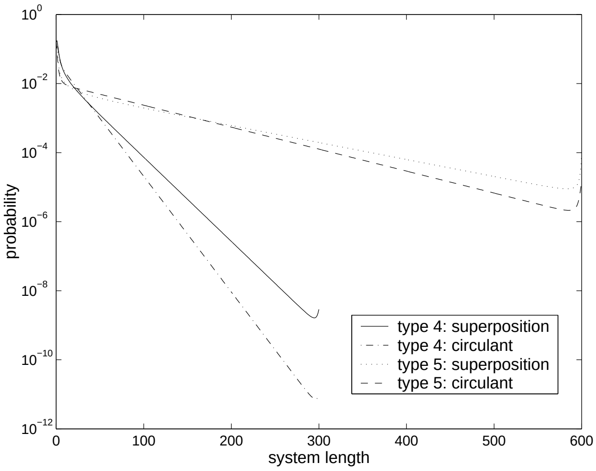

Figure 3.9: System length distribution for the exact superposition and for the circulant match of 30 type 1 or type 2 on/off sources.

input rate process into account, its result might badly reflect the queueing behavior of the sources it replaces.

Consider now the superposition of 30 identical on/off sources with parameters and characteristics as mentioned in Table 3.2. Again it is easily seen that for example a source of type 2 or a source of type 4 is more bursty than a source of type 1. Figures 3.8 and 3.9 show the system length distributions obtained with the D-BMAP/D/1/K queues, again when the input is either the exact superposition, or the circulant match of the superposition. For sources of type 1, 2, 4 and 5 respectively, the system size K is chosen equal to 150, 200, 300 and 600 respectively.

Remark that no queueing results are shown for sources of type 3 or 6. The reason is that for these types of sources, the circulant matching method does not find a solution. More in particular, no solution exists for the minimization problem (2.74) such that all conditions are fulfilled, i.e., such that all components of the input rate vector $\boldsymbol{\gamma}$ of the

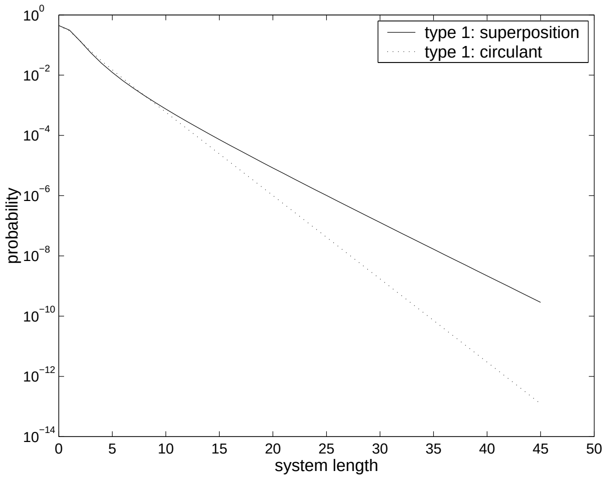

Figure 3.10: System length distribution for the exact superposition and for the circulant match of 20 type 1 on/off sources.

circulant D-BMAP are positive. An explanation for this has to be found in the fact that the input rate distribution of an on/off source takes two values: p_1 and 0, with probabilities $\beta/(\alpha+\beta)$ and $\alpha/(\alpha+\beta)$. So the input rate distribution of the superposition of M on/off sources then takes $M+1$ values, i.e., $0, p_1, 2p_1, \ldots, Mp_1$, where the probability that it takes value 0 is given by $(\alpha/(\alpha+\beta))^M$. The larger this probability, the more components of $\boldsymbol{\gamma}$ have to lie 'close to zero'. But there are less values that lie close to zero than to another value, since negative values are not allowed. All on/off sources struggle with this problem, but for sources of type 3 and 6 the probability that the input rate takes value 0 is so large, i.e., 0.32, that this causes the matching of the input rate distribution to fail. For the other types of on/off sources considered, a solution exists, but the input rate distribution of the resulting circulant is certainly no perfect match of that of the superposition, which explains why the match between the system length distributions is not very well. For a superposition of 30 sources, the matches maybe are not too bad, but when decreasing the number of sources, the matches become much worse, because then the probabilities $(\alpha/(\alpha+\beta))^M$ become larger. An example of this is shown in Figure 3.10 for 20 sources of type 1.

When considering the match of the autocorrelation sequence for the scenario with 20 type 1 on/off sources, the match can be considered as perfect: both the largest absolute and relative differences between both sequences are smaller than 10^{-14}. This illustrates that when a matching method would only take a second order characteristic (e.g., the autocorrelation sequence) into account and neglects the first order distribution, its result might badly reflect the queueing behavior of the sources it replaces. This fact is illustrated more extensively in [40]. This article explores the variations in the mean queue length when arrival processes with the same mean and autocorrelation function are applied to a ./D/1 queue. It is observed that the mean queue length can vary substantially, so the behavior of a queue cannot be predicted solely based on the mean and autocorrelation function of its arrival process.

3.3 Multiplexing MPEG video sources

In this section, the circulant matching method is applied to the superposition of an MPEG source model developed by B. Helvik in [42]. We extensively used this model in a series of connection admission control (CAC) experiments performed at the ATM[1] testbed in Basle, Switzerland. These experiments were carried out within the EXPERT project [47] of the European telecommunications research program ACTS. Because CAC experiments are in fact multiplexing experiments, a comparison between the experimentally and the theoretically obtained results is possible.

3.3.1 MPEG encoding and the MPEG model of Helvik

Due to the high bandwidth needs of uncompressed video data, several video encoding algorithms were developed to compress this data. A widely used coding scheme that is independent of a particular application is MPEG (Moving Picture Experts Group) [67]. Several MPEG schemes exist: MPEG-1, MPEG-2 and MPEG-4. The scheme that is dealt with here is MPEG-1.

The MPEG compression algorithm reduces both spatial and temporal redundancy of a video data stream, thereby generating three different frame types of a constant duration: I-frames, P-frames and B-frames. In all three frame types, spatial redundancy is removed. I-frames or intrapictures are typically the largest of the three frame types, since only intra frame encoding is used, i.e., only spatial redundancy is removed. P-frames or predicted pictures have also temporal redundancy with reference to the previous I-frame or P-frame removed. P-frames are typically the second largest. B-frames or bidirectional pictures provide the highest amount of compression since they have temporal redundancy with reference to both the previous and the next I-frame or P-frame removed. After the compression, the frames are mostly arranged in a periodic deterministic sequence, e.g., 'IBBPBBPBBPBB'. One such sequence is referred to as a group of pictures (GOP).

Following the standardization of the MPEG algorithm and its wide acceptance, researchers started investigating the characteristics of MPEG coded video traffic and developing source models specific to this type of traffic. The result is a wide variety of models (see for example [49] and the references therein). One of these models is the MPEG model of Helvik [42], which is especially designed for the type of traffic generator available at the EXPERT

[1] ATM stands for asynchronous transfer mode, and is a connection-oriented packet switching transfer mode based on asynchronous time division multiplexing. It uses fixed length packets of 53 bytes (48 bytes payload + 5 bytes header), called cells, to transport traffic. An important property of ATM is its notion of quality of service (QoS). The QoS parameter that is considered in this section is the cell loss ratio (CLR), the ratio of lost cells to the total number of transmitted cells. We come back to some aspects of ATM in the second part of this thesis, but much more information about it can be found in for example [24].

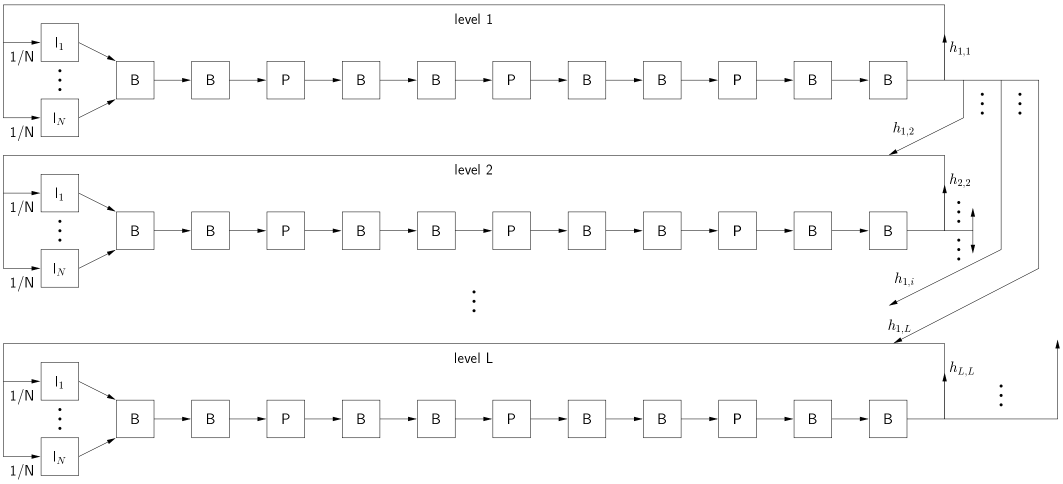

Figure 3.11: Structure of the MPEG model proposed by Helvik.

testbed.

The Helvik model is a periodic Markovian model at the MPEG frame level. Its transition diagram is shown in Figure 3.11. State sojourn times are deterministic, with as length a frame duration. With each state a (mean) load is associated. As can be seen from Figure 3.11, the Helvik model is level-oriented. A level i models the activity of the MPEG source when the sum of the loads generated by the B-frames and P-frames of a GOP is between two values l_i and l_{i+1}. Because of the variation in the loads produced by the different frames in a GOP, a smooth transition over the frames cannot be assumed. Therefore, within each level of the model the I-frame, B-frame and P-frame activities are modeled. For the B-frames and P-frames a single state is used. As the I-frames are the largest, they are modeled in more detail using multiple states. For further details about the Helvik model, we refer to [42].

The parameters of the Helvik model, i.e., the different load level intervals $[l_i, l_{i+1}[$, the transition matrix $\mathbf{H} = (h_{i,j})$, where $h_{i,j}$ describes the probability of going from level i to level j, and the load that is associated with each state, are obtained from MPEG frame size data. Details about how this is done can also be found in [42]. First one has to decide how many levels L and how many I-states N per level to use. This decision is a trade-off between the model accuracy and the number of states 'budget'. When M denotes the number of frames in a GOP, the number of states in the Helvik model is given by $(M - 1 + N)L$. The two Helvik sources that are used further on are based on frame size trace data of the James Bond movie 'Goldfinger' (referred to as 'bond'), and on a trace of an Asterix cartoon (referred to as 'asterix'). These are two of the many MPEG-1 frame size traces made publicly available by the University of Würzburg at http://nero.informatik.uni-wuerzburg.de/MPEG/. The GOP pattern of these traces is 'IBBPBBPBBPBB', such that the parameter M of the Helvik model equals 12. Each trace consists of 40 000 frames, which corresponds to approximately half an hour of video. The duration of a frame is 45 ms[2]. The Helvik source bond is implemented with five load levels, and two I-states per level, such that it is a 65 state model. The asterix source is also implemented with two I-states per level, but now four load levels are used, such that it has 52 states. With this number of states, we were able to multiplex both models in one traffic generator, without exceeding the upper limits of what the equipment can handle.

3.3.2 Experimentally obtained CAC boundaries

Connection admission control (CAC) is the traffic control function which has to determine whether a new connection setup request can be accepted or should be rejected. This decision is based on the constraint to meet the negotiated quality of service (QoS) requirements of all existing connections as well as that of the new connection. Besides this basic function of CAC, there is the secondary goal to maximize the system utilization by allowing for a statistical multiplexing gain, i.e., an efficient CAC method should accept as many connections as possible without violating any QoS guarantees.

Experimental multiplexing results were obtained in the EXPERT project by using a traffic generator and analyzer instrument, called ATM-100, which gives the possibility to generate and analyze quite general random traffic. The ATM-100 is equipped with a traffic generator module that is used for generating the artificial MPEG traffic. The periodicity of the traffic is compromised in the sense that the duration in the individual states of the Helvik model is assumed exponential instead of constant, which is a requirement if more than one MPEG source is to be generated by the traffic generator. This traffic is then multiplexed on the output port of a Fore ASX-200 ATM switch with a buffer of 100 cells, or on an output port of a Cisco LS1010 ATM switch with a buffer of 256 cells. Due to hardware constraints in the traffic generator, a pacing function has been used to limit the output port capacity to 37.44 Mbit/s, thereby reducing the number of sources required to adequately load the system. The aggregate traffic stream is then analyzed in the ATM-100 analyzer module, which permits cell loss measurements. CAC boundaries are obtained from these multiplexing experiments by changing the traffic mix until a cell loss ratio (CLR) below, but as close as possible to a fixed value is obtained. All CAC boundaries were obtained with a target CLR of 10^{-4}. More details about the experimental setup are given in [1, 2].

[2] Remark that originally on the website where the MPEG traces are made available, it was mentioned that “the capture rate of the video system was between 19 and 25 frames per second”. Since it was not clear what the exact capture rate for each of the traces was, Helvik used in [42] the average of these two numbers, i.e., 22 frames per second, resulting in a frame time of 45 ms. In the experiments, we adopted this number. Later on, this indistinctness was clarified, and now it is mentioned that “the capture rate of the video system was 25 frames per second”, so resulting in a frame time of 40 ms. Since all experimental results were obtained assuming that a frame time has a duration of 45 ms, also the theoretical results are generated based on this assumption.

3.3.3 Theoretically obtained CAC boundaries

The Markovian MPEG model of Helvik is mapped onto a D-BMAP $(\mathbf{D}_k)_{k\geq 0}$ in a rather straightforward way. The transition matrix $\mathbf{D}$ is easily read from the transition diagram shown in Figure 3.11 and from the values in the matrix $\mathbf{H}$, which describes the transition probabilities among the different levels of the Helvik model. The transition matrix of the D-BMAP has period 12, due to the periodic GOP structure in the MPEG traces and in the Helvik model. For each state, the Helvik model gives the number of bits that should be generated during the time that the model is in that state. First this number is transformed from bits into cells ($\lceil$number of bits / $(8 \times 48)\rceil$). When for a certain state i this results in y_i cells, then $\forall k \geq 0$ and $\forall j$, $(\mathbf{D}_k)_{ij}$ is defined as

$$(\mathbf{D}_k)_{ij} = \begin{cases} \mathbf{D}_{ij} & \text{if } k = y_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.22}$$

Based on the parameters of the Helvik sources, a D-BMAP for the bond and the asterix sources can thus be constructed.

To obtain results comparable with the experimental results, a superposition of these D-BMAPs should be offered to a single server queueing system. This system should have a finite buffer capacity of 100 or 256 cells and a deterministic service time equal to the time needed to place one cell on a link of 37.44 Mbit/s (call this time a *slot*).

Because of the size of the bond and asterix D-BMAPs, i.e., 65 or 52 states, it is obviously that the exact superposition of these sources cannot be used (a superposition of two of such sources has already 4225 resp. 2704 states). So the circulant matching method is applied to these sources, resulting in a circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$. For a superposition of asterix or bond D-BMAPs, the resulting circulant has 132 states. For a superposition of bond and asterix sources, the result is a circulant of dimension 276. These dimensions stay the same irrespective of the number of sources that is multiplexed, the differences are in the rate vector $\boldsymbol{\gamma}$ of the circulant D-BMAPs.

Since the Helvik model is a model at the MPEG frame level, the underlying time unit of the circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ is also a frame time, or 45 ms. Because the circulant D-BMAP will be used as input to a queueing system with a constant service time of one slot, it has to be transformed into a D-BMAP with one slot as underlying time unit. Suppose that the state sojourn time of this new D-BMAP is geometrically distributed with mean x, where x is the number of slots in a frame time. Then $p = 1 - 1/x$ is the probability of staying in the same state after one slot. Transform the circulant D-BMAP $(\mathbf{Q}_k)_{k\geq 0}$ into the circulant D-BMAP $(\mathbf{R}_k)_{k\geq 0}$ with one slot as underlying time unit, by defining the elements of its transition matrix $\mathbf{R}$ as

$$\mathbf{R}_{ij} = \begin{cases} (1-p)\mathbf{Q}_{ij} & \text{if } i \neq j, \\ p & \text{if } i = j. \end{cases} \tag{3.23}$$

Remark that the periodicity of the input traffic is in this way compromised in a similar way as in the experiments, and that the matrix $\mathbf{R}$ is stochastic, since for all i, $\mathbf{Q}_{ii} = 0$, because

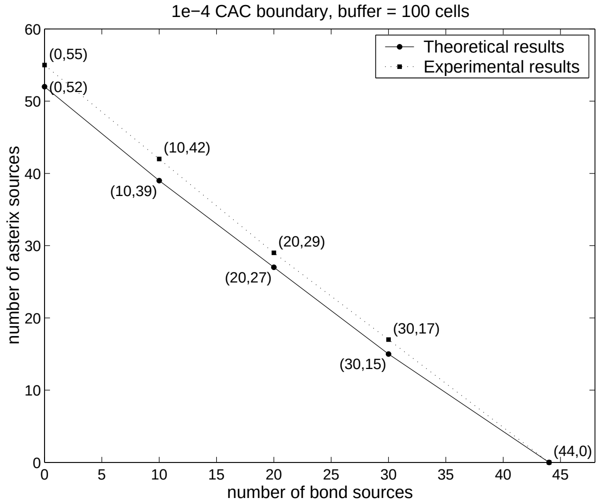

Figure 3.12: Comparison of theoretically and experimentally obtained 10^{-4} CAC boundary with a buffer capacity of 100 cells.

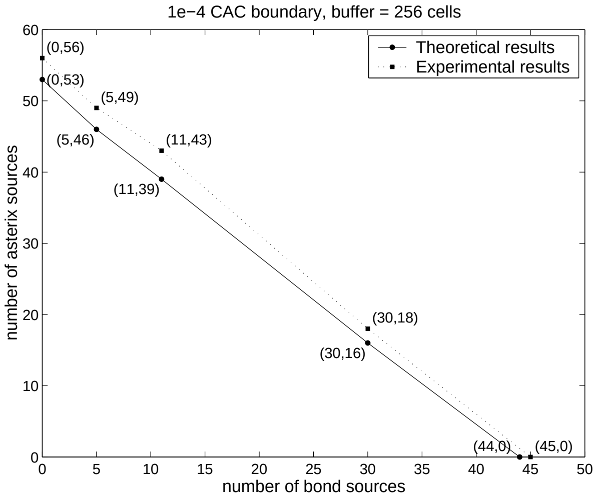

Figure 3.13: Comparison of theoretically and experimentally obtained 10^{-4} CAC boundary with a buffer capacity of 256 cells.

$\mathbf{Q}$ is periodic. Construct a vector $\hat{\boldsymbol{\gamma}}$ from the rate vector $\boldsymbol{\gamma} = \sum_{k=1}^{\infty} k\mathbf{Q}_k\mathbf{e}$ by dividing all its elements by x, i.e., $\hat{\gamma}_i = \gamma_i/x$. The matrices $\mathbf{R}_k$ are then obtained from $\mathbf{R}$ and $\hat{\boldsymbol{\gamma}}$ as in (2.31).

By using the D-BMAP $(\mathbf{R}_k)_{k\geq 0}$ as input for the D-BMAP/D/1/$K+1$ queueing system, where $K = 100$ or $K = 256$, and calculating the cell loss probability for this system using formula (1.13), theoretical CAC boundaries can be obtained in a similar way as the experimental CAC boundaries, i.e., by changing the traffic mix until a CLR below, but as close as possible to 10^{-4} is obtained.

3.3.4 Numerical results

All results presented here are obtained with the values already mentioned before: a buffer capacity of 100 or 256 cells, an outgoing link of capacity 37.44 Mbit/s, which implies that one slot equals 11.325 μs, and a target CLR of 10^{-4}. Figure 3.12[3] shows results for the buffer of 100 cells, Figure 3.13 for that of 256 cells. If the theoretically and experimentally

[3] Remark that this figure is also shown in [89], but that the theoretically obtained results shown now are slightly better than these shown in [89]. The reason is that when we generated the results of [89], the minimization problem (2.74) of the circulant matching method was implemented using the MATLAB function *fmins*, which uses the Nelder-Meade simplex method [78]. Because *fmins* implements unconstrained minimization, while we require that all components of the rate vector are positive, we adapted this method as suggested in [78] for constrained optimization: we let the goal function take a large positive value when a component of the rate vector is negative. Later we had access to the MATLAB optimization toolbox, in which the function *fmincon*, which finds the constrained minimum of a function of several variables, is available. It is this function that is used to generate the results shown in Figure 3.12.

obtained points shown in the figures are compared, it is seen that the theoretical results are more conservative than the experimental results, with a larger deviation if the number of asterix sources grows. For the D-BMAPs of the MPEG sources, the parameters as obtained from the Helvik model are used, which gives rise to a mean arrival rate of 58.1812 cells/45 ms, or 0.54820 Mbit/s for the asterix source, and 63.3247 cells/45 ms or 0.59666 Mbit/s for the bond source. If the Helvik sources are implemented in the traffic generator however, their parameters are automatically slightly changed to adapt them to the hardware limitations of this device. Depending on the number of sources generated, these changes may become more important. The first limitation is that only transition probability values in integer multiples of 1/256 are allowed. Secondly, the peak rate generated in a state of the model must divide the link rate, such that in a state the interarrival time between cells is always the same integer number of slots. As a result, the mean arrival rate for an experimental asterix source is 0.51318 Mbit/s, and 0.59221 Mbit/s for a bond source. The experimental model for the asterix source thus generates 0.03502 Mbit/s less than the theoretical model, which means that for a certain experimental point the corresponding theoretical CLR is worse, depending on the number of asterix sources used. This explains partially why the theoretical CAC boundary lies below the experimental one, with a larger difference when more asterix sources are involved. Analogue observations are made in [2] when the experimental results are compared with results obtained by simulation.

3.4 Conclusions

Numerical examples and applications of the circulant matching method were described in this chapter. A first application discussed in Section 3.2 is the superposition of M identical two dimensional MMBP sources. For these types of sources, it is possible to compare the system lengths obtained when using the circulant approximation of the superposition and the exact superposition as input to a queueing system, because the exact superposition of M identical two dimensional sources is also exactly described by an $(M+1)$-dimensional Markov source. First general MMBP sources are considered, and the system length distribution obtained with a circulant as input matched the exact system length distribution rather well. Then a special type of MMBP sources is considered, namely on/off sources. For these type of sources the agreement between the system length distribution obtained with the circulants as input and the exact distribution is bad. The reason is that the rate distribution of the circulant very badly matches that of the exact superposition, because a large part of the probability mass of the rate distribution is located at rate zero. The same fact sometimes even causes the circulant matching method to fail in finding a valid rate distribution for the circulant. Using the two dimensional sources it is also illustrated that it is necessary for a matching method to take both first and second order statistics of the arrival process into account, since when considering only one of both, the result of the matching process might badly reflect the queueing behavior of the sources it replaces.

A second application, considered in Section 3.3, is the superposition of a periodic MPEG source model. Using the circulant matching method, we obtained a theoretical CAC boundary for a mix of two types of MPEG sources. Remark that due to the dimension of the MPEG source models (52 and 65 states) and the realistic number of such sources considered, it is impossible to obtain the exact queueing results using the exact superposition. So we compared the theoretically obtained results with experimentally obtained results.

Part II

Frame aware buffer acceptance schemes

Chapter 4

Introduction

In the late 1980s, the asynchronous transfer mode (ATM) [24, 80] was developed in order to provide a network that was capable of handling a virtually unlimited range of user applications independent of their bandwidth requirements. The major organizations responsible for developing standards and specifications for ATM are the ITU-T (International Telecommunications Union-Telecommunication Standardization Sector) and the ATM Forum. ATM is a fast connection-oriented packet switching transfer mode based on asynchronous time division multiplexing. It uses fixed length packets of 53 bytes (48 bytes payload + 5 bytes header), called cells, to transport data. Based on some information in the header of each cell, cells belonging to the same virtual channel (VC) can be identified. Cell sequence integrity is preserved per VC.

When ATM came on the scene, it was thought to be the beginning of a new era in networking, because it was both a local area network and a wide area network technology that could start at the desktop. In addition, ATM's ability to provide end-to-end quality of service (QoS) was highly praised. However, ATM never became the magic end-to-end solution. But it has been successfully deployed in the backbone network, because of its ability to provide QoS. The ATM framework for providing QoS guarantees is described in the ATM Forum's Traffic Management Specification [5]. Different ATM service categories and traffic control functions which relate traffic characteristics and QoS requirements to network behavior have been defined.

Four service categories are intended for non-real-time data traffic: non-real-time variable bit rate (nrt-VBR), available bit rate (ABR), unspecified bit rate (UBR) and guaranteed frame rate (GFR). Data applications using the most widely used protocol suite in computer communications, i.e., TCP/IP (transport control protocol/internet protocol), increase their rate if extra bandwidth is available in the network, and reduce it if congestion builds up. As a result, this type of traffic may be highly unpredictable, extremely bursty and very hard to characterize in terms of a peak cell rate, sustainable cell rate and maximum burst size, as is needed to set up a nrt-VBR connection. An additional weakness of nrt-VBR in the context of transporting TCP/IP traffic is its inability to define a QoS guarantee in terms of frames[1]. The ABR service category was developed especially for traffic sources that are willing to adapt their rate to changing network conditions and available resources, but can only characterize their traffic in a rather 'vague' way. ABR uses a feedback flow control scheme to provide information about congestion inside the network to the sources, and it expects sources to adapt their traffic in accordance to this feedback. This feedback algorithm is however fairly complex, especially in the endsystems. Further, when ATM is not deployed end-to-end, ATM's traffic control terminates at the access nodes and it becomes very difficult to explicitely control a non-ATM source. So the most suited service categories for TCP/IP traffic are UBR and GFR.

The best way to characterize the UBR service category is as ATM's 'best effort' service category: UBR is not subject to a specific traffic contract, so no specification of the traffic that will be sent over a UBR connection is needed, but also no QoS commitments are made to UBR connections. To perform end-to-end congestion control, UBR depends entirely on a higher layer protocol such as TCP.

Where UBR was developed as a way to accommodate traffic that is difficult to characterize to the early ATM market, GFR, which was initially called UBR+, was developed especially for this kind of packet data (i.e., TCP/IP traffic). The main motivation behind the introduction of GFR was to retain the simplicity of UBR at the user network interface, while providing GFR connections with a minimum cell rate guarantee at the frame level: if frames smaller than a specified maximum frame size are sent in a burst of cells that does not exceed a maximum burst size, then these frames are expected to get delivered across the network with minimum losses.

The absence of congestion control mechanisms for the basic UBR service can lead to a low throughput for this type of connections. As a result, competitive UBR implementations enhance the basic UBR service with intelligent frame aware buffer acceptance schemes. For GFR, it is explicitely required in the definition of the GFR service category that this type of traffic is transmitted as frames of cells, and that the ATM switches supporting GFR need to be frame aware and accept or discard entire frames instead of individual cells. Frame aware buffer acceptance schemes, also often called packet discarding mechanisms, are the topic of the second part of this thesis.

A literature overview of the most important frame aware buffer acceptance schemes proposed for UBR and GFR is given in the next chapter. In Chapter 6, a theoretical model is developed and applied to study the transient performance of the selective drop buffer acceptance algorithm. This model is slightly modified in Chapter 7 to study the fair buffer allocation acceptance scheme. The remainder of the current chapter contains short introductory descriptions on AAL5 frames, TCP congestion control, the UBR and GFR service guarantees and on performance measures that are important to assess the performance of TCP over UBR or GFR.

[1] The term 'frame' means an AAL5 frame, and is discussed further on in this chapter. Roughly spoken, it corresponds to an IP packet which holds a TCP segment.

4.1 Some concepts related to buffer acceptance

A buffer acceptance scheme decides about which cells are allowed to enter the buffer of a network element, and which cells have to be dropped. This decision is very often taken based on buffer accounting information, i.e., on the counters and states associated with the buffer.

Together with the scheduling algorithm, the buffer acceptance scheme determines the throughput and fairness guarantees a network element can offer to the different virtual circuits. The scheduling algorithm is the algorithm that decides about the order in which the accepted cells will leave the buffer.

Closely related to scheduling is the queueing strategy used, i.e., the internal organization of the buffer. The queueing strategy can be a global one, most of the time resulting in FIFO scheduling, or it can be per-class or per-VC, which makes scheduling schemes like round robin, priority scheduling etc. possible. Important to note is that the accounting strategy used does not imply a queueing strategy: many of the schemes which are considered further on use per-VC accounting combined with global queueing.

Although strictly speaking the term 'buffer acceptance scheme' as defined above covers only the decision rules about which cells to accept in the buffer, it is also often used to denote the totality of buffer acceptance (in the strict sense), accounting, queueing and scheduling. Throughout this thesis, the term is also used in both meanings. Sometimes, buffer acceptance is also called buffer management (e.g., in [32, 5]).

4.2 AAL5 aware buffer acceptance

The most widely used ATM adaptation layer (AAL) for data traffic is AAL5. The GFR service guarantee is even explicitely based on the use of AAL5. AAL5 provides to the upper layer protocols an unassured transfer of variable-sized service data units (SDU) over the underlying ATM network. Each such variable-sized SDU is encapsulated in an AAL5 frame which consists of a payload field of up to 65 535 bytes, some padding bytes and a 8-byte long trailer (see Figure 4.1). The padding aligns the AAL5 frame on a multiple of 48 bytes. The segmentation of the AAL5 frame in cells by the AAL5 segmentation and reassembly (SAR) sublayer does not introduce any new overhead, but relies on the payload type indicator (PTI) field in the ATM header. The ATM user-to-user (AUU) bit in the PTI field for user data cells is set to zero by the SAR sublayer for all cells, except for the last cell of each AAL5 frame, which is transmitted with the AUU bit set to one. Buffer acceptance schemes can thus detect frame boundaries by inspecting the AUU bit in the header of the ATM cells.

Buffer acceptance schemes in ATM networks decide in principle about the acceptance of cells. But with AAL5, buffer acceptance schemes preferentially are AAL5 frame aware, because the destination AAL5 entity checks each reassembled AAL5 frame for message

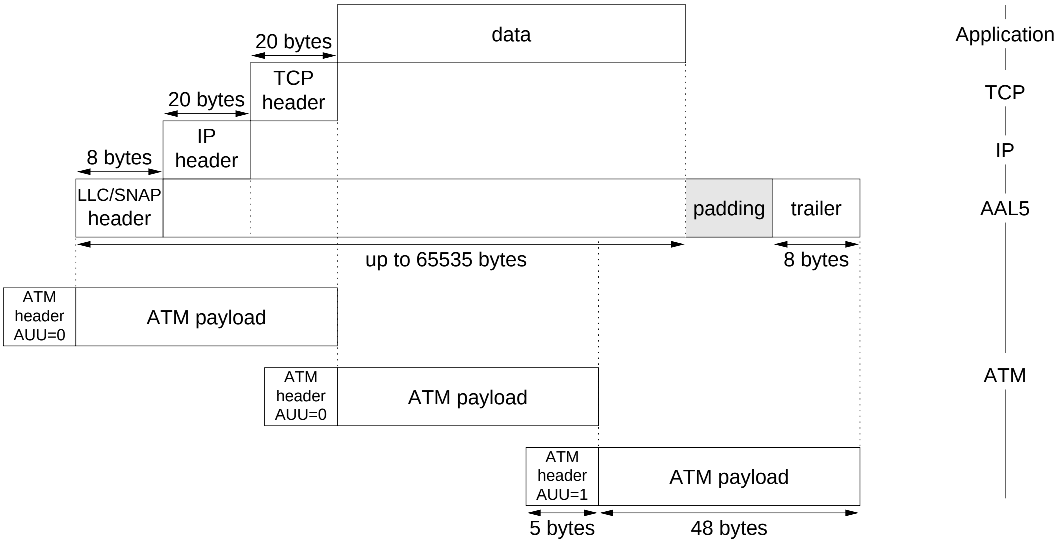

Figure 4.1: Encapsulation of data in AAL5 and AAL5 segmentation into ATM cells.

length and cyclic redundancy check field, and discards corrupted frames. The loss of a single cell of an AAL5 frame at a network element thus leads to the loss of a whole frame at the destination. Buffer acceptance schemes without frame awareness consequently give rise to a flow of ATM cells that is very likely to transport incomplete frames which are of no use. This can degrade the data throughput significantly. To support GFR, buffer acceptance schemes are required to be AAL5 aware, since the GFR service guarantee is based on AAL5 frames.

4.3 TCP congestion control

If the UBR or GFR service category is used to transport TCP/IP traffic, network elements perform congestion control (i.e., packet discarding) based on local information. For end-to-end congestion control, these service categories depend entirely on TCP.

The congestion control of TCP is window-based. TCP's window size corresponds to the amount of data the TCP source can send in one round trip time (RTT), and is the minimum of the receiver's advertised window (RCVWND) and the sender's congestion window (CWND).

The congestion control scheme of TCP includes four algorithms, i.e., 'slow start', 'congestion avoidance', 'fast retransmit' and 'fast recovery' [50, 3]. The slow start and congestion avoidance algorithms control TCP's window size. A variable SSTRESH is maintained for each connection to switch between the two algorithms. When a TCP connection starts or has been idle for a time longer than the retransmission timeout, the slow start mechanism is used. At the beginning of the slow start algorithm, CWND is set to 1 maximum segment size (MSS). Each time an acknowledgement (ACK) for new data is received, CWND is increased by 1 MSS. If CWND reaches SSTRESH, the congestion avoidance algorithm takes over, and now CWND is increased by 1/CWND on receipt of a new ACK. Slow start corresponds with an exponential increase of the congestion window every round trip time, congestion avoidance with a linear increase.

TCP's congestion control relies on segment loss as the indication of congestion. On detection of a segment loss by expiration of the retransmission timer, half the current window size is recorded in SSTRESH, CWND is set to 1 MSS and slow start is initiated. The triggering of the retransmission timer is affected by the TCP timer granularity. Most real TCP implementations use a 100 to 500 ms timer granularity, although some simulations use a much lower granularity (e.g., 0.1 ms in [84]). Since the timer granularity determines the amount of time lost during congestion, lowering the TCP granularity results in faster recovery after a loss and thus a higher throughput.

Since a TCP receiver should send an immediate duplicate ACK when an out-of-order segment arrives, the sender can also detect losses based on incoming duplicate ACKs. After receiving three duplicate ACKs, the fast retransmit algorithm sets SSTRESH to half the current window size, and retransmits the segment that appears to be the missing one without waiting for the retransmission timer to expire. If the fast retransmit algorithm is implemented in combination with the fast recovery algorithm, CWND is set to SSTRESH plus 3$*$MSS. Otherwise, CWND is set to 1 MSS and slow start is initiated.

The fast recovery algorithm governs the transmission of new data until a non-duplicate ACK arrives: it increments CWND by 1 MSS for each additional duplicate ACK received, and transmits a segment if allowed by TCP's window size. When an ACK for new data arrives, CWND is set to SSTRESH, which implies that the congestion avoidance algorithm is triggered. The reason for not performing slow start is that the receipt of the duplicate ACKs does not only indicate that a segment has been lost, but also that segments are most likely arriving at the destination.

The two most common reference implementations for TCP are Tahoe TCP and Reno TCP [27]. Tahoe TCP refers to TCP with the slow start, congestion avoidance and fast retransmit algorithms implemented, while Reno TCP implements also the fast recovery algorithm.

Since the fast retransmit and recovery algorithms are known to generally not recover very efficiently from multiple losses in a single window of packets [27], the selective acknowledgement (SACK) strategy was proposed in [74]. With selective acknowledgements, the data receiver can inform the sender about all segments that have arrived successfully, so the sender needs to retransmit only the segments that have actually been lost.

4.4 Buffer acceptance and TCP congestion control

The rate at which a TCP source can send data into the network depends on its window size. If the network would have direct control over this window size, it could control the source's rate. The network however does not have this direct control. But since TCP's congestion control scheme manipulates the window size by increasing it while there are no losses, and decreasing it on detection of a lost TCP segment, the network could have indirect control over a source's rate by means of dropping.

This dropping occurs automatically in case of congestion because of buffer overflow, but can lead to very low effective throughput. Dropping can however also be done in a more intelligent way, by trying to drop complete frames prior to congestion, preferentially from a connection which is using more bandwidth than one would call fair. To determine which connections are getting more than a fair share of the bandwidth, the number of cells each connection has in the buffer is taken into account, and the principle is used that connections which use more than a fair share of the buffer capacity will also get more than a fair share of the bandwidth. Connections from which cells are dropped will decrease their rate because of the TCP congestion control mechanism. As a result, these connections will probably have the fewest cells in the buffer next time the cell dropping condition is satisfied, and their frames have the least chance of being discarded. So it is unlikely that in buffer acceptance schemes which try to drop frames in a 'fair' manner, frames from the same VC get discarded all the time.

4.5 The UBR and GFR service guarantees

The UBR service guarantee as defined in [5] is simple to describe: UBR offers *no* traffic related service guarantee. No commitment is made about the cell loss ratio experienced by a UBR connection, or about the cell transfer delay experienced by cells on the connection. Fairness among connections cannot be assumed, although a local policy in some network elements may have this effect.

The description of the GFR service guarantee is not so easy. It is explicitely based on AAL5 frames. Before it can be formulated, some parameters and terms need to be introduced.

For a GFR connection, a traffic contract is specified that is composed of the following parameters: a minimum cell rate (MCR) and associated cell delay variation tolerance $\mathrm{CDVT}_{\mathrm{MCR}}$, a peak cell rate (PCR) and associated cell delay variation tolerance $\mathrm{CDVT}_{\mathrm{PCR}}$, a maximum frame size (MFS) and a maximum burst size (MBS). The MFS is the maximum AAL5 frame size in cells.

A cell with its cell loss priority (CLP) bit set to one is called *marked* when the originator of the cell has set the CLP bit. When it is the network that has set the CLP bit, the cell is called *tagged*. Any source or network element that sets the CLP bit of a cell to one shall set the CLP bit of every other cell of the same frame to one as well, since no partial frame marking or tagging is allowed by the GFR definition. There are two types of GFR connections: GFR.1 and GFR.2 connections. In either type of GFR connection, less important frames may be marked by the source. Tagging by the network is however only allowed for GFR.2 connections. Networks may only tag frames that are ineligible. A frame is *eligible* if and only if it is conforming, and it passes the F-GCRA test.

A frame is *conforming* if all its cells are conforming. The cell conformance is checked by the GFR usage parameter control (UPC), which verifies if

- the cell is either the last cell of a frame, or no more than MFS-1 cells of the same frame have preceded it, i.e., the frame length is limited to MFS cells,
- the end systems send traffic at a cell rate that conforms to PCR and $\mathrm{CDVT}_{\mathrm{PCR}}$, i.e., the cell does not violate PCR,
- the cell has the same CLP value as the first cell of the frame to which it belongs, i.e., CLP should be set uniformly in a frame.

The UPC discards or tags (if allowed) cells of non-conforming frames. Because the three tests performed by the UPC are applied on cell level, the UPC is unable to predict the conformance of succeeding cells when the first cell of a frame is received. Since no partial tagging is allowed, the tails of non-conforming frames are therefore usually discarded.

To be eligible, a frame must additionally pass the frame-based generic cell rate algorithm F-GCRA(T, L) [5, p.72], which is the reference algorithm used to identify the QoS eligibility of a frame with respect to the minimum cell rate MCR = $1/T$, assuming that a tolerance $L = (\mathrm{MBS} - 1)(1/\mathrm{MCR} - 1/\mathrm{PCR})^{-1}$, is allowed. The F-GCRA is an adaptation of the well-known GCRA used with the VBR service category [5, p.31]. The main difference between the GCRA and the F-GCRA is that the F-GCRA declares entire CLP=0 frames to be eligible or non-eligible. Reasons for frames to fail the F-GCRA test are that the frames are CLP=1 frames, the frame interarrival times are too small, or traffic was sent at PCR for longer than the MBS. Because CLP=1 frames cannot pass the F-GCRA test, all CLP=1 frames are ineligible. A classification of the frames of a GFR connection in terms of marking/tagging, conformance and eligibility is shown in Table 4.1.

The GFR service guarantee provides a low cell loss ratio (CLR) for a number of cells in complete CLP=0 frames, at least equal to the number of cells in eligible frames. Since CLP=1 frames are not subject to the CLR objective, buffer acceptance schemes in network elements will treat them with lower priority. Note that since the GFR service guarantee is with respect to a number of cells in complete frames, and not precisely to the frames that are considered eligible, the network is not required to perform the F-GCRA test, although some switch elements may rely on it to satisfy the GFR service guarantee.

Cells may always be sent at a rate up to the PCR. Apart from the MCR guarantee, the GFR service also includes the expectation that traffic in excess of MCR and MBS will be delivered within the limits of available resources, and that each connection will be provided

CLP frame	conforming frame	frame passes F-GCRA	type of frame
0	no	no	ineligible nonconforming
0	no	yes	ineligible nonconforming
0	yes	no	ineligible conforming
0	yes	yes	*eligible conforming*
1	no	no	ineligible nonconforming
1	yes	no	ineligible conforming

Table 4.1: Classification of GFR frames.

with a fair share of those available resources. So buffer acceptance schemes for GFR need to be designed in such a way that they can deliver both rate and fairness guarantees to the GFR connections.

Because the GFR service guarantee applies to complete AAL5 frames, buffer acceptance schemes used with GFR decide about the acceptance of a frame on arrival of its first cell: if this first cell is accepted, they try to accept all cells of the frame; if the first cell is discarded, all cells of the frame are discarded. So all these schemes need at least one per-VC state to indicate if the next cell on the connection will be the first one of a frame.

4.6 Performance measures

To decide about the performance of TCP over UBR or GFR using a certain buffer acceptance scheme, the throughput obtained by the different connections at the destination TCP layer is measured. Throughput is defined as the number of bytes delivered to the destination application divided by the time needed to deliver these bytes. This measure is also called goodput or effective throughput, as it is the throughput that is 'good' or 'effective' in terms of the higher layer protocol. If the sources are not persistent (i.e., they have only a limited amount of data to send), the total time needed by each source to deliver all its data to the destination application is converted into a throughput value by dividing the amount of data by the measured time. Two important performance measures are obtained from the throughput values: efficiency and fairness.

4.6.1 Efficiency

The efficiency of TCP over UBR or GFR is defined as:

$$\text{efficiency} = \frac{\text{sum of TCP throughputs}}{\text{maximum possible TCP throughput}}, \tag{4.1}$$

where the maximum possible TCP throughput is the throughput attainable by the TCP layer on a link. This throughput is lower than the link capacity because of header, trailer and padding overhead added to the data by different layers (see Figure 4.1). Considering for example a TCP MSS of 512 bytes, the maximum possible throughput is approximately 125.2 Mbps on a 155.52 Mbps link [36].

4.6.2 Fairness

To decide about the fairness of a certain buffer acceptance scheme, a fairness criterion is needed. Several example fairness criteria, such as equal allocation, weighted allocation, MCR plus equal share, allocation proportional to MCR, etc. are given in [5]. Of course, fairness criteria based on MCRs are only applicable to GFR connections. The most used criterion for UBR traffic is the equal allocation.

Once a fairness criterion is defined, the distance between the resource allocation and the desired goal needs to be assessed. In the case of equal allocation, this is often done visually, by plotting the effective throughput of the different connections versus time (e.g., in [28, 69, 25]). Another way to judge the fairness is by making use of a fairness index. Several fairness indices are used in the literature (e.g., indices based on the coefficient of variation of the goodputs [25, 85]). The following one, which is defined by the ATM-Forum in [4] and used in e.g., [36, 35, 55], is used in Chapter 6.

If the goodputs of N virtual circuits are found to be $\{T_1, \dots, T_N\}$, where the ideal goodputs according to the chosen fairness criterion should be $\{\hat{T}_1, \dots, \hat{T}_N\}$, then

$$\text{fairness index} = \left(\sum_{i=1}^{N} x_i\right)^2 \Bigg/ \left(N \sum_{i=1}^{N} {x_i}^2\right), \tag{4.2}$$

where $x_i = T_i/\hat{T}_i$ is the relative goodput allocation of connection i. This fairness index ranges between zero (minimum fairness) and one (maximum fairness) and can be given the following interpretation: if $N - k$ of the N x_i's are zero, while the remaining k x_i's are equal and non-zero, the fairness index will be k/N, or the fraction of users favored. More properties of this fairness index can be found in [53, 52].

Chapter 5

An overview of buffer acceptance schemes for UBR and GFR

A literature overview of the most representative frame aware buffer acceptance schemes defined for use with the UBR or GFR ATM service categories is given in this chapter. The overview is kept rather descriptive, but in [91] pseudo code of all schemes that are discussed, using a uniform notation and level of detail, can be found. Also the general conclusions of performance evaluation studies of the schemes by simulation or by experiments are summarized, without going into details concerning the simulation/experimental configuration, exact parameter settings and TCP implementations used in the various studies. These details and the detailed results can be found in the references this chapter is based on. At the end of this chapter, a summary of the queueing and scheduling strategy used in each scheme, and of the accounting information that needs to be kept, is given (see Table 5.1 for the UBR schemes, and Table 5.2 for the GFR schemes).

5.1 Buffer acceptance schemes for UBR

5.1.1 Some of the first buffer acceptance schemes for UBR

In this section three of the very first buffer acceptance schemes defined for UBR are considered: tail drop, partial packet discard and early packet discard. Although tail drop is not frame aware, and its performance is not satisfactory at all, the scheme is included here since it illustrates the problems which need to be resolved by 'better' buffer acceptance schemes. Partial and early packet discard are important schemes since they are widely implemented in commercial ATM switches, and the principles behind these schemes keep cropping up in almost all of the more sophisticated schemes discussed in the following sections.

Tail drop

The simplest buffer acceptance scheme for use with UBR is called tail drop (TD). This scheme accepts cells into a global buffer as long as the buffer is not full. Upon buffer overflow, cells are dropped.

Applying tail drop while providing sufficiently large buffers to minimize cell loss, and letting the higher layer protocol (i.e., TCP) handle recovery after an occasional loss, was the initial idea about how to deal with the absence of congestion control for UBR. However, it has been shown by simulations [84, 36] that this approach can give low efficiency results. Main reasons for this are:

- Delivery of useless cells. Dropping a single cell at a network element results in the discarding at the receiver of all other cells of the same frame. Although these cells are thus useless, they are still transmitted over the network, resulting in lower goodput. Furthermore, these useless cells consume bandwidth and buffer space, such that in times of congestion they may cause other frames to lose also some cells. This problem becomes worse with smaller buffers, larger frames, increased number of active connections and increased TCP window sizes [84].
- Link idle time due to TCP synchronization effects. Because cells from several connections usually arrive interleaved at a switch, cells from all sources are dropped when the dropping condition is satisfied, i.e., the buffer is full. As a result, all sources timeout and go through slow start at roughly the same time. This is called TCP synchronization. While the sources wait for a timeout, they stop sending data into the network. So occasionally, the congested link can be idle.

Although TCP synchronization is an important factor that affects TCP's performance, it is not a significant problem in the scenarios explored in [84]. This is because in the simulations this paper reports on, the TCP timer granularity was changed to 0.1 ms, which is much lower than the value used in real TCP implementations (100 to 500 ms). A criticism of [36] on TCP simulations which use a timer granularity lower than 100 ms is that the throughput obtained in these simulations is artificially increased.

TCP synchronization does not necessarily result in the link being idle for a while. It is also possible that one or two 'lucky' sources escape synchronization, and these sources can then send their next window and keep filling up the buffer while the other sources have stopped sending data. The lucky sources thus get most of the bandwidth, which results in unfairness between the goodput of the various connections.

Results in [35] show that fairness is better if TCP's fast retransmit and recovery algorithms are enabled, since those algorithms help in mitigating the TCP synchronization effects. The efficiency can however be worse for links with large bandwidth delay products. Because multiple segments are dropped during congestion, and fast retransmit and recovery cannot recover from multiple segment losses, some segments are retransmitted during slow start, even though they have already been successfully received. In links with large bandwidth delay products, the number of retransmitted segments can be significant.

TCP performs best when there is zero loss. The connections then achieve 100% of the possible throughput and perfect fairness. For a switch to guarantee zero loss for TCP over UBR with tail drop, [54] concludes from simulation results that the amount of buffering required is at least equal to the sum of the TCP maximum window sizes for all TCP connections. This is in particular true for connections with small round trip times, since for large round trip times the switch has more time to clear out the buffer before data of the next TCP window arrives. In any case, the increase in buffer requirements is proportional to the number of sources in the simulation. This implies that UBR with tail drop and almost no loss is thus not scalable.

Partial packet discard

If a cell of an AAL5 frame is dropped because of buffer overflow, there is no reason to transmit the remaining cells of this frame. The partial packet discard (PPD) scheme [84] drops all cells of a frame subsequent to a cell loss, apart from the last one. The discard is ‘partial’ because at the time of cell loss, some cells of the corrupted frame may already be stored in the buffer or even be transmitted. The PPD scheme does not search for cells possibly stored in the buffer. The implementation of PPD requires per-VC accounting, since for each VC a state must be kept to indicate if the VC currently has to drop cells.

The last cell of a corrupted frame is not dropped, because this cell is needed at succeeding network elements and at the destination to delineate the beginning of a new frame. If the last cell is dropped anyway because there is no place left in the buffer, the cells of the corrupted frame which arrive at the destination get merged there with the next frame. This merged frame fails the cyclic redundancy check and is dropped. The source thus needs to retransmit both frames. Therefore, if the last cell of a frame is dropped, PPD also drops the next frame to avoid the useless transmission of the cells of this frame.

Simulations in [84] and experiments in [55] compare the efficiency obtained when using PPD with that obtained when using tail drop. Results with PPD are better, but the improvements are limited because still a significant amount of useless cells is transmitted over the link.

The PPD scheme is often used in conjunction with buffer acceptance schemes which try to drop complete frames. These schemes start from the principle that cells of a frame are only dropped if the first cell of the frame to which they belong was dropped. But in case a non-first cell of a frame is dropped because of buffer overflow, although the first cell of the frame was not dropped, the PPD scheme is used to ensure that the remaining cells of the frame are also dropped.

Early packet discard

Early packet discard (EPD) is a buffer acceptance scheme that has been widely implemented in commercial ATM switches. It tries to avoid the transmission of useless cells over the network by dropping complete AAL5 frames (i.e., all their cells) when the buffer becomes in danger of overflowing. This is implemented by setting a threshold, the EPD threshold, and discarding the first cell of any incoming frame if on arrival of this cell the buffer occupancy exceeds the threshold. Once the first cell of a frame has been discarded, its remaining cells are also discarded on their arrival. As with PPD, the implementation of the scheme requires a state per VC to indicate if the VC currently has to drop cells. Because frame boundaries are indicated by the last cell of each frame, a per-VC state to indicate if the next cell on the connection will be the first cell of a frame is also needed.

The EPD threshold splits the capacity of the buffer into an effective and an excess buffer capacity. The excess buffer capacity is used to accommodate cells from frames whose first cell has arrived before the EPD threshold was exceeded. In the worst case, the switch could have received the first cell of a frame from all connections before reaching the EPD threshold. To make it possible for the buffer to accept all these frames, [36] suggests to set the EPD threshold at (buffer capacity - $N*$ maximum frame size), where N is the expected number of connections active at the same time.

Simulation results in [36, 84] and experimental results in [55] show that EPD normally improves the efficiency of TCP over UBR. However, the results of [55] also show that the position of the EPD threshold is a critical point for small buffers, since in this case the excess capacity has to be considered as a reduction of the effective buffer size, which results in an increasing frame loss. So the worst case setting of the EPD threshold as suggested in [36], if possible, is not in all cases a good idea, and also not necessary, as is shown in [83]. In that paper it is illustrated by simulations that when all sources are highly synchronized, an excess buffer capacity of 50-75% of the worst case excess buffer requirement is large enough to obtain a reasonable performance. In the case where no inherent synchronization is present, this number can be reduced to 25-50%, and in both cases the percentages may be reduced even further if the number of VCs is very large.

The reason that EPD normally improves the efficiency of TCP over UBR is because the link now only carries complete frames, and because EPD concentrates the cell loss to a lower number of frames. In this way, EPD increases the likelihood that during congestion at least some of the VCs succeed in transferring a complete frame, and get a chance to further increase their window. It is however exactly this behavior which makes that EPD cannot guarantee fairness. Since EPD does not take the current rate or buffer utilization of the different VCs into account while discarding frames, it is very well possible that frames from connections causing the congestion are accepted, resulting in a possible rate increase for these connections, while frames of other connections are dropped, which forces these connections to decrease their rate.

Simulations in [28, 36, 70] show that the degree of unfairness increases as the buffer capacity is reduced. It is also shown in [70] that the effective throughput is much lower for connections which traverse more congested links than other connections. Simulations in [35] show that the combination EPD and fast retransmit and recovery improves the fairness, but hurts the efficiency for links with large bandwidth delay products.

5.1.2 Buffer acceptance schemes for UBR based on FBA

All buffer acceptance schemes discussed in the previous section fail in offering fair allocation of bandwidth among competing connections. The schemes discussed in the current section are all based on the fair buffer allocation scheme proposed by Heinanen and Kilkki in [41]. The principle behind this scheme is that a connection that gets more than its fair share of buffer space will also get more than its fair share of the bandwidth. This principle is true, irrespective of the scheduling algorithm used, since buffer space is always limited. So to provide bandwidth fairness to connections, it is necessary to at least allocate the buffer capacity fairly among the connections. The fairness offered in this way is fairness at the time that first cells of frames are accepted in the buffer. Scheduling can add to that by letting the cells of the different VCs leave the buffer in a fair order, such that the fair buffer allocation is also maintained throughout transmissions.

Fair buffer allocation

The fair buffer allocation (FBA) scheme proposed in [41] attempts to improve bandwidth fairness between competing VCs by allocating the buffer capacity fairly among them. A frame from a connection is discarded if the buffer occupancy exceeds a certain fixed threshold while the connection takes more than its fair share of the buffer. As with EPD, the decision about the acceptance of a frame is taken upon arrival of its first cell.

FBA is implemented with a global FIFO buffer and per-VC accounting. Besides per-VC states which indicate if the next cell on the connection will be the first cell of a frame, and if cells on the VC currently have to be dropped, also a counter is kept for each VC. This counter represents the number of cells that the VC has in the buffer, and is used to decide if the VC exceeds its fair share (FS). The fair share of a connection is calculated as the product of its so called *fair allocation* and the *acceptable load ratio*:

$$\text{FS} = \text{fair allocation} \times \text{acceptable load ratio}. \tag{5.1}$$

The fair allocation indicates how many cells the VC would have in the buffer if the total number of cells in the buffer was divided fairly among the various active connections, where a connection is called active if it has at least one cell in the buffer. For the FBA scheme, the fair allocation is chosen as the average number of cells per active connection:

$$\text{fair allocation} = \frac{Q}{N}, \tag{5.2}$$

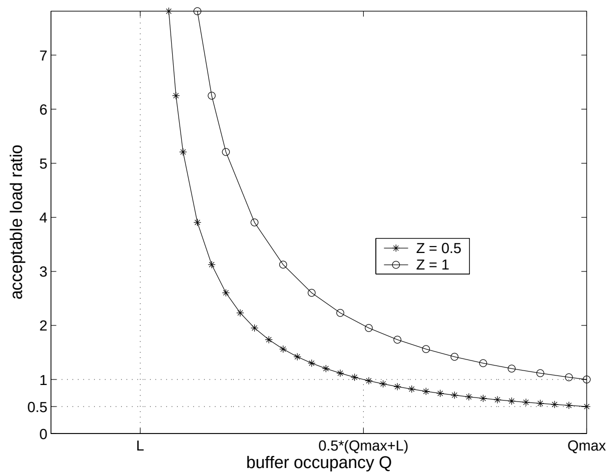

Figure 5.1: Acceptable load ratio versus the buffer occupancy for the FBA scheme.

where Q is the total buffer occupancy and N is the number of active connections.

The ratio of the number of cells that a VC has in the buffer to the fair allocation is called the *load ratio* of the VC, and gives a measure of how much the VC exceeds the fair allocation. The acceptable load ratio is the highest load ratio at which frames are still accepted in the buffer. The acceptable load ratio used in the FBA scheme is a smooth function of the buffer occupancy:

$$\text{acceptable load ratio} = Z\left(1 + \frac{Q_{\max} - Q}{Q - L}\right), \tag{5.3}$$

where Z is a linear scaling factor, typically between 0.5 and 1, $Q_{\max}$ is the buffer capacity, Q is the buffer occupancy and L is the fixed threshold below which all frames are accepted. Figure 5.1 shows the acceptable load ratio versus the buffer occupancy for $Z = 0.5$ and $Z = 1$. It can be seen from this figure that if Q is close to the threshold L, a VC can exceed the fair allocation considerably, but the acceptable load ratio decreases very fast when Q increases. For all Z less than one, the acceptable load ratio becomes smaller than one when Q gets larger than $Z(Q_{\max} - L) + L$. This means that if the buffer is almost full, a new frame can be dropped even when its connection occupies less of the buffer than the fair allocation. As stated in [41], this property is desirable since some of the buffer capacity should be left for the remaining cells of already accepted frames.

As expected, simulations [28, 36] show that the fairness results obtained with the FBA scheme are better than those obtained with the EPD scheme, since now frames of overloading connections are dropped in preference to underloading ones. Most of the time, also the efficiency results obtained with FBA are better, because the dropping condition is not fulfilled for all connections at the same time anymore, such that TCP synchronization is easier broken. However, it is also shown in [36] that the FBA scheme is sensitive to changes in the parameters L and Z. Higher efficiency values have either L or Z high, since higher buffer utilization is allowed in these cases. But there is a considerable variation in the fairness numbers obtained, since not all combinations of the parameter values are equally effective in breaking the TCP synchronization. If for example almost all connections exceed their fair share at roughly the same time, the buffer occupancy will presumably be no longer above the threshold L at the moment that the few connections that did not exceed their fair share earlier finally do exceed it, since the other sources have stopped sending meanwhile because of dropped segments. The frames of the 'lucky' sources are thus not dropped, resulting in unfairness.

Selective drop

Selective drop (SD) [36], also called EPD with per-VC accounting [69], is a simpler version of the FBA scheme. The principle of the SD scheme is exactly the same as that of the FBA scheme: it is implemented with a global FIFO buffer and per-VC accounting, and new frames of a connection are discarded if on arrival of their first cell the buffer occupancy exceeds a certain fixed threshold while the connection takes more than its fair share of the buffer. The only difference with the FBA scheme is in the definition of the 'fair share'. It is again calculated as the product of the fair allocation and the acceptable load ratio, and the fair allocation is again chosen as the average number of cells per connection, but the acceptable load ratio is now a simple parameter K independent of the buffer occupancy.

$$\mathrm{FS} = \text{fair allocation} \times \text{acceptable load ratio} = \frac{Q}{N}K. \tag{5.4}$$

Since this scheme also drops frames of overloading connections in preference to underloading ones, it improves fairness over the EPD scheme, as is shown by simulations in [36, 69]. Also the efficiency values obtained are slightly better. Compared to the FBA scheme (for optimal parameter values), the efficiency results obtained with SD are slightly lower than with FBA, while the fairness results are comparable. The simulations in [36] also show that the fairness of the scheme decreases with an increasing number of sources, and those in [69] indicate that the more hops a VC traverses, the lower its effective throughput is.

EPD with per-VC queueing

In both the FBA and SD schemes discussed above, a fair allocation of the buffer capacity is maintained only at the moments of acceptance of the first cell of a frame. But since in both schemes all VCs share a single FIFO buffer, this fair buffer allocation cannot be maintained throughout transmissions. The EPD with per-VC queueing mechanism [69] uses the same criteria as the SD scheme to decide about the buffer acceptance of cells, but cells from the different VCs are placed into different queues (per-VC queueing). All these VC queues are then served using round robin scheduling, such that the accepted cells are emitted from the buffer in a fair manner. The EPD with per-VC queueing scheme thus does not only provide fair buffer allocation at moments of frame acceptance, but also throughout transmissions. In this way, throughput fairness can be achieved as long as each VC has some cells in its queue.

In [69], simulation results obtained by using EPD with per-VC queueing are compared to results obtained by using the SD scheme. It appears that EPD with per-VC queueing achieves almost perfect fairness, also for VCs which traverse more hops. However, the efficiency obtained with the scheme is somewhat lower than that obtained with SD, which is explained as a synchronization effect.

5.1.3 Buffer acceptance schemes for UBR based on RED

The random early detection (RED) algorithm was first proposed in [30], and applies for IP gateways. RED thus deals with IP packets, not ATM cells. The RED algorithm has some attractive properties: RED gateways keep the average queue size low, while allowing occasional bursts of packets in the queue; during congestion, the probability that a packet from a particular connection is dropped is roughly proportional to that connection's share of the bandwidth through the gateway; RED avoids TCP synchronization since packet dropping will probably concern the most greedy connections. Because of these attractive properties, some adaptations of the RED scheme for use with ATM were developed: cell-based RED (C-RED) [25], packet-based RED (P-RED) [25] and ATM-RED [85].

First the RED algorithm is shortly described. Then the P-RED and ATM-RED proposals are discussed. The C-RED proposal is not considered here since it is outperformed in both efficiency and fairness results by P-RED, which is also less complex to implement.

The RED algorithm

The objective of the RED algorithm as proposed in [30] is to keep the throughput of an IP gateway high, but its delay low. This is done by dropping arriving packets with a certain probability each time the average queue size of the gateway exceeds a certain threshold. The RED algorithm is applied on a global FIFO queue, for which only global accounting information is kept.

On each packet arrival, the average queue size is estimated by[1]

$$Avg = (1 - w_q)Avg + w_q Q, \tag{5.5}$$

where w_q is a weight between 0 and 1, and Q is the actual queue size. This average queue size is then compared with two thresholds L (low) and H (high): when it is less than L, the packet is accepted; when it is greater than H, the packet is dropped; if it is between the

[1] This equation is actually used only when the queue is not empty on packet arrival. When it is empty, the average queue size is calculated based on the number of packets that might have arrived during the idle time.

two thresholds, the packet is dropped with a probability p_a. The initial drop probability p_b is calculated as a linear function of the average queue size:

$$p_b = max_p \frac{Avg - L}{H - L}, \tag{5.6}$$

where max_p is the maximum value for p_b. The final drop probability p_a is calculated such that when the average queue size is constant, the random variable describing the number of packets that arrive after a dropped packet, until the next dropped packet, is uniformly distributed over $\{1, 2, \ldots, 1/p_b\}$ (assuming that $1/p_b$ is an integer):

$$p_a = \frac{p_b}{1 - p_b\, count}. \tag{5.7}$$

The parameter *count* counts the number of accepted packets since the last dropped packet or since Avg exceeded L.

Packet-based RED

Packet-based RED (P-RED) [25] is an AAL5 aware buffer acceptance algorithm for use with ATM-UBR. The decision about the acceptance of an AAL5 frame is taken upon arrival of the first cell of the frame: if the first cell of a frame is discarded, its remaining cells are also discarded on their arrival. P-RED is implemented using a global FIFO buffer and two fixed thresholds L and H. The algorithm implements EPD with the threshold H: if the buffer occupancy Q is above H on arrival of the first cell of a frame, this frame is discarded. Otherwise, the average queue length is estimated by equation (5.5), and compared with the thresholds L and H exactly as in the RED algorithm described above. When the average queue size is below L, the frame is accepted; when it is above H, the frame is dropped. When the average queue size is between L and H, the frame is dropped with a probability p_a calculated by equation (5.7) as in the RED algorithm. There is however a difference in the calculation of the initial drop probability p_b. If the new packet belongs to VC i, p_b is weighted by the load ratio of VC i:

$$p_b = max_p \frac{Avg - L}{H - L} \times \text{ load ratio of VC } i, \tag{5.8}$$

where the load ratio of VC i is defined by $Q_i N/Q$ as in the FBA algorithm. Remark that due to the weights that appear in (5.8), the P-RED algorithm needs to keep per-VC buffer accounting information.

Simulations in [25] compare the performance of P-RED with that of EPD using a simulation configuration with different propagation delays for the various TCP sources. It appears clearly that efficiency and fairness obtained with P-RED are better than with EPD, and that both the queue size and the average queue size of P-RED is low compared to EPD. This means that the P-RED algorithm is an attractive buffer acceptance algorithm for interactive applications like Telnet. Moreover, the drop probability for this type of connections is very low since their load ratio is expected to be very low.

ATM-RED

ATM-RED [85] is also an adaptation of the original RED algorithm for use with ATM. It uses a global FIFO buffer and two thresholds L and H. Typical to the ATM-RED algorithm is that a drop probability is calculated on cell level. Since the algorithm accepts or discards entire AAL5 frames, each discard decision concerns the next frame (with relation to the cell that caused the decision), except if the decision is made on arrival of the first cell of a frame. This approach allows to obtain smaller values for the overall frame dropping probability when frames are smaller.

ATM-RED calculates a drop probability when a first cell of a frame arrives at the buffer and it has not yet been decided if this frame is to be dropped, and when a non-first cell of a frame is accepted in the buffer and it has not yet been decided if the next frame on the same connection will be dropped. To calculate the drop probability, first an average queue size Avg is calculated using (5.5). Then, if $Avg > L$, p_c is calculated as

$$p_c = \begin{cases} max_c\,(Avg - L)/(H - L) & \text{if } L < Avg \leq H, \\ 1 + (1 - max_c)(Avg - 2H)/H & \text{if } H < Avg \leq 2H, \\ 1 & \text{if } Avg > 2H, \end{cases} \tag{5.9}$$

where max_c is the maximum value for p_c when $Avg \leq H$. With a probability p_c it is then decided to drop the next AAL5 frame.

Remark that from the viewpoint of per-VC complexity, this algorithm needs to keep only three bits of state for each VC, while the P-RED algorithm needs also a counter per VC. The price for this is however the requirement to compute the p_c probability upon cell arrivals instead of once for each packet as in P-RED.

In [85], the performance of ATM-RED has been compared to the performance of EPD and FBA in several quite different environments. It is shown by simulation that FBA and ATM-RED are almost always superior to EPD. ATM-RED has in general by far the lowest mean buffer occupancy, which gives low delays, while offering high goodputs and link utilization. ATM-RED is also a good solution as regards the fairness among similar sources (same characteristics, same RTTs, crossing the same hops), but is poor at achieving fairness under a heterogeneous traffic mix. In particular, sources with higher RTTs or crossing more hops, have lower goodputs.

5.1.4 Related work

The buffer acceptance schemes discussed until now are a number of representative schemes for UBR. Of course, more schemes exist. In [60] for example, two drop from front schemes are proposed: pure drop from front and partial frame drop at front. These schemes are similar to the tail drop and partial packet discard schemes, but cells are dropped at the front of the buffer instead of at the end. This policy causes TCP's congestion control actions being invoked approximately one buffer drain time earlier.

The early selective packet discard (ESPD) scheme introduced in [21] tries to avoid the link idle time due to synchronization by concentrating the frame discarding on a few connections only. Spread over times longer than a frame duration, ESPD makes connections to take turns to access the network resources. This is unlike EPD where the dropping/accepting status of a connection is released upon the arrival of the last cell of a frame. In [21, 22], it is demonstrated that ESPD slightly improves the effective throughput over EPD and provides better overall fairness since it provides more throughput enhancement to a long round trip time session than to a short round trip time session.

In [58] the Fair Buffering (FB) mechanism is proposed, which allocates buffer space for the different connections in proportion to their bandwidth delay products, and spreads out the discarding of frames from the same connection over time. FB needs to know however each connection's RTT, a value which in practice is not known by the switches [33].

A modification of the FBA scheme for supporting weighted bandwidth allocation is proposed in [41]. However, no performance evaluation of the scheme is performed. In the modified FBA version, the fair allocation for a connection i is calculated as W_iQ/W, where W_i is a weighting coefficient associated with connection i and W is the sum of the weighting coefficients of all active connections.

Virtual queueing, a technique which is discussed in [97], emulates an acceptance scheme similar to the EPD with per-VC queueing scheme on a global queue. This is done by maintaining a counter M_i for each VC i. Cells leave the buffer in FIFO order, but regardless of which VC a transmitted cell actually belongs to, the counters M_i are decremented in a round robin fashion as if per-VC queueing and round robin scheduling were implemented. Every time a cell of connection i is accepted in the buffer, the counter M_i is incremented. To avoid loss of buffer allocation to active connections with temporarily empty virtual queue, the per-VC counters are allowed to be negative. The scheme achieves nearly perfect fairness.

5.2 Buffer acceptance schemes for GFR

In general, buffer acceptance schemes for GFR can be classified in three categories:

- schemes relying on a tagging function,
- schemes using per-VC accounting and per-VC queueing,
- schemes using per-VC accounting in a global FIFO buffer.

For each of these categories, an informative example implementation is given in the ATM Forum Traffic Management Specification [5], while also other schemes have been defined in the literature. Some of these schemes are discussed in this section. Remark that the terminology that is used relies heavily on the terms introduced in Section 4.5.

5.2.1 Buffer acceptance schemes for GFR relying on a tagging function

The schemes in this first category rely on the fact that network based tagging is performed at the entrance of the network to provide the per-VC minimum rate guarantees to the different connections. The tagging function, which is typically based on the F-GCRA algorithm, identifies the eligible and ineligible frames of each connection and sets their CLP bit correspondingly, while the buffer acceptance scheme then uses this CLP information to treat the eligible frames preferentially. When deciding about the acceptance of a new frame, these schemes usually take their decision based on global buffer accounting information and the CLP priority of the frame. Remark that because these buffer acceptance schemes rely on network based tagging, they cannot support GFR.1 connections.

Implementation using tagging and a FIFO queue

The buffer acceptance scheme that is considered here is one of the informative example implementations given in the ATM Forum Traffic Management Specification [5]. The acceptance scheme relies on two fixed buffer thresholds L (low) and H (high) in a global FIFO queue. Those thresholds are used as EPD thresholds, L for the CLP=1 frames and H for the CLP=0 frames. The L threshold is used to limit the amount of CLP=1 frames in the buffer. The scheme is very simple, but it is immediately clear that no attempt is made to divide the excess bandwidth in a fair manner between the different connections. As already mentioned, this scheme is of no use if it is not preceded by a tagging function.

Simulation experiments with this implementation are performed in [81, 14, 15]. It is shown that the performance of TCP is never satisfactory, since not all TCP sources are able to benefit from the minimum guaranteed bandwidth. VCs with higher MCRs get throughputs which are much lower than their MCRs, while the VCs with lower MCRs get bandwidth in excess of their MCRs. The reason is that when the buffer occupancy goes below L, all frames are accepted into the buffer. The acceptance rate of cells of the different connections into the buffer is therefore not proportional to their MCR, implying that their respective service rates are also not proportional to their MCRs. Whenever the buffer occupation exceeds L, the cell acceptance rate into the buffer is bounded by the rate at which cells pass the F-GCRA tagging function without being tagged. But since TCP traffic is bursty, the F-GCRA tags a large fraction of the frames, even when the long term average throughput of a VC is smaller than its MCR. Furthermore, the F-GCRA has the tendency to mark TCP traffic in bursts. The tagged frames are dropped when the buffer occupancy is above L, and the large number of bursty losses combined with TCP's congestion control algorithms force the congestion window of the TCP sources down such that less traffic is sent into the network. For sources with high MCR, the average congestion window can be much lower than their on average required value to fill the minimum reserved throughput. In [15] it is shown that the performance is much better in scenarios where each ATM VC carries the traffic of more than one TCP connection, since when a burst of frames is marked by the

F-GCRA, and later on discarded by the buffer acceptance scheme, not all TCP connections are affected.

Implementation using tagging and per-VC queueing

Since the results obtained with the GFR implementation using tagging and a FIFO queue were not satisfactory, also an implementation based on tagging and per-VC queueing is investigated in [81]. The same criteria as in the first scheme are used to decide about the buffer acceptance of cells, but the cells from the different VCs are now buffered in different queues (per-VC queueing). These VC queues are scheduled using round robin scheduling or weighted round robin scheduling with the weights set in proportion to the MCRs of the connections.

The simulation results in [81] indicate that with round robin scheduling, the throughput of all VCs again does not always reach the MCR, for the same reasons as when using a FIFO buffer. With the weighted round robin scheduling, the MCR for each VC is guaranteed in the buffer region below L because of the scheduler, and in the region between L and H because only the MCR gets through the F-GCRA tagging function. In the example simulated in [81], the throughput of each VC is above its MCR, and it is then concluded that a rate guaranteeing service discipline such as weighted round robin in conjunction with a tagging function can make the guarantees for the GFR service discipline. However, we can imagine that the fact that the TCP traffic is not able to adapt its behavior to the F-GCRA tagging function, and the resulting problem as described above of sources having a too low average window size to use their guaranteed rate, could also occur here.

5.2.2 Buffer acceptance schemes for GFR using per-VC accounting and per-VC queueing

Since per-VC queueing maintains a separate queue for each VC, it isolates frames from different VCs. A suitable per-VC scheduling mechanism can then select between the queues at each scheduling instant to provide all active connections with their reserved bandwidth. When it is however not sure that ineligible frames are tagged at the entrance of the network, or if GFR.1 connections need to be supported, it must be ensured that a single VC is not able to saturate the switch buffers. For this, also per-VC accounting needs to be implemented, because if an unbalanced distribution of the buffer occupancy is allowed, then the resulting output will also be unbalanced since the total buffer space is limited.

Implementation using per-VC accounting and WFQ-like scheduling

This buffer acceptance scheme for GFR is again one of the schemes proposed in [5]. It uses per-VC queueing, per-VC scheduling and a per-VC counter R_i representing the number of CLP=0 cells VC i has in its queue. Individual connections are scheduled at a rate of at least their MCR using a WFQ-like scheduler. This guarantees that when active, each VC is allocated its reserved bandwidth as well as some fair share of the excess bandwidth.

The scheme uses two global thresholds, L (low) and H (high), and a threshold T_i for each VC queue, which is typically set to the MBS of connection i. The threshold L is used as EPD threshold for CLP=1 frames. CLP=0 frames are accepted if the total buffer occupancy is below the second threshold H, or if R_i is below T_i for a frame of connection i.

In [14], simulation experiments were performed with this buffer acceptance scheme. The WFQ-like scheduler used is a virtual spacing scheduler. With GFR.1 connections, the performance was much better than with the implementation using tagging and a FIFO queue. The goodput achieved by the TCP sources is much closer to the expected goodput, although sources with a high MCR are again somewhat penalized. With GFR.2 connections where the F-GCRA tags frames at the entrance of the network, the TCP performance was lower than with the GFR.1 connections. This is again because the TCP traffic is bursty, implying that many frames are tagged by the F-GCRA. Further, these CLP=1 frames are already discarded by the buffer acceptance scheme when the buffer occupancy is relatively small.

Global FIFO scheduling

Global FIFO scheduling (GFS) is proposed in [19]. In contrast to other buffer acceptance schemes for GFR, GFS does not use the CLP information in the cells, but integrates the buffer acceptance scheme and the eligibility test.

GFS uses per-VC queueing, a global FIFO buffer containing references from the VC queues, a fixed global threshold L and a threshold T_i for each VC queue. The decision about the eligibility of a frame and about the acceptance of the cells of the frame in the buffer is taken on arrival of the first cell of the frame. When the frame is considered eligible, or when the total buffer occupancy is below the global threshold L, or when the occupancy of the queue i corresponding to the connection on which the frame arrives is below T_i, then the first cell of the frame and all its following cells are accepted. Otherwise, all cells of the frame are discarded. Each time a cell arrives from a frame that is accepted, and this frame is considered eligible, a reference to the VC it belongs to is put into the FIFO buffer. The FIFO buffer thus maintains the order in which the VC queues have to be served according to the order of the arrivals of cells from frames which have been chosen eligible for the GFR MCR service guarantee. When the global FIFO queue is empty, a round robin scheduling scheme is performed among all VC queues. The excess bandwidth is thus equally shared among the excess traffic, in contrast to the previous WFQ based scheme that shares it in proportion to the MCRs of the connections.

Simulations in [19] evaluate the performance of GFS. When there are no losses, GFS shows good performance and provides each connection with its guaranteed bandwidth. The excess bandwidth is shared equally among the different VCs. When cell losses occur, GFS cannot always guarantee the reserved bandwidth in a fair manner, but the results are close to what might be expected (MCR plus an equal share of the left over bandwidth).

5.2.3 Buffer acceptance schemes for GFR using per-VC accounting in a global FIFO buffer

For a service like GFR, the cost of per-VC queueing and per-VC scheduling may be considered too high, making an implementation using a global FIFO buffer for all VCs desirable. In contrast to per-VC queueing, FIFO queueing cannot isolate frames from different VCs at the egress of the buffer, since the cells are scheduled in the order in which they entered the buffer. So an intelligent buffer acceptance algorithm based on per-VC accounting is needed to provide the minimum rate guarantees to the various connections. Several buffer acceptance schemes for GFR using per-VC accounting in a global FIFO buffer have been proposed, but were not able to deliver GFR guarantees. Examples of these are weighted buffer allocation (WBA) in [34] and a scheme based on dynamic per-VC thresholds in [7]. In this section, two acceptance schemes that are more successful in delivering GFR guarantees are considered.

Differential fair buffer allocation

Differential fair buffer allocation (DFBA) is also one of the example GFR implementations of the ATM Forum Traffic Management Specification [5]. It is designed for use with a global FIFO buffer and tries to allocate buffer capacity fairly amongst competing connections. This allocation is proportional to the MCRs of the connections, by assigning to each connection a weight W_i corresponding to its MCR.

DFBA uses two thresholds L (low) and H (high). If on arrival of the first cell of a frame the total buffer occupancy Q falls below L, the scheme attempts to bring the system to efficient utilization by accepting the frame. When Q is above L, it drops new CLP=1 frames to ensure that sufficient buffer capacity is available for CLP=0 frames. The threshold L is thus an EPD threshold for low priority frames. The threshold H does the same for CLP=0 frames, so when Q is above H, all new frames are discarded. When Q is between L and H, DFBA attempts to allocate buffer space proportional to the MCRs: when Q_i, the number of cells of connection i in the buffer, is below its fair share, then new CLP=0 frames of connection i are accepted. In DFBA, the fair share of connection i equals its fair allocation (i.e., the acceptable load ratio is 1), which is defined as

$$\text{fair allocation for connection } i = \frac{W_i}{W} Q,$$

where W is the sum of the weighting coefficients of all active connections. If Q_i exceeds the fair share of connection i, then new CLP=0 frames of connection i are dropped with a certain probability. The purpose of this probabilistic drop is to notify TCP sources of congestion, but in such a way that they back off without a timeout, and thus without temporal inactivity.

The DFBA drop probability consists of an efficiency and a fairness component. The efficiency component increases linearly when Q increases from L to H, and the fairness component increases linearly with an increase of Q_i from $(W_i/W)Q$ to Q:

$$p = \mathrm{P}\{\text{drop}\} \; = Z_i \left(\alpha \frac{Q_i - (W_i/W)Q}{Q(1 - W_i/W)} + (1 - \alpha) \frac{Q - L}{H - L} \right). \tag{5.10}$$

In this formula, the parameter α is used to assign appropriate weights to the fairness and efficiency components. The parameter Z_i defines the maximum drop probability enforceable for connection i.

Simulations with the DFBA scheme for GFR.1 VCs carrying multiple TCP/IP connections are performed in [33, 16]. They show that DFBA meets the MCR guarantees, but fails to share the excess bandwidth among the VCs in proportion to their MCR: the smaller MCR a connection has, the larger the proportion 'goodput/MCR' for that connection becomes. In [16], it is illustrated that the poor fairness obtained with DFBA results from the fact that DFBA fails to provide a fair share of the buffer to the various VCs. Tuning the parameter Z_i carefully with respect to the MCR of VC i can alleviate this problem a bit, but not entirely.

Although DFBA treats CLP=0 and CLP=1 frames differently, as far as we are aware of, no results are published with the DFBA scheme using GFR.2 connections. The same problem as previously discussed is however to be expected: since TCP is not able to adapt its behavior to the F-GCRA function, a large percentage of the frames will be tagged. These tagged frames are discarded by DFBA when Q exceeds L, resulting in some sources having for long times a congestion window smaller than their MCR times their round trip time, which implies that these sources cannot use their minimum bandwidth guarantee.

Token-based buffer allocation

Like the GFS scheme, the token-based buffer allocation (TBA) scheme proposed in [13] tests the eligibility of the frames at the switching element, without using the CLP information in the cells. A main difference between the eligibility test used with TBA in [13] and the one used with GFS in [19], is that in this last scheme the buffer acceptance algorithm uses per-VC counters to count the number of cells each VC has in the buffer, while also the eligibility test keeps F-GCRA alike counters for each VC. In TBA, an approximate token based solution is used for the eligibility test which increments and decreases one of the per-VC counters of the buffer acceptance algorithm: a counter C_i is associated with VC i and decreased every time a cell of VC i is accepted in the FIFO buffer; this counter is incremented at a rate corresponding to the MCR of connection i. Further, the excess bandwidth is divided among the active VCs by distributing excess tokens which also increment the C_i's. The distribution of these excess tokens can be done equally among all VCs, in proportion to their MCRs, but also completely uncoupled from the MCRs.

The buffer acceptance part of the TBA scheme is implemented in a FIFO buffer. The scheme takes a different acceptance decision for a frame of connection i depending on if connection i is a GFR.1 or a GFR.2 connection. The switch has this information about each connection available, since it has been signaled during connection establishment. For a GFR.2 connection, no CLP=0 frames should be discarded, so they are always accepted. CLP=1 frames of a GFR.2 connection are accepted based on the RED algorithm, thus with a certain probability. For GFR.1 connections, CLP=0 frames are accepted if the C_i counter is positive, or based on the RED algorithm with a certain probability; CLP=1 frames are accepted if the C_i counter is larger than some positive value (e.g., $\mathrm{MBS}_i/2$), or again based on the RED algorithm with a certain probability. When the acceptance decision is based on the RED algorithm, the average queue length Avg, which is calculated using equation (5.5), is compared to two fixed thresholds L (low) and H (high): the frame is accepted if Avg is below L; when it is above H, the frame is dropped. When Avg is between both thresholds, a packet dropping probability p_b is calculated. This p_b is calculated as a linear function of the average buffer occupancy and the C_i counter:

$$p_b = \alpha \frac{Avg - L}{H - L} - \beta \frac{\min(0, C_i)}{H}. \tag{5.11}$$

The relative influence of the average buffer occupancy and the C_i counter on the dropping probability depends on the values of α and β[2].

Simulations in [13] compare the performance of the TBA scheme with that of the implementation using tagging and a FIFO queue and the implementation using per-VC accounting and WFQ-like scheduling. For GFR.1 connections that carry the traffic of a single TCP connection, the performance of TBA is rather disappointing and highly dependent on the values of the parameters and on the TCP implementations used. Mostly the scheme does not bring any significant benefit compared to the the implementation using tagging and a FIFO queue: also with TBA VCs with higher MCRs get throughputs below their reserved bandwidth, while VCs with lower MCR get bandwidth in excess of their MCRs. If each VC carries the traffic of several TCP connections, the performance of TBA is better: each VC can efficiently utilize its minimum bandwidth, and the performance does not appear to depend heavily on the chosen values of the parameters. Compared with the implementation using per-VC accounting and WFQ-like scheduling, the performance of TBA is comparable or even slightly better. Although the TBA scheme provides a different treatment to GFR.1 and GFR.2 connections, no simulations with GFR.2 traffic were performed in [13].

5.2.4 Related work

As with the acceptance schemes for UBR, more schemes than the ones presented exist also for GFR. In [26], three acceptance schemes closely related to the ones already discussed are proposed. The first one is the implementation using tagging and a FIFO queue, but

[2] When p_b needs to be calculated for a frame from a GFR.1 connection, it is proposed to use a larger β for the CLP=1 frames than for the CLP=0 frames, such that the discard probability for CLP=1 frames is higher than for CLP=0 frames.

combined with the drop from front strategy. The second scheme falls into the category of schemes using per-VC accounting in a FIFO buffer. It uses a fixed threshold L for the CLP=1 frames and a fixed threshold H for the CLP=0 frames, together with separate accounting information about the number of CLP=0 and CLP=1 cells each VC has in the buffer, to decide about the acceptance of a CLP=1, resp. CLP=0, frame. The third scheme combines the second scheme with the drop from front strategy. The conclusion of the study in [26] is that the combination of a buffer acceptance scheme with drop from front improves its fairness, but can negatively impact its efficiency because when cells of a given frame should be dropped, some cells of that frame might already have left the buffer.

In [88], a scheme which belongs to the category of schemes using per-VC accounting in a FIFO buffer is presented. The scheme relies on the virtual queueing technique of [97] and divides the time in intervals of length T. In each period T, the virtual scheduling mechanism consists of two phases. In the first phase, the scheduler virtually serves $T \times \mathrm{MCR}_i$ cells from each VC i to guarantee to each connection its MCR. In the second phase, the scheduler virtually serves each VC in a round robin fashion to achieve fair allocation of the excess bandwidth. The actual order in which cells leave the buffer is FIFO. Although the principles behind the scheme are sound, we think that a more complex implementation of these principles than presented in [88] is needed, since the implementation of [88] can lead to the loss of connections from the list identifying the active VCs.

A packet-discard push-out scheme which belongs to the category of schemes using per-VC accounting in a global FIFO buffer is proposed in [20]. As long as a certain dynamic variable C which estimates the available buffer space is positive, all frames are accepted. Since C takes the buffer space needed by cells of frames whose first cell has already been accepted into account, this policy does not hurt already accepted frames. If C is negative, a new frame of connection i arrives at the buffer, and the cells of connection i do not occupy more than a certain share of the buffer, then the queue manager selects another (or more than one) connection that occupies too much space in the buffer, and the last frame of this VC is pushed out of the buffer such that space comes available for the new frame of connection i. The performance of this scheme is only compared with the performance of EPD in [20]. Simulation results illustrate that the buffer utilization is kept at 100% with the packet-discard push-out scheme, while this is not always the case for EPD, and that the scheme can prevent an ill-behaved source from obtaining an arbitrary share of the bandwidth.

5.3 Conclusions

In this chapter, an overview of the most representative buffer acceptance schemes that have been proposed for use with the UBR and GFR ATM service categories was presented. Characteristic of all schemes is their AAL5 frame awareness: if the scheme decides to accept, respectively discard, the first cell of a frame, it will try to accept, respectively drop, all cells of the same frame, since incomplete frames are of no use at the destination.

The principles of two of the earliest proposed schemes, namely partial packet discard and early packet discard, are found back in many of the more sophisticated schemes. To be able to accept the non-first cells of a frame from which the first cell was accepted, most acceptance schemes use a threshold, as in EPD, to provide some excess capacity in the buffer. If in spite of this excess capacity a cell is lost because of buffer overflow, the remaining cells of its frame are discarded as in PPD.

No QoS commitments are made by the network to UBR connections, but most recent buffer acceptance schemes for UBR try to provide a fair allocation of the bandwidth to competing connections. This is done by aiming at a fair allocation of the buffer capacity among the connections, using the principle behind the FBA scheme that a connection that gets more than its fair share of the buffer space will also get more than its fair share of the bandwidth. The same principle is used in some of the buffer acceptance schemes for GFR, although the fairness is an issue then only to the excess capacity. The first concern of buffer acceptance schemes for GFR is to provide each connection with its MCR service guarantee.

Relying on the attractive properties of the RED scheme in IP gateways, some schemes for ATM using the principles behind RED are proposed. The most important feature of these schemes is their ability to keep the average buffer size, and thus also the average queueing delay, low.

Most buffer acceptance schemes proposed to support GFR connections can be grouped in one of the three main categories, as is done in Section 5.2. The first category contains schemes relying on the tagging of ineligible frames by a F-GCRA function to provide the per-VC minimum rate guarantees to the different connections. This implies that those schemes can only support GFR.2 connections. Schemes that support GFR.1 connections are found in the second and the third category. The schemes in the second category use per-VC accounting and per-VC queueing, making per-VC scheduling possible. With an appropriate per-VC scheduling algorithm, each VC is, when active, allocated its reserved bandwidth. The schemes in the third category use per-VC accounting in a FIFO buffer, since the cost of per-VC queueing and per-VC scheduling may be too high for a service category like GFR.

In general, buffer acceptance schemes for GFR have problems in providing GFR.2 connections with their minimum guaranteed bandwidth. This is because the GFR service guarantee applies only to the CLP=0 frames of a connection. So the buffer acceptance schemes have to treat the CLP=1 frames with a lower priority. But TCP congestion control mechanisms react on the loss of the frames by reducing the windows of the sources, resulting in some TCP sources sending at a rate which is much too low to obtain their reserved throughput. The main cause of this problem is found in the fact that TCP is not able to adapt its behavior to the F-GCRA tagging function used with GFR.2 connections. It is shown in [13] that when the F-GCRA function is preceded by a shaping function, a significant gain in performance is noticed for the GFR implementation using tagging and a FIFO queue. It seems however logical to expect the same improvement of the performance when the shaping function is used in combination with other buffer acceptance schemes.

For buffer acceptance schemes not only the principles behind the acceptance algorithm are important, but also the accounting information the algorithm can base its decisions on and the queueing and scheduling strategy used. In Tables 5.1 and 5.2 a summary of this information for the main buffer acceptance schemes discussed in this chapter is provided.

Finally, remark that the ATM-Forum has recently proposed an optional minimum desired cell rate (MDCR) indication for UBR [6], by which UBR connections can indicate to the network a preference for a minimum bandwidth objective. Regardless of the presence and/or value of this MDCR, this does not define a service commitment of the network to the UBR connection. However, network specific QoS commitments for such connections are not precluded. When a network wants to provide such QoS commitments, it will need to implement a buffer acceptance scheme which relates closely to the schemes discussed for GFR in this chapter.

Buffer acceptance scheme	Queueing strategy	Scheduling strategy	Accounting information: per-VC states[a]	Accounting information: counters[b]
Tail drop	global	FIFO	-	global (Q)
Partial packet discard	global	FIFO	drop$_i$	global (Q)
Early packet discard	global	FIFO	drop$_i$ new_frame$_i$	global (Q)
Fair buffer allocation	global	FIFO	drop$_i$ new_frame$_i$	global (Q,N) per-VC (Q_i)
Selective drop	global	FIFO	drop$_i$ new_frame$_i$	global (Q,N) per-VC (Q_i)
EPD with per-VC queueing	per-VC	round robin	drop$_i$ new_frame$_i$	global (Q,N) per-VC (Q_i)
Packet-based RED	global	FIFO	drop$_i$ new_frame$_i$	global (Q,N,$count$,Avg) per-VC (Q_i)
ATM-RED	global	FIFO	drop$_i$ new_frame dropnext$_i$	global (Q,Avg)

Table 5.1: Buffer acceptance schemes for UBR: overview of the queueing, scheduling and accounting strategies.

[a] Drop$_i$ is a per-VC state that indicates if the next cell on connection i needs to be dropped. On arrival of the last cell of a frame on connection i, it is reset. New_frame$_i$ is a per-VC state which indicates that the next cell on connection i is the first one of a frame. Dropnext$_i$ is a per-VC state that indicates if the next frame on connection i needs to be dropped.

[b] Q: total buffer occupancy; N: number of active connections; Q_i: number of cells of VC i in the buffer; Avg: average buffer occupancy, calculated by equation (5.5); $count$: number of accepted frames since the last dropped frame, or since Avg exceeded a threshold L.

Buffer acceptance scheme	Queueing strategy	Scheduling strategy	Accounting information: per-VC states[a]	Accounting information: counters[b]
Tagging + FIFO	global	FIFO	drop_i new_frame_i	global (Q)
Tagging + per-VC queueing	per-VC	(weighted) round robin	drop_i new_frame_i	global (Q)
Per-VC accounting and WFQ-like scheduling	per-VC	WFQ-like (virtual spacing)	drop_i new_frame_i	global (Q) per-VC (R_i)
Global FIFO scheduling	per-VC[c]	FIFO + round robin[d]	drop_i new_frame_i eligible_i	global (Q) per-VC (Q_i)
Differential fair buffer allocation	global	FIFO	drop_i new_frame_i	global (Q,N,W) per-VC (Q_i,W_i)
Token based buffer allocation	global	FIFO	drop_i new_frame_i	global(Q,Avg) per-VC (C_i)

Table 5.2: Buffer acceptance schemes for GFR: overview of the queueing, scheduling and accounting strategies.

[a] Drop_i is a per-VC state that indicates if the next cell on connection i needs to be dropped. On arrival of the last cell of a frame on connection i, it is reset. New_frame_i is a per-VC state which indicates that the next cell on connection i is the first one of a frame. Eligible_i is a per-VC state that indicates if the current frame on connection i is considered eligible.

[b] Q: total buffer occupancy; R_i: number of CLP=0 cells of VC i in the buffer; Q_i: number of cells of VC i in the buffer; N: number of active connections; W_i: weight of VC i; W: sum of the weights of all active VCs; Avg: average buffer occupancy, calculated by equation (5.5); C_i: token counter associated with VC i.

[c] Although the name of the scheme suggests otherwise, cells are queued per-VC. There is also a FIFO queue in which references to the per-VC queues are queued.

[d] As long as the FIFO queue contains references to the VC queues, scheduling is FIFO. If the FIFO queue is empty, scheduling is round robin.

Chapter 6

Transient performance analysis of the selective drop buffer acceptance algorithm with responsive traffic

The selective drop and EPD with per-VC queueing buffer acceptance schemes discussed in Chapter 5 use the same frame[1] aware buffer acceptance rules to decide about which cells are allowed to enter the buffer and which cells are discarded. A flowchart of these rules is shown in Figure 6.1. In the selective drop scheme these rules are combined with a global queueing strategy and the FIFO scheduling strategy, while in the EPD with per-VC queueing scheme they are used in combination with per-VC queueing and round robin scheduling. In the current chapter we consider them in combination with three scheduling algorithms: FIFO, round robin (RR) and a variant of probabilistic longest queue first (PLQF). For the sake of simplicity, throughout this chapter only the term 'selective drop (SD)' is used, completed when needed with the specific scheduling algorithm considered.

The transient performance of SD is analyzed when traffic is generated by sources which respond to the presence or absence of losses (as TCP sources do). For this goal a theoretical model is developed, where two responsive sources send traffic in fixed-sized packets of cells, via a buffer on which the SD buffer acceptance algorithm is implemented. Transient efficiency and fairness results are obtained from the model, most of the time under an unfair start condition, which corresponds to a situation where one source alone has been sending traffic for some time, and suddenly the second source starts also sending traffic, leading to a bottleneck.

Where performance oriented studies typically rely on the assumption that the stochastic process modeling the phenomenon of interest is already in steady state, transient performance results are addressed in this chapter. Transient analysis is important when the life cycle of the phenomenon under study is not long enough, since usually a stochastic process

[1] Throughout this chapter, the terms 'frame' and 'packet' are used interchangeably.

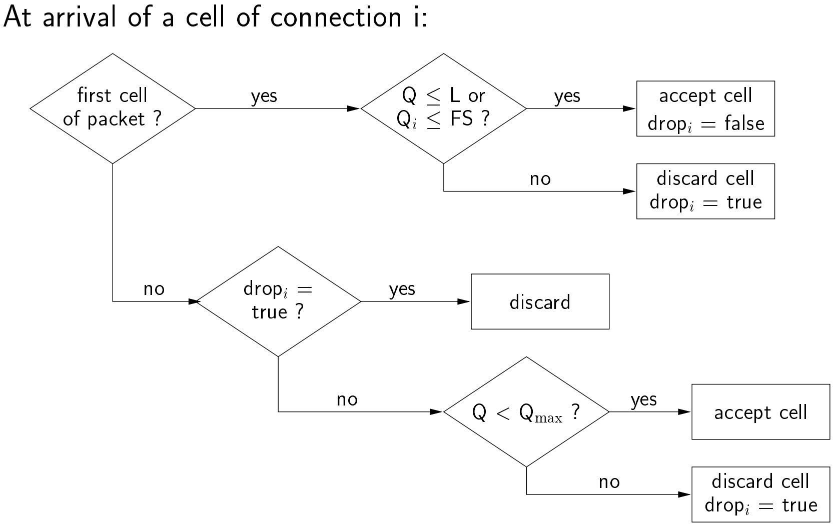
At arrival of a cell of connection i:
first cell of packet ?
yes
Q ≤ L or Q_i ≤ FS ?
yes
accept cell drop_i = false
no
discard cell drop_i = true
no
drop_i = true ?
yes
discard
no
Q < Q_max ?
yes
accept cell
no
discard cell drop_i = true

Figure 6.1: Flowchart of the acceptance rules used by the SD and EPD with per-VC queueing scheme. The following notation is used: Q: buffer occupancy; $Q_{\max}$: capacity of the buffer; Q_i: number of cells of connection i in the buffer; L: a fixed threshold; FS: Fair Share. FS is calculated as $(Q/N) * K$, where N is the number of active connections and K is a fixed parameter of the SD algorithm.

cannot reach steady state unless time evolves towards infinity, or when its behavior before reaching steady state is important. So when observing the reaction upon an unfair start situation of a buffer acceptance scheme which aims at fairness, a transient approach is required.

The results presented in this chapter are an extension to the results we already presented in [92, 93]. The structure of the chapter is as follows: the theoretical model is described in Section 6.1, and results obtained with the model are presented and discussed in Section 6.2. This last section is further subdivided in three subsections: identical scenarios under three different start conditions are considered in Section 6.2.1, while the influence of the responsive traffic, resp. of the SD parameters, under an unfair start situation, is considered in Section 6.2.2, resp. Section 6.2.3. Conclusions are drawn each time at the end of the subsections.

6.1 Model description

6.1.1 System configuration

The performance of the SD buffer acceptance scheme will be observed using the configuration of Figure 6.2. Traffic is generated in fixed-sized packets of cells by two respon-

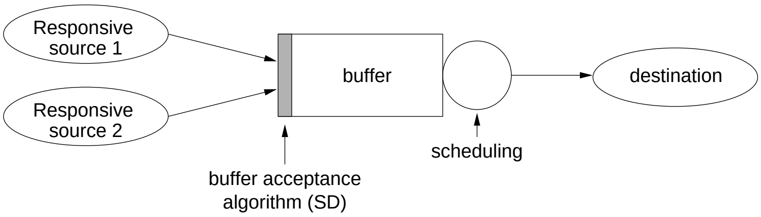

Figure 6.2: System configuration.

sive sources, which respond to the presence/absence of losses of their traffic by decreasing/increasing the amount of packets they send in a certain time. All traffic is sent to the same destination via the output port of a network element. The links in the scenario all have the same capacity, which makes this output port a bottleneck at which buffering is needed. The decision about which packets are allowed to enter the buffer is made using the SD buffer acceptance algorithm. The queueing in the buffer can be global, or per-VC, and the order in which the cells leave the buffer depends on the scheduling algorithm used.

6.1.2 Source behavior

The traffic in the system is generated by two independent but identical sources, which send their traffic in fixed-sized packets of D back-to-back cells (for modeling simplicity it is assumed that D is even). The time needed to place D cells onto the links is considered as time unit of the system, and is called a 'slot'. On the input links, a slot thus equals the time to place a packet of cells onto the links, while on the output link the D cells that may be put onto the link in a slot can belong to both connections, depending on the output of the scheduling algorithm. The sources are persistent sources that have always traffic to send, but the amount of packets they send in a time of x slots (where x is a parameter of the source model) is limited by their window size. The window sizes of the sources are updated every x slots, based on the number of packets a source has lost at the buffer in the previous x slots. The following rules are used for the window updates:

- if a source did not lose any packets during the previous x slots, then its window size is increased by one packet, except if it has already reached its maximum window size of x packets,

- if a source has lost one packet during the previous x slots, then its window size is approximately halved, by setting it to the smallest integer not smaller than half its current window size,

- if a source has lost two or more packets during the previous x slots, then its window size is reduced to one packet.

Further, it is assumed that a source with a window size of r packets ($1 \leq r \leq x$) sends these packets during the first r slots of an interval of x slots.

6.1.3 Buffer acceptance

When a packet arrives at the buffer, the decision about if it is allowed to enter the buffer or not is made based upon the SD buffer acceptance algorithm. Denote by Q_1, resp. Q_2, the number of cells of connection 1, resp. connection 2, in the buffer, by $Q = Q_1 + Q_2$ the total buffer occupancy, by L the fixed threshold of the SD algorithm, by K the fixed parameter of the SD algorithm, and by N the number of active connections. Because of the assumption in the model that the sources send the D cells of a packet back-to-back, packet boundaries correspond to slot boundaries. Since the acceptance rule ($Q \leq L$ or $Q_i \leq (Q/N) * K$) of the SD algorithm is only tested for the first cell of a packet (see Figure 6.1), a decision about the acceptance or discarding of the complete packet can be made in the model at slot boundaries. If packets from both sources arrive at the same time and they both pass the acceptance rules, but there is only place in the buffer for one packet, then it is assumed that each packet has equal probability of being the one that is dropped.

6.1.4 Scheduling

Three scheduling algorithms are considered: FIFO, round robin (RR) and probabilistic longest queue first (PLQF). In a FIFO system, if the D cells of a packet arrive back-to-back at the buffer, D cells of one connection (when *upon arrival* of this packet no packet of the other connection arrived), or $D/2$ cells of each connection (when a packet of both connections arrived at the same time) leave the buffer in a slot. In a RR system on the other hand, when at *departure instants* no cells of the other connections are present, D cells of one connection leave the buffer in a slot. Otherwise, $D/2$ cells of each connection leave the buffer in a slot. The system is also considered with a PLQF scheduling discipline, which selects for service a connection with a probability proportional to the contribution of this connection to the total queue length. Where the aim of RR scheduling is to let an equal amount of cells of each connection leave the buffer per scheduling cycle, PLQF scheduling strives to an equal amount of cells of each connection in the buffer. Corresponding to the FIFO and RR system, also in the PLQF system we let $D/2$ cells of each connection or D cells of one connection leave the system in a slot, and this with the following probabilities:

- $D/2$ cells of each connection, with probability S/Q,
- D cells of connection 1, with probability $\frac{Q_1 - S/2}{Q}$,
- D cells of connection 2, with probability $\frac{Q_2 - S/2}{Q}$,

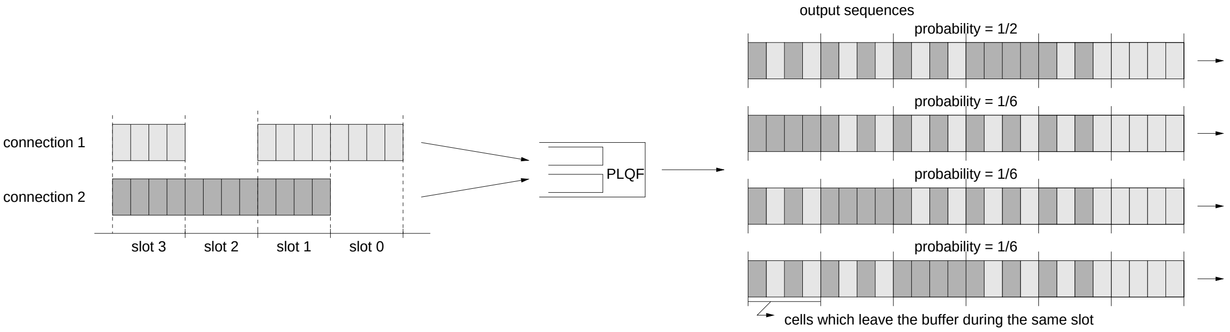

Figure 6.3: In the PLQF system, different output sequences occur probabilistically.

	arrs. slot 0	depts. slot 0	arrs. slot 1	depts. slot 1	arrs. slot 2	depts. slot 2		arrs. slot 3		depts. slot 3		depts. slot 4		depts. slot 5
Q_1	2	0	2	1	1	0	1	2	3	1	2	0	1	0
Q_2	0	0	2	1	3	2	1	4	3	3	2	2	1	0
S	0	0	4	2	2	0	2	4	6	2	4	0	2	0
with prob.	1	1	1	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{5}{6}$	1

Table 6.1: Values of the parameters Q_1, Q_2 and S (expressed as a multiple of $D/2$) at different times for the PLQF system shown in Figure 6.3.

where S is the number of cells in the buffer belonging to packets that have been accepted at the same time as other packets.

To illustrate the probabilistic character of PLQF scheduling and to make the meaning of S more clear, a small example is shown in Figure 6.3 and Table 6.1. Figure 6.3 shows for four slots packets that are accepted in the buffer and the possible output sequences that can occur when PLQF scheduling is applied. Table 6.1 shows the values of the parameters Q_1, Q_2 and S (expressed as a multiple of $D/2$) at different times.

6.1.5 System evolution

Define the following random variables at discrete-time slot boundaries k, where $k = lx+m$, $l \in \mathbb{N}$ and $m \in \{0,\ldots,x-1\}$, for $i = 1,2$:

- $Q_i(k)$: number of cells of connection i in the buffer at time k,

- $W_i(k)$: number of packets source i sends in the interval of length x slots that starts at time lx,

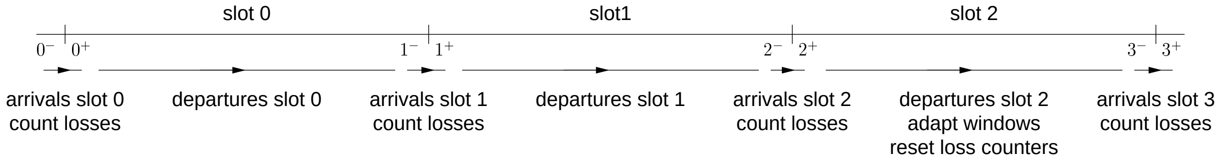

Figure 6.4: Evolution of the system during the first 3 slots for $x = 3$.

- $L_i(k)$: number of lost (i.e., not accepted in the buffer) packets of source i at time k since the beginning of the interval of length x slots that started at time lx (0, 1, or 2, where '2' means 'more than one').

For the PLQF system, define also

- $S(k)$: number of cells in the buffer at time k belonging to packets that have been accepted at the same time as other packets

Remark that because in a slot 0 cells, D cells of one connection or $D/2$ cells of both connections leave the buffer, and because in a slot 0 or D cells of each connection enter the buffer, the number of cells of each connection in the buffer at slot boundaries is always a multiple of $D/2$. So the values the $Q_i(k)$'s can take are always multiples of $D/2$, and the values $S(k)$ can take are always multiples of D. Further, corresponding to the assumption made in Section 6.1.2 that the window sizes of the sources are updated every x slots, the definition of $W_i(k)$ implies that $W_i(lx) = W_i(lx + 1) = \cdots = W_i(lx + x - 1)$.

To simplify the description of the evolution over time of the defined system, split the time instant k virtually into k^- and k^+, where k^- represents the moment just before the arrivals, if any, of slot k occur, and k^+ represents the moment just before the departures, if any, of slot k occur. The result is an evolution $0^- \to 0^+ \to 1^- \to 1^+ \to \cdots \to (k-1)^+ \to k^- \to k^+ \to (k+1)^- \to \ldots$. Two types of evolution over time can then be distinguished:

1. Evolution of the type $k^- \to k^+$: the arrivals in the buffer during slot k are taken into account, resulting in a possible increase of the random variables $Q_i(k^+)$, $S(k^+)$ and $L_i(k^+)$ compared to $Q_i(k^-)$, $S(k^-)$ and $L_i(k^-)$.
2. Evolution of the type $k^+ \to (k+1)^-$: the departures from the buffer during slot k are taken into account, resulting in a possible decrease of the $Q_i\left((k+1)^-\right)$ and $S\left((k+1)^-\right)$ compared to $Q_i(k^+)$ and $S(k^+)$. When $k+1$ is a multiple of x, also the window size $W_i\left((k+1)^-\right)$ of each source is updated from $W_i(k^+)$, using the value $L_i(k^+)$. Further, the loss counters $L_i\left((k+1)^-\right)$ are reset to zero.

Figure 6.4 illustrates for $x = 3$ the evolution of the system during the first three slots.

Remark that the tuple $(W_1(k^-), W_2(k^-))$ provides enough information to know from which connections packets arrive in slot k, while based on $(Q_1(k^-), Q_2(k^-))$ it can be determined by means of the SD algorithm which of these packets are accepted in the buffer. Using the information $(W_1(k^+), W_2(k^+), L_1(k^+), L_2(k^+))$, where $k+1$ is a multiple of x, window updates can be performed. To decide which cells depart from the buffer during slot k, the information that is needed depends on the scheduling algorithm used:

- RR: $(Q_1(k^+), Q_2(k^+))$,
- PLQF: $(Q_1(k^+), Q_2(k^+), S(k^+))$.

For FIFO, which is one of the simplest scheduling schemes to implement, it is necessary to keep track of the order in which the cells of the different connections have entered the buffer. In an analytical model this is difficult to incorporate, since even for two sources this makes the number of states in the model very large, leading to an unattractive model. Because of that, we do not describe the system evolution of the FIFO system theoretically, but obtain it by simulation. In these simulations, the same source behavior and modeling assumptions are used as in the theoretical models for RR and PLQF. A formal description of the system evolution corresponding to the informal description above is now given for the RR and PLQF system.

At time k, the state of the PLQF system is given by a seven dimensional element $(Q_1(k), Q_2(k), S(k), W_1(k), W_2(k), L_1(k), L_2(k))$. The evolution over time of the system is described by a multidimensional discrete-time random process

$$\{(Q_1(k), Q_2(k), S(k), W_1(k), W_2(k), L_1(k), L_2(k)), k \geq 0\}, \tag{6.1}$$

whose future, given the presence, is independent of the past for all time instants k. Hence the process is a Markov chain. It is however a nonstationary Markov chain, since the probability of going from one state to another depends on the time at which the transition is made (multiple of x or not).

Analoguously, the state of the RR system at time k is described by a six dimensional element $(Q_1(k), Q_2(k), W_1(k), W_2(k), L_1(k), L_2(k))$, and the evolution over time of the system is then given by the Markov chain

$$\{(Q_1(k), Q_2(k), W_1(k), W_2(k), L_1(k), L_2(k)), k \geq 0\}. \tag{6.2}$$

In the sequel, when the only difference between an equation for the RR system and for the PLQF system is that the random variable $S(k)$ needs to be omitted for the RR system, as is the case in equations (6.1) and (6.2), only the equation for the PLQF system is written out formally.

Denote the state space of the Markov chains by Ω and define $X_k = (Q_1(k), Q_2(k), S(k), W_1(k), W_2(k), L_1(k), L_2(k))$. The random variables that constitute the multidimensional states X_k take values in the following range (assume that $Q_{\max}$ is a multiple of D):

- $Q_1(k) : 0, D/2, D, 3D/2, \ldots, Q_{\max}$,
- $Q_2(k) : \begin{cases} 0, D, 2D, \ldots, Q_{\max} - Q_1(k), & \text{if } Q_1(k) \text{ is an even multiple of } D/2, \\ D/2, 3D/2, 5D/2, \ldots, Q_{\max} - Q_1(k), & \text{if } Q_1(k) \text{ is an odd multiple of } D/2, \end{cases}$
- $S(k) : \begin{cases} 0, 2D, 4D, \ldots, 2\min\{Q_1(k), Q_2(k)\}, & \text{if } Q_1(k) \text{ is an even multiple of } D/2, \\ D, 3D, 5D, \ldots, 2\min\{Q_1(k), Q_2(k)\}, & \text{if } Q_1(k) \text{ is an odd multiple of } D/2, \end{cases}$
- $W_i(k) : 1, 2, \ldots, x, \qquad \text{for } i = 1, 2,$
- $L_i(k) : 0, 1, 2, \qquad \text{for } i = 1, 2.$

Assuming the probabilities that the system is in a certain state $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \in \Omega$ at time k^- are known, the probability that the system is in a state $(q_1, q_2, s, w_1, w_2, l_1, l_2) \in \Omega$ at time k^+ is calculated using the complete probability formula:

$$\begin{aligned} P\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\} = \\ \sum_{(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \in \Omega} P\left\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2) \mid X_{k^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\} \\ P\left\{X_{k^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\}. \end{aligned} \tag{6.3}$$

For simplicity, denote the conditional probability in equation (6.3) by P_1. Let L be the fixed threshold of the SD algorithm, and denote by $\mathrm{FS}(\hat{q}_1, \hat{q}_2)$ the fair share as calculated by the SD algorithm, i.e.,

$$\mathrm{FS}(\hat{q}_1, \hat{q}_2) = K \frac{\hat{q}_1 + \hat{q}_2}{I_{\{\hat{q}_1 \neq 0\}} + I_{\{\hat{q}_2 \neq 0\}}}, \tag{6.4}$$

where I_A denotes the indicator function of an event A.

To compute P_1, assume that $k = lx + m$, $l \in \mathbb{N}$ and $m \in \{0, \ldots, x-1\}$. Since at time $(k-1)^+$ the departures of slot $k-1$ were taken into account, at time k^- there is always place in the buffer for at least D cells. Different cases can be distinguished:

1. $\hat{w}_1 > m$ and $\hat{w}_2 > m$, i.e., arrivals occur on both connections.
   (a) $\hat{q}_1 + \hat{q}_2 \leq L$, or $\big(\hat{q}_1 + \hat{q}_2 > L$, $\hat{q}_1 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)$ and $\hat{q}_2 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)\big)$, i.e., both packets are accepted.
      i. $\hat{q}_1 + \hat{q}_2 \leq Q_{\max} - 2D$, i.e., there is place in the buffer for both packets. Then $P_1 = 1$ if $(q_1, q_2, s, w_1, w_2, l_1, l_2) = (\hat{q}_1 + D, \hat{q}_2 + D, \hat{s} + 2D, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)$.
      ii. $\hat{q}_1 + \hat{q}_2 = Q_{\max} - D$, i.e., there is place in the buffer for only one packet. Each packet has equal probability of being the one that is dropped. Then $P_1 = 1/2$ if $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, and

- $q_1 = \hat{q}_1 + D$, $q_2 = \hat{q}_2$, $l_1 = \hat{l}_1$, $l_2 = \hat{l}_2 + 1$ or $l_2 = \hat{l}_2 = 2$, or
- $q_1 = \hat{q}_1$, $q_2 = \hat{q}_2 + D$, $l_1 = \hat{l}_1 + 1$ or $l_1 = \hat{l}_1 = 2$, $l_2 = \hat{l}_2$.

(b) $\hat{q}_1 + \hat{q}_2 > L$, $\hat{q}_1 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)$ and $\hat{q}_2 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$, i.e., only the packet of connection 1 is accepted, that of connection 2 is dropped. Then $P_1 = 1$ if $q_1 = \hat{q}_1 + D$, $q_2 = \hat{q}_2$, $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, $l_1 = \hat{l}_1$, $l_2 = \hat{l}_2 + 1$ or $l_2 = \hat{l}_2 = 2$.

(c) $\hat{q}_1 + \hat{q}_2 > L$, $\hat{q}_1 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$ and $\hat{q}_2 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)$, i.e., only the packet of connection 2 is accepted, that of connection 1 is dropped. Then $P_1 = 1$ if $q_1 = \hat{q}_1$, $q_2 = \hat{q}_2 + D$, $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, $l_1 = \hat{l}_1 + 1$ or $l_1 = \hat{l}_1 = 2$, $l_2 = \hat{l}_2$.

(d) $\hat{q}_1 + \hat{q}_2 > L$, $\hat{q}_1 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$ and $\hat{q}_2 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$, i.e., both packets are dropped. Then $P_1 = 1$ if $q_1 = \hat{q}_1$, $q_2 = \hat{q}_2$, $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, $l_1 = \hat{l}_1 + 1$ or $l_1 = \hat{l}_1 = 2$, $l_2 = \hat{l}_2 + 1$ or $l_2 = \hat{l}_2 = 2$.

2. $\hat{w}_1 > m$ and $\hat{w}_2 \leq m$, i.e., only on connection 1 an arrival occurs.

(a) $\hat{q}_1 + \hat{q}_2 \leq L$, or $\left(\hat{q}_1 + \hat{q}_2 > L \text{ and } \hat{q}_1 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)\right)$, i.e., the packet is accepted. Then $P_1 = 1$ if $(q_1, q_2, s, w_1, w_2, l_1, l_2) = (\hat{q}_1 + D, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)$.

(b) $\hat{q}_1 + \hat{q}_2 > L$ and $\hat{q}_1 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$, i.e., the packet is dropped. Then $P_1 = 1$ if $q_1 = \hat{q}_1$, $q_2 = \hat{q}_2$, $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, $l_1 = \hat{l}_1 + 1$ or $l_1 = \hat{l}_1 = 2$, $l_2 = \hat{l}_2$.

3. $\hat{w}_1 \leq m$ and $\hat{w}_2 > m$, i.e., only on connection 2 an arrival occurs.

(a) $\hat{q}_1 + \hat{q}_2 \leq L$, or $\left(\hat{q}_1 + \hat{q}_2 > L \text{ and } \hat{q}_2 \leq \mathrm{FS}(\hat{q}_1, \hat{q}_2)\right)$, i.e., the packet is accepted. Then $P_1 = 1$ if $(q_1, q_2, s, w_1, w_2, l_1, l_2) = (\hat{q}_1, \hat{q}_2 + D, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)$.

(b) $\hat{q}_1 + \hat{q}_2 > L$ and $\hat{q}_2 > \mathrm{FS}(\hat{q}_1, \hat{q}_2)$, i.e., the packet is dropped. Then $P_1 = 1$ if $q_1 = \hat{q}_1$, $q_2 = \hat{q}_2$, $s = \hat{s}$, $w_1 = \hat{w}_1$, $w_2 = \hat{w}_2$, $l_1 = \hat{l}_1$, $l_2 = \hat{l}_2 + 1$ or $l_2 = \hat{l}_2 = 2$.

4. $\hat{w}_1 \leq m$ and $\hat{w}_2 \leq m$, i.e., no arrivals occur. Then $P_1 = 1$ if $(q_1, q_2, s, w_1, w_2, l_1, l_2) = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)$.

In all other cases, $P_1 = 0$.

To compute the probabilities that the system is in a certain state $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \in \Omega$ at time k^-, again the complete probability formula is used:

$$P\left\{X_{k^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\} = \sum_{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega} P\left\{X_{k^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \mid X_{(k-1)^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\} P\left\{X_{(k-1)^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\}. \quad (6.5)$$

Denote the conditional probability in equation (6.5) by P_2, and assume that $k = lx + m$, $l \in \mathbb{N}$ and $m \in \{0, \ldots, x-1\}$. Remark that when $m \neq 0$, only the departures of slot $k-1$ are taken into account. When $m = 0$, also the window sizes are adapted and the loss counters are reset. Define by f the window update function that corresponds to the source behavior rules described in Section 6.1.2:

$$f(w_i, l_i) = \begin{cases} \min\{w_i + 1, x\} & \text{if } l_i = 0, \\ \lceil \frac{w_i}{2} \rceil & \text{if } l_i = 1, \\ 1 & \text{if } l_i = 2. \end{cases} \tag{6.6}$$

Again, different cases can be distinguished:

1. $q_1 + q_2 = 0$, i.e., the system is empty. Then $P_2 = 1$ if $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) = (q_1, q_2, s, w_1, w_2, l_1, l_2)$.
2. $q_1 + q_2 \neq 0$, i.e., the system is not empty. If
   - $m \neq 0$, $\hat{w}_1 = w_1$, $\hat{w}_2 = w_2$, $\hat{l}_1 = l_1$, $\hat{l}_2 = l_2$, or
   - $m = 0$, $\hat{w}_1 = f(w_1, l_1)$, $\hat{w}_2 = f(w_2, l_2)$, $\hat{l}_1 = 0$, $\hat{l}_2 = 0$,

   then

   (a) for PLQF scheduling,

   i. $P_2 = s/(q_1 + q_2)$, i.e., $D/2$ cells of each connection leave the buffer, if $\hat{q}_1 = q_1 - D/2$, $\hat{q}_2 = q_2 - D/2$, $\hat{s} = s - D$.

   ii. $P_2 = (q_1 - s/2)/(q_1 + q_2)$, i.e., D cells of connection 1 leave the buffer, if $\hat{q}_1 = q_1 - D$, $\hat{q}_2 = q_2$, $\hat{s} = s$.

   iii. $P_2 = (q_2 - s/2)/(q_1 + q_2)$, i.e., D cells of connection 2 leave the buffer, if $\hat{q}_1 = q_1$, $\hat{q}_2 = q_2 - D$, $\hat{s} = s$.

   (b) for RR scheduling, $P_2 = 1$ if

   i. $q_1 . q_2 \neq 0$, $\hat{q}_1 = q_1 - D/2$, $\hat{q}_2 = q_2 - D/2$, i.e., the buffer contains cells of both connections, so $D/2$ cells of each connection leave the buffer.

   ii. $\hat{q}_1 = q_1 - D$, $\hat{q}_2 = q_2 = 0$, i.e., the buffer contains only cells of connection 1, so D cells of that connection leave the buffer.

   iii. $\hat{q}_1 = q_1 = 0$, $\hat{q}_2 = q_2 - D$, i.e., the buffer contains only cells of connection 2, so D cells of that connection leave the buffer.

In all other cases, $P_2 = 0$.

Using alternately the equations (6.3) and (6.5), the probabilities that the system is in a certain state of Ω at time k^- or k^+ can be calculated when starting values $P\{X_{0^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\}$ at time 0^- for all states $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \in \Omega$ are given. Remark that by definition of the random variables $L_1(k)$ and $L_2(k)$, $L_1(0^-) = L_2(0^-) = 0$, i.e., the starting probabilities $P\{X_{0^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\}$ should be zero when $\hat{l}_1 \neq 0$ or when $\hat{l}_2 \neq 0$.

6.1.6 Transient performance measures

From the state of the system at time k^+, the following random variables ($i = 1, 2$) are obtained:

- $O_i(k)$: number of cells of connection i that leave the buffer during slot k.

Remark that the $O_i(k)$'s take values 0, $D/2$ or D. Further on in this section, $E\left[O_i(k)\right]$ is used, which is the average over all realizations of the random process $O_i(k)$, and is thus per definition given by

$$E\left[O_i(k)\right] = (D/2)\, P\left\{O_i(k) = D/2\right\} + D\, P\left\{O_i(k) = D\right\}. \tag{6.7}$$

For the systems with FIFO and RR scheduling, due to the fact that this are non-probabilistic scheduling schemes, it is possible that the random processes $O_i(k)$ are deterministic (i.e., have only one realization), such that $E\left[O_i(k)\right] = O_i(k)$. This is the case when the initial state of the system is deterministic, and over time it *never* occurs that two packets that arrive at the same time are both accepted by the acceptance rules, while there is only place in the buffer for one packet. When the latter would happen anyhow, the sample paths of the processes $O_i(k)$ split into two branches at such moments.

For the PLQF system, the probability that $O_i(k)$ equals $D/2$ is the probability that during slot k, $D/2$ cells of each connection leave the buffer:

$$P\left\{O_i(k) = D/2\right\} = \sum_{\substack{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega \\ q_1+q_2\neq 0}} \frac{s}{q_1+q_2}\, P\left\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\}. \tag{6.8}$$

The probability that $O_i(k)$ equals D with PLQF scheduling is given by the probability that D cells of connection i leave the buffer during slot k:

$$P\left\{O_i(k) = D\right\} = \sum_{\substack{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega \\ q_1+q_2\neq 0}} \frac{q_i - s/2}{q_1+q_2}\, P\left\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\}. \tag{6.9}$$

With RR scheduling, the probability that $O_i(k)$ equals $D/2$ is the probability that at time k^+ the buffer contains cells of both connections:

$$P\left\{O_i(k) = D/2\right\} = \sum_{\substack{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega \\ q_1+q_2\neq 0,\, q_1.q_2\neq 0}} P\left\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\}. \tag{6.10}$$

The probability that at time k^+ the buffer contains only cells of connection i equals the probability that $O_i(k) = D$ for the RR system:

$$P\left\{O_i(k) = D\right\} = \sum_{\substack{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega \\ q_1+q_2\neq 0,\, q_1.q_2=0,\, q_i\neq 0}} P\left\{X_{k^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\}. \tag{6.11}$$

For the FIFO system, the probability distribution of the $O_i(k)$'s is obtained by simulation.

Both the efficiency and fairness performance of the system in transient state are of interest, and are obtained from the effective throughputs $T_i(k)$ of the connections after k slots. Since the effective throughput of a connection is defined as the average number of packets of that connection that have arrived at the destination, divided by the time needed to deliver these packets, $T_i(k)$ is calculated as

$$T_i(k) = \frac{1}{Dk} \sum_{j=0}^{k-1} E\left[O_i(j)\right]. \qquad (6.12)$$

Remark the factor 'D' in the denominator, since we want the throughput to be expressed in packets per slot time.

The efficiency after k slots is defined as the sum of the effective throughputs of all connections after k slots, divided by the maximum possible effective throughput after k slots (which is one packet per slot time), resulting in

$$\text{efficiency}(k) = T_1(k) + T_2(k). \qquad (6.13)$$

To decide about the fairness performance of the system, the fairness index of equation (4.2) is used. Since the equal division of the total effective throughput among both connections is considered as the perfectly fair situation, the fairness index after k slots, denoted by $F(k)$, reduces to

$$F(k) = \frac{\left(T_1(k) + T_2(k)\right)^2}{2\left(T_1(k)\right)^2 + 2\left(T_2(k)\right)^2}. \qquad (6.14)$$

Remark that for two sources, $F(k)$ ranges between one half (minimum fairness) and one (maximum fairness).

6.2 Numerical results and discussion

6.2.1 Different start conditions

The scenarios in this section are all considered with the following three deterministic start conditions: $P\{X_{0^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\} = 1$, where

1. $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) = (0, 0, 0, 1, 1, 0, 0)$,
2. $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) = (0, 0, 0, x, x, 0, 0)$,
3. $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) = (0, 0, 0, x, 1, 0, 0)$.

Start condition 1 corresponds to a set-up where the two sources start probing the network at the same time with a window size of one packet. The buffer of the system is empty at that time. Under start condition 2, the two sources start also sending traffic to an empty system at the same time, but now they do it in a very aggressive way, by both starting at their maximum window size. With start condition 3, the window setting is the most unfair situation possible, but also the most realistic one. Start condition 3 can be seen as the result of a situation where only one source is sending traffic, and because there is no bottleneck then, all traffic of this source passes through the system without building up a queue and without losses. The window size of this source can thereby grow until its maximum. At time 0^-, the second source starts also sending traffic, starting with a window size of one packet.

Scenario 6.2.1. Consider a system with following parameters:

- $x = 10$ (slots), $Q_{\max} = 12 \times D$ (cells), $L = 7 \times D$ (cells), $K = 1$,
- PLQF scheduling.

The evolution over time of the mean window size of the two sources and the mean buffer occupation of the two connections under the three start conditions is shown in Figures 6.5 and 6.6. Because the input traffic in the system is generated by two identical sources, none of which is offered a preferential treatment by the buffer acceptance or the scheduling scheme, the *mean* window sizes and the *mean* buffer occupations coincide under identical start values for both connections, although of course the two window sizes and the two buffer occupations at time k are often different. This is not only the case in this example, but is true in general, as is shown in the appendix. For start condition 3, there is a difference in the start value of the two window sizes. In Figures 6.5 and 6.6, a difference between the curves of both connections is clearly seen at the beginning, while afterwards the curves for both connections become more and more the same (i.e., the differences between the curves become invisible on the plots after approximately 500 slots). It is observed very often in scenarios with PLQF scheduling that the curves for the mean window size and the mean queue size of both connections coincide more and more when time progresses. This can be explained as follows: when at time k the mean buffer occupation and window size of both connections would be calculated over only these sample paths on which it occurred at a certain time instant l before k that the parameters (buffer occupation, window size, loss counter) of both connections were the same, then these means would be identical for both connections (cfr. property 6.3.1). The more time progresses, the more likely it becomes that it has happened on more and more sample paths that the parameters of both connections were the same once, and thus that the difference between the means for connection 1 and connection 2 become smaller. This is of course true for all scheduling schemes considered, but since PLQF scheduling aims at equal buffer occupation for all connections, the probability that the parameters of the connections come together at a certain time instant is higher with PLQF scheduling.

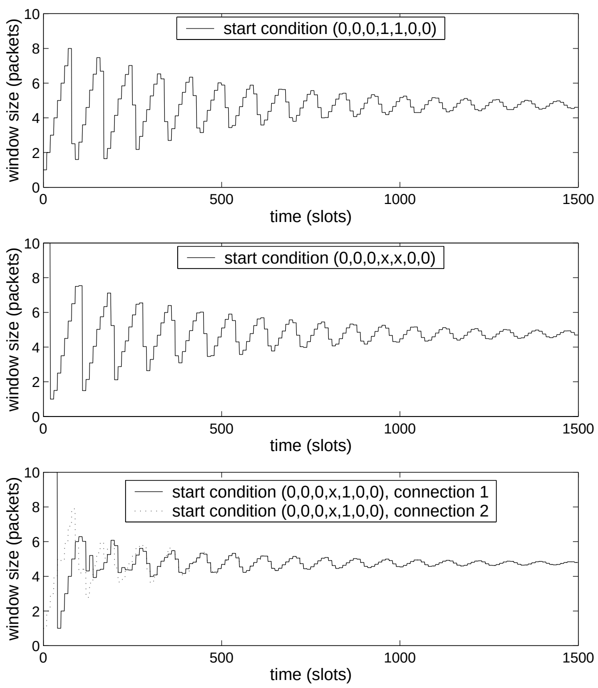

Figure 6.5: Evolution of the mean window sizes when $x = 10$, $Q_{\max} = 12 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (PLQF scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

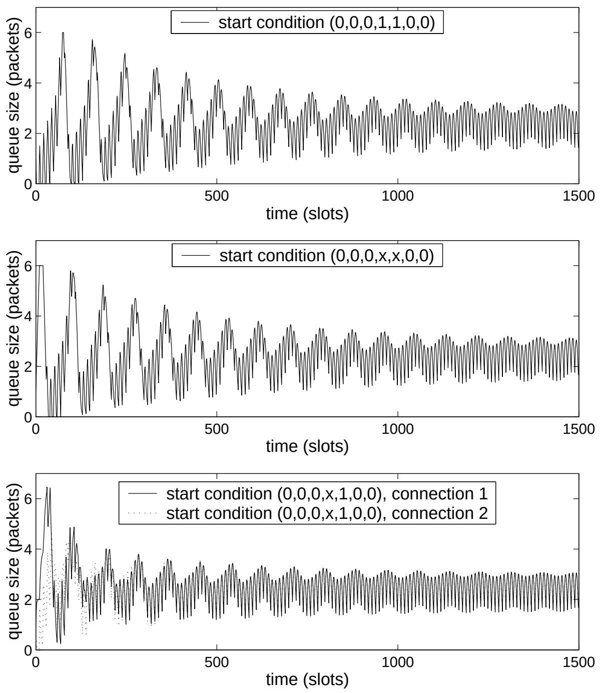

Figure 6.6: Evolution of the mean buffer occupations when $x = 10$, $Q_{\max} = 12 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (PLQF scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

Figure 6.5 shows clearly the oscillating behavior of the window size of the sources, which is caused by their responsiveness: the window is allowed to grow as long as no losses occur, and it is reduced after losses. The window size of a source keeps on oscillating (except if it is allowed to stay at its maximum value, which occurs when there is no bottleneck in the system), because the source's behavior is such that it keeps on trying to let its window grow. Remark again that Figure 6.5 does not show the evolution of the window sizes $W_i(k)$ over time, but $E[W_i(k)]$. In the beginning, $E[W_i(k)]$ is equal to $W_i(k)$, since the start conditions used are deterministic ones (one start vector with probability 1), but during time more and more sample paths are explored because of the probabilistic character of the system. From Figure 6.5 it is also seen that for the different start conditions, the mean window sizes oscillate around the same value ($4.7 \times D$ cells) in the long run.

In Figure 6.6 it is seen that the oscillating behavior of the windows of the sources is reflected in the low-frequency oscillations of the queue sizes. The high-frequency oscillations of the queue sizes have a length of x slots, and are caused by the fact that when a source has a window size of r packets, it sends packets during the first r slots of an interval of x slots, letting the queue grow during these slots and go down afterwards when packets leave but

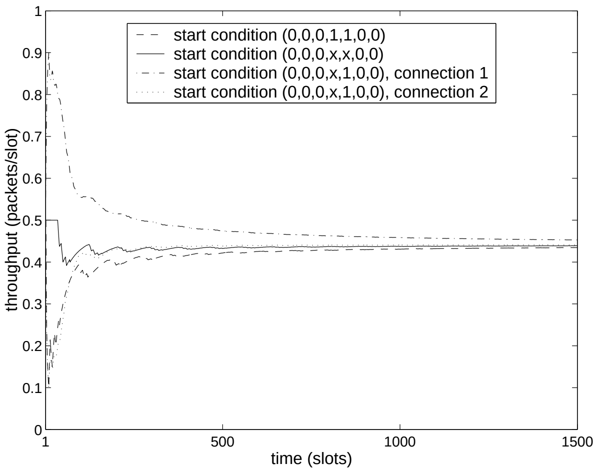

Figure 6.7: Evolution of the throughputs when $x = 10$, $Q_{\max} = 12 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (PLQF scheduling). For start conditions $(0, 0, 0, 1, 1, 0, 0)$ and $(0, 0, 0, x, x, 0, 0)$, the curves of connection 1 and 2 coincide.

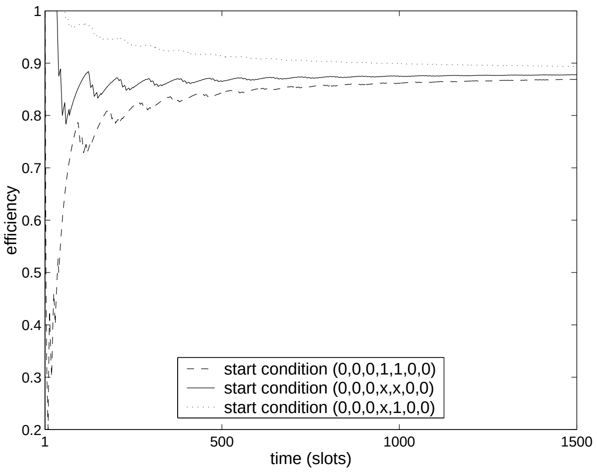

Figure 6.8: Evolution of the efficiency when $x = 10$, $Q_{\max} = 12 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (PLQF scheduling).

do not arrive.

Figure 6.7 shows the evolution over time of the throughput of the different connections. Because of the identical mean output of the two connections under start conditions 1 and 2, their throughput curves coincide. Under start condition 1, the throughput of a connection is lower than under start condition 2, especially in the beginning. This difference is mainly caused by the difference in the mean number of packets that leave the system during the first slots. This number is clearly larger with start condition 2, since then the window sizes and consequently the buffer occupations are larger. Under start condition 1, the system is under-utilized in the beginning (remark from Figure 6.6 that the mean buffer occupation becomes often zero then), because the sources need time to build up their window. In line with definition (6.12), this difference in output of the system at the beginning stays perceptible for some while in the throughput values. When comparing the throughput of the two connections under start condition 3, it is seen that in the beginning the throughput of connection 1 is higher than that of connection 2, since then source 1 sends more packets than source 2, and they are all accepted, at least until $Q > L$. Figures 6.6 and 6.7 together illustrate clearly how an initial difference between the output from the system of both connections stays perceptible in the throughput values: the buffer occupations of both connections coincide from a certain moment on, while this is not the case for the throughput values. It is however the buffer occupation which determines with some delay the output values, since all packets that are accepted in the buffer also leave the buffer (packets that do not pass the acceptance rules of the acceptance scheme do not enter the buffer either).

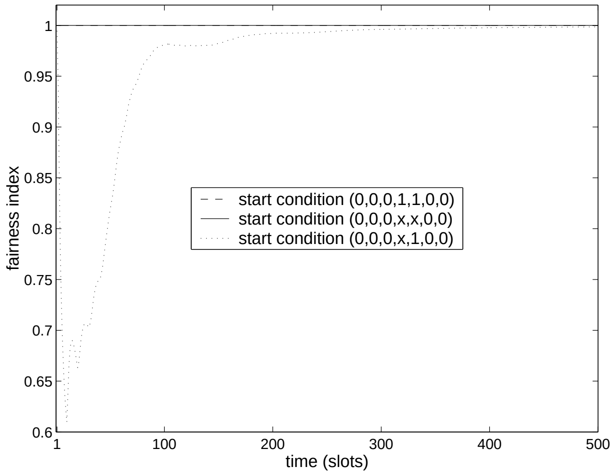

Figure 6.9: Evolution of the fairness index when $x = 10$, $Q_{\max} = 12 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (PLQF scheduling). For start conditions $(0, 0, 0, 1, 1, 0, 0)$ and $(0, 0, 0, x, x, 0, 0)$ the curves coincide.

The evolution of the efficiency over time under the three start conditions is shown in Figure 6.8. The efficiency obtained with start condition 3 is the highest, while that obtained with start condition 1 is the lowest. Remark that for start conditions 1 and 2 the efficiency is exactly twice the throughput of a connection, since the throughput for both connections coincides. This explains why the efficiency is higher under start condition 2 than under start condition 1. Under start condition 3, the efficiency is even higher in the beginning, because while the window of source 2 is still growing, source 1 sends enough traffic into the system to keep its utilization high. When the window of source 1 needs to go down, that of source 2 is large enough to keep the system at high utilization. The fact that the efficiency curves are below one from a certain moment on indicates that on some of the sample paths the system gets temporarily empty with a strict positive probability: formulas (6.12) and (6.13) indicate that for efficiency(k) to be equal to one, the mean output of the system needs to be D cells in all slots until slot k, and since the maximum output per slot is D cells, the mean can only be D cells if the output is D cells with probability 1 (i.e., the system is always non-empty with probability 1). The moment that the efficiency drops below one occurs first for start condition 1, then for start condition 2 and the latest for start condition 3. This explains why the first dropping of the efficiency is the largest for start condition 1 and the smallest for start condition 3: the longer the efficiency stays one, the smaller the difference between the numerator and the denominator of equation (6.12) and consequently the larger the efficiency is when the average output becomes smaller than one for the first time.

The evolution of the fairness index is shown in Figure 6.9. For start conditions 1 and 2 this index is constantly one, since under these start conditions the throughput of the two connections is exactly the same. With start condition 3, some time is needed to approach a fairness index close to one. Remark that the results are only shown for the first 500 time slots, since as soon as the system recovers from the unfairness caused by the unfair start situation, the fairness index approaches one, since the behavior of the two sources is the same, and they are treated equally by the buffer acceptance and the scheduling algorithm. This illustrates the importance of a transient analysis when observing the behavior of the SD scheme towards an unfair start situation. In real systems, unfair (start)situations are constantly created when connections appear and disappear.

Figures 6.8 and 6.9 illustrate that the efficiency and the fairness of a scenario give complementary information about the throughput of the different connections. Efficiency looks at how well the outgoing capacity of the system is used, without caring by which connection it is used, while fairness looks at how fair the outgoing capacity is used by the different connections, independent of how much or how little of the outgoing capacity is used.

To illustrate that the observations made about scenario 6.2.1 under the three different start conditions are more generally valid, a lot of other scenarios were considered, two of which are added here for further illustration.

Scenario 6.2.2. Consider a system with following parameters:

- $x = 13$ (slots), $Q_{\max} = 15 \times D$ (cells), $L = 7 \times D$ (cells), $K = 1$,
- RR scheduling.

In this scenario, the evolution of the system is deterministic under the third start condition. Under the first and second start condition, there are each time two sample paths that are symmetric with respect to connection 1 and 2. These two sample paths coincide until the first time that the buffer overflows, because the SD acceptance rules are always fulfilled until then due to the identical behavior of both connections. When the buffer overflows, the sample path splits into two symmetrical sample paths with equal probability. Afterwards, the behavior of the two connections differs, and the SD algorithm is always able to force the window of one of the connections to go down on time such that the buffer never overflows again and the sample paths never split again. A problem of the SD algorithm is illustrated here: when there are constantly a fair amount of cells of each connection present in the buffer, the SD algorithm does not discard packets, and so it cannot be avoided that the buffer occupation grows until the buffer overflows. Of course this problem is less severe in reality than in the model. In the model it is assumed that packets arrive at slot boundaries and that the windows of both sources are updated at the same time, whereas in reality there will be some jitter in the arrival of the packets at the network elements and in the updating of the windows of the sources.

Figure 6.10 shows the evolution of the mean window sizes of the sources, and Figure 6.11 that of the mean buffer occupations of the connections. As can be seen from these figures, the evolutions of the average window and queue sizes with the first and second start condition become the same after a while. More in particular, the average behavior of the system with start condition 1 at time k is identical to the average behavior of the system

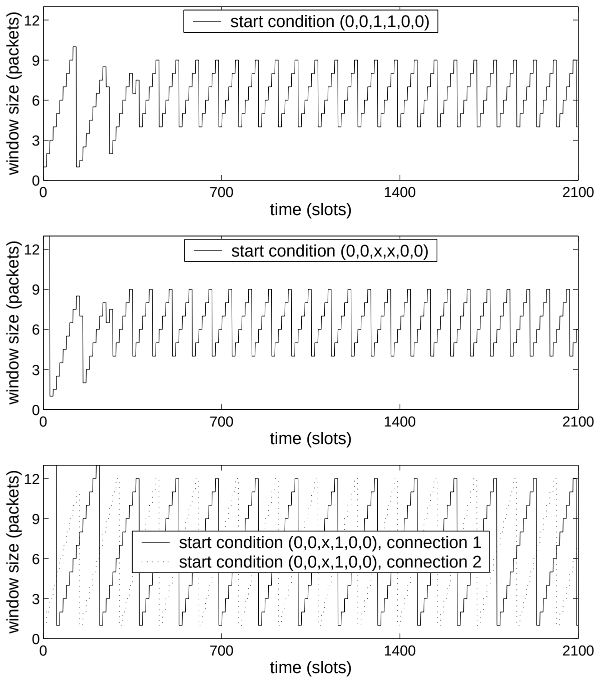

Figure 6.10: Evolution of the mean window sizes when $x = 13$, $Q_{\max} = 15 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (RR scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

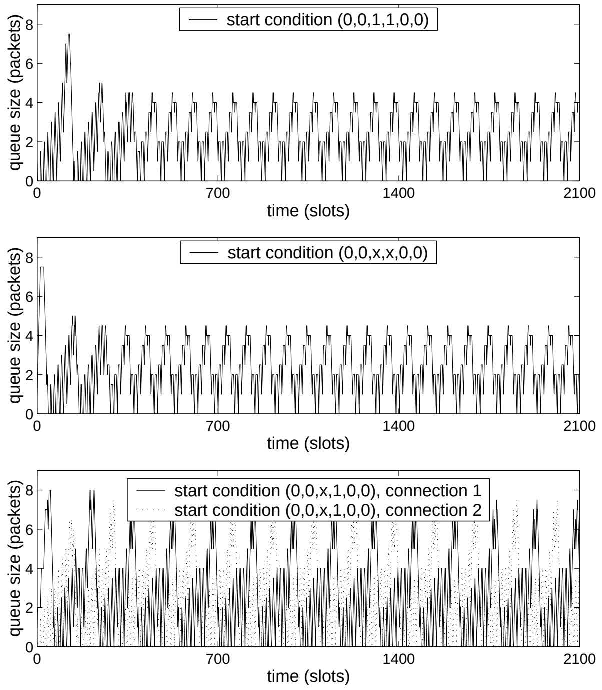

Figure 6.11: Evolution of the mean buffer occupations when $x = 13$, $Q_{\max} = 15 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (RR scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

with start condition 2 at time $k - 104$, for all $k \geq 157$. From Figures 6.12 and 6.13 it is seen that the throughput of the connections and the efficiency is higher under start condition 2 than under start condition 1. Since the behavior of both systems becomes the same with some delay, this illustrates again the influence of a difference in output from the system at the beginning. The efficiency is the highest with start condition 3, since then it occurs only rarely that the buffer is empty, and there is thus only rarely no output during some slots. Figure 6.14 shows the evolution of the fairness index. Again, the fairness is constantly one for equal start values, and approaches one after some time when the start values are not equal. Remark that the fairness index always approaches one when the behavior of the two sources is the same, and they are treated equally by the buffer acceptance and the scheduling algorithm. So one could wonder what the influence of the SD algorithm is. Because of that, the evolution of the fairness index with the third start condition, but now without the implementation of the SD algorithm, is also shown in Figure 6.14. As can be seen, when SD is not implemented, it takes much more time before the fairness index approaches one.

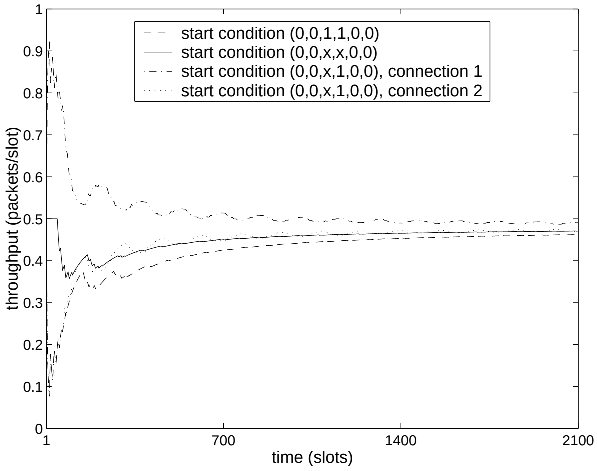

Figure 6.12: Evolution of the throughputs when $x = 13$, $Q_{\max} = 15 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (RR scheduling). For start conditions $(0, 0, 1, 1, 0, 0)$ and $(0, 0, x, x, 0, 0)$, the curves of connection 1 and 2 coincide.

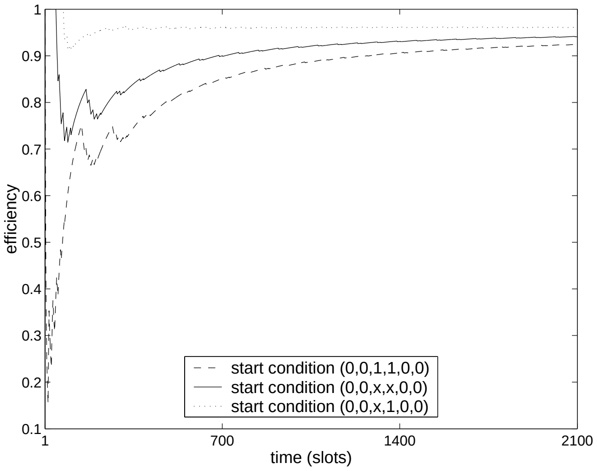

Figure 6.13: Evolution of the efficiency when $x = 13$, $Q_{\max} = 15 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (RR scheduling).

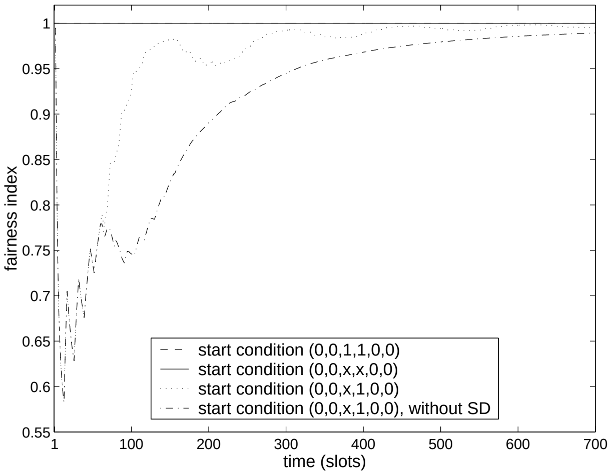

Figure 6.14: Evolution of the fairness index when $x = 13$, $Q_{\max} = 15 \times D$, $L = 7 \times D$ and $K = 1$ under different start conditions (RR scheduling). For start conditions $(0, 0, 1, 1, 0, 0)$ and $(0, 0, x, x, 0, 0)$ the curves coincide. For the third start condition, also a comparison with the fairness index when SD is not implemented is shown.

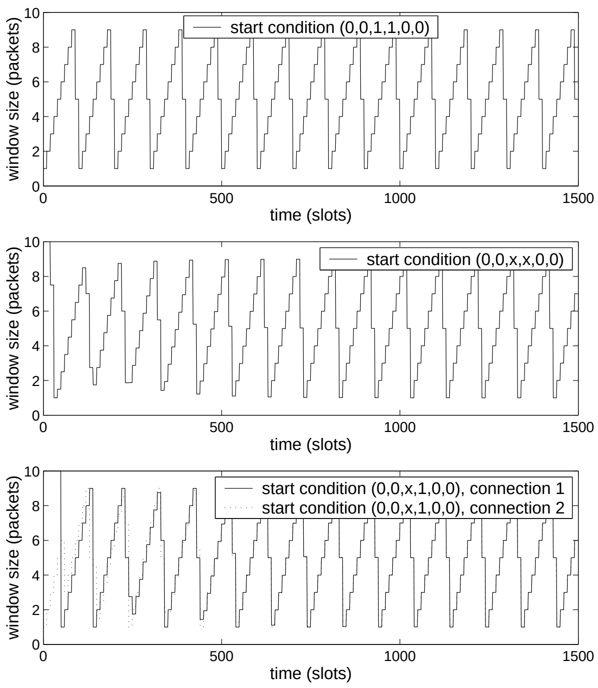

Figure 6.15: Evolution of the mean window sizes when $x = 10$, $Q_{\max} = 20 \times D$, $L = 13 \times D$ and $K = 1$ under different start conditions (FIFO scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

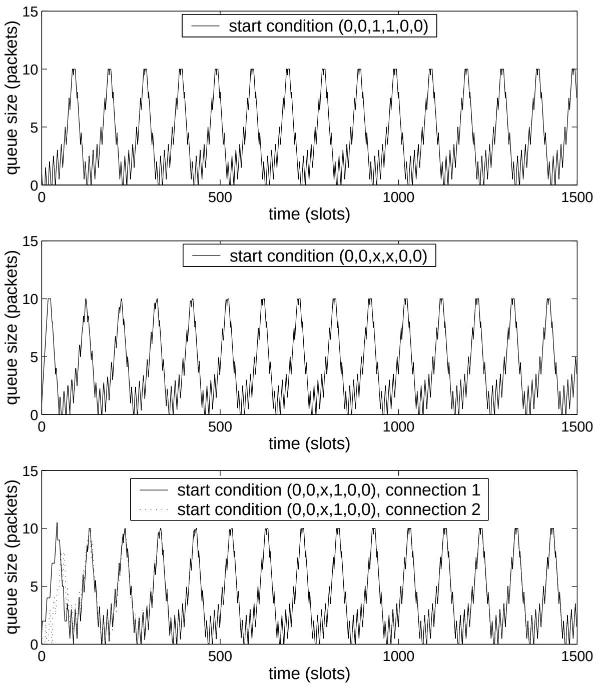

Figure 6.16: Evolution of the mean buffer occupations when $x = 10$, $Q_{\max} = 20 \times D$, $L = 13 \times D$ and $K = 1$ under different start conditions (FIFO scheduling). In the two topmost plots, the curves of connection 1 and 2 coincide.

Scenario 6.2.3. Consider a system with following parameters:

- $x = 10$ (slots), $Q_{\max} = 20 \times D$ (cells), $L = 13 \times D$ (cells), $K = 1$,
- FIFO scheduling.

Figures 6.15 until 6.18 show the evolution of the window sizes, the buffer occupations, the throughput and the efficiency obtained with this system under the three start conditions. Again it can be seen from the figures that under start condition 3, the throughput of connection 1 is higher than that of connection 2, because of its initial higher window, which makes that source 1 can send more packets than source 2 during the initial period where all packets are accepted because the total buffer occupation is not yet above L. A same explanation can be given to the fact that the efficiency is the highest under start condition 3, and the lowest under start condition 1. Remark that the efficiency stays equal to one under the second and third start condition for a much longer time in this scenario than in the previous scenarios, because enough packets (more than x per x slots) are sent in the beginning to let the queue grow, and since L and $Q_{\max}$ are larger here than in

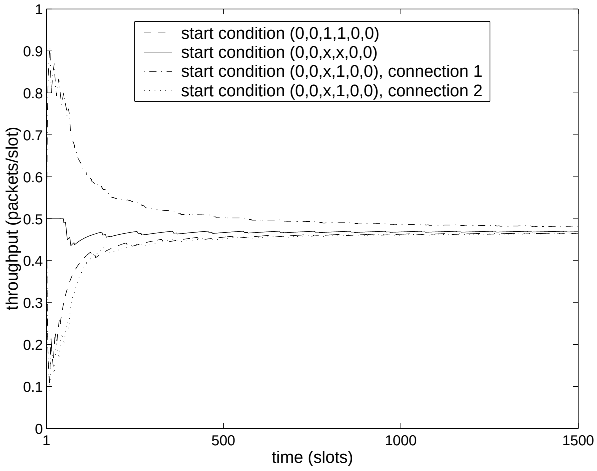

Figure 6.17: Evolution of the throughputs when $x = 10$, $Q_{\max} = 20 \times D$, $L = 13 \times D$ and $K = 1$ under different start conditions (FIFO scheduling). For start conditions $(0, 0, 1, 1, 0, 0)$ and $(0, 0, x, x, 0, 0)$, the curves of connection 1 and 2 coincide.

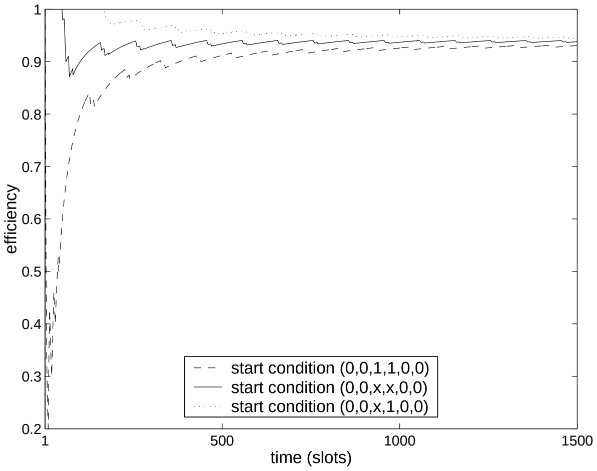

Figure 6.18: Evolution of the efficiency when $x = 10$, $Q_{\max} = 20 \times D$, $L = 13 \times D$ and $K = 1$ under different start conditions (FIFO scheduling).

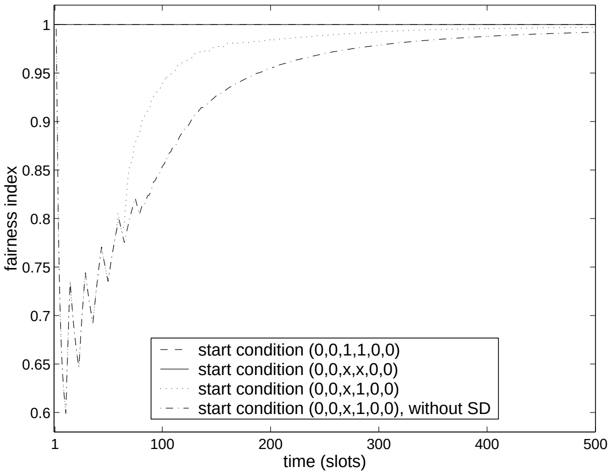

Figure 6.19: Evolution of the fairness index when $x = 10$, $Q_{\max} = 20 \times D$, $L = 13 \times D$ and $K = 1$ under different start conditions (FIFO scheduling). For start conditions $(0, 0, 1, 1, 0, 0)$ and $(0, 0, x, x, 0, 0)$ the curves coincide. For the third start condition, also a comparison with the fairness index when SD is not implemented is shown.

the other scenarios, which means that more packets can be buffered to keep the efficiency longer equal to one.

Figure 6.19 shows the evolution of the fairness index under the three start conditions, and also for the same system without SD implemented under the third start condition. It is again seen that under start conditions 1 and 2 the fairness is perfect because of the equal throughput of both connections. Under the third start condition, the fairness index approaches one sooner when SD is implemented. So the SD algorithm clearly influences the fairness results in a positive way. However, the problem of the SD algorithm mentioned already in scenario 6.2.2 (when there are a fair amount of cells of each connection present in the buffer, the SD algorithm cannot discard packets, and so it occurs that the buffer grows until it overflows) appears also in this scenario, as can be seen from Figure 6.16. It happens that the mean buffer occupation of both connections together equals $Q_{\max}$, which means that on all sample paths the buffer occupation of both connections together equals $Q_{\max}$ at these times. This suggests that the buffer has overflowed on these times, which is only possible when there are a fair amount of cells of each connection present in the buffer from the moment that Q exceeds L until it reaches $Q_{\max}$.

Conclusions

The conclusions of this section are that when the input traffic is generated by two identical sources, none of which is offered a preferential treatment by the buffer acceptance or the scheduling scheme, then

- The *mean* window sizes and the *mean* buffer occupations coincide under identical start values for both connections, resulting in equal throughput for both connections and thus perfect fairness.

- The fairness index approaches one as soon as the system has recovered from the unfairness caused by an unfair start situation. This illustrates the importance of a *transient* analysis when observing the behavior of the SD scheme towards an unfair start situation.

- A difference in the amount of output from the buffer at the beginning due to different start conditions for the system stays perceptible in the efficiency values. A difference in the amount of output of the two connections at the beginning due to unequal start values for both connections stays perceptible for some while in the throughput and fairness values.

From now on, all scenarios are considered with start condition 3, to observe the behavior of the SD scheme towards an unfair start situation.

6.2.2 Influence of the responsive traffic

In this section it is illustrated with numerical examples that due to the responsiveness of the sources, it is not necessarily true anymore that being more conservative in accepting packets implies a lower efficiency, as would be the case when non-responsive sources would be used. As a result, there is not necessarily a trade-off between efficiency and fairness, as is also illustrated by the examples. All examples are considered with start condition 3 of the previous subsection.

Scenario 6.2.4. Consider the two systems with following parameters:

- $x = 10$ (slots), $Q_{\max} = 12 \times D$ (cells),
- FIFO scheduling,
- (1) with SD implemented: $L = 7 \times D$ (cells), $K = 1$, (2) without SD implemented.

When packets arrive at the first system, they are accepted as long as the SD acceptance rules are fulfilled, while in the second system they are accepted as long as there is place in the buffer. Figure 6.20 shows the efficiency obtained with both systems. The highest efficiency is obtained when SD is implemented, so when being more conservative in accepting packets. The evolution of the first system is deterministic, while that of the second system is not. So the efficiency of the system without SD is the average efficiency over all sample paths. The efficiency of two such sample paths is shown in Figure 6.21. One of these paths is a *most-likely* path, which is a path obtained by following always the branch that has the highest probability associated with it (or one of these branches when there are more of them) when the sample path of the system evolution splits.

The evolution of the window sizes corresponding to the two sample paths of Figure 6.21 is shown in Figures 6.22 and 6.23. Figure 6.24 shows the evolution of the mean window sizes for the system without SD implemented, and Figure 6.25 that for the system with SD. Figure 6.24, together with the Figures 6.22 and 6.23, illustrates clearly that although the mean window sizes of both connections coincide after a while and stay almost constant in the long run, this is certainly not the case on the single sample paths. That the mean window sizes for both connections stay different and highly variable for the system with SD implemented (Figure 6.25) is because this system is deterministic, such that the mean window sizes correspond to the window sizes on the single sample path that occurs. It can be seen that in the system with SD, when the window of a source goes down, it is most of the time only halved, which indicates that only one packet of the source was dropped during the previous x slots. In the system without SD on the other hand, the windows generally can grow larger, but both windows afterwards go down at the same moment, often both to a size of one, which indicates that two or more packets per connection were lost during the previous x slots. The result of this is that during the time the windows need to grow again, the buffer which first overflowed becomes empty, which results in a decrease of efficiency, since there is no output during some slots.

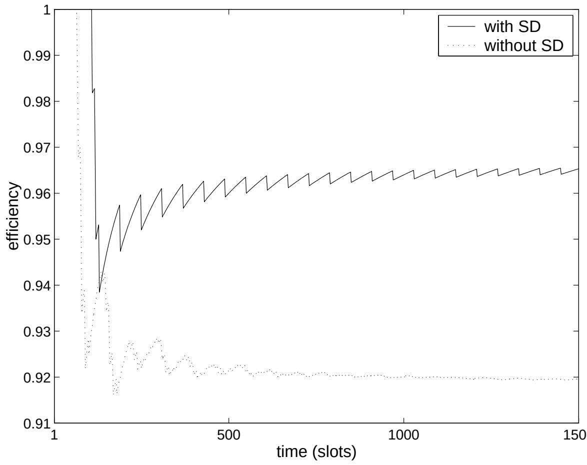

Figure 6.20: Evolution of the efficiency when $x = 10$, $Q_{\max} = 12 \times D$, with and without SD implemented (FIFO scheduling). SD parameters: $L = 7 \times D$, $K = 1$.

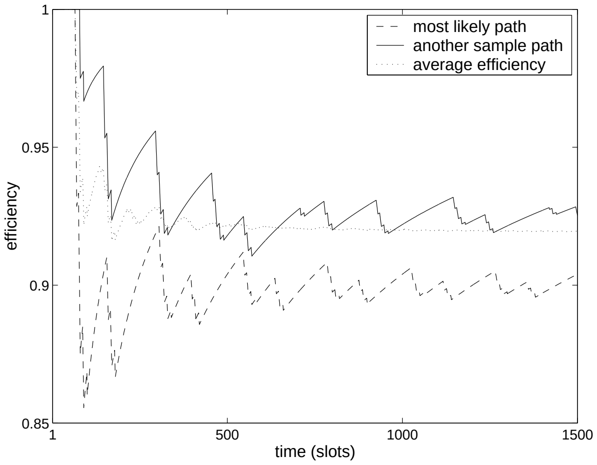

Figure 6.21: Average efficiency and efficiency obtained on two sample paths when $x = 10$, $Q_{\max} = 12 \times D$, SD not implemented (FIFO scheduling).

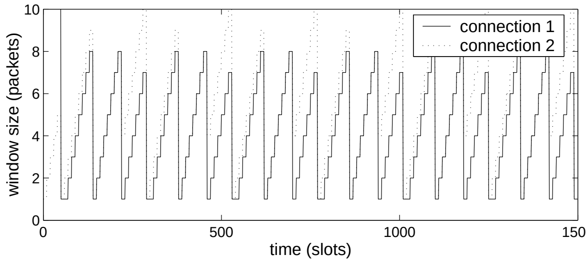

Figure 6.22: Evolution of the window sizes corresponding to the most likely path shown in Figure 6.21 (system without SD, $x = 10$, $Q_{\max} = 12 \times D$, FIFO scheduling).

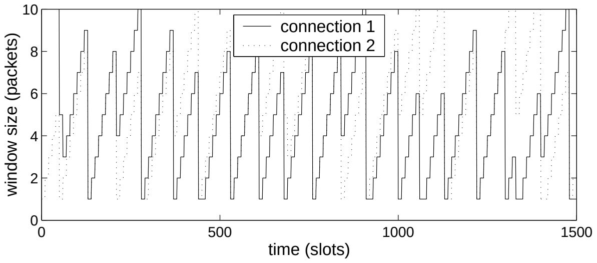

Figure 6.23: Evolution of the window sizes corresponding to the 'other' sample path shown in Figure 6.21 (system without SD, $x = 10$, $Q_{\max} = 12 \times D$, FIFO scheduling).

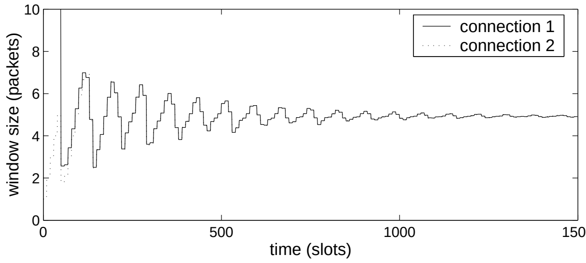

Figure 6.24: Evolution of the mean window sizes when $x = 10$, $Q_{\max} = 12 \times D$, SD not implemented (FIFO scheduling).

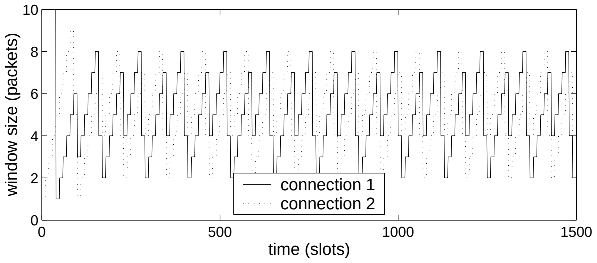

Figure 6.25: Evolution of the window sizes when $x = 10$, $Q_{\max} = 12 \times D$, SD implemented, $L = 7 \times D$ and $K = 1$ (FIFO scheduling).

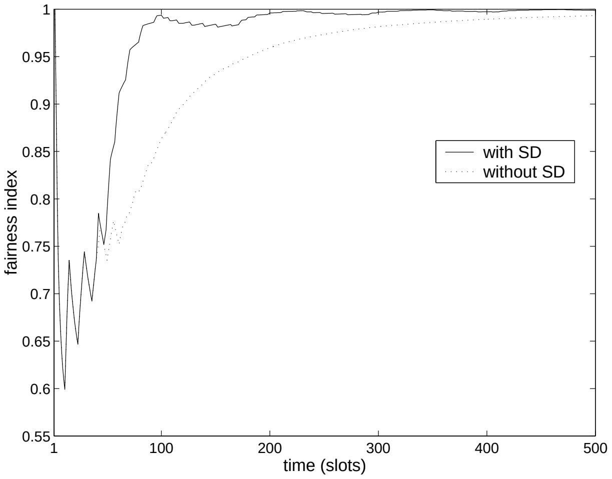

Figure 6.26: Evolution of the fairness index when $x = 10$, $Q_{\max} = 12 \times D$, with and without SD implemented (FIFO scheduling). SD parameters: $L = 7 \times D$, $K = 1$.

From the fairness results in Figure 6.26 for the two systems it is seen that the fairness is much better when the SD algorithm is applied. This is a result that also appeared from the scenarios in Section 6.2.1, and that is to be expected, since the SD algorithm aims at fairness. In the very beginning, the two fairness curves are the same, since as long as the acceptance rules of the SD algorithm are fulfilled, the output of the systems depends only on the scheduling algorithm and on the traffic that is offered, which is equal then since the traffic that is offered stays the same for both systems as long as no losses occur. Because both the efficiency and fairness are larger for the system with SD implemented, this scenario illustrates that there is not necessarily a trade-off between efficiency and fairness.

Remark that no fairness curves are shown for single sample paths, as was done for the efficiency, since the global fairness is *not* just the mean of the fairness obtained on all different sample paths, whereas for the efficiency this relation is true, as can be seen from the definitions in Section 6.1.6.

Scenario 6.2.5. Consider the two systems with following parameters:

- $x = 10$ (slots), $Q_{\max} = 12 \times D$ (cells), $L = 9 \times D$ (cells),
- PLQF scheduling,
- (1) $K = 1.2$, (2) $K = 1.4$.

Figure 6.27 shows the average efficiency obtained with these systems. The highest efficiency is obtained when K is set to 1.2, so when being more conservative in accepting packets. The same figure shows also for both settings of K the efficiency of a single sample path.

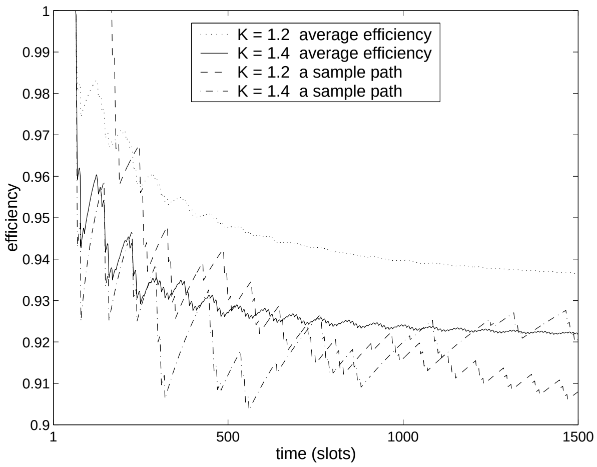

Figure 6.27: Evolution of the average efficiency and the efficiency obtained on two sample paths when $x = 10$, $Q_{\max} = 12 \times D$, $L = 9 \times D$, $K = 1.2$ or $K = 1.4$ (PLQF scheduling).

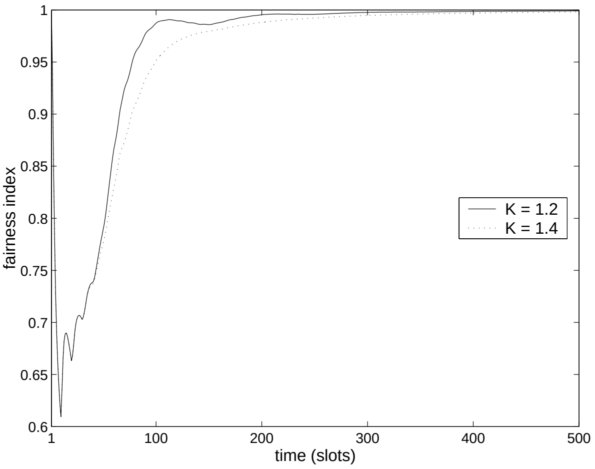

Figure 6.28: Evolution of the fairness index when $x = 10$, $Q_{\max} = 12 \times D$, $L = 9 \times D$, $K = 1.2$ or $K = 1.4$ (PLQF scheduling).

As can be seen, although the average efficiency is the highest with $K = 1.2$, a sample path obtained with $K = 1.4$ does not have to lie the whole time below one obtained with $K = 1.2$. Figures 6.29 and 6.30 show the evolution of the window sizes and queue sizes of the two connections corresponding to the sample paths whose efficiency is shown in Figure 6.27. It can be seen from the plots that a decrease in efficiency occurs at moments that the buffer becomes empty, and these moments occur with some delay after moments on which the sum of both windows was low. The delay is caused by the fact that the windows are forced to go down due to losses, and losses only occur when the queue size is at least $9 \times D$ cells, since $L = 9 \times D$. During the time the windows need to grow again, the queue can flow empty.

On the sample path shown for $K = 1.2$, the queue does not become empty during approximately the first 160 slots, keeping the efficiency equal to 1. The reason is that in the beginning, the queue occupation is completely unfair, due to the unfair start situation, so the SD algorithm drops packets from the first connection. As a result, the window of the first source is forced to one, but the influx of packets into the queue keeps assured since the second source was able to grow its window by that time. Later on, the queue size of both connections is more equal over time, making that losses occur then due to buffer overflow. As can be seen from Figure 6.29, most of the time one of the windows is forced down completely, while the other is only halved. This window is then mostly forced down further at the next window adaptation by the SD algorithm, since its corresponding connection has more packets in the queue. For $K = 1.4$, the acceptance rules of the SD algorithm are less conservative, such that in the beginning only one packet of the first connection is dropped, and the window of that connection is halved. Afterwards, also losses due to

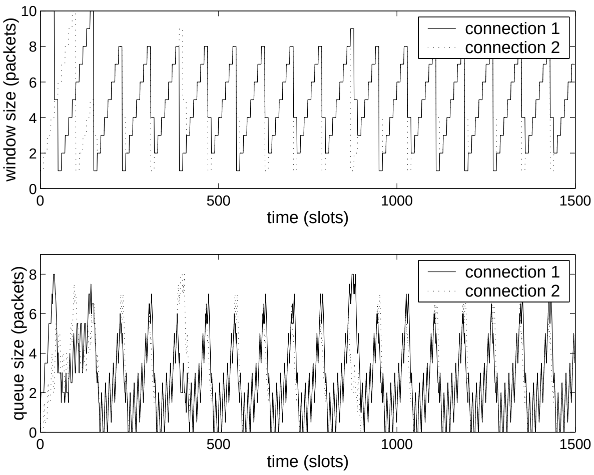

Figure 6.29: Evolution of the window and queue sizes corresponding to the sample path shown in Figure 6.27 for $K = 1.2$ ($x = 10$, $Q_{\max} = 12 \times D$, $L = 9 \times D$, PLQF scheduling).

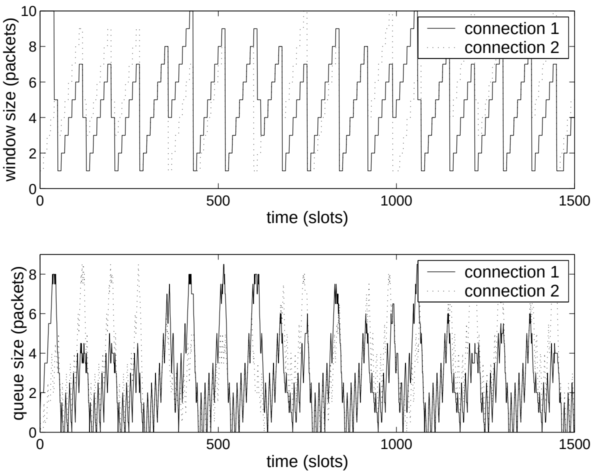

Figure 6.30: Evolution of the window and queue sizes corresponding to the sample path shown in Figure 6.27 for $K = 1.4$ ($x = 10$, $Q_{\max} = 12 \times D$, $L = 9 \times D$, PLQF scheduling).

buffer overflow occur, such that the windows of the two connections need to go down (that of connection 2 less than that of connection 1). While the windows grow again, the queue becomes empty, resulting in a drop of the efficiency. When looking at the average efficiency in Figure 6.27, the moment of the first drop of efficiency occurs at approximately the same time for $K = 1.2$ and $K = 1.4$, which indicates that for both K's there are sample paths for which the queue becomes empty at that time. However, the total probability mass of the sample paths on which this occurs when K equals 1.2 is much less than when $K = 1.4$.

From Figure 6.28 it is seen that the unfairness of the start situation is solved a bit earlier for $K = 1.2$ than for $K = 1.4$. This is because with $K = 1.2$, packets of the connection which has most cells in the buffer are dropped sooner than with $K = 1.4$, since the acceptance condition $Q_i \leq \text{FS}$ is sooner not fulfilled anymore. Also this scenario illustrates that there is not necessarily a trade-off between efficiency and fairness, because the efficiency and the fairness are higher for $K = 1.2$ than for $K = 1.4$.

The two previous examples have illustrated that being more conservative in accepting packets does not necessarily result in lower efficiency, due to the responsiveness of the sources. That this is not always the case is illustrated by many of the examples in the next section, and by the following example:

Scenario 6.2.6. Consider the two systems with following parameters:

- $x = 10$ (slots), $Q_{\max} = 12 \times D$ (cells), $K = 1$,
- RR scheduling,
- (1) $L = 3 \times D$ (cells), (2) $L = 5 \times D$ (cells).

The efficiency obtained with these systems is shown in Figure 6.31. The evolution of the window sizes is shown in Figures 6.33 and 6.34. Both system evolutions are deterministic, since the queue occupation never reaches the maximum buffer size and RR scheduling is applied. It is seen from Figure 6.31 that the highest efficiency is obtained for $L = 5 \times D$. The reason is that for $L = 3 \times D$, both windows are at nearly the same time large, and at nearly the same time small, while when the windows are large, their sum is never very large (never larger than 13 packets) because the low setting of L causes already losses at low buffer occupations. So the queue is often empty during the times that both windows are small, as can be seen from Figure 6.35, because then there are not enough packets sent to keep the buffer full, and before when the windows were high, only a small reserve could be collected. When L equals $5 \times D$ on the other hand, the first window is large when the other is small, and vice versa, such that the buffer becomes only rarely empty (see Figure 6.36), so the efficiency remains high.

The fairness obtained for both settings of L is already very soon high. This is because the fairness condition of the SD algorithm is already tested very soon, and the RR scheduling algorithm lets the cells leave the buffer in a very fair way as long as there are cells of both connections present in the buffer. The reason that the fairness curve for $L = 5 \times D$ slowly oscillates around that of $L = 3 \times D$ is that for $L = 5 \times D$ there are alternately periods that there are no cells of the first, respectively the second, connection in the buffer when the window of that particular connection is low. During such period, more cells of one connection leave the buffer, such that the fairness goes slightly down, but during the following period more cells of the other connection leave the buffer, such that the fairness increases again.

Conclusions

For this section, the conclusions are that

- Due to the responsiveness of the sources, it is not necessarily true anymore that being more conservative in accepting packets implies a lower efficiency, as would be the case when non-responsive sources would be used.
- There is not necessarily a trade-off between efficiency and fairness, so it should be possible to find parameter settings for the SD scheme that result in both good efficiency and good fairness results.

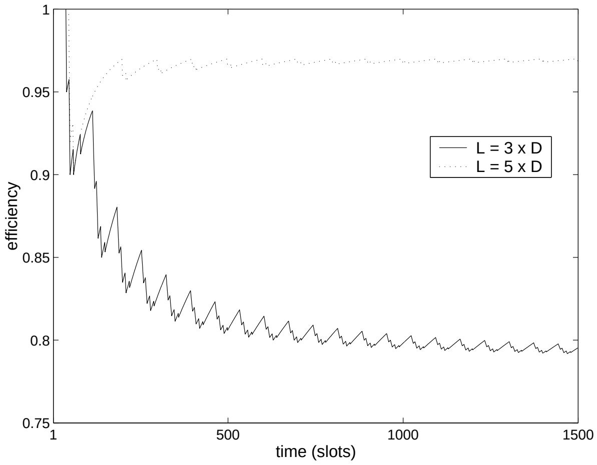

Figure 6.31: Evolution of the efficiency when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$, $L = 3 \times D$ or $L = 5 \times D$ (RR scheduling).

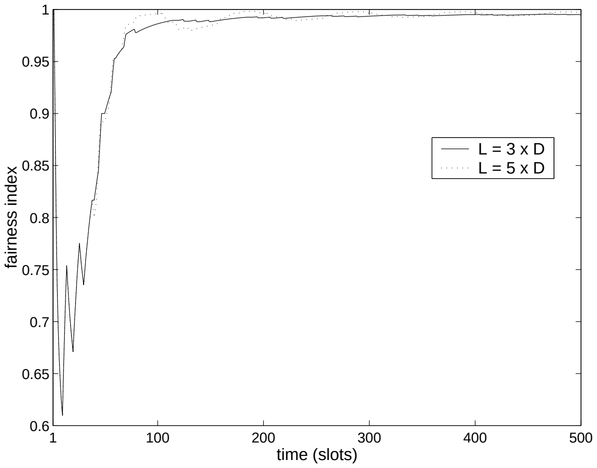

Figure 6.32: Evolution of the fairness index when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$, $L = 3 \times D$ or $L = 5 \times D$ (RR scheduling).

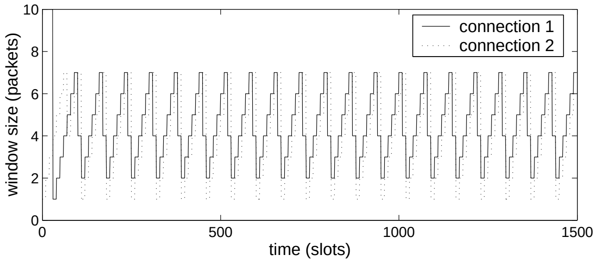

Figure 6.33: Evolution of the window sizes when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$ and $L = 3 \times D$, (RR scheduling).

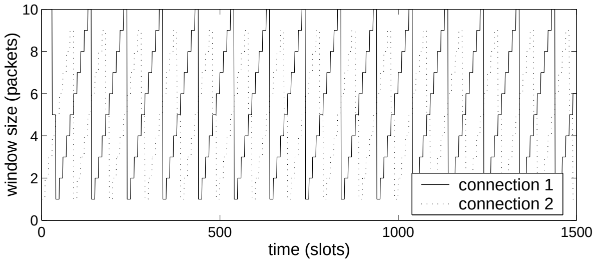

Figure 6.34: Evolution of the window sizes when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$ and $L = 5 \times D$, (RR scheduling).

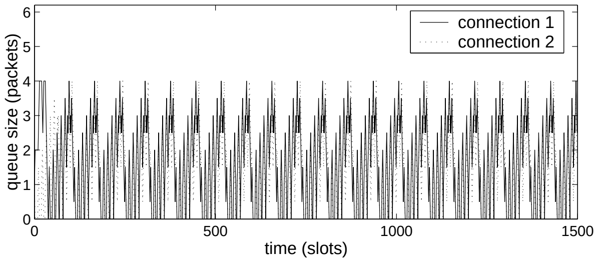

Figure 6.35: Evolution of the buffer occupations when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$ and $L = 3 \times D$, (RR scheduling).

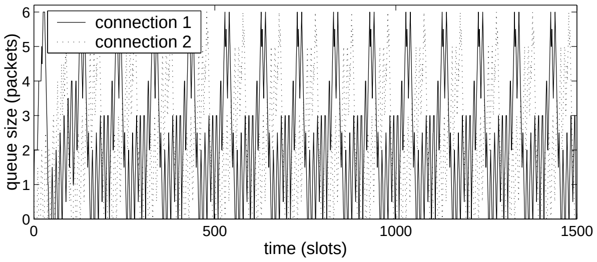

Figure 6.36: Evolution of the buffer occupations when $x = 10$, $Q_{\max} = 12 \times D$, $K = 1$ and $L = 5 \times D$, (RR scheduling).

x (slots)	$Q_{\max}$ (cells)	$Q_{\max}/x$
6	$5 \times D$	0.83
	$7 \times D$	1.17
	$12 \times D$	2.00
10	$5 \times D$	0.50
	$8 \times D$	0.80
	$12 \times D$	1.20
	$20 \times D$	2.00
13	$7 \times D$	0.54
	$10 \times D$	0.77
	$16 \times D$	1.23
	$26 \times D$	2.00

Table 6.2: Parameter settings for the scenarios considered in Section 6.2.3.

6.2.3 Influence of the SD parameters

In this section the influence of the parameters of the SD algorithm on the efficiency and fairness results, when starting from the unfair start condition 3 of Section 6.2.1, is studied. Because of the unfair start situation, all fairness curves have a typical shape: in the beginning they go very fast down, because then an unfair amount of packets of each connection is offered to the system; since the buffer is empty then, all packets are accepted until Q exceeds L, so also the output of the system is unfair in the beginning. Afterwards, the fairness increases again in a rather steep and fluctuating way, and finally it slowly grows towards one. When discussing fairness results further on, this last part of a fairness curve is called the 'horizontal' part, the other part the 'steep' part.

Scenarios that are considered in this section have parameters as shown in Table 6.2. These parameters are chosen such that the ratio of $Q_{\max}$ to x approximately takes the same values (0.50, 0.80, 1.20 and 2.00) for the different settings of x. Remark that during x slots, x packets may leave the buffer. So ideally, the windows of both sources should be fixed a little below $x/2$. But this is not how responsive but greedy sources work: they always try to let their window grow, and decrease it only when losses occur. So on one hand the buffer is needed to accommodate packets that arrive simultaneously, and on the other hand to build up some reserve of packets to keep the efficiency high when the windows of the sources are low. With a setting of $Q_{\max}/x$ equal to 0.50 and two sources, in the ideal situation there is only place in the buffer to accommodate packets that arrive simultaneously. With a setting of $Q_{\max}/x$ larger than 0.50, some reserve can be built up. When $Q_{\max}/x$ equals 2, in the worst case scenario that both sources send together at their maximum rate, the buffer can accommodate their packets during $2x$ slots. Remark however that it are the parameters of the SD algorithm which determine in a large way how many packets eventually are accommodated in the buffer.

Influence of the threshold L

Consider the systems of Table 6.2 and set the SD parameter K equal to 1. The threshold L is varied between $L = 1 \times D$ and $L = Q_{\max} - D$. Remark that the maximal setting of L corresponds to the case where the SD algorithm is not implemented, since at the moments that packets arrive, there is always place in the buffer for at least one packet, because D cells have just left it. So when L is set at D cells before $Q_{\max}$, it is always true that $Q \leq L$ when packets arrive, and the test $Q_i \leq \text{FS}$ is never performed. First the observations made based on this extensive set of scenarios are summarized. Afterwards they are illustrated by representative examples.

The following observations are made:

- For RR and PLQF scheduling, the efficiency generally increases when L increases. This seems natural because increasing L implies that more packets are accepted, but as has been mentioned before, this is not always true due to the responsiveness of the sources. The exceptions to this general rule are:
  - There are always settings of L through which higher efficiency values are obtained than when L is set to its maximal value $Q_{\max} - D$ (i.e., SD is not implemented). With RR scheduling, there are more of these settings than with PLQF scheduling. Sometimes even perfect efficiency values (i.e., constantly equal to 1) are obtained with RR. With the implementation of the SD algorithm, whose main intention is to increase the fairness, there are thus settings of L that allow to obtain also a higher efficiency than when SD is not implemented.
  - With RR scheduling, in case that the efficiency results obtained are very high, it is possible that a larger L leads to a lower efficiency. Probably because these results are so close to optimal, a change of L becomes less significant.
  - With PLQF scheduling, for efficiency results which are among the highest obtained with a particular scenario, sometimes a larger L gives lower efficiency results.
  - A few examples are found with RR scheduling where the efficiency is drastically lower than what would be expected when looking at the results obtained with neighboring examples (i.e., examples where the difference in the setting of L is only D cells). In these examples the windows of both sources synchronize after a while, but in such a way that the buffer becomes often empty, which pulls the efficiency down. None of such examples occur with PLQF scheduling, because of the probabilistic character of such systems.
- With FIFO scheduling, the statement that the efficiency increases when L increases is true when x is small ($x = 6$), and for very small values of L for the other x's. In the other cases, no real relation can be found between a change of L and the corresponding change of the efficiency, but in general large L values (a few packet sizes before the

end of the buffer) give better efficiency results than small L values. As with RR and PLQF scheduling, also with FIFO scheduling there are always settings of L with which higher efficiency values are obtained than when SD is not implemented, but sometimes these results are not the whole time above these obtained when no SD is implemented, but only in the long run. As with RR scheduling, also with FIFO scheduling some examples are found where the windows of both sources synchronize in such a way that the buffer becomes often empty, implying decreasing efficiency results.

- A main observation that can be made about the fairness for the systems with RR and PLQF scheduling is that it is always much better when SD is implemented than when SD is not implemented, irrespective of the exact setting of L. For all settings of L such that $L < Q_{\max} - D$ (i.e., SD implemented), no specific setting of L can really be judged to give results that are the whole time better than with another L. With FIFO scheduling, in general the same observation can be made. However, a few exceptions are found now where the fairness is worse in a scenario where SD is used than when it is not used.

- In the very beginning, the fairness curves coincide for all L, since the behavior of all systems is the same as long as $Q \leq L$. Later, the curves split. The smaller L, the sooner a curve splits from the other curves, since the smaller L, the sooner the SD scheme starts to solve the initial unfairness.

- In general, the larger x, the longer the steep part lasts, when time is expressed in multiples of x. This indicates that the longer between adapting the windows, the longer it takes before the initial unfairness is more or less solved.

The observations summarized above are now illustrated by numerical examples. Figures 6.37 until 6.39 show some of the efficiency results obtained with RR scheduling. In Figure 6.37 results obtained with the system with parameters $x = 13$ slots and $Q_{\max} = 10 \times D$ cells are shown. As can be seen, the efficiency increases when L increases, except for $L = 3 \times D$ cells and $L = 9 \times D$ cells. $L = 9 \times D$ cells is the scenario in which the SD algorithm is not applied. A higher efficiency than in this scenario is obtained with L larger or equal to $5 \times D$ cells. The scenario with L set equal to $3 \times D$ cells is one of the few examples where the efficiency curve is different than expected. Analyzing the results obtained with this scenario learns that in this scenario the windows of both sources synchronize after a while, but in such a way that the sum of both windows is always much smaller than x. This means that every x slots, there will be some slots that the buffer is empty, which pulls the efficiency down. Figure 6.38 shows results obtained with the scenario where $x = 13$ slots and $Q_{\max} = 16 \times D$ cells. Again values for L can be found such that the efficiency is larger than when SD is not implemented ($L = 15 \times D$ cells). In general, the efficiency increases when L increases, but for $L = 12 \times D$ cells, the efficiency is smaller than when $L = 10 \times D$ cells, except in the beginning. But for both settings of L the efficiency is high (above 0.98). The results shown in Figure 6.39 are all obtained with the scenario where x

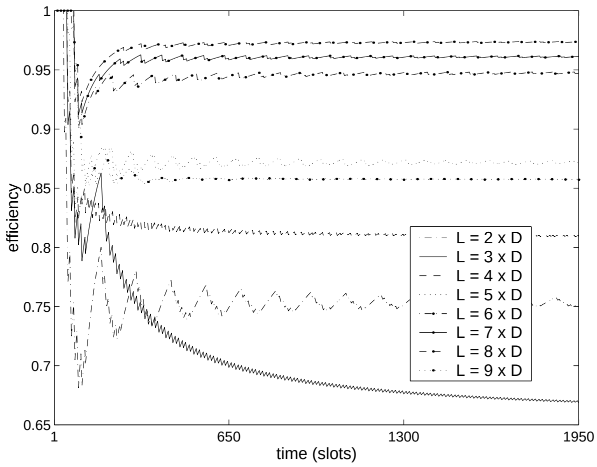

Figure 6.37: Efficiency results obtained when $x = 13$, $Q_{\max} = 10 \times D$ and $K = 1$ for different settings of L (RR scheduling).

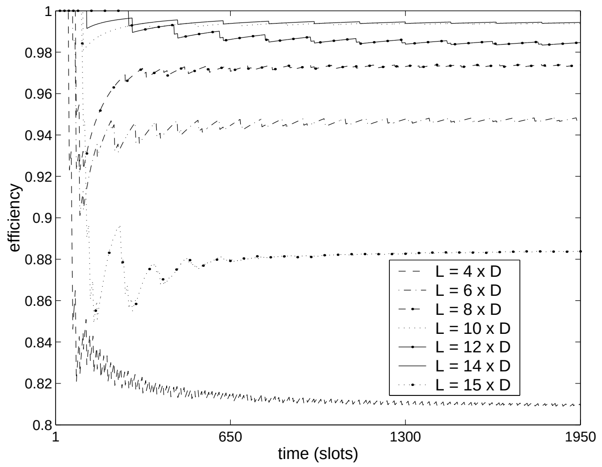

Figure 6.38: Efficiency results obtained when $x = 13$, $Q_{\max} = 16 \times D$ and $K = 1$ for different settings of L (RR scheduling).

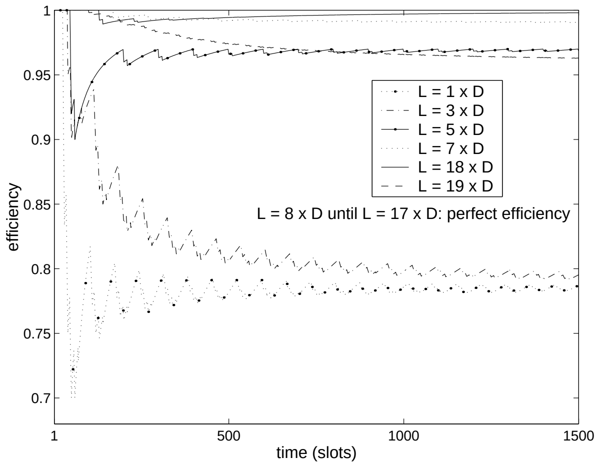

Figure 6.39: Efficiency results obtained when $x = 10$, $Q_{\max} = 20 \times D$ and $K = 1$ for different settings of L (RR scheduling).

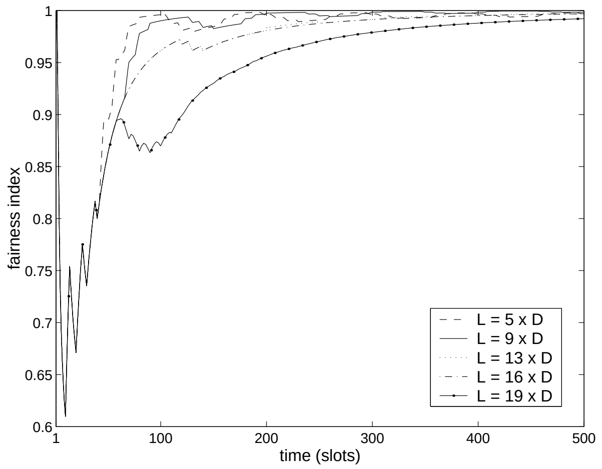

Figure 6.40: Fairness results obtained when $x = 13$, $Q_{\max} = 20 \times D$ and $K = 1$ for different settings of L (RR scheduling).

equals 10 slots and $Q_{\max}$ equals $20 \times D$ cells. When L is set between $8 \times D$ and $17 \times D$, the efficiency obtained is always perfect (i.e., constantly equal to one), which means that the buffer never becomes empty. Also the buffer never overflows under these scenarios. For $L = 18 \times D$, the efficiency curve is a little below one, so this is again an example where the efficiency is very high, but smaller than with a scenario where L is smaller (i.e., smaller than all scenarios that lead to a perfect efficiency).

Some fairness results obtained with the systems with RR scheduling are shown in Figures 6.40 and 6.41. In these figures only a few curves are shown to keep the figures clear,

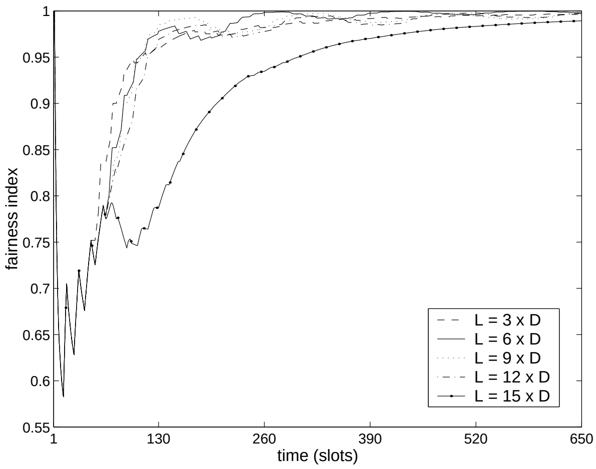

Figure 6.41: Fairness results obtained when $x = 13$, $Q_{\max} = 16 \times D$ and $K = 1$ for different settings of L (RR scheduling).

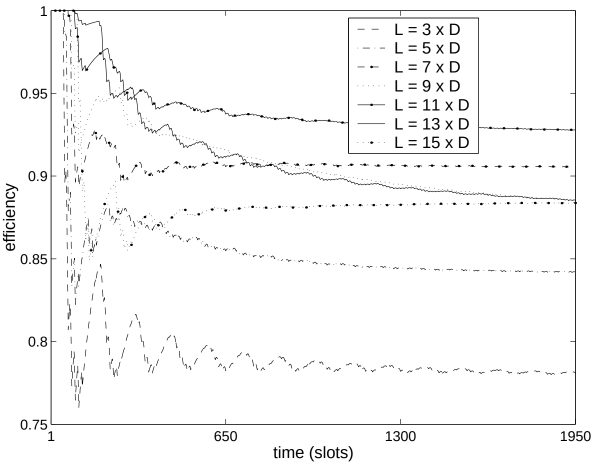

Figure 6.42: Efficiency results obtained when $x = 13$, $Q_{\max} = 16 \times D$ and $K = 1$ for different settings of L (PLQF scheduling).

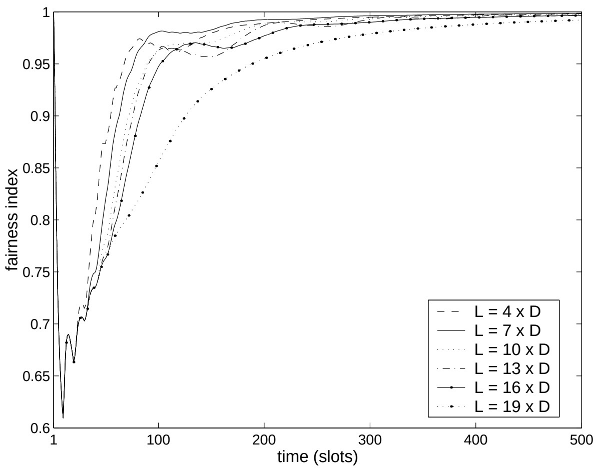

Figure 6.43: Fairness results obtained when $x = 10$, $Q_{\max} = 20 \times D$ and $K = 1$ for different settings of L (PLQF scheduling).

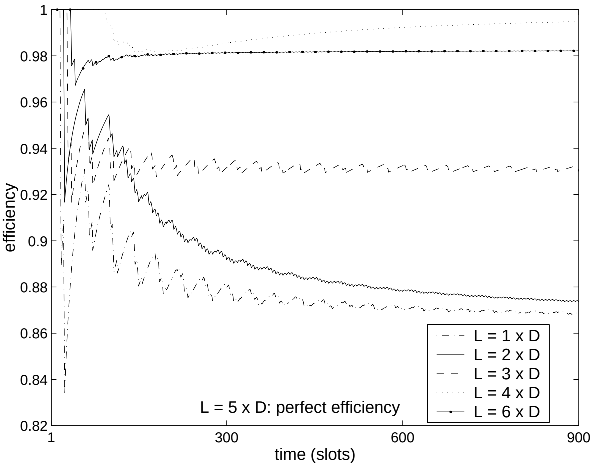

Figure 6.44: Efficiency results obtained when $x = 6$, $Q_{\max} = 7 \times D$ and $K = 1$ for different settings of L (FIFO scheduling).

but all fairness curves for $L \neq Q_{\max} - D$ lie in the same region as the curves shown in the figures. This illustrates that the fairness is much better when SD is implemented than when it is not implemented. Remark also the dips (around 60-100 slots in Figure 6.40, 78-130 slots in Figure 6.41) in the fairness curves when SD is not implemented. Before losses occur, the second source is allowed to let its window grow and sends more and more cells in the system, such that the initial unfairness is slowly solved. Because the SD algorithm is not implemented, losses occur due to buffer overflow, and the most likely situation is that both connections experience losses, and have to reduce their window. This makes that the fairness goes down again (the dip), because the first connection still has the largest reserve of cells in the queue from before, and because the window of the first source was still the largest when losses occurred, it is very likely that this window now still is larger than that of the second source, although not as extremely anymore as in the beginning. The figures illustrate also that the curves coincide in the beginning, and one by one branch off. The smaller L, the sooner this happens. Comparing Figure 6.40 with Figure 6.41 shows that in Figure 6.41, where x is 13, it takes longer (approximately 10 times x slots) before the horizontal part of the curves starts than in Figure 6.40 where $x = 10$ and it takes approximately 7 times x slots.

Figures 6.42 and 6.43 show some results obtained when PLQF scheduling is used. In Figure 6.42, efficiency results are shown for $x = 13$ slots and $Q_{\max} = 16 \times D$ cells. This figure illustrates that also with PLQF scheduling there are settings of L with which higher efficiency values are obtained than when SD is not implemented ($L = 15 \times D$). Further it can be seen from the figure that in general, the efficiency increases when L increases, although this is not always the case. For example, for $L = 11 \times D$ the efficiency is larger than for $L = 13 \times D$. When looking at the most-likely path for this last case, it is seen that from a certain time on (around 420 slots) approximately once every 520 slots, the window of one connection is forced down until its minimum, while that of the second connection, which at that time was not too large, is halved, such that both connections end up with a small window. Some time is needed to let these windows grow again, during which the buffer flows empty for a few slots. Figure 6.43 shows fairness results when x equals 10 slots and $Q_{\max}$ is $20 \times D$. The figure illustrates clearly that the fairness obtained when SD is not implemented ($L = 19 \times D$) is worse than when it is implemented and that the fairness curves coincide in the beginning, and branch off one by one, first for the smallest L. This branching off happens sooner here than in the corresponding case with RR scheduling (Figure 6.40), since with RR a difference in fairness occurs only from the moment that there is a difference in the output for the scenarios with different L. This happens when there is a difference in which queue is empty at the particular moment.

Results obtained when FIFO scheduling is used are shown in Figures 6.44 until 6.46. Figure 6.44 shows efficiency results obtained when $x = 6$ and $Q_{\max} = 7 \times D$. Here it is true that the efficiency increases when L increases, and that again there are settings of L ($L = 4 \times D$ and $L = 5 \times D$) such that higher efficiency values can be obtained when SD is implemented than when SD is not implemented. In Figure 6.45, efficiency results obtained for $x = 13$ and $Q_{\max} = 10 \times D$ are shown. It can be seen from the figure that for small values of L, the efficiency increases when L is larger. For larger values of L, no real relation seems to exist between a change of L and a corresponding change of the efficiency, but except for $L = 8 \times D$, all settings of L above $5 \times D$ give reasonable efficiency results. Analyzing the results obtained when L equals $8 \times D$ learns that all sample paths will eventually reach a state in which the windows of both sources synchronize, and once the system has reached this state, it keeps returning to it. During such a cycle, which lasts 117 slots, in 31 of these slots the buffer is empty, implying that the efficiency keeps going down. No figure is shown here where no setting of L is found such that the corresponding

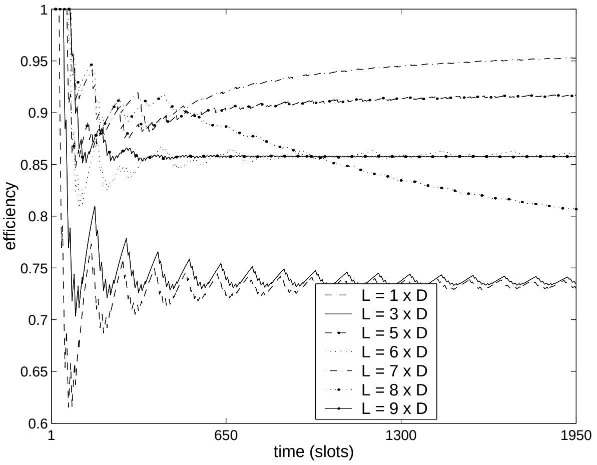

Figure 6.45: Efficiency results obtained when $x = 13$, $Q_{\max} = 10 \times D$ and $K = 1$ for different settings of L (FIFO scheduling).

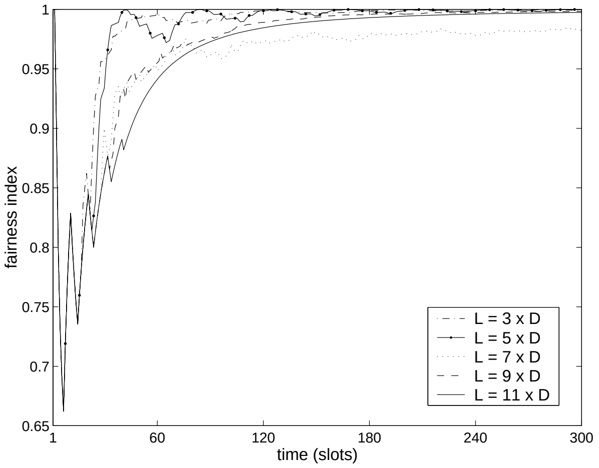

Figure 6.46: Fairness results obtained when $x = 6$, $Q_{\max} = 12 \times D$ and $K = 1$ for different settings of L (FIFO scheduling).

efficiency curve lies the whole time above that obtained when SD is not implemented. But for example when $x = 10$ and $Q_{\max} = 8 \times D$, only efficiency curves which lie in the long run above that obtained when $L = 7 \times D$ (no SD implemented) exist. Figure 6.46 shows fairness results obtained when $x = 6$ and $Q_{\max} = 12 \times D$. Mostly, the fairness obtained when SD is implemented is better than when SD is not implemented ($L = 11 \times D$), but this figure shows also one of the very few exceptions found. When $L = 7 \times D$, it looks from the figure that the fairness curve does not approach one. Investigating the numerical results learns however that it does, but much slower than normal. The reason is that in this particular scenario, the following occurs frequently: the window of connection two shows with a slight delay the same behavior as that of connection one; at the moment that packets of connection one are dropped by the SD algorithm (obviously, W_1 has reached a maximum at that time, and the buffer content is above L), then also the window of connection two is fairly high, such that Q will stay above L; during the following interval of x slots, W_2 reaches its maximum, but W_1 is low now, and thus connection one sends much less than connection two, implying that connection two will loose a lot of packets; the result is that the window of connection two has to go down, such that connection two can send less than connection one; because this occurs repeatedly, the unfairness is higher now than in other scenarios.

Influence of the parameter *K*

For the scenarios of Table 6.2 with $x = 10$ and different settings of L, K will now be chosen from $K = 1$, $K = 1.2$ and $K = 1.4$. The larger K, the less severe the SD algorithm is in dropping packets. The following observations are made based on the results:

- With RR scheduling, it is true in general that when K increases, then the efficiency stays equal or increases also. The larger L, the smaller the positive effect of increasing K becomes. Some exceptions are found where the efficiency obtained with $K = 1.2$ or $K = 1.4$ is the lowest. This occurs when the efficiency results are very large or in scenarios where the windows synchronize in such a way that the buffers become often empty. On the fairness results almost no influence of K is noticed. The steep parts of the fairness curves for the different K mostly coincide, since as long as cells are present in both queues, the output of the RR scheduling algorithm is the same for the different scenarios.

- With PLQF scheduling, for small L (approximately $L < x/2$) a larger K gives a larger efficiency. The larger $Q_{\max}$ is, for the larger L values this stays true. When L is increased, the results evolve through the following situations: (i) a larger K gives still a larger efficiency in the long run, but in the transient phase the efficiency curves cross. (ii) $K = 1.2$ gives a higher efficiency than $K = 1$, but for $K = 1.4$ the efficiency is below that obtained with $K = 1$, (iii) a larger K implies a lower efficiency. For the fairness, some differences are noticed when changing K, but the different fairness curves still stay very close to each other. The largest difference is noticed in the steep parts of the curves, where the smallest K value gives the best result.

- When FIFO scheduling is applied, as with PLQF scheduling the efficiency increases when K increases for small L. The more L grows, an evolution towards the fact that a larger K gives a lower efficiency is seen. Concerning the fairness results, curves obtained for different K values coincide in the beginning, after which they one by one branch off. In the steep part the best fairness is obtained when K equals 1. In the long run, it is difficult to judge which K value gives best results in a scenario. What is seen often in the horizontal parts of the fairness curves is a slowly oscillating behavior. Curves which show this behavior correspond often to scenarios with which perfect efficiency values are obtained.

In Figure 6.47 some efficiency results are shown for different settings of L and K when $x = 10$, $Q_{\max} = 12 \times D$ and RR scheduling is applied. The curves for $K = 1$ and $K = 1.2$ when $L = 5 \times D$, and those for the three K values when $L = 7 \times D$ or $L = 9 \times D$ coincide. For all L, except $L = 1 \times D$, the efficiency increases when K grows, but the differences between the curves for $K = 1$ and these of $K = 1.4$ decrease when L increases. For $L = 1 \times D$ and $K = 1$ or $K = 1.2$, the evolution of both systems is deterministic, since the queue occupation never reaches the maximum buffer size. The efficiency curve obtained when K equals 1.2 lies below that when K equals 1, since in the first case the buffer becomes more often empty. Some fairness results obtained with different K for $x = 10$, $Q_{\max} = 20 \times D$ and $L = 3 \times D$ can be found in Figure 6.48. The steep part of the

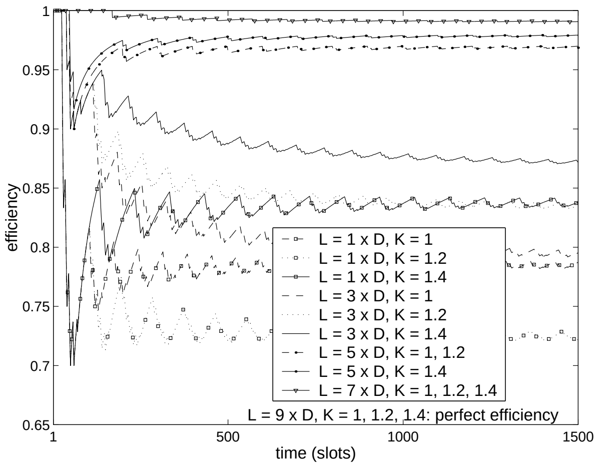

Figure 6.47: Efficiency results obtained when $x = 10$, $Q_{\max} = 12 \times D$ for different settings of L and K (RR scheduling).

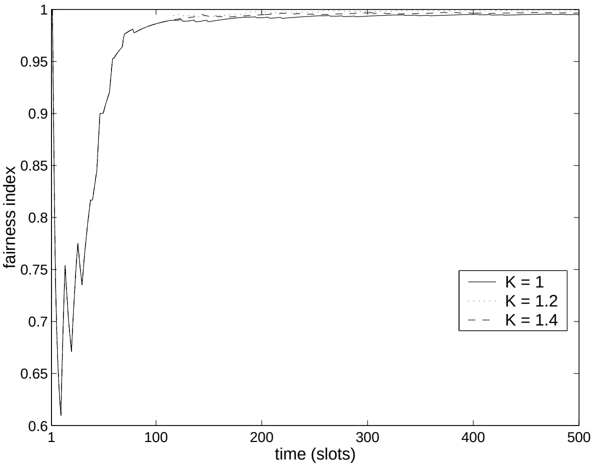

Figure 6.48: Fairness results obtained when $x = 10$, $Q_{\max} = 20 \times D$ and $L = 3 \times D$ for different settings of K (RR scheduling).

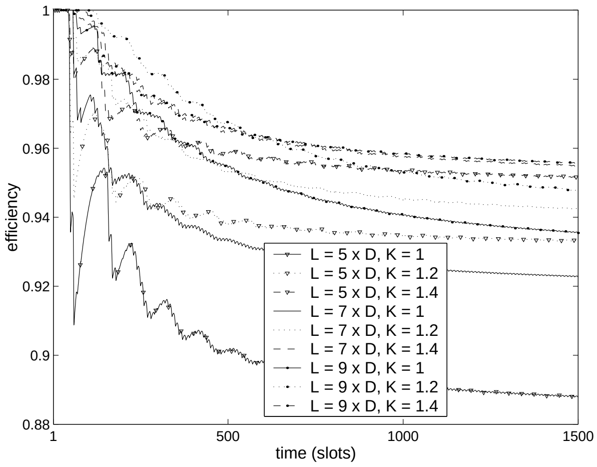

Figure 6.49: Efficiency results obtained when $x = 10$, $Q_{\max} = 20 \times D$ for different settings of L and K (PLQF scheduling).

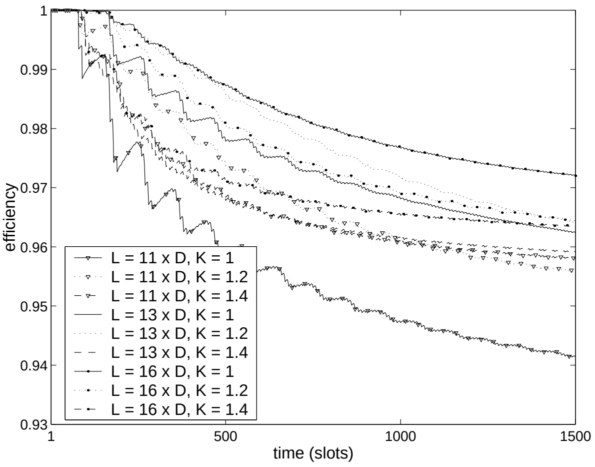

Figure 6.50: Efficiency results obtained when $x = 10$, $Q_{\max} = 20 \times D$ for different settings of L and K (PLQF scheduling).

three curves coincides, in the horizontal part a slight difference is seen.

Figures 6.49 and 6.50 show efficiency results obtained with PLQF scheduling when $x = 10$ and $Q_{\max} = 20 \times D$. The figures illustrate the evolution from 'a larger K gives a larger efficiency' for small L towards 'a larger K gives a smaller efficiency' for large L. For $L = 5 \times D$ or $L = 7 \times D$, the efficiency increases when K increases. For $L = 9 \times D$ and $L = 11 \times D$, in the long run this stays true, but in the transient phase the largest efficiency is obtained when $K = 1.2$. When L equals $13 \times D$, the largest efficiency is obtained with $K = 1.2$, the lowest with $K = 1.4$. Finally, with $L = 16 \times D$, a larger K gives a

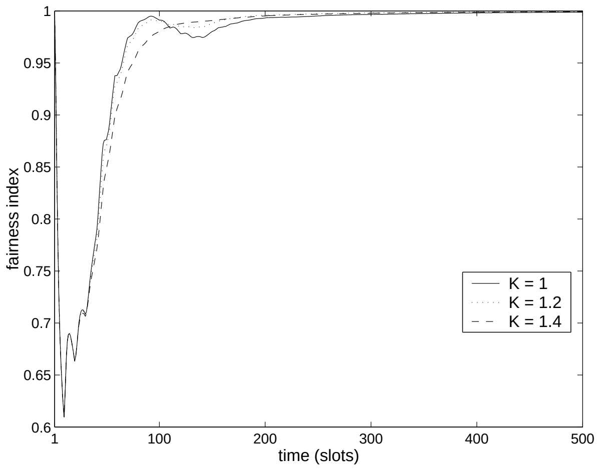

Figure 6.51: Fairness results obtained when $x = 10$, $Q_{\max} = 20 \times D$ and $L = 5 \times D$ for different settings of L and K (PLQF scheduling).

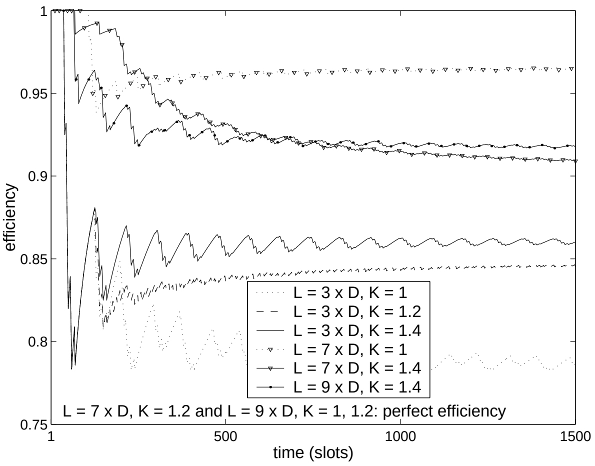

Figure 6.52: Efficiency results obtained when $x = 10$, $Q_{\max} = 12 \times D$ for different settings of L and K (FIFO scheduling).

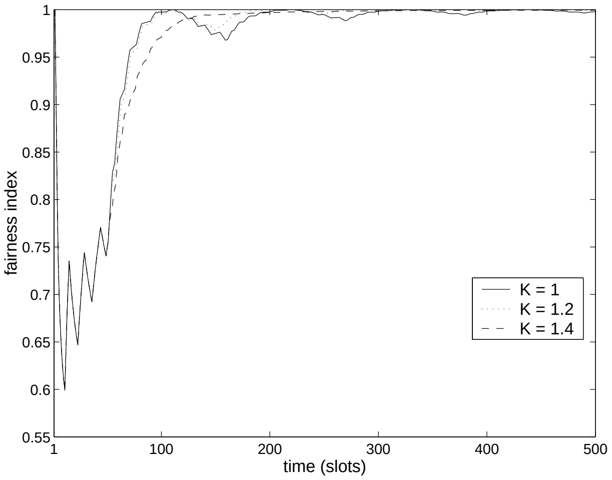

Figure 6.53: Fairness results obtained when $x = 10$, $Q_{\max} = 12 \times D$ and $L = 9 \times D$ for different settings of L and K (FIFO scheduling).

smaller efficiency. Figure 6.51 shows some fairness results when PLQF scheduling is used for $x = 10$, $Q_{\max} = 20 \times D$ and $L = 5 \times D$. As the figure illustrates, no large differences occur between the fairness curves for different K. The dip in the fairness around 100-150 slots corresponds to the first moment that the window of connection 2 is forced down (until then it has been growing). Because connection 2 still has to make up some of its initial arrears, and now temporarily can send less than connection 1, the fairness goes a bit down again.

Figures 6.52 and 6.53 show results for the different values of K when FIFO scheduling is applied. In Figure 6.52 efficiency values are shown for a scenario in which $x = 10$ and $Q_{\max} = 12 \times D$. As can be seen, when L equals $3 \times D$, then the efficiency increases when K increases. For $L = 7 \times D$, the efficiency is the largest (i.e., perfect) when $K = 1.2$. The efficiency obtained with $K = 1$ is larger than that obtained with $K = 1.4$, while in the transient phase curves cross. When L equals $9 \times D$, the lowest efficiency is obtained when $K = 1.4$. Figure 6.53 shows fairness results for different K when L equals $9 \times D$ for the same scenario as used in Figure 6.52. The three fairness curves coincide in the beginning and then branch off one by one. In the horizontal part, the curves for $K = 1$ and $K = 1.2$ show an oscillating behavior caused by the fact that the window of one connection is high while that of the other connection is low. Because of the FIFO scheduling, this implies that during a period more cells of connection one will leave the buffer, such that the fairness goes down. During a following period, more cells of connection two leave the buffer, such that the fairness grows again, and so on. Comparing with Figure 6.52 learns that perfect efficiency values are obtained when $L = 9 \times D$ and $K = 1$ or $K = 1.2$.

Conclusions

The most important conclusion of this section is that the presence of the SD algorithm has a large positive effect on the fairness results, irrespective of the exact setting of the parameters of the algorithm. On the efficiency results however, these parameters have more influence.

With RR and PLQF scheduling, the efficiency generally increases when the threshold L increases, and choosing L at a few packet sizes less than the size of the buffer results in a good setting. With RR scheduling, the chance is rather high that the efficiency values obtained are then even above these obtained when SD is not implemented (so there is no trade-off between efficiency and fairness then). With PLQF scheduling, this chance is reasonable. Remark however that with RR scheduling, sometimes the efficiency is lower than expected because of synchronization effects. When using PLQF scheduling, no lasting synchronization will occur because of the probabilistic character of the scheduling algorithm in these scenarios. Also with FIFO scheduling, synchronization can occur. With FIFO scheduling it is much harder to make a conclusion about the best setting of the threshold L, since no real relation was found between a change of L and a corresponding change of the efficiency. But choosing it a few packet sizes less than the size of the buffer as with RR and PLQF scheduling gave in most scenarios rather good results.

The parameter K of the SD algorithm has also more influence on the efficiency results than on the fairness results. Increasing K has principally a positive effect on the efficiency when L is set at a small value. When the setting of L is larger, this positive effect is still seen with RR scheduling, but with PLQF and FIFO scheduling the probability is rather high that the efficiency will be lower than when K is chosen equal to one.

As a general conclusion, it is recommended to implement SD to increase the fairness, but with a parameter setting focusing on the efficiency results.

6.3 Appendix

In this appendix it is shown that under identical start values for both connections (i.e., $Q_1(0^-) = Q_2(0^-)$, $W_1(0^-) = W_2(0^-)$, $L_1(0^-) = L_2(0^-)$), the mean window size and the mean buffer occupation at an arbitrary time instant l is identical for both connections. For the PLQF and the RR system, this is a special case of the property below, which is proven formally. For the FIFO system, only an intuitive explanation is given. Because the results in this chapter for the FIFO system are obtained by simulation, no mathematical description of the evolution over time of this system was developed before. Developing it here would only introduce more notation to describe the order in which the cells of the different connections have entered the buffer, after which a formal proof would be almost analogue to that for the PLQF and RR system.

Property 6.3.1. *Let k be an element of the ordered set $\{0^-, 0^+, 1^-, 1^+, 2^-, 2^+, \dots\}$. For the PLQF and the RR system as defined in Section 6.1, if for all $(q_1, q_2, s, w_1, w_2, l_1, l_2) \in \Omega$ for which $P\{X_k = (q_1, q_2, s, w_1, w_2, l_1, l_2)\} \neq 0$, it is true that $q_1 = q_2$, $w_1 = w_2$ and $l_1 = l_2$, then $E[Q_1(l)] = E[Q_2(l)]$ and $E[W_1(l)] = E[W_2(l)]$, for all $l \geq k$.*

Remark that the random variable S is only present in the states of Ω when needed, i.e., when PLQF scheduling is applied.

Proof. From the computations in Section 6.1.5 of

$$P_1 = P\left\{X_{h^+} = (q_1, q_2, s, w_1, w_2, l_1, l_2) \mid X_{h^-} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\}, \text{ and} \tag{6.15}$$

$$P_2 = P\left\{X_{h^-} = (q_1, q_2, s, w_1, w_2, l_1, l_2) \mid X_{(h-1)^+} = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\}, \tag{6.16}$$

it is easily seen that when $P_1 = p_1$, then also

$$P\left\{X_{h^+} = (q_2, q_1, s, w_2, w_1, l_2, l_1) \mid X_{h^-} = (\hat{q}_2, \hat{q}_1, \hat{s}, \hat{w}_2, \hat{w}_1, \hat{l}_2, \hat{l}_1)\right\} = p_1, \tag{6.17}$$

and when $P_2 = p_2$, then also

$$P\left\{X_{h^-} = (q_2, q_1, s, w_2, w_1, l_2, l_1) \mid X_{(h-1)^+} = (\hat{q}_2, \hat{q}_1, \hat{s}, \hat{w}_2, \hat{w}_1, \hat{l}_2, \hat{l}_1)\right\} = p_2. \tag{6.18}$$

By induction it is now shown that for all $l \geq k$,

$$P\{X_l = (q_1, q_2, s, w_1, w_2, l_1, l_2)\} = P\{X_l = (q_2, q_1, s, w_2, w_1, l_2, l_1)\}. \tag{6.19}$$

For $l = k$, (6.19) is trivially true by the assumption in property 6.3.1. Assume that (6.19) is also true for $l > k$ (inductionhypothesis). If $l = h^-$, then define $l^* = h^+$. If l equals h^+, then define $l^* = (h+1)^-$. So it should be shown now that (6.19) is also true for l^*:

$$\begin{aligned}
&P\left\{X_{l^*} = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\} = \\
&\sum_{(\hat{q}_1,\hat{q}_2,\hat{s},\hat{w}_1,\hat{w}_2,\hat{l}_1,\hat{l}_2)\in\Omega} P\left\{X_{l^*} = (q_1, q_2, s, w_1, w_2, l_1, l_2) \mid X_l = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\} \\
&\qquad\qquad\qquad\qquad P\left\{X_l = (\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\right\} \\
&= \sum_{(\hat{q}_2,\hat{q}_1,\hat{s},\hat{w}_2,\hat{w}_1,\hat{l}_2,\hat{l}_1)\in\Omega} P\left\{X_{l^*} = (q_2, q_1, s, w_2, w_1, l_2, l_2) \mid X_l = (\hat{q}_2, \hat{q}_1, \hat{s}, \hat{w}_2, \hat{w}_1, \hat{l}_2, \hat{l}_1)\right\} \\
&\qquad P\left\{X_l = (\hat{q}_2, \hat{q}_1, \hat{s}, \hat{w}_2, \hat{w}_1, \hat{l}_2, \hat{l}_1)\right\} = P\left\{X_{l^*} = (q_2, q_1, s, w_2, w_1, l_2, l_1)\right\}, \quad (6.20)
\end{aligned}$$

where the first and third equalities use the complete probability formula, and the second equality uses that when $(\hat{q}_1, \hat{q}_2, \hat{s}, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) \in \Omega$, then also $(\hat{q}_2, \hat{q}_1, \hat{s}, \hat{w}_2, \hat{w}_1, \hat{l}_2, \hat{l}_1) \in \Omega$, together with equations (6.15) and (6.17) when $l^* = h^+$, or equations (6.16) and (6.18) when $l^* = h^-$, and the inductionhypothesis.

By definition of the mean, it follows now immediately that for each $l \geq k$,

$$\begin{aligned}
E\left[Q_1(l)\right] &= \sum_{t=0}^{2Q_{\max}/D} \frac{tD}{2} \sum_{\substack{(q_1,q_2,s,w_1,w_2,l_1,l_2)\in\Omega \\ q_1 = tD/2}} P\left\{X_l = (q_1, q_2, s, w_1, w_2, l_1, l_2)\right\} \\
&= \sum_{t=0}^{2Q_{\max}/D} \frac{tD}{2} \sum_{\substack{(q_2,q_1,s,w_2,w_1,l_2,l_1)\in\Omega \\ q_1 = tD/2}} P\left\{X_l = (q_2, q_1, s, w_2, w_1, l_2, l_1)\right\} = E\left[Q_2(l)\right], \quad (6.21)
\end{aligned}$$

and analogously that $E\left[W_1(l)\right] = E\left[W_2(l)\right]$. ■

Under identical start values for both connections, property 6.3.1 can be applied for $k = 0^-$, such that for the PLQF and the RR system the mean window size and the mean buffer occupation at an arbitrary time instant l is identical for both connections.

Property 6.3.1 is also valid for the FIFO system under the extra condition that the equal amounts of cells in the buffer of connection 1 and connection 2 at time k are in such an order present in the buffer that $D/2$ cells of each connection leave the buffer per slot under FIFO scheduling. Because the condition of property 6.3.1 and this extra condition stay fulfilled until the first time instant later than time k that the buffer overflows on one of the sample paths, property 6.3.1 is already certainly true until that time instant. Because with FIFO scheduling a sample path only splits at times that buffer overflow occurs, and it was assumed that then each connection has *equal* probability of being the one from which the packet is lost, for each sample path there is always another sample path with identical probability such that the number of cells of connection 1 in the buffer on the first sample path equals the number of cells of connection 2 in the buffer on the second sample path, and vice versa, and the same is true for the window sizes and the values of the loss counters. So property 6.3.1 stays also true after the first time instant that the buffer overflows on a sample path.

Chapter 7

Extensions to the SD model

In this chapter two extensions to the model developed in Chapter 6 are considered. In Section 7.1 the parameter x of the source model, which represents the time after which the responsive sources update their window, is taken differently for both sources. The motivation behind this extension is to introduce another aspect of unfairness in the model than the unfair start situation, and observe the behavior of the SD scheme under this kind of unfairness. Where in Chapter 6 the SD buffer acceptance rules were considered, in Section 7.2 the definition of the fair share is changed such that now the fair buffer allocation (FBA) acceptance rules are considered. A comparison with the results obtained in this case and the results obtained before using SD is made. Section 7.3 concludes this chapter with a short overview of other methods used in the literature to model frame aware buffer acceptance schemes.

7.1 Use of a different parameter x for both sources

With real TCP sources, there is an inherent unfairness to connections with longer roundtrip times [29]. This unfairness originates from the fact that in the absence of congestion, each connection increases its window every roundtrip time, so the window and thus also the throughput increases at a faster rate for connections with shorter roundtrip times.

In the model developed in the previous chapter, a source increases its window after x slots when no losses occurred during these x slots. Since two identical sources were considered, both sources used the same value for x. In this section we extend the source model such that both connections use a different value for x, i.e., x_1 for the first source and x_2 for the second source, with $x_1 \neq x_2$. The source behavior stays the same, except that the window size of source i ($i = 1, 2$) now can take values in the range $1, \ldots, x_i$.

A scenario with identical start conditions for both connections is considered:

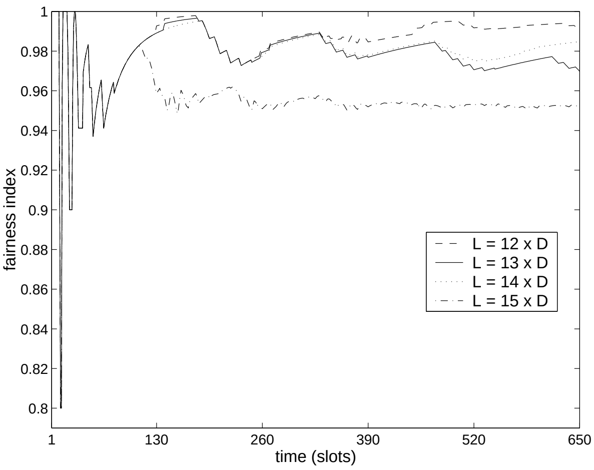

Figure 7.1: Fairness results obtained when $x_1 = 10$, $x_2 = 13$, $Q_{\max} = 16 \times D$ and $K = 1$ for different settings of L (RR scheduling). Identical start conditions.

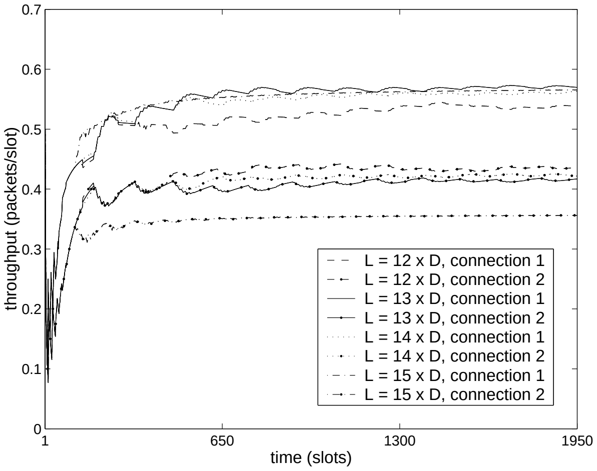

Figure 7.2: Throughput results obtained when $x_1 = 10$, $x_2 = 13$, $Q_{\max} = 16 \times D$ and $K = 1$ for different settings of L (RR scheduling). Identical start conditions.

Scenario 7.1.1.

- Start condition: $P\{X_{0^-} = (\hat{q}_1, \hat{q}_2, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2)\} = 1$, where $(\hat{q}_1, \hat{q}_2, \hat{w}_1, \hat{w}_2, \hat{l}_1, \hat{l}_2) = (0, 0, 1, 1, 0, 0)$,
- $x_1 = 10$ (slots), $x_2 = 13$ (slots), $Q_{\max} = 16 \times D$ (cells), $K = 1$,
- RR scheduling,
- (1) $L = 12 \times D$ (cells), (2) $L = 13 \times D$ (cells), (3) $L = 14 \times D$ (cells), (4) $L = 15 \times D$ (cells).

Remark that when identical start conditions were considered in the previous chapter, this resulted in equal throughputs for both connections and thus perfect fairness. But since now both sources are not identical anymore, this does not need to be true anymore (i.e., Property 6.3.1 does not need to hold). Setting the threshold L equal to $15 \times D$ cells is again the same as not implementing the SD algorithm.

Figure 7.1 shows fairness results. In the beginning, when the buffer content has not yet reached the threshold L, the behavior of all systems is the same: the fairness goes down for the first time after the first source increases its window, and thus sends new packets; because $x_2 > x_1$, the second source cannot send traffic at that time, so the fairness goes down; after the second source has also increased its window, it also sends new packets, and after they have left the queue the fairness equals 1 again. This goes on until the second source cannot catch up anymore with the amount of packets the first source has already sent. So when both sources have different intervals after which they update their

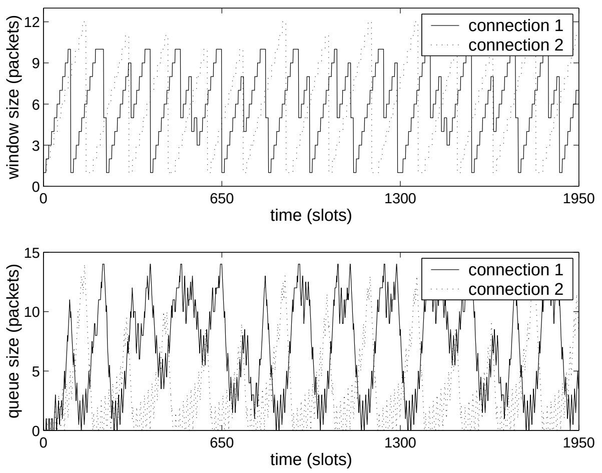

Figure 7.3: Evolution of the window and queue sizes when $x_1 = 10$, $x_2 = 13$, $Q_{\max} = 16 \times D$, $K = 1$ and $L = 13 \times D$ (RR scheduling). Identical start conditions.

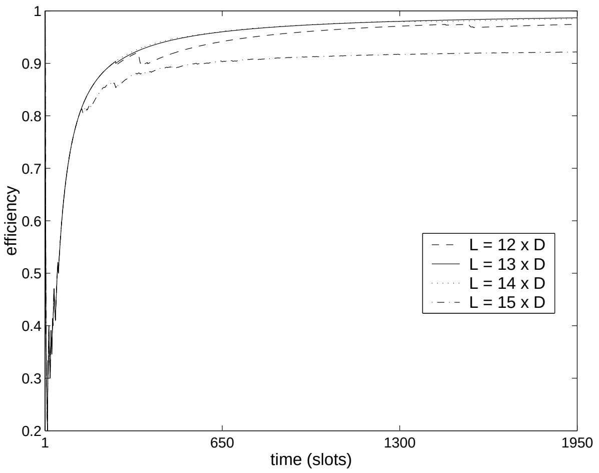

Figure 7.4: Efficiency results obtained when $x_1 = 10$, $x_2 = 13$, $Q_{\max} = 16 \times D$, and $K = 1$ for different settings of L (RR scheduling). Identical start conditions.

window, fairness is not perfect anymore. But with SD implemented, fairness again is better than when SD is not implemented. So SD partially resolves the bias against the shorter updating interval x_1 of connection 1. Figure 7.2 translates the unfairness to the difference in throughput for both connections. As can be seen, although the maximum window size of the first source is smaller than that of the second source, the first connection obtains a higher throughput than the second one. This is because the first source can send more traffic over time than the second one, since the window of the first source recuperates faster after it needed to go down.

Figure 7.3 shows the evolution of the window and queue sizes for both connections when $L = 13 \times D$. Remark that both window settings are the same at the start, but the window of the first source grows faster. Because the first source can thus send more traffic, the SD algorithm drops packets of the first connection once the threshold L is exceeded, and the window of the first source is forced down. Meanwhile, the second source can let its window grow further, and thus sends more traffic, such that the next time packets of the second connection are dropped. Remark however that the second source can never let its window grow until its maximum window size of 13 packets, while that of the first source at regular times reaches its maximum window size of 10 packets. The reason is that the first source benefits from a small window of the second source, since then the second connection sends only a few packets, plus after it has sent them, it takes a long time before the next packets arrive. So the first source has for a long period the buffer almost for its own, with periods where there is no bottleneck, such that the total buffer occupancy will be below the threshold L. Quite the reverse happens when the window of the first source is small. The second connection benefits from this, but only for a short time, since the time before new packets from the first source arrive is not so long. And when then the queue occupancy grows above L, the second connection looses packets and has to reduce its window, such that it never reaches its maximum value.

In Figure 7.4 the evolution of the efficiency for the four systems is shown. In the beginning the efficiency is low, since both sources still have to let their windows grow. After both windows are large enough to keep the buffer non-empty, the efficiency increases. Remark that all settings of L considered were chosen a few packet sizes before the end of the buffer. These settings of L were found in the previous chapter to be good settings for the threshold to obtain good efficiency and good fairness results. Also here this is the case: both the efficiency and fairness results obtained with them are better than the results obtained when SD is not implemented ($L = 15 \times D$).

7.2 Comparison with fair buffer allocation

From Chapter 5 it is known that the selective drop buffer acceptance scheme is a simpler version of the fair buffer allocation (FBA) buffer acceptance scheme. The difference between both schemes is in the calculation of the fair share (FS), and more in particular in the acceptable load ratio. For both schemes the FS is calculated as the product of the fair allocation and the acceptable load ratio, but for SD the acceptable load ratio is a simple parameter K, while for the FBA scheme it is given by

$$\text{acceptable load ratio} = Z\left(1 + \frac{Q_{\max} - Q}{Q - L}\right), \tag{7.1}$$

where Z is a scaling factor, $Q_{\max}$ the buffer capacity, L a fixed threshold and Q the buffer occupancy. Because this is the only difference between both schemes, the flowchart of the acceptance rules shown in Figure 6.1 is also valid for the FBA scheme, where the FS is now calculated using the acceptable load ratio given above. As a consequence, if in the model developed in Chapter 6 for the SD scheme the definition of the FS in equation (6.4) is replaced by

$$\text{FS}(\hat{q}_1, \hat{q}_2) = Z\left(1 + \frac{Q_{\max} - \hat{q}_1 - \hat{q}_2}{\hat{q}_1 + \hat{q}_2 - L}\right)\left(\frac{\hat{q}_1 + \hat{q}_2}{I_{\{\hat{q}_1 \neq 0\}} + I_{\{\hat{q}_2 \neq 0\}}}\right), \tag{7.2}$$

then this model can also be used for the FBA scheme.

In Table 7.1 the meaning of the acceptable load ratio is illustrated with a small example. The table shows for $Q_{\max} = 6 \times D$ and for different values of the buffer occupation Q and different settings of the threshold L the FBA acceptable load ratio when Z equals 1. For other values of Z, the numbers in this table need to be multiplied by Z. The meaning of the acceptable load ratio is the following: when a new packet of connection i arrives at the buffer, it is accepted in the buffer if the number of cells Q_i of connection i in the buffer at that moment is not larger than the number in the table corresponding to Q and L, times Q/N, the average number of cells per active connection in the buffer. Remark that the packet is always accepted if $Q \leq L$, so the value '∞' is put in the table on these positions.

	$L = 1 \times D$	$L = 2 \times D$	$L = 3 \times D$	$L = 4 \times D$	$L = 5 \times D$
$Q = 2 \times D$	5	∞	∞	∞	∞
$Q = 3 \times D$	5/2	4	∞	∞	∞
$Q = 4 \times D$	5/3	2	3	∞	∞
$Q = 5 \times D$	5/4	4/3	3/2	2	∞
$Q = 6 \times D$	1	1	1	1	1

Table 7.1: Acceptable load ratio for the FBA scheme when $Q_{\max} = 6 \times D$ (cells) and $Z = 1$ for different values of the buffer occupation Q and different settings of the threshold L.

From the columns of the table it is read that the closer Q is to the threshold L, the more times a connection is allowed to exceed the fair allocation before its packet is dropped. The closer Q is to $Q_{\max}$, the smaller this value becomes. When $Z = 1$, the values in the table never become smaller than 1, while when $Z < 1$, the values decrease when Q increases and become smaller than 1 before $Q_{\max}$ is reached. On the rows of the table it is seen that for a certain buffer occupation Q, the value in the table increases for increasing L, meaning that the closer L is to $Q_{\max}$, the more times a connection's occupation of the buffer may exceed the fair allocation before cells of its new packets are dropped. For the SD scheme, a similar table would have the value of K on all positions where now a number stands. So with SD the threshold L is only indicating from which buffer occupation level on the scheme should test on the fairness before accepting cells of a new packet, while with FBA the threshold is also used in the calculation of the acceptable load ratio.

The goal of this section is to compare the performance obtained when using the FBA acceptance scheme with the performance obtained when using the SD acceptance scheme. The starting point is again the unfair start situation used in the previous chapter where at time 0^- the window of source 1 is at its maximum, while the window of source 2 is at its minimum. As for SD, the FBA scheme is originally defined with a global queueing and FIFO scheduling strategy, but we consider it also in combination with RR and PLQF scheduling.

Consider systems with parameters as shown below:

- $x = 10$ (slots), $Q_{\max} = 12 \times D$ (cells),
- $x = 10$ (slots), $Q_{\max} = 20 \times D$ (cells),
- $x = 13$ (slots), $Q_{\max} = 16 \times D$ (cells).

Similar results as in the previous chapter (evolution of the throughputs, efficiency, fairness index over time) are obtained when considering these systems with FBA. To make an easy comparison of the results with these for the SD scheme possible, we present the results in a slightly different way as before. Figures 7.5 until 7.10 show the results. In each figure one of the systems mentioned before combined with one of the scheduling schemes is considered, for different settings of the threshold L and for three settings of the FBA parameter Z,

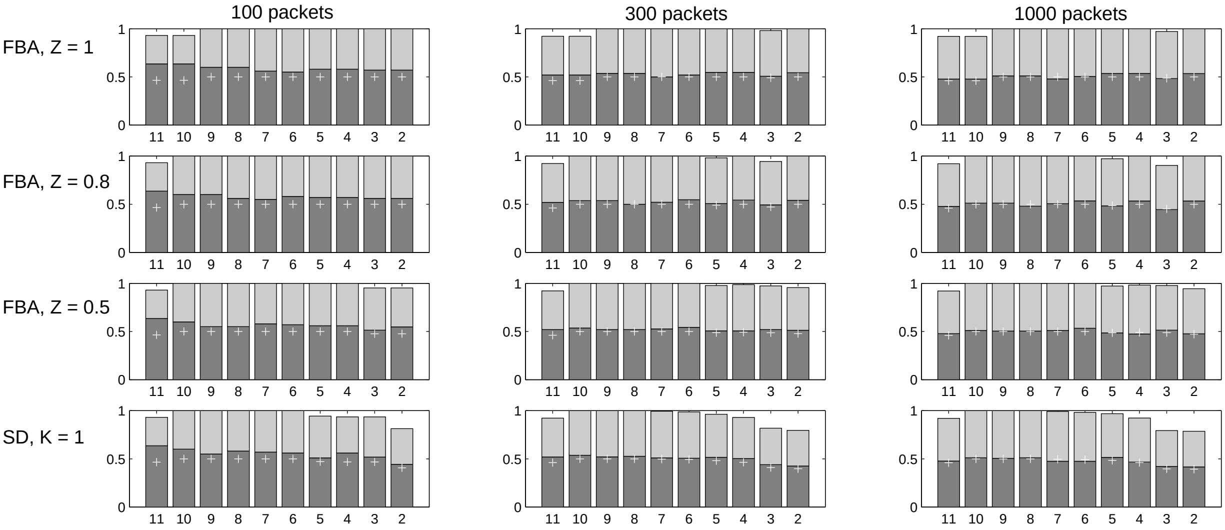

Figure 7.5: Efficiency and fairness results obtained with RR scheduling when $x = 10$ and $Q_{\max} = 12 \times D$.

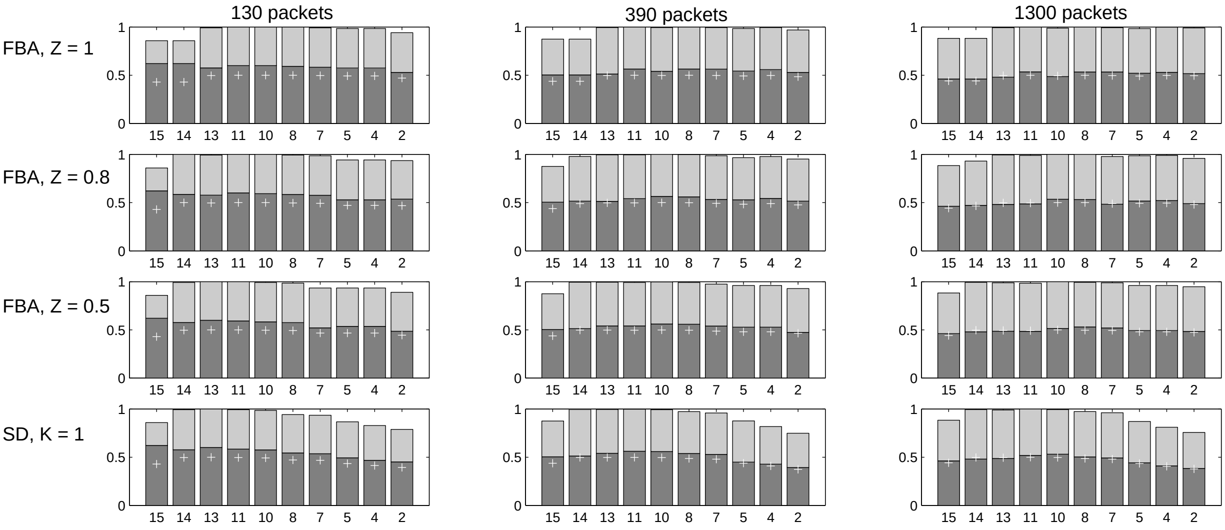

Figure 7.6: Efficiency and fairness results obtained with RR scheduling when $x = 13$ and $Q_{\max} = 16 \times D$.

i.e., $Z = 1$, $Z = 0.8$ and $Z = 0.5$. Results obtained with SD when $K = 1$ are also shown. Subplots in the figures on the same column show results obtained after a certain fixed number N of packets of cells ($N = 10x, 30x$ or $100x$ packets) have left the buffer. The height of the bars represents the efficiency obtained with a scenario at that time, while the dark, resp. light gray areas represent the throughput of connection 1, resp. connection 2 at that time. The higher the bars, the less time was needed to successfully deliver the N packets to the destination. When the height of the bar is 1, this means that the N packets have left the buffer after the minimum time possible of N slots. The horizontal part of the white plus-sign in each bar indicates half the height of the bar. Since the efficiency after k slots is the sum of the throughputs of the two connections after k slots, the plus-sign indicates what the throughputs of both connections should be to have perfect fairness. The numbers on the horizontal axis of each subplot indicate the setting of the threshold in the buffer, expressed as a multiple of D cells. Remark that for comparison, the leftmost bar in each plot gives results obtained when neither SD, neither FBA is implemented. So in subplots in the same column of a figure, the leftmost bar is always the same.

From Figures 7.5 until 7.10, and based on the main observations already made in the previous chapter about the SD algorithm, the following similarities and differences between the FBA and SD acceptance schemes are noticed:

- Like the SD algorithm, also the FBA algorithm has a large positive effect on the fairness results, irrespective of the exact settings of the parameters L and Z of the algorithm. In the figures this is mainly seen when $N = 10x$ packets and when $N = 30x$ packets. Afterwards also the system without acceptance scheme approaches perfect fairness because the two sources considered are equal. The fairness obtained with FBA is comparable to that obtained by SD.

- Also for the FBA scheme the parameter setting has more influence on the efficiency, although not so extreme as for the SD scheme. With SD, the efficiency generally increases with increasing L, as is also clearly seen on the figures in this section. For SD it was concluded that a setting of L a few packet sizes before the end of the buffer resulted most of the time in good efficiency, so the results obtained with FBA should be compared with the results obtained in this case. For the FBA scheme, no strict relation between the setting of L and the efficiency appears from the results. Most settings give efficiency results that are comparable with good results obtained with the SD algorithm.

- With RR scheduling, the efficiency obtained when FBA is used is in most cases above that obtained when no acceptance scheme is implemented, and also often perfect (i.e., constantly equal to 1). So FBA with RR scheduling results in both good efficiency and fairness. When FBA is combined with FIFO or PLQF scheduling, the efficiency values obtained are often above these obtained without acceptance scheme, but not always. With PLQF scheduling, usually the highest efficiency results are obtained when a higher L is combined with a lower Z, or vice versa.

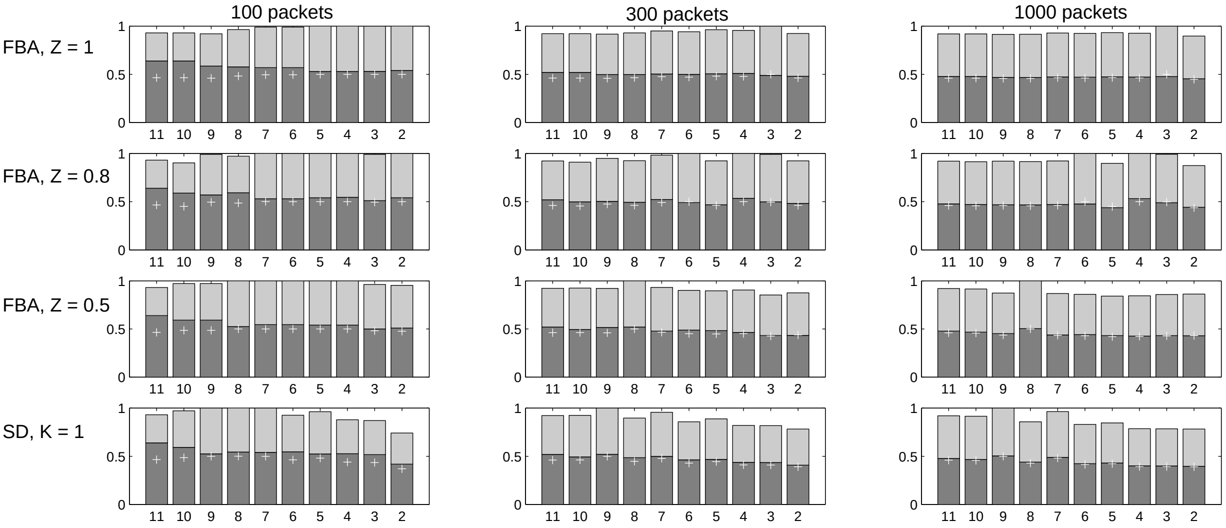

Figure 7.7: Efficiency and fairness results obtained with FIFO scheduling when $x = 10$ and $Q_{\max} = 12 \times D$.

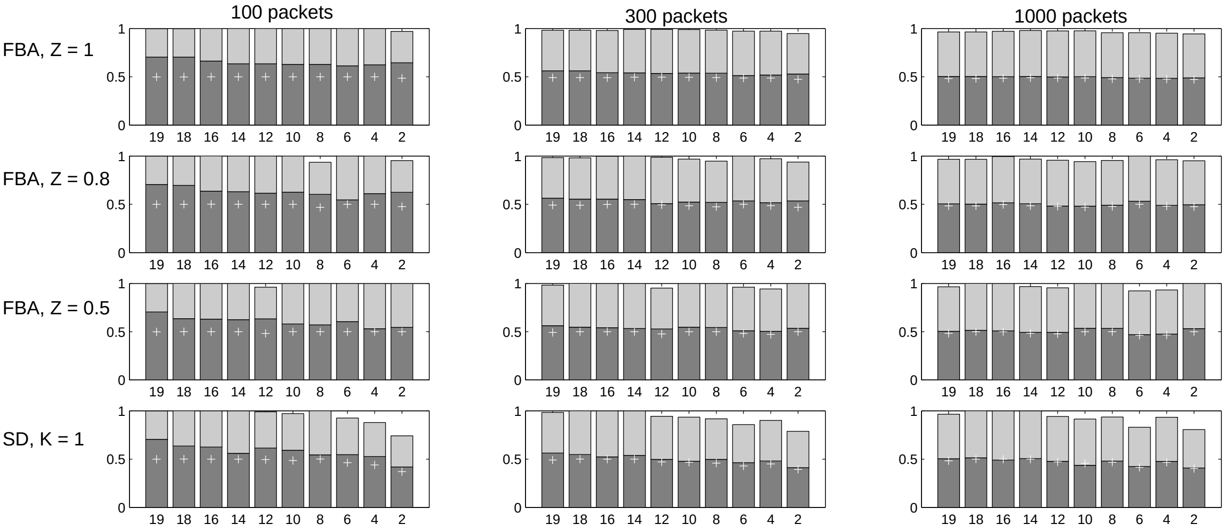

Figure 7.8: Efficiency and fairness results obtained with FIFO scheduling when $x = 10$ and $Q_{\max} = 20 \times D$.

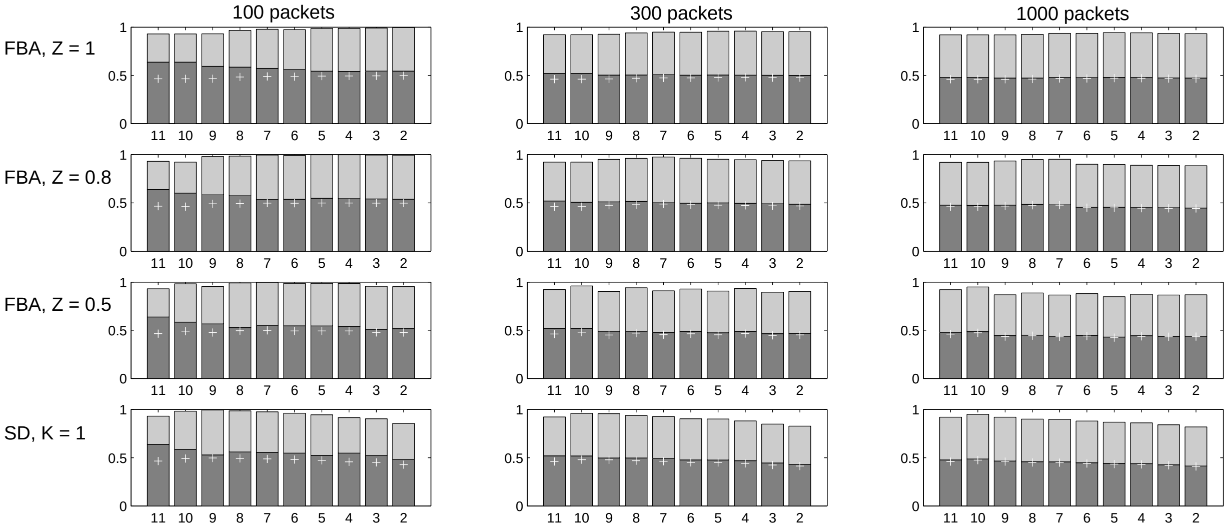

Figure 7.9: Efficiency and fairness results obtained with PLQF scheduling when $x = 10$ and $Q_{\max} = 12 \times D$.

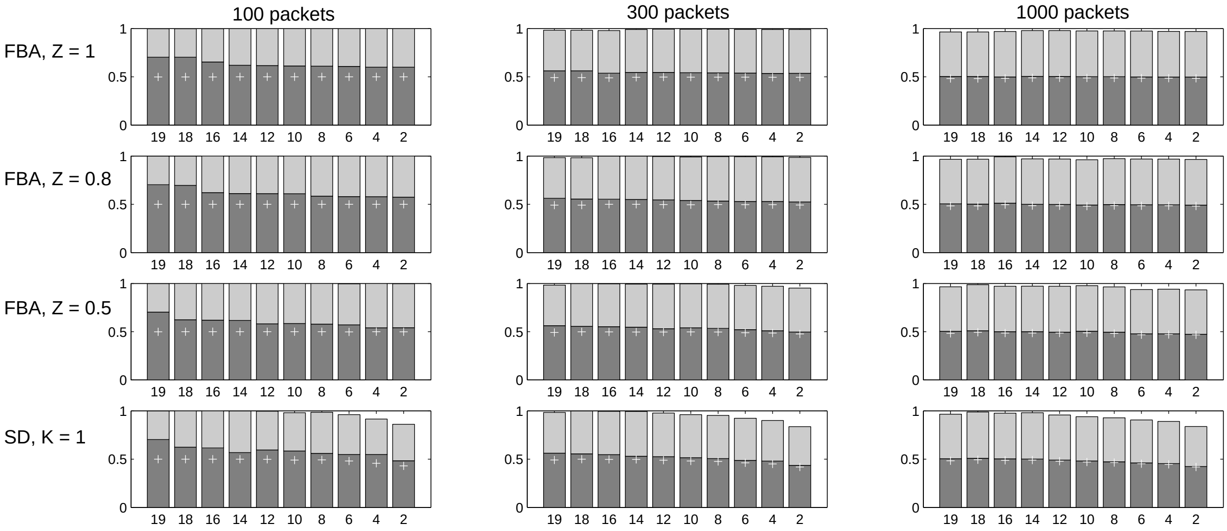

Figure 7.10: Efficiency and fairness results obtained with PLQF scheduling when $x = 10$ and $Q_{\max} = 20 \times D$.

7.3 Conclusions and related work

In this chapter two extensions to the model developed in Chapter 6 were considered. First the model was adapted such that the interval after which the sources update their window is different for both sources. This implies that a kind of inherent unfairness is introduced in the model, since the window of the source with the smallest update interval can grow faster and thus this source can recuperate faster after its window is forced down due to losses. The scenario we considered illustrated clearly that because the two sources are now different, some unfairness in the throughputs of the two connections stays present for ever. However, SD can partially resolve the bias that exists against the source with the shorter updating interval and improve the fairness results.

Secondly the definition of the fair share was modified such that the fair buffer allocation (FBA) scheme could be considered. Results obtained with this scheme show that as the SD scheme, also the FBA scheme has a large positive effect on the fairness results, irrespective of the exact setting of its parameters. These parameters have, again as with the SD scheme, more influence on the efficiency results, although their influence is not so large as with the SD scheme. Both the fairness and the efficiency results obtained with the FBA scheme are comparable with the results that are obtained with the SD scheme, for well-chosen parameters.

Most theoretical models of frame aware buffer acceptance schemes in the literature are about partial packet discard (PPD) and early packet discard (EPD). In [17] the behavior of the EPD scheme is studied by considering the evolution of the buffer level over time using a deterministic model where r sources continuously send packets of cells, whose boundaries are offset from one another by an equal amount. [83] analyzes the worst-case excess buffer capacity requirement for the EPD and for SD-like schemes (the difference is in the calculation of the fair share (FS), which is calculated as FS $= KL/N$ instead of FS $= KQ/N$) that use global queueing and FIFO scheduling, or per-VC queueing and RR scheduling. An upperbound value on the total buffer occupancy is derived for all schemes. With EPD this upper bound is reached when all sources begin sending a new packet just one cell time before the queue occupation exceeds the EPD threshold. With the SD-like schemes the upper bound is reached with the staggered input schedule where VC j begins sending a new packet just before the buffer occupation of VC $j-1$ reaches its maximum value.

In [63], PPD and EPD are compared with tail drop (TD) using a single source. The TD system is modeled as an M/M/1/N queueing system. Each of the Poisson arrivals representing a cell is assumed to belong to the same packet as the previous arrival with a certain probability p, and is the first cell of a new packet with probability $1-p$. The same system is considered to model PPD and EPD, by distinguishing between two modes: the normal mode 0 in which packets are admitted to the system and the discarding mode 1 in which packets are discarded. The states of the M/M/1/N system representing that a certain number j of cells are present in the system are now split into two states $(j,0)$ and

$(j, 1)$, where 0 and 1 indicate the mode the system is in. With PPD, when the system is in state $(N, 0)$, the buffer is full. A cell that arrives at this state is discarded and the system enters state $(N, 1)$. Since the following cells belonging to the same packet must be discarded, the system stays in mode 1 until a new packet starts and the system is not full on arrival of its first cell. The EPD scheme is modeled analoguously, except that an additional threshold K is defined. If a first cell of a packet arrives when the system occupation is $j \geq K$, the cell is not admitted to the system and the system enters state $(j, 1)$ in discarding mode.

An exact analysis of the packet loss probability obtained with TD, PPD and EPD for M homogeneous sources is carried out in [57]. Each source generates cells according to a two state discrete-time Markovian on/off source, where a packet is compounded of cells that are generated in the same on period. An off period represents the inactive period between adjacent packets. For packet level performance analysis each two state Markov source is expanded into a three state source by adding an on* state, in order to distinguish successful packets from corrupted/lost packets while the source is active. A very similar analysis is performed in [59], except that now continuous time on/off sources are used.

In [90] we developed a model with similar input sources as in [57], in a first attempt to model the SD scheme with per-VC queueing and RR scheduling. Remark that important differences from the modeling point of view between this scheme and EPD or PPD are that this scheme uses per-VC accounting information to decide if cells of a new packet may enter the buffer or not, and that the queues are served according to a cyclic service strategy. The analysis we made was approximate in the sense that a queueing system with one tagged queue and repeated server vacations together with the occupation of all per-VC queues was considered. Although the results of this approximate model are in accordance with results obtained by simulating the exact model (i.e., the model that considers all queues together and exact RR scheduling), this model lacks the possibility to obtain other results than packet discard ratios, queue length distributions and cell loss ratios. So an important aim of the acceptance scheme considered, namely fairness, could not be assessed with the model. Furthermore, as in the other models described above, the sources were also not responsive to losses.

Summary

As its title suggests, this thesis consists of two parts, since it focuses on two separate topics that are related to the performance evaluation of telecommunication network elements: (i) the superposition of Markovian traffic sources, and (ii) frame aware buffer acceptance schemes.

Part I: Circulant matching of the superposition of D-BMAPs

A basic problem in the dimensioning and performance evaluation of telecommunication network elements is the computation of the buffer occupancy and waiting time distribution of a single server queue, whose input consists of a superposition of processes modeling traffic streams. Starting from the assumption that a traffic stream is modeled by a D-BMAP (discrete-time batch Markovian arrival process), which is a quite general discrete-time Markov model, a representation of the aggregation or superposition of D-BMAPs is needed, since the input to network elements generally consists of multiple traffic streams. In theory, this aggregation is exactly described by a new D-BMAP. A major problem however is the explosion of the state space of this new D-BMAP when the number of input streams takes values that are typical for real life situations. In the first part of the thesis, a method called *circulant matching* is proposed, which constructs another D-BMAP with a smaller state space to replace the exact superposition.

Chapter 1 reviews some definitions and results about finite-state stationary Markov chains and their eigenstructure, that are used in the following chapters. It also gives the definition and some properties of the D-BMAP and the D-BMAP/D/1/K queue and motivates why the exact superposition of D-BMAPs should be avoided.

Chapter 2 presents the details of the circulant matching method. The purpose of this method is to construct another D-BMAP to replace the exact superposition of independent D-BMAPs, while matching the autocorrelation sequence (characterized in the frequency domain by the power spectrum) and the stationary cumulative distribution of the input rate process of the exact superposition. The transition matrix of the D-BMAP is chosen to be circulant, in order to avoid solving an inverse spectrum problem. First expressions for the autocorrelation sequence, power spectrum and stationary cumulative distribution of a single D-BMAP are derived. For the autocorrelation sequence and the power spectrum,

these expressions are written as a function of the eigenvalues and eigenvectors of the transition matrix of the D-BMAP. Then the circulant D-BMAP is introduced, and based on the results obtained before, formulas for its autocorrelation sequence, power spectrum and stationary cumulative distribution are obtained. Also the condition for a circulant transition matrix to be irreducible and some properties about periodic circulants are proven. Finally expressions for the power spectrum and stationary cumulative distribution of the exact superposition of independent D-BMAPs, that can be calculated without explicitely constructing the exact superposition, are derived. All these results lay the foundation for the description of how the circulant D-BMAP that replaces the exact superposition is constructed. This construction consists of two steps: the matching of the power spectra and the matching of the stationary cumulative distribution of the input rate process of the circulant D-BMAP and of the exact superposition of D-BMAPs. First the transition matrix of the circulant D-BMAP is constructed, in such a way that it has as eigenvalues among others all eigenvalues of the D-BMAPs in the superposition, since it are these eigenvalues which contribute to the power spectrum of the superposition. Then the factors by which each eigenvalue of the circulant contributes to the power spectrum of the circulant D-BMAP are fixed, such that the power spectrum of the circulant matches that of the exact superposition. Secondly the input rate vector of the circulant D-BMAP is constructed, taking into account the parameters that were already fixed in the previous step, such that also the stationary distribution of the circulant D-BMAP matches that of the exact superposition.

The circulant matching method for D-BMAPs is based on a component of a measurement-based tool developed by San-qi Li et al. [46]. An important difference with the method of San-qi Li is that he works in continuous time, while a D-BMAP is a discrete-time model. So to replace the superposition of D-BMAPs by a new circulant D-BMAP, we had to adapt the method for discrete time. Simultaneously, the method was extended such that the periodicity which is present in the transition matrix of D-BMAPs that model periodic traffic streams, and which is thus also noticed in their superposition, is preserved.

The circulant matching method allows us to solve some realistic queueing problems, as is illustrated in Chapter 3. But it also has its limitations. A first problem is in the construction of the circulant transition matrix, and more in particular in the number of possible choices that need to be investigated for its dimension and the indices of its eigenvalues. When the predefined set of eigenvalues the circulant should have becomes large (say more than 10, after some reductions we proposed), it might take a long time before a circulant with these values as eigenvalues is found. So the circulant matching method is only useful when all D-BMAPs in the superposition are identical, or can be divided into a limited group of identical ones, since then many of their eigenvalues are identical. A positive point on the other hand is that the same circulant transition matrix can be used when considering a superposition of another number of the same D-BMAPs. The difference is in the rate vector associated with the circulant D-BMAP, not in its transition matrix. A second possible problem is in the construction of this rate vector when a large part of the probability mass of the rate distribution of the exact superposition is situated at the value zero, or very close to it, as can occur when considering the superposition of on/off sources. In that case, it can happen that no solution for a constrained minimization problem that needs to be solved when constructing the rate vector of the circulant D-BMAP, exists.

Chapter 3 presents numerical examples and applications of the circulant matching method. First the rather theoretical description of the different steps of the method in the previous chapter is illustrated by commenting upon a numerical example where a circulant D-BMAP is constructed to replace the superposition of 50 identical 16-state D-BMAPs of period 3. Then the superposition of identical two dimensional MMBPs (Markov modulated Bernouilli arrival processes) is considered. For these types of sources, it is possible to compare the system lengths obtained when using either the circulant approximation of the superposition or the exact superposition as input to a queueing system, because the exact superposition of M identical two dimensional sources is also exactly described by an $(M+1)$-dimensional Markov source. First general MMBP sources are considered, and the system length distribution obtained with a circulant as input matched the exact system length distribution rather well. Then a special type of MMBP sources is considered, namely on/off sources. For these type of sources the agreement between the system length distribution obtained with the circulants as input and the exact distribution is bad. The reason is that the rate distribution of the circulant very badly matches that of the exact superposition, because a large part of the probability mass of the rate distribution is located at rate zero. The same fact sometimes even causes the circulant matching method to fail in finding a valid rate distribution for the circulant. Using the two dimensional sources it is also illustrated that it is necessary for a matching method to take both first and second order statistics of the arrival process into account, since when considering only one of both, the result of the matching process might badly reflect the queueing behavior of the sources it replaces. Another application that is considered in Chapter 3 is the superposition of a periodic MPEG source model. Using the circulant matching method, we obtained a theoretical CAC boundary for a mix of two types of MPEG sources. Remark that due to the dimension of the MPEG source models (52 and 65 states) and the realistic number of such sources considered, it is impossible to obtain the exact queueing results using the exact superposition. So we compared the theoretically obtained results with experimentally obtained results. The results confirm the accuracy of the circulant matching method.

Part II: Frame aware buffer acceptance schemes

In the second part of the thesis frame aware buffer acceptance schemes are considered. When packet or frame based data is transported over an ATM (asynchronous transfer mode) network, these packets are segmented into cells. A buffer acceptance scheme in a network element decides about which cells are allowed to enter its buffer, and which cells have to be dropped. Because the loss of a single cell of a frame leads to a corrupted frame that is in any case discarded at the destination, buffer acceptance schemes that are frame aware, i.e., try to accept or discard all cells of a same frame, thus improve the efficiency.

Not only efficiency is an issue, but also the fairness among the effective throughputs of the different connections. So also schemes that preferentially drop frames from connections that use more bandwidth than one would call fair have been defined.

Chapter 4 reviews some concepts related to buffer acceptance and gives a more exact definition of a frame. Since most non-real-time packet based data traffic in a network is TCP traffic, also a short introduction on TCP and on the two ATM service categories that are most suited to transport TCP traffic, i.e., unspecified bit rate (UBR) and guaranteed frame rate (GFR), is given. Also the definition of some performance measures that are considered in the following chapters is given.

Chapter 5 gives an overview of the most representative buffer acceptance schemes that have been proposed in the literature for use with the UBR and GFR ATM service categories. Characteristic of all schemes is their AAL5 frame awareness: if the scheme decides to accept, respectively discard, the first cell of a frame, it will try to accept, respectively drop, all cells of the same frame, since incomplete frames are of no use at the destination. The principles of two of the earliest proposed schemes, namely partial packet discard (PPD) and early packet discard (EPD), are found back in many of the more sophisticated schemes. To be able to accept the non-first cells of a frame from which the first cell was accepted, most acceptance schemes use a threshold, as in EPD, to provide some excess capacity in the buffer. If in spite of this excess capacity a cell is lost because of buffer overflow, the remaining cells of its frame are discarded as in PPD.

No QoS commitments are made by the network to UBR connections, but most recent buffer acceptance schemes for UBR try to provide a fair allocation of the bandwidth to competing connections. This is done by aiming at a fair allocation of the buffer capacity among the connections, using the principle behind the fair buffer allocation (FBA) scheme that a connection that gets more than its fair share of the buffer space will also get more than its fair share of the bandwidth. The same principle is used in some of the buffer acceptance schemes for GFR, although the fairness is an issue then only to the excess capacity. The first concern of buffer acceptance schemes for GFR is to provide each connection with its minimum cell rate service guarantee.

Relying on the attractive properties of the random early detection (RED) scheme in IP gateways, some schemes for ATM using the principles behind RED are proposed. The most important feature of these schemes is their ability to keep the average buffer size, and thus also the average queueing delay, low.

Most buffer acceptance schemes proposed to support GFR connections can be grouped in one of three main categories. The first category contains schemes relying on the tagging of ineligible frames to provide the per-VC minimum rate guarantees to the different connections. The schemes in the second category use per-VC accounting and per-VC queueing, making per-VC scheduling possible. With an appropriate per-VC scheduling algorithm, each VC is, when active, allocated its reserved bandwidth. The schemes in the third category use per-VC accounting in a FIFO buffer, since the cost of per-VC queueing and per-VC scheduling may be too high for a service category like GFR.

For buffer acceptance schemes not only the principles behind the acceptance algorithm are important, but also the accounting information the algorithm can base its decisions on and the queueing and scheduling strategy used. In Chapter 5 also a summary of this information for the main buffer acceptance schemes discussed is provided.

Chapter 6 considers one of the schemes discussed in the previous chapter, namely selective drop (SD), that aims at discarding frames in a fair way. The transient performance of SD is analyzed when traffic is generated by sources for which the amount of traffic they can send is controlled by a window that responds to the presence or absence of losses (as TCP sources do). For this goal a theoretical model is developed, where two responsive sources send traffic in fixed-sized packets of cells, via a buffer on which the SD buffer acceptance algorithm is implemented. Transient efficiency and fairness results are then obtained from the model.

First some identical scenarios are considered under different start conditions, among which an unfair start condition, which corresponds to a situation where one source alone has been sending traffic for some time, and suddenly the second source starts also sending traffic. Conclusions are that: (i) When the input traffic is generated by two identical sources, none of which is offered a preferential treatment by the buffer acceptance or the scheduling scheme, then the mean window sizes and the mean buffer occupations coincide under identical start values for both connections, resulting in equal throughput for both connections and thus perfect fairness. (ii) The fairness approaches perfect fairness as soon as the system has recovered from the unfairness caused by an unfair start situation. This illustrates the importance of a transient analysis when observing the behavior of the SD scheme towards an unfair start situation. (iii) A difference in the amount of output from the buffer at the beginning due to different start conditions for the system stays perceptible in the efficiency values. A difference in the amount of output of the two connections at the beginning due to unequal start values for both connections stays perceptible for some while in the throughput and fairness values.

Then it is illustrated with some examples that due to the responsiveness of the sources, it is not necessarily true anymore that being more conservative in accepting packets implies a lower efficiency, as would be the case when non-responsive sources would be used. There is also not necessarily a trade-off between efficiency and fairness.

Also the influence of the parameters of the SD algorithm (SD has two parameters, a threshold L and another parameter K) on the efficiency and fairness results is studied when starting from the unfair start situation. The most important conclusion of this study is that the presence of the SD algorithm has a large positive effect on the fairness results, irrespective of the exact setting of the parameters of the algorithm. On the efficiency results however, these parameters have more influence. The SD algorithm is considered in combination with three scheduling algorithms. With round robin (RR) and probabilistic longest queue first (PLQF) scheduling, the efficiency generally increases when the threshold L increases, and choosing L at a few packet sizes less than the size of the buffer results in a good setting. With RR scheduling, the chance is rather high that the efficiency values obtained are then even above these obtained when SD is not implemented (so there is no trade-off between efficiency and fairness then). With PLQF scheduling, this chance is reasonable. Remark however that with RR scheduling, sometimes the efficiency is lower than expected because of synchronization effects. When using PLQF scheduling, no lasting synchronization will occur because of the probabilistic character of the scheduling algorithm in these scenarios. Also with FIFO scheduling, synchronization can occur. With FIFO scheduling it is much harder to make a conclusion about the best setting of the threshold L, since no real relation was found between a change of L and a corresponding change of the efficiency. But choosing it a few packet sizes less than the size of the buffer as with RR and PLQF scheduling gave in most scenarios rather good results. The parameter K of the SD algorithm has also more influence on the efficiency results than on the fairness results. Increasing K has principally a positive effect on the efficiency when L is set at a small value. When the setting of L is larger, this positive effect is still seen with RR scheduling, but with PLQF and FIFO scheduling the probability is rather high that the efficiency will be lower than when K is chosen equal to one. As a general conclusion, it is recommended to implement SD to increase the fairness, but with a parameter setting focusing on the efficiency results.

Chapter 7 considers two extensions to the model developed in Chapter 6. First the model was adapted such that the interval after which the sources update their window is different for both sources. This implies that a kind of inherent unfairness is introduced in the model, since the window of the source with the smallest update interval can grow faster and thus this source can recuperate faster after its window is forced down due to losses. The scenario we considered illustrated clearly that because the two sources are now different, some unfairness in the throughputs of the two connections stays present for ever. However, SD can partially resolve the bias that exists against the source with the shorter updating interval and improve the fairness results.

Secondly the definition of the fair share was modified such that also the fair buffer allocation (FBA) scheme, another frame aware buffer acceptance scheme that aims at fairness, could be considered. Results obtained with this scheme show that as the SD scheme, also the FBA scheme has a large positive effect on the fairness results, irrespective of the exact setting of its parameters. These parameters have, again as with the SD scheme, more influence on the efficiency results, although their influence is not so large as with the SD scheme. Both the fairness and the efficiency results obtained with the FBA scheme are comparable with the results that are obtained with the SD scheme, for well-chosen parameters.

Bibliography

[1] E. Aarstad, S. Blaabjerg, F. Cerdan, S. Peeters, and K. Spaey. Experimental investigation of CAC and effective bandwidth for video and data. In *Proceedings ATM Traffic Symposium*, Mykonos, Greece, 1997.

[2] E. Aarstad, S. Blaabjerg, F. Cerdan, S. Peeters, and K. Spaey. CAC investigation for video and data. In *Proceedings IFIP BC'98*, pages 356–367, Stuttgart, Germany, 1998.

[3] M. Allman, V. Paxson, and W. Stevens. *TCP Congestion Control*, Apr. 1999. RFC 2581.

[4] The ATM Forum. *Performance Testing Specification*, Oct. 1999. AF-TEST-TM-0131.000.

[5] The ATM Forum. *Traffic Management Specification, Version 4.1*, Sept. 1999. AF-TM-0121.000.

[6] The ATM Forum. *Addendum to Traffic Management V4.1 for an Optional Minimum Desired Cell Rate Indication for UBR*, July 2000. AF-TM-0150.000.

[7] D. Basak and S. K. Pappu. *GFR Implementation Alternatives with Fair Buffer Allocation Schemes*, July 1997. ATM Forum Contribution 97-0528.

[8] D. A. Bini, B. Meini, and V. Ramaswami. Analyzing M/G/1 paradigms through QBDs: the role of the block structure in computing the matrix G. In *Proceedings Third Conference on Matrix Analytic Methods*, pages 73–86, 2000.

[9] C. Blondia. A discrete-time batch Markovian arrival process as B-ISDN traffic model. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32(3,4):3–23, 1993.

[10] C. Blondia and O. Casals. Statistical multiplexing of VBR sources: A matrix-analytical approach. *Performance Evaluation*, 16:5–20, 1992.

[11] C. Blondia and F. Panken. Traffic profile of a connection in an ATM network with application to traffic control. In *Proceedings BRAVE workshop*, Milan, Italy, 1995.

[12] C. Blondia and T. Theimer. A discrete-time model for ATM traffic. Technical Report PRLB_123_0018_CD_CC / UST_123_0022_CD_CC, RACE, Oct. 1989.

[13] O. Bonaventure. *Integration of ATM under TCP/IP to provide Services with Minimum Guaranteed Bandwidth.* PhD thesis, Université de Liège, Oct. 1998.

[14] O. Bonaventure. A simulation study of TCP with the GFR service category. In A. Danthine, O. Spaniol, W. Effelsberg, and D. Ferrari, editors, *High Performance Networks for Multimedia Applications.* Kluwer Academic Publishers, 1998.

[15] O. Bonaventure and J. Nelissen. Guaranteed frame rate: a better service for TCP/IP in ATM networks. *IEEE Network*, 15(1):46–54, Jan. 2001.

[16] V. Bonin, F. Cerdán, and O. Casals. A simulation study of differential fair buffer allocation. In *Proceedings ICAM 2000*, Heidelberg, Germany, 2000.

[17] M. Casoni and J. S. Turner. On the performance of early packet discard. *IEEE Journal on Selected Areas in Communications*, 15(5):892–902, June 1997.

[18] E. Çinlar. *Introduction to Stochastic Processes.* Prentice-Hall, 1975.

[19] F. Cerdán and O. Casals. Performance of different TCP implementations over the GFR service category. *Interoperable Communications Network Magazine*, 2:273–286, Jan. 2000.

[20] C.-T. Chan, Y.-C. Chen, and P.-C. Wang. An efficient traffic control approach for GFR services in IP/ATM internetworks. In *Proceedings Globecom'98*, Sydney, Australia, 1998.

[21] K. Cheon and S. S. Panwar. Early selective packet discard for alternating resource access of TCP over ATM-UBR. In *Proceedings LCN'97*, Minneapolis, MN, 1997.

[22] K. Cheon and S. S. Panwar. On the performance of ATM-UBR with early selective packet discard. In *Proceedings ICC'98*, Atlanta, GA, 1998.

[23] M. Conti, S. Ghezzi, and E. Gregori. Aggregation of Markovian sources: Approximations with error control. In *Proceedings Networking 2000*, pages 350–361, Paris, France, 2000.

[24] M. De Prycker. *Asynchronous Transfer Mode.* Prentice-Hall, third edition, 1995.

[25] O. Elloumi and H. Afifi. RED algorithm in ATM networks. In *Proceedings IEEE ATM'97 Workshop*, Lisbon, Portugal, 1997.

[26] O. Elloumi and H. Afifi. Evaluation of FIFO-based buffer management algorithms for TCP over guaranteed frame rate service. In *Proceedings IEEE ATM'98 Workshop*, Fairfax, VA, 1998.

[27] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno and SACK TCP. *ACM Computer Communications Review*, 26(3):5–21, July 1996.

[28] C. Fang and A. Lin. *On TCP Performance of UBR with EPD and UBR-EPD with a Fair Buffer Allocation Scheme*, Dec. 1995. ATM Forum Contribution 95-1645.

[29] S. Floyd and V. Jacobson. Traffic phase effects in packet-switched gateways. *Journal of Internetworking: Research and Experience*, 3(3):115–156, Sept. 1992.

[30] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug. 1993.

[31] Y. Gachoud. A recursive adaptive D-BMAP parameters estimation based on information measure. In *Proceedings ITC-15*, volume 2a, pages 663–673, Washington, WA, 1997.

[32] N. Ghani, S. Nananukul, and S. Dixit. ATM traffic management considerations for facilitating broadband access. *IEEE Communications Magazine*, pages 98–105, Nov. 1998.

[33] R. Goyal, R. Jain, S. Fahmy, B. Vandalore, and M. Goyal. Buffer management for TCP over the ATM GFR service. Submitted to Computer Communications.

[34] R. Goyal, R. Jain, S. Fahmy, B. Vandalore, and S. Kalyanaraman. Design issues for providing minimum rate guarantees to the ATM unspecified bit rate service. In *Proceedings IEEE ATM'98 Workshop*, Fairfax, VA, 1998.

[35] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and S.-C. Kim. *Further Results on UBR+: Effect of Fast Retransmit and Recovery*, dec 1996. ATM Forum Contribution 96-1761.

[36] R. Goyal, R. Jain, S. Kalyanaraman, S. Fahmy, and S.-C. Kim. UBR+: Improving performance of TCP over ATM-UBR service. In *Proceedings ICC'97*, volume 2, pages 1042–1048, Montreal, Canada, 1997.

[37] A. Graham. *Kronecker Products and Matrix Calculus: with Applications.* Ellis Horwood Limited, 1981.

[38] W. Grassmann, M. Taksar, and D. Heyman. Regenerative analysis and steady state distribution for Markov chains. *Operations Research*, 33:1107–1116, 1985.

[39] R. Gusella. Characterizing the variability of arrival processes with indexes of dispersion. *IEEE Journal on Selected Areas in Communications*, 9(2):203–211, Feb. 1992.

[40] B. Hajek and L. He. On variations of queue response for inputs with the same mean and autocorrelation function. *IEEE/ACM Transactions on Networking*, 6(5):588–598, Oct. 1998.

[41] J. Heinanen and K. Kilkki. A fair buffer allocation scheme. *Computer Communications*, 21(3):220–226, 1998.

[42] B. E. Helvik. MPEG source type models for the STG (Synthesized Traffic Generator). Technical Report STF40 A96016, SINTEF, Feb. 1996.

[43] D. Heyman and D. Lucantoni. Modeling multiple IP traffic streams with rate limits. In *Proceedings ITC-17*, Salvador da Bahia, Brazil, 2001.

[44] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Stochastic Processes*. Houghton Mifflin, 1972.

[45] C.-L. Hwang and S. Q. Li. On input state space reduction and buffer noneffective region. In *Proceedings Infocom'94*, pages 1018–1028, Toronto, Canada, 1994.

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

[47] ATM in Europe, ACTS trials. A Publication of the InfoWin Project.

[48] D. L. Isaacson and R. W. Madsen. *Markov Chains Theory and Applications*. John Wiley & Sons, 1976.

[49] M. R. Izquierdo and D. S. Reeves. A survey of statistical source models for variable-bit-rate compressed video. *Multimedia Systems*, 7:199–213, 1999.

[50] V. Jacobson. Congestion avoidance and control. In *Proceedings SIGCOMM'88*, Stanford, CA, 1988.

[51] D. L. Jagerman, B. Melamed, and W. Willinger. Stochastic modeling of traffic processes. In J. Dshalalow, editor, *Frontiers in Queueing: Models, Methods and Problems*. CRC Press, 1996.

[52] R. Jain, D. Chiu, and W. Hawe. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. Technical Report DEC-TR-301, DEC, Sept. 1984.

[53] R. Jain, A. Durresi, and G. Babic. *Throughput Fairness Index: An Explanation*, Feb. 1999. ATM Forum Contribution 99-0045.

[54] R. Jain, R. Goyal, S. Kalyanaraman, and S. Fahmy. *Performance of TCP over UBR and Buffer Requirements*, Apr. 1996. ATM Forum Contribution 96-0518.

[55] L. Jaussi, M. Lorang, and J. Nelissen. A detailed experimental performance evaluation on TCP over UBR. In *Proceedings ICATM'98*, pages 214–223, Colmar, France, 1998.

[56] A. E. Kamal. Efficient solution of multiple server queues with application to the modeling of ATM concentrators. In *Proceedings Infocom'96*, pages 248–254, San Francisco, CA, 1996.

[57] K. Kawahara, K. Kitajima, T. Takine, and Y. Oie. Packet loss performance of selective cell discard schemes in ATM switches. *IEEE Journal on Selected Areas in Communications*, 15(5):903–913, June 1997.

[58] W.-J. Kim and B. G. Lee. The FB mechanism for TCP over UBR in subnet ATM models. *IEICE Transactions on Communications*, 82(3):481–488, Mar. 1999.

[59] Y. Kim and S. Q. Li. Performance analysis of data packet discarding in ATM networks. In *Proceedings ITC-15*, pages 89–100, Washington, WA, 1997.

[60] T. Lakshman, A. Neidhardt, and T. J. Ott. The drop from front stragegy in TCP over ATM. In *Proceedings Infocom'96*, pages 1242–1250, San Francisco, CA, 1996.

[61] R. Landry and I. Stavrakakis. Multiplexing generalized periodic Markovian sources with an application to the study of VBR video. In *Proceedings ICC'94*, New Orleans, LA, 1994.

[62] R. Landry and I. Stavrakakis. Non-deterministic periodic packet streams and their impact on a finite-capacity multiplexer. In *Proceedings Infocom'94*, Toronto, Canada, 1994.

[63] Y. Lapid, R. Rom, and M. Sidi. Analysis of discarding policies in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 16(5):764–777, June 1998.

[64] G. Latouche and G. W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*. Kluwer, 1995.

[65] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems.* SIAM, second edition, 1995.

[66] J.-Y. Le Boudec. An efficient solution method for Markov models of ATM links with loss priorities. *IEEE Journal on Selected Areas in Communications*, 9(3), Apr. 1991.

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

[68] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering.* Addison-Wesley Publishing Company, second edition, 1994.

[69] H. Li, K.-Y. Siu, H.-Y. Tzeng, C. Ikeda, and H. Suzuki. On TCP performance in ATM networks with per-VC early packet discard mechanisms. *Computer Communications*, 19(13):1065–1076, Nov. 1996.

[70] H. Li, K.-Y. Siu, H.-Y. Tzeng, C. Ikeda, and H. Suzuki. A simulation study of TCP performance in ATM networks with ABR and UBR services. In *Proceedings Infocom'96*, San Francisco, CA, 1996.

[71] S. Q. Li and C.-L. Hwang. Queue response to input correlation functions: Continuous spectral analysis. *IEEE/ACM Transactions on Networking*, 1(6):678–692, Dec. 1993.

[72] S. Q. Li, S. Park, and D. Arifler. SMAQ: A measurement-based tool for traffic modeling and queueing analysis, part I - design methodologies and software architecture. *IEEE Communications Magazine*, Aug. 1998.

[73] S. Q. Li, S. Park, and D. Arifler. SMAQ: A measurement-based tool for traffic modeling and queueing analysis, part II - network applications. *IEEE Communications Magazine*, Aug. 1998.

[74] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *TCP Selective Acknowledgement Options*, Oct. 1996. RFC 2018.

[75] B. Meini. Solving M/G/1 type Markov chains: recent advances and applications. *Communications in statistics: stochastic models*, 14:479–496, 1998.

[76] H. Michiel and K. Laevens. Teletraffic engineering in a broad-band era. *Proceedings of the IEEE*, 85(12):2007–2033, Dec. 1997.

[77] H. Minc. *Nonnegative Matrices*. John Wiley & Sons, 1988.

[78] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[79] M. F. Neuts. *Probability*. Allyn and Bacon, 1973.

[80] A. S. Pandya and E. Sen. *ATM technology for broadband telecommunications networks*. CRC Press, 1999.

[81] S. K. Pappu and D. Basak. *TCP over GFR Implementation with Different Service Disciplines: A Simulation Study*, Apr. 1997. ATM Forum Contribution 97-0310.

[82] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Communications in statistics: stochastic models*, 4(1), 1988.

[83] W. Ren, K.-Y. Siu, and H. Suzuki. Excess buffer requirement for EPD schemes in ATM networks. *Computer Communications*, 22:1367–1381, 1999.

[84] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on Selected Areas in Communications*, 13(4):633–641, May 1995.

[85] V. Rosolen, O. Bonaventure, and G. Leduc. A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic. *ACM Computer Communications Review*, 29(3), July 1999.

[86] P. Salvador and R. Valadas. A fitting procedure for Markov modulated Poisson processes with an adaptive number of states. In *Proceedings IFIP ATM & IP 2001*, pages 62–73, Budapest, Hungary, 2001.

[87] T. B. Senior. *Mathematical Methods in Electrical Engineering.* Cambridge University Press, 1986.

[88] K.-Y. Siu, Y. Wu, and W. Ren. Virtual queueing techniques for UBR+ service in ATM with fair access and minimum bandwidth guarantee. In *Proceedings Globecom'97*, Phoenix, AZ, 1997.

[89] K. Spaey and C. Blondia. Circulant matching method for multiplexing ATM traffic applied to video sources. In *Proceedings IFIP PICS'98*, pages 234–245, Lund, Sweden, 1998.

[90] K. Spaey and C. Blondia. Analysis of the early packet discard with per-VC queueing mechanism. In *Proceedings IFIP ATM'99 Workshop*, Antwerp, Belgium, 1999.

[91] K. Spaey and C. Blondia. Buffer acceptance schemes for the UBR and GFR ATM service categories. In *Proceedings IFIP ATM & IP 2000*, Ilkley, UK, 2000.

[92] K. Spaey and C. Blondia. Observations of the transient performance of the selective drop buffer acceptance algorithm with responsive traffic. In *Proceedings IFIP ATM & IP 2001*, pages 89–100, Budapest, Hungary, 2001.

[93] K. Spaey and C. Blondia. Transient performance analyse of the selective drop buffer acceptance scheme with responsive traffic. In *Proceedings ICCCN 2001*, pages 361–366, Scottsdale, AZ, 2001.

[94] K. Sriram. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):833–846, Sept. 1986.

[95] M. M. Syslo, N. Deo, and J. S. Kowalik. *Discrete Optimization Algorithms with Pascal Programs.* Englewood Cliffs, 1983.

[96] J. V. Uspensky and M. A. Heaslet. *Elementary Number Theory.* McGraw-Hill Book Company, 1939.

[97] Y. Wu, K.-Y. Siu, and W. Ren. Improved virtual queueing and dynamic EPD techniques for TCP over ATM. In *Proceedings IEEE ICNP'97*, Atlanta, GA, 1997.

[98] K. Wuyts and R. K. Boel. A matrix geometric algorithm for finite buffer systems with B-ISDN applications. In *Proceedings 10th ITC Specialist Seminar*, pages 265–276, Lund, Sweden, 1996.

Nederlands overzicht

Zoals gesuggereerd wordt door de titel, bestaat deze thesis uit twee delen. Twee afzonderlijke onderwerpen die gerelateerd zijn met de prestatieanalyse van telecommunicatienetwerkelementen worden beschouwd: (i) de superpositie van Markov verkeersbronnen, en (ii) pakketbewuste bufferacceptatie.

Een fundamenteel probleem bij het dimensioneren en de prestatieanalyse van telecommunicatienetwerkelementen is het berekenen van de distributie van de wachtrijbezetting en van de wachttijd in een discrete-tijd wachtrijsysteem met als input een superpositie van processen die verkeersstromen modelleren. Een belangrijke klasse van veelgebruikte verkeersmodellen zijn de Markov verkeersbronnen, enerzijds omdat deze bronnen het grillig ('bursty') en variabel karakter van netwerkverkeer kunnen beschrijven, en anderzijds omdat ze analytisch bruikbaar zijn. Omdat de input van een netwerkelement meestal uit meerdere verkeersstromen bestaat, moet ook de superpositie van Markov verkeersstromen gekarakteriseerd kunnen worden. In theorie wordt deze superpositie exact beschreven door een nieuw Markov model. Een probleem is echter de explosie van de toestandsruimte van dit Markov model indien het aantal inputstromen realistische waarden aanneemt.

Het eerste deel van deze thesis stelt een methode voor, *circulant matching* genoemd, die een nieuwe Markov aankomststroom met een kleinere toestandsruimte construeert ter vervanging van de exacte superpositie. Twee statistische functies van het exacte inputsnelheidsproces die de prestaties van wachtrijen beïnvloeden, namelijk de autocorrelatiesequentie en de stationaire distributie, worden gematcht door dit nieuwe Markov model. De transitiematrix van de Markov keten is een circulante matrix, om het oplossen van een omgekeerdspectrumprobleem te vermijden. Deel I van de thesis bestaat uit drie hoofdstukken. Hoofdstuk 1 illustreert het probleem van de explosie van de toestandsruimte en introduceert enkele definities en resultaten. Een gedetailleerde beschrijving van de 'circulant matching' methode is te vinden in Hoofdstuk 2. Hoofdstuk 3 bespreekt numerieke voorbeelden en toepassingen van de methode, waaronder de superpositie van een model voor MPEG bronnen.

Het tweede deel van de thesis handelt over pakketbewuste bufferacceptatieschema's. Indien pakketgebaseerde data getransporteerd wordt over een ATM ('asynchronous transfer mode') netwerk, dan worden deze pakketten opgedeeld in cellen, de kleine data-eenheden met een vaste lengte waarin ATM per definitie alle data verstuurt. Een bufferacceptatieschema beslist welke cellen de buffer van een netwerkelement binnen mogen, en welke niet. Omdat het verlies van een enkele cel van een pakket al resulteert in een corrupt pakket dat sowieso weggegooid wordt aan de bestemming, verbeteren pakketbewuste bufferacceptatieschema's de efficiëntie. Niet enkel efficiëntie is belangrijk, maar ook hoe rechtvaardig de totale effectieve 'throughput' onder de verschillende verbindingen verdeeld is. Daarom werden ook schema's gedefinieerd die bij voorkeur pakketten van verbindingen die meer bandbreedte gebruiken dan wat eerlijk is, laten verloren gaan.

Deel II van de thesis bestaat uit vier hoofdstukken. Hoofdstuk 4 definieert wat exact onder de term pakket moet verstaan worden. Omdat het overgrote deel niet-tijdskritisch pakketgebaseerd dataverkeer in een netwerk TCP verkeer is, wordt ook een korte inleiding over TCP en over de twee ATM servicecategorieën die het meest geschikt zijn om TCP verkeer te transporteren, toegevoegd. Hoofdstuk 5 maakt een overzicht van de belangrijkste pakketbewuste bufferacceptatieschema's die voorgesteld worden in de literatuur voor gebruik in combinatie met deze twee servicecategorieën. In Hoofdstuk 6 wordt een theoretisch model opgesteld en toegepast om de vergankelijke ('transient') prestaties te bestuderen van één van deze schema's, namelijk 'selective drop'. Selective drop is een voorbeeld van een schema dat probeert om het verloren gaan van pakketten eerlijk te verdelen onder de verschillende verbindingen. Door het aanbrengen van een kleine wijziging aan het model uit Hoofdstuk 6, wordt in Hoofdstuk 7 de prestatie van het 'fair buffer allocation' schema, een ander schema dat streeft naar een rechtvaardige verdeling van de totale effectieve 'throughput' onder de verschillende verbindingen, bestudeerd.

Acknowledgements

The last lines of this text are some words of gratitude to all those who helped to make this thesis possible. First of all, I would like to thank my promotor, Prof. Chris Blondia, for giving me the opportunity to participate in several research projects, attend conferences and of course write this thesis.

I also want to thank

> all members and former members of our research group, and colleagues at the department, for the pleasant and motivating atmosphere,
>
> Egil, Søren, Fernando, Laurent and Martin, with whom I enjoyed several experiment weeks at the EXPERT test platform, and the many other people I have met at project meetings. Thanks also to those people who on simple request provided me promptly with some paper I was looking for, such that many time-consuming document orders could be avoided.
>
> Tim, for his advices, motivating interest and ability to relativize,
>
> my parents, brothers and friends for their support. A special thanks to my mother, for her patience and understanding, especially during the weeks of writing this text.

This work has been supported by projects sponsored by the IWT (Institute for the Promotion of Innovation by Science and Technology in Flanders) and by the European telecommunications research program ACTS.