Queues with Congestion-dependent Feedback

ISBN 90-365-2116-5

QUEUES WITH CONGESTION-DEPENDENT FEEDBACK

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. F.A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 17 december 2004 om 16.45 uur

door

Jonkheer Nicolaas Dirk van Foreest
geboren op 12 oktober 1967
te Utrecht

Dit proefschrift is goedgekeurd door de promotor en assistent promotor

prof.dr. M.R.H. Mandjes
dr.ir. W.R.W. Scheinhardt

Contents

Queues with Congestion-dependent Feedback

Chapter 1

Introduction

It is all too obvious that queueing situations abound. Consider for instance the number of customers in a barber shop to get a haircut, or the waiting time in queue to have one's passport renewed. Such examples provide ample motivation, in the author's opinion, for the existence of a mathematical theory called *queueing theory*. In general, queueing theory aims to obtain quantitative information about queue length, waiting time, the work per server, and so on, as a function of the inter-arrival times between customers, the type and amount of service required, the order in which customers receive service, the number of servers (such as hairdressers, public servants working), the number of processing steps per customer (for instance, washing hair, cutting, drying), and so on. Since most of the required information, such as the inter-arrival time between customers, can only be formulated in *probabilistic* terms, queueing theory is a part of applied probability.

Often the amount of work in queue is not a continuous process. For instance, in the example of the barber shop customers arrive individually. Consequently, the workload changes abruptly when customers decide to stay and wait, rather than leave. Hence, at the occurrence of arrival events, the amount of unfinished work changes *discontinuously*.

In other types of queue the workload changes *gradually*. An illustrative example is the amount of water in a bathtub with children playing with the tap and the plug simultaneously. As long as the children do not pour water over the edge, the water level in- and decreases gradually. Even so, pueri pueri, pueri puerilia tractant: the water level is wildly stochastic. Queueing processes of this second type are called *stochastic fluid queues*.

The study of stochastic fluid queues occupies the larger part of this thesis. As a primary application we analyze a stochastic fluid model of the Transmission Control Protocol (TCP), which is an important protocol used in the operation of the Internet.

In this preparatory chapter we assemble and discuss the material required for the sequel of this monograph. In Section 1.1 we introduce the formal concepts of stochastic

fluid queues, address basic analytic results available for such queues, and provide an overview of some of the relevant references. Then, in Section 1.2, we describe the main characteristics of TCP, and summarize in Section 1.3 the most influential TCP models. Finally, Section 1.4 gives an overview of the results obtained in this thesis.

In the sequel we assume the reader to be familiar with basic probability and queueing theory. Among others, Feller (1968), Ross (1993, 1996), Shiryaev (1996), or Grimmett & Stirzaker (2001), provide the required background on probability theory. Kleinrock (1975, 1976), and Harrison & Patel (1993), treat the queueing theory we need. Asmussen (2003) covers both subjects at a somewhat higher level of abstraction. Below we use, supposedly, well-known results from these sources without explicit reference.

1.1 Stochastic Fluid Queues

For the sake of exposition we consider the content of a water reservoir behind a dam as a second example of a stochastic fluid queue—less entertaining perhaps than children in a bath, but economically more interesting, and most probably easier to characterize. The water level in the reservoir changes dynamically as a function of weather conditions, the release of water to generate electricity, and so on. To guarantee (within reasonable limits) a minimal supply of electricity, it is necessary to have a release strategy, which, in turn, depends on a model that relates the dynamics of the content to the in- and output process of water. In this section we present one such model and call this the *standard fluid model* or *standard fluid queue*. We concentrate in Sections 1.1.1–1.1.3 on the analysis of the standard fluid queue with *unlimited* buffer capacity because of its relative simplicity. Nevertheless, the assumption of unlimited buffer capacity is often quite unrealistic. For instance, a water reservoir is occasionally full. Moreover, all our fluid models of TCP have finite buffers. Therefore we present in Section 1.1.4 the main results for the standard fluid model with *limited* buffer capacity.

The survey paper of Kulkarni (1997) is our prime reference for this section.

1.1.1 The Standard Fluid Model

We start by modeling the process that determines the rate at which the content changes. The state of this *background* or *modulating* process represents momentary weather conditions, maintenance, and so on. As, clearly, the state of the background process changes randomly, we describe it as a stochastic process $\{W(t)\} \equiv \{W(t), t \geq 0\}$ with state space $\mathscr{W}$. For the sake of tractability we suppose $\mathscr{W} = \{1, \ldots, N\}$, for some finite N.

Next, consider the transitions among the states of $\{W(t)\}$ and the time it stays in a certain state. In the sequel we model $\{W(t)\}$ as a continuous-time Markov chain with

generator matrix Q. Hence, for all $t \geq 0$ and h small,

$$\begin{aligned} \mathbb{P}\{W(t+h) = j \,|\, W(t) = i\} &= Q_{ij}h + o(h), \\ \mathbb{P}\{W(t+h) = i \,|\, W(t) = i\} &= 1 + Q_{ii}h + o(h), \end{aligned} \tag{1.1}$$

where $Q_{ij} \geq 0$ if $i \neq j$ and

$$Q_{ii} := -\sum_{j \in \mathscr{W} \setminus \{i\}} Q_{ij} < \infty.$$

Let us define *time-dependent state probabilities* $\pi_i(t) = \mathbb{P}\{W(t) = i \,|\, W(0)\}$ for $\{W(t)\}$. The evolution of $\boldsymbol{\pi}(t) = (\pi_1(t), \ldots, \pi_N(t))$ is given by the system of ordinary differential equations

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)Q. \tag{1.2}$$

Now we model the content process of the reservoir itself. Clearly, the *net input rate* or *drift function*, which is the difference between the input rate and the output rate, determines the rate of change of the content. (We prefer to use the term 'drift' over 'rate' to avoid confusion with the term 'transition rate' of the background process.) The drift is a function $r : \mathscr{W} \to \mathbb{R}$,

$$r : i \mapsto r_i := r(i). \tag{1.3}$$

The content process is also stochastic, as it depends on the stochastic process $\{W(t)\}$. We denote the content process by $\{C(t)\} \equiv \{C(t), t \geq 0\}$. It follows from (1.3) that $\{C(t)\}$ satisfies the differential equation

$$\frac{dC(t)}{dt} = \begin{cases} \max\{r_i, 0\}, & \text{if } C(t) = 0, \\ r_i, & \text{if } C(t) > 0, \end{cases} \tag{1.4}$$

when $W(t) = i$. Here, dC/dt denotes the right-hand derivative.

Clearly, (1.2) and (1.4) provide insight in the infinitesimal behavior of $\{W(t)\}$ and $\{C(t)\}$ separately. To obtain information on the transient behavior of $\{C(t)\}$ we need to study the transient distributions of the *joint process*

$$\{W(t), C(t)\} \equiv \{W(t), C(t), t \geq 0\}.$$

The study of this bivariate Markov process is the subject of the next two subsections.

1.1.2 Kolmogorov Forward Equations

Here we focus on the transient analysis of $\{W(t), C(t)\}$.

On the state space $\mathscr{S} = \mathscr{W} \times [0, \infty)$ of $\{W(t), C(t)\}$ we define functions

$$F_i(y,t) = \mathbb{P}\left\{W(t) = i, C(t) \leq y \,|\, W(0), C(0)\right\}, \quad \text{for } (i,y) \in \mathscr{S},\ t \geq 0. \tag{1.5}$$

To keep the notation concise, we suppress the dependence of $F_i(y,t)$ on the initial conditions $W(0)$ and $C(0)$. The functions F_i relate to the distributions of the separate processes $\{W(t)\}$ and $\{C(t)\}$ in a simple manner:

$$\pi_i(t) = \mathbb{P}\{W(t) = i\} = F_i(\infty, t); \qquad \mathbb{P}\{C(t) \leq y\} = \sum_{i \in \mathscr{W}} F_i(y,t).$$

To derive the Kolmogorov forward equation for $\{W(t), C(t)\}$, we express the functions $F_i(y, t+h)$ in terms of $F_j(y,t)$ for $j \in \mathscr{W}$. Specifically, using (1.1) and (1.4), it follows that when $y > 0$ and $h > 0$ small enough that also $y - r_i h > 0$ for all $i \in \mathscr{W}$,

$$F_i(y, t+h) = (1 + Q_{ii}h)F_i(y - r_i h, t) + h\sum_{j \neq i} Q_{ji}F_j(y,t) + o(h). \tag{1.6}$$

Without loss of generality—for details, consult Kella & Stadje (2002)—we can assume that $\partial_y F_i$ exists. Therefore the above becomes, after some rearrangements,

$$\frac{F_i(y,t+h) - F_i(y,t)}{h} = Q_{ii}F_i(y,t) - \frac{\partial F_i(y,t)}{\partial y} r_i + \sum_{j \neq i} Q_{ji}F_j(y,t) + \frac{o(h)}{h}. \tag{1.7}$$

If we further assume that $\partial_t F_i$ exists, the limit $h \to 0$ yields

$$\frac{\partial F_i(y,t)}{\partial t} = \sum_{j \in \mathscr{W}} Q_{ji}F_j(y,t) - \frac{\partial F_i(y,t)}{\partial y} r_i. \tag{1.8}$$

In matrix form this becomes:

$$\frac{\partial \mathbf{F}(y,t)}{\partial t} = \mathbf{F}(y,t)Q - \frac{\partial \mathbf{F}(y,t)}{\partial y} R, \quad y > 0, \tag{1.9}$$

where $\mathbf{F}(y,t) = (F_1(y,t), \ldots, F_N(y,t))$, and R is the N-dimensional *drift matrix* with the drifts r_i at its diagonal, i.e.,

$$R = \text{diag}(r_1, \ldots, r_N). \tag{1.10}$$

The last step of the derivation concerns the behavior of $\{W(t), C(t)\}$ at the boundary $y = 0$. To this end we define two subsets of $\mathscr{W}$ and their respective cardinalities:

$$\begin{aligned} \mathscr{W}_- &= \{i \in \mathscr{W} \mid r_i < 0\}, \quad N_- = |\mathscr{W}_-|, \\ \mathscr{W}_+ &= \{i \in \mathscr{W} \mid r_i > 0\}, \quad N_+ = |\mathscr{W}_+|. \end{aligned}$$

We assume for ease that $\mathscr{W}_- \cup \mathscr{W}_+ = \mathscr{W}$, which in turn implies that $\mathscr{W}_- \cap \mathscr{W}_+ = \varnothing$. Including the case that $r_i = 0$ for some i is a technical, but not particularly difficult, point, cf. Mitra (1988). Thus, we may focus on each of the disjoint sets $\mathscr{W}_+$ and $\mathscr{W}_-$ successively.

It follows from (1.4) that $C(t)$ only spends an infinitesimally small amount of time at $y = 0$ whenever $W(t) \in \mathscr{W}_+$, hence,

$$F_i(0,t) \equiv 0, \quad \text{if } i \in \mathscr{W}_+,\ t > 0. \tag{1.11}$$

However, for $i \in \mathscr{W}_+$ the right-hand partial derivative

$$\partial_y F_i(0+,t) := \lim_{h \downarrow 0} \frac{F_i(h,t)}{h}$$

is not necessarily identically 0. To see this we reason as follows. As no probability mass can accumulate at $y = 0$, the probability flux out of $(i,0)$ should equal the flux into it. In other words,

$$(1 - Q_{ii}h)F_i(r_i h, t+h) = h \sum_j Q_{ji} F_j(0,t) + o(h)$$

where, because of (1.11), it is not necessary to exclude $j = i$ in the summation. This becomes, again using (1.11),

$$\frac{\partial F_i(0+,t)}{\partial y} r_i h = h \sum_j Q_{ji} F_j(0,t) + o(h).$$

Thus, for $i \in \mathscr{W}_+$ we obtain in the limit $h \downarrow 0$

$$0 = -\frac{\partial F_i(0+,t)}{\partial y} r_i + \sum_j Q_{ji} F_j(0,t).$$

When $i \in \mathscr{W}_-$ we get for $y = 0$ and h sufficiently small (recall $r_i < 0$),

$$F_i(0,t+h) = (1 + Q_{ii}h)F_i(-r_i h, t) + h \sum_{j \neq i} Q_{ji} F_j(0,t) + o(h).$$

(Now the case $j = i$ should be excluded.) After rearranging, taking limits, and combining with the results obtained on $\mathscr{W}_+$ we find for $y = 0$:

$$\frac{\partial \mathbf{F}(0,t)}{\partial t} = \mathbf{F}(0,t)Q - \frac{\partial \mathbf{F}(0+,t)}{\partial y} R. \tag{1.12}$$

The next theorem summarizes the results obtained up to now.

Theorem 1.1. *The functions $F_i(y,t)$ defined by (1.5) satisfy the partial differential equation (1.9) when $y > 0$, and (1.11–1.12) at $y = 0$.*

Remark 1.2. We can now see that although $\{W(t)\}$ is a Markov process, it is *not* probabilistically independent of $\{C(t)\}$, for, with (1.11),

$$0 = \mathbb{P}\{C(t) = 0, W(t) \in \mathscr{W}_+\} \neq \mathbb{P}\{C(t) = 0\}\mathbb{P}\{W(t) \in \mathscr{W}_+\},$$

as for all sufficiently large $t > 0$, $\mathbb{P}\{C(t) = 0\} > 0$ and $\mathbb{P}\{W(t) \in \mathscr{W}_+\} > 0$.

Remark 1.3. A point of theoretical interest is whether a well-defined set of functions F_i exists that satisfy the conditions of Theorem 1.1. (The derivation above takes the existence for granted.) To resolve this issue, we construct $\{W(t), C(t)\}$ with (1.1) and (1.4) as a piecewise-deterministic Markov process (PDP) in the sense of Davis (1984, 1993). As a consequence the probability law of the process, and hence $F_i(y)$, is well-defined. Therefore, we can actually use the properties of the process itself to derive the Kolmogorov forward equation, which we indeed do in (1.6).

As Kulkarni (1997) remarks, solving the fluid system of Theorem 1.1 is difficult. The study of the distribution function in steady-state is considerably easier.

1.1.3 The Stationary System

From now on we concentrate on the steady-state limit of the distribution of $\{W(t), C(t)\}$.

First, let us assume that the generator Q of the background process is irreducible. As $N < \infty$, $\boldsymbol{\pi}(t) \to \boldsymbol{\pi}$ for $t \to \infty$, independent of initial conditions, and $\pi_i > 0$ for all $i \in \mathscr{W}$. Moreover, this vector satisfies $\boldsymbol{\pi}Q = \mathbf{0}$. It is well-known that, as a consequence, the Markov chain $\{W(t)\}$ is ergodic. Concerning the fluid model we assume furthermore that $\{W(t), C(t)\}$ is stable in the following sense:

$$\boldsymbol{\pi}R\mathbf{1}' \equiv \sum_{i=1}^{N} \pi_i r_i < 0, \tag{1.13}$$

where $\mathbf{1} = (1, \ldots, 1)$ and $\mathbf{v}'$ denotes the transpose of the vector $\mathbf{v}$.

Kulkarni (1997) proves that for a stable fluid queue driven by an ergodic background process a steady-state limit $F_i(y)$ of the functions $F_i(y, t)$ always exists, i.e.,

$$F_i(y) := \lim_{t \to \infty} F_i(y, t).$$

We consider the system in steady state, and write W and C for $W(t)$ and $C(t)$, respectively, at an arbitrary moment in time. Thus,

$$\mathbb{P}\{W = i, C \leq y\} = F_i(y). \tag{1.14}$$

With the random variable W we may interpret condition (1.13) as $\mathbb{E}\{r(W)\} < 0$, i.e., the expected drift of the fluid queue is negative.

Clearly, in steady state $F_i(y, t) \equiv F_i(y)$, so that as consequence $\partial_t F_i(y, t) \equiv 0$. Thus, (1.9) reduces to

$$\frac{d\mathbf{F}(y)}{dy} R = \mathbf{F}(y)Q, \quad y > 0. \tag{1.15a}$$

We can generalize this to hold for all $y \geq 0$, rather than just for $y > 0$, by using the convention that $d\mathbf{F}(0)/dy = d\mathbf{F}(0+)/dy$, thereby, in passing, also covering (1.12). Equations (1.11) reduce to the boundary conditions

$$F_i(0) \equiv 0, \quad \text{if } i \in \mathscr{W}_+. \tag{1.15b}$$

Solving (1.15a) is, in a sense, a straightforward exercise in the theory of ordinary differential equations. Since we excluded drift functions such that $r_i = 0$ for some i, the drift matrix R is invertible. Hence, formally, the solution of (1.15a) is

$$\mathbf{F}(y) = \mathbf{a}\, e^{QR^{-1}y}, \tag{1.16}$$

for some vector $\mathbf{a} = (a_1, \ldots, a_n)$. The boundary conditions (1.15b) determine N_+ components of the coefficients vector $\mathbf{a}$. We derive the rest of the conditions presently.

For a considerable number of models that appeared in the literature, the matrix QR^{-1} is simple, i.e., a nonsingular linear transformation T exists such that $T^{-1}QR^{-1}T$ is diagonal, in which case the solution for $\mathbf{F}$ can be written in the form, see, e.g., Lancaster & Tismenetsky (1985: Section 9.10),

$$\mathbf{F}(y) = \sum_{i=1}^{N} a_i e^{\theta_i y} \mathbf{v}_i, \tag{1.17}$$

where $(\theta_i, \mathbf{v}_i)$ is a (left) eigenpair of the equation

$$\theta_i \mathbf{v}_i R = \mathbf{v}_i Q. \tag{1.18}$$

Equation (1.17) is known as the *spectral representation* of $\mathbf{F}$.

When the matrix QR^{-1} cannot be diagonalized, the algebraic multiplicity of at least one of the eigenvalues is larger than one. In that case the solution (1.16) is not of the form (1.17), but instead,

$$\mathbf{F}(y) = \sum_{i=1}^{N} a_i p_i(y) e^{\theta_i y} \mathbf{v}_i, \tag{1.19}$$

where $p_i(y)$ is some polynomial in y (with degree strictly smaller than the algebraic multiplicity of θ_i) and the vectors $\mathbf{v}_i$, $1 \leq i \leq N$ form a set of independent (generalized) left eigenvectors of QR^{-1}, cf. Lancaster & Tismenetsky (1985: Section 9.10).

Concerning the structure of the spectrum of the eigenvalue problem (1.18), Kulkarni states, for irreducible Q and R possibly including zero drifts, that the eigenvalues satisfy the following properties.

Theorem 1.4. *When (1.13) is true, i.e., $\boldsymbol{\pi} R \mathbf{1}' < 0$, the eigenvalues of (1.18) can be ordered as*

$$\Re(\theta_1) \leq \ldots \leq \Re\left(\theta_{N_+}\right) < \Re\left(\theta_{N_+ + 1}\right) = 0 < \Re\left(\theta_{N_+ + 2}\right) \leq \ldots \leq \Re(\theta_N), \tag{1.20}$$

where $\Re(\theta)$ denotes the real part of θ. In particular, the eigenvalue $\theta_{N_+ + 1}$ is simple. It is immediate from (1.18) and the singularity of Q that $\theta_{N_+ + 1} = 0$, and $\mathbf{v}_{N_+ + 1} = \boldsymbol{\pi}$.

From the expansion shown in (1.19) it is easily seen that the contributions to $F_i(y)$ of the eigenvectors $\mathbf{v}_{N_+ +2}, \ldots, \mathbf{v}_N$ grow beyond bound when $y \to \infty$. Since the functions F_i are bounded between 0 and 1 for all y, we should therefore set $a_j = 0$ whenever $\Re(\theta_j) > 0$. Furthermore, it can be proved that the (algebraic) multiplicity of the eigenvector $\mathbf{v}_{N_+ +1} = \boldsymbol{\pi}$ is one. As a result, the decomposition (1.19) reduces to

$$\mathbf{F}(y) = \sum_{i \le N_+} a_i p_i(y) e^{\theta_i y} v_i + a_{N_+ +1} \boldsymbol{\pi}.$$

With the N_+ boundary conditions (1.15b) we are one short of the required number to specify $\mathbf{a}$, and thereby the solution, uniquely. This last condition follows from considering the limit $y \to \infty$ of $F_i(y)$. Since $\Re(\theta_i) < 0$ for $i \le N_+$, $\lim_{y\to\infty} F_i(y) = a_{N_+ +1}\pi_i$, which implies $a_{N_+ +1} = 1$. This completes the number of conditions.

Theorem 1.5. *The functions $F_i(y)$ in (1.14) satisfy the system of ordinary differential equations (1.15a) with:*

1. *boundary conditions (1.15b);*
2. *$a_{N_+ +1} = 1$, $a_{N_+ +2} = \ldots = a_N = 0$.*

1.1.4 Finite Buffer Sizes

We can also carry out the above analysis for a fluid queue in which $\{C(t)\}$ is bounded by some finite $B > 0$. Here we present the main consequences of including this constraint.

First, as the content cannot increase beyond B, (1.4) changes to

$$\frac{dC(t)}{dt} = \begin{cases} \max\{r_i, 0\}, & \text{if } C(t) = 0, \\ r_i, & \text{if } C(t) \in (0, B), \\ \min\{r_i, 0\}, & \text{if } C(t) = B, \end{cases} \tag{1.21}$$

when $W(t) = i$.

An immediate implication of this differential equation is that, besides $F_i(0, t) = 0$ for $i \in \mathscr{W}_+$, also

$$\mathbb{P}\{W(t) \in \mathscr{W}_-, C(t) = B\} = 0.$$

Thus,

$$F_i(B-, t) \equiv \pi_i(t), \quad \text{if } i \in \mathscr{W}_-,\ t > 0. \tag{1.22}$$

Second, the atoms at $y = B$ satisfy similar dynamic behavior as those at $y = 0$. Subtracting

$$\frac{\partial \mathbf{F}(B-, t)}{\partial t} = \mathbf{F}(B-, t)Q - \frac{\partial \mathbf{F}(B-, t)}{\partial y} R$$

from

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t)Q$$

yields the desired forward equation at $y = B$.

To find the steady-state limit of $F_i(y,t)$ we note that nearly all of the analysis of the unlimited buffer case carries over, except that extra boundary conditions at $y = B$ are involved. Now the number of boundary conditions mentioned in (1.15b) and (1.22) add up to N, thus equaling to the number of components (the unknowns) in the coefficients vector $\mathbf{a}$ of the solution. Hence,

Theorem 1.6. *For finite B, the steady-state functions $F_i(y)$ satisfy*

$$\frac{d\mathbf{F}(y)}{dy}R = \mathbf{F}(y)Q, \tag{1.23a}$$

with boundary conditions

$$F_i(0) \equiv 0, \quad \textit{if } i \in \mathscr{W}_+, \qquad\qquad F_i(B-) = \pi_i, \quad \textit{if } i \in \mathscr{W}_-. \tag{1.23b}$$

This problem is not an initial value problem such as the problem specified in Theorem 1.5. Rather, it is a *two-point boundary value problem* with conditions at $y = 0$ and $y = B$. Thus, the fluid model with constant Q, R, and finite buffer is considerably more difficult to solve than the infinite-buffer model.

The stability condition (1.13) is not required when $B < \infty$. Nevertheless, the sign of this condition influences the number of eigenvalues with positive (and, hence, negative) real part. Only when $\boldsymbol{\pi}R\mathbf{1}' < 0$ the decomposition of the spectrum is as in (1.20). Rather than restating Theorem 1.4 to cover also the case $\boldsymbol{\pi}R\mathbf{1}' \geq 0$, we simply assume in the sequel that $\boldsymbol{\pi}R\mathbf{1}' < 0$. (The other cases are not more difficult, cf. Kulkarni's paper.)

Clearly, the expansions of $\mathbf{F}(y)$ as in (1.17) or (1.19) involve all eigenvectors. It can be proved that all eigenvectors, except $\boldsymbol{\pi}$ and the eigenvector associated to the eigenvalue θ_{N_+}, have positive and negative components. Thus, the components of a vector i associated to a positive eigenvalue θ_i will become large in absolute value when multiplied by $\exp(\theta_i B)$. Hence, matching the boundary conditions at $y = B$, i.e., (1.23b), which is in essence a linear algebra problem of the type $A\,\mathbf{x} = \mathbf{0}$, involves adding large positive and negative numbers. This procedure is well-known to be sensitive to round-off errors, which complicates the numerical evaluation of the system considerably.

1.1.5 State-dependent Drift and Feedback

In the example of the reservoir behind a dam, the type of failure and maintenance typically depend on the momentary content of the reservoir. More accurate models of the queueing process should therefore allow the drift function and the transition rates to become functions of $\{C(t)\}$. As the 'standard' fluid model of Section 1.1.1 cannot capture

such intricate interaction—R and Q are constant, rather than functions of $\{C(t)\}$—it is necessary to extend this fluid model.

Here we summarize two interesting extensions of the standard fluid model capable of including the required dependency. A first extension allows only the drift to depend on $\{C(t)\}$. In a second extension the generator of the background process itself becomes also a function of $\{C(t)\}$. We refer to these extensions as *state-dependent drift* and *feedback*, respectively. To avoid any confusion that may arise with respect to the term 'feedback', we note that *in the context of fluid queues* feedback denotes the transfer of information from the buffer about its content to the modulating process. Thus, fluid itself is *not* fed back to the buffer for a second round of service, say. We remark that feedback fluid queues prove their usefulness in Chapters 2 and 3 as models of the interaction between one or two TCP sources (i.e., traffic sources that use the Transmission Control Protocol, cf. Section 1.2) and a bottleneck buffer in the Internet.

State-dependent Drift In a sense the drift in the standard fluid model is already state dependent. To see this, observe from (1.4) that most of the time during which $C(t) = 0$, its derivative $dC(t)/dt = 0$ rather than $r(W(t))$. Therefore we may as well define the left-continuous drift function on $[0, \infty)$, or $[0, B]$ when $B < \infty$,

$$y \mapsto r(i, y) := r_i(y) = \begin{cases} 0, & \text{if } i \in \mathscr{W}_-,\ y = 0, \\ r_i, & \text{elsewhere.} \end{cases} \tag{1.24}$$

Note that the *function* $r_i(y)$ is not identical to the constant r_i as used in Sections 1.1.1–1.1.4. With this drift function we can replace (1.4) by

$$\frac{dC(t)}{dt} = r_i(C(t)), \quad \text{if } W(t) = i. \tag{1.25}$$

When $B < \infty$ the drift function becomes

$$r(i, y) := r_i(y) = \begin{cases} 0, & \text{if } i \in \mathscr{W}_-, y = 0, \\ 0, & \text{if } i \in \mathscr{W}_+, y = B, \\ r_i, & \text{elsewhere.} \end{cases}$$

Thus, in this case, r_i is left- or right-continuous when $i \in \mathscr{W}_-$ or $i \in \mathscr{W}_+$, respectively.

The above changes do not seem to make much of a difference: the differential equation for $C(t)$ becomes simpler indeed, but this comes at the expense of a more complicated definition for the drift function. Conceptually, though, we feel this makes considerable difference. The differential equation (1.25) for $C(t)$ is more natural than (1.4), and therefore invites to investigate the behavior of fluid models with more complicated drift functions than (1.24). This we do now.

Suppose, as a *first extension*, that we distinguish $K+1$ *buffer thresholds*

$$0 = B^{(0)} < B^{(1)} < \ldots < B^{(K-1)} < B^{(K)} = B \leq \infty, \tag{1.26}$$

and, as a consequence, K *buffer regimes*, i.e., intervals $(B^{(k-1)}, B^{(k)})$, $1 \leq k \leq K$. On each regime we suppose that the drift matrix $R^{(k)} = \text{diag}(r_1^{(k)}, \ldots, r_N^{(k)})$, where the drift function $r_i(y)$ satisfies:

$$r_i(y) = r_i^{(k)}, \quad \text{if } y \in (B^{(k-1)}, B^{(k)}), \tag{1.27}$$

and $r_i^{(k)} \neq 0$ for all i, k. In other words, the state-dependent drift function is *piecewise constant*.

Assuming that $R^{(k)}$ is invertible on each regime k, we can interpret the fluid model at regime k as a model with finite buffer. Then the steady-state solution has a form similar to (1.16). Finally we should 'glue together' the solutions of each separate regime such that the 'threshold conditions', i.e., the boundary conditions of each regime, are satisfied.

In contrast to the standard case, identifying threshold conditions similar to (1.23b) for the steady-state solution is more elaborate. Now we have to distinguish four scenarios for the signs of the drifts at two adjacent regimes $k-1$ and k, say. These scenarios are shown in Figure 1.1. As is apparent from the figure, $\mathscr{W}$ splits into:

a. ascending: $\mathscr{W}_a^{(k)} = \{i \in \mathscr{W} \mid r_i^{(k-1)}, r_i^{(k)} > 0\}$,

b. bifurcating: $\mathscr{W}_b^{(k)} = \{i \in \mathscr{W} \mid r_i^{(k-1)} < 0 < r_i^{(k)}\}$,

c. confluence: $\mathscr{W}_c^{(k)} = \{i \in \mathscr{W} \mid r_i^{(k-1)} > 0 > r_i^{(k)}\}$,

d. descending: $\mathscr{W}_d^{(k)} = \{i \in \mathscr{W} \mid r_i^{(k-1)}, r_i^{(k)} < 0\}$.

In cases 'a' and 'd' no complications arise: the content just drifts at another rate upwards or downwards when it passes the threshold $B^{(k)}$. Therefore, no atom is present at $B^{(k)}$, as indicated by the open circles in the figure. Case 'c' implies that an atom, shown by the bullet, exists at $B^{(k)}$, comparable to the atoms at 0 when $i \in \mathscr{W}_-$ for the standard model.

Case 'b' is, perhaps, the most interesting, as it is not clear what the fluid process should do when it enters $(i, B^{(k)})$. We see at least three ways to deal with this ambiguity. First, it is possible to simply exclude such cases in the model, i.e., to assume that $\mathscr{W}_b^{(k)} = \varnothing$ for all k. Second, the problem does not occur when $Q_{ji} = 0$ for all $j \in \bigcup_{k=1}^K \mathscr{W}_c^{(k)}$ and $i \in \bigcup_{k=1}^K \mathscr{W}_b^{(k)}$. The third method, which we conjecture to be the resolution of the ambiguity, is to choose the drift function as a left or right continuous function for $i \in \mathscr{W}_b^{(k)}$.

To comment on this third option, observe that, formally, the drift function is only specified by (1.27) *within* the regimes, but not *on* the thresholds. To complete the model

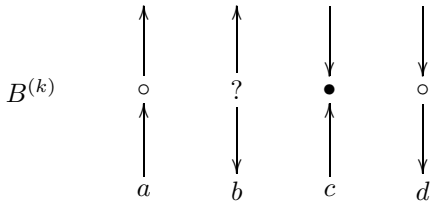

Figure 1.1: *Here we show the different configurations of the rates below and above the threshold at $B^{(k)}$. The direction of the arrows indicate the direction at which the fluid flows immediately below and above $B^{(k)}$, that is, the sign of $r_i^{(k-1)}$ and $r_i^{(k-1)}$. A bullet (open circle) denotes that (no) probability mass is present in steady state in the state $\big(i, B^{(k)}\big)$. The question mark means that it is not clear what will happen with the fluid process when it enters this state.*

it is necessary to define the drift also on the thresholds $B^{(k)}$. A straightforward solution is to define the drift function everywhere on $\mathscr{S}$ as either a left or right continuous function. The actual choice will supposedly not have any consequence for the cases 'a', 'c' and 'd'. (There are some details to resolve at $y = 0$ and $y = B$ if $B < \infty$. These are easy to provide, cf. (1.24).) However, when the drifts 'bifurcate', the choice determines which way the content drifts after a transition into the bifurcating state. Clearly, these observations lead us to conjecture that a drift function that is either left or right continuous is sufficient to tackle the problematic 'b' states and, in fact, necessary to complete the model.

A *second extension* of the standard fluid queue allows the drift function to be *piecewise continuous*, rather than piecewise constant. The existing literature requires that the sign of $r_i(\cdot)$ does not change on the interval $[0, \infty)$ (or $[0, B]$ when $B < \infty$) as this appears difficult. Consequently, the set $\mathscr{W}$ again splits into two disjoint proper subsets $\mathscr{W}_+$ and $\mathscr{W}_-$, respectively.

When the drift function (1.24) is piecewise continuous and has no sign changes, it is possible to show that (1.9) becomes

$$\frac{\partial \mathbf{F}(y,t)}{\partial t} = \mathbf{F}(y,t)Q - \frac{\partial \mathbf{F}(y,t)}{\partial y} R(y), \tag{1.28}$$

where the drift matrix is similar to (1.10) except that here the drift is state dependent, that is, $R(y) = \operatorname{diag}(r_1(y), \ldots, r_N(y))$. When a stationary distribution exists, the above system of partial differential equations reduces to the analog of (1.15a) with $R(y)$ replacing

R:

$$\mathbf{F}(y)Q = \frac{d\mathbf{F}(y)}{dy}R(y). \tag{1.29}$$

Whereas the boundary conditions (1.15b) carry over unchanged, it appears difficult to obtain comparable conditions on the coefficients vector $\mathbf{a}$ as used in Theorem 1.5, mainly because the solution of (1.29) does not have the form (1.16). Therefore, we require besides (1.15b) that

$$\lim_{y\to\infty} F_i(y) = \pi_i, \quad \text{for } i \in \mathscr{W}_+. \tag{1.30}$$

Feedback Now we consider *feedback fluid queues*, i.e., queues for which the buffer content actually changes the generator matrix Q. In a feedback fluid queue the buffer content behaves as a modulated fluid queue as before, but an infinitesimal generator $Q(y)$ whose entries *depend on the current buffer content* y governs the state of the background process. Formally, this means that

1. $\mathbb{P}\{W(t+h) = i \,|\, W(t) = i, C(t) = y\} = 1 + Q_{ii}(y)h + o(h)$,
2. $\mathbb{P}\{W(t+h) = j \,|\, W(t) = i, C(t) = y\} = Q_{ij}(y)h + o(h)$,
3. $\mathbb{P}\{W \text{ makes more than one transition in } [t, t+h] \,|\, W(t) = j, C(t) = y\} = o(h)$.

Here the function $Q_{ij}(y), j \neq i$, is said to be the transition rate at which the source process jumps from state i to j when $C(t) = y$, and $Q_{ii}(y) := -\sum_{j\neq i} Q_{ij}(y)$. In this sense, the buffer provides *feedback* to the source about the content level so that the source may adapt both the drift as well as the transition rates. As an immediate consequence of introducing feedback, the background process $\{W(t)\}$ is *not* a Markov process any longer. To see this, observe first that the knowledge of $C(t)$ is required to evaluate the generator $Q(C(t))$, which, in turn, dictates $\{W(s), s > t\}$. Second, as $C(t)$ depends on $\{W(s), s \leq t\}$, (rather than on momentary values of $\{W(t)\}$), through the integration of the differential equation (1.4), $\{W(s), s > t\}$ depends on $\{W(s), s \leq t\}$ through $C(t)$, and not just on its present state. However, $\{W(t), C(t)\}$ is still a Markov process as we can construct it as a piecewise deterministic Markov process, cf. Chapter 4.

Let us see how the two previous extensions for the fluid queue with state-dependent drift work out here. First we consider the case in which both the drift and the generator are *piecewise-constant* matrices. Then we concentrate on *piecewise-continuous* drift and generator matrices.

As in (1.26) we distinguish $K+1$ buffer thresholds, and K regimes $(B^{(k-1)}, B^{(k)})$. Within the regimes the drift matrix $R^{(k)}$ determines the net input rate and the generator $Q^{(k)}$ governs the source process. At threshold k the source behaves as a Markov chain with generator $\tilde{Q}^{(k)}$. For the joint process $\{W(t), C(t)\}$ it is again possible to derive differential equations and boundary conditions for the transient and stationary distribution

functions. In particular, the differential equations for the steady-state, cf. (1.15a), become slightly (but not fundamentally) more difficult in that they involve inhomogeneous, but constant, terms. Thus, we can still find the solution by standard methods of the theory of ordinary differential equations. Obtaining the necessary conditions is possible, but a bit troublesome, cf. Mandjes *et al.* (2003a). In Chapters 2 and 3 we show how to do this for one threshold at B, i.e., the case $K = 1$.

Finally we consider the extension in which (the entries of) Q and R are (left or right) continuous functions of the content level. A problem with deriving the forward equations is that we cannot simply replace Q by $Q(y)$ in (1.28), as we did before when migrating from (1.9) to (1.28). In Chapter 4 we show that the (differential) operators (in this form) apply to the *density* $\mathbf{f}(y,t) = \partial_y \mathbf{F}(y,t)$, rather than the distribution. This gives, instead of (1.28):

$$\frac{\partial \mathbf{f}(y,t)}{\partial t} = \mathbf{f}(y,t)Q(y) - \frac{\partial}{\partial y}\left(\mathbf{f}(y,t)R(y)\right). \tag{1.31}$$

It is necessary to impose regularity conditions on $R(y)$ and $Q(y)$, cf. Chapter 4 and Boxma *et al.* (2005).

The above reduces to a system of ordinary differential equations for the steady-state distribution function by setting $\partial_y f_i(y,t) \equiv 0$. This results in the system

$$\mathbf{f}(y)Q(y) = \frac{d}{dy}\left(\mathbf{f}(y)R(y)\right).$$

Unfortunately, general closed-form solutions are not available for $N > 2$. Moreover, when the buffer size is unlimited and $N > 2$, no simple conditions exist to guarantee the stability of the system. For instance, in Section 4.7.2 we show that the intuitive condition $\boldsymbol{\pi}(y)R(y)\mathbf{1} < 0$, for all y, where $\boldsymbol{\pi}(y)$ solves $\boldsymbol{\pi}(y)Q(y) = 0$, does not suffice. However, when B is finite, a stability condition is not needed. Hence, for finite B the analysis is somewhat simpler.

1.1.6 Literature

We now summarize some of the more influential papers that appeared on stochastic fluid queues and their applications. As the literature on these topics, and their ramifications, is vast, we make no pretense about the completeness of the discussion below. We also do not adhere to a strict ordering of the articles as they appeared over time.

For general introductions to Markov-modulated fluid queues we refer the reader to Schwartz (1996) and Kulkarni (1997). Schwartz provides also a thorough overview of the applications of fluid queues to the field of telecommunication such as the Internet and Asynchronous Transfer Mode; De Prycker (1995) explains the latter technology in detail. Roberts *et al.* (1996) also treat some of this material, but on a higher level of abstraction and with an emphasis on performance evaluation.

Stationary state Anick *et al.* (1982) consider as a background process the superposition of J identical, independent on/off sources that feed into a buffer with unlimited size. The on (off) times are independent and identically distributed random variables with exponential distribution with rate λ (μ). The on and off times are also assumed to be mutually independent. This model, commonly abbreviated as the 'AMS' model, has become one of the standard models in the teletraffic literature. The model discussed by Kosten (1974) can be seen as a limiting case of the AMS model in which the number of sources J and the average off period $1/\lambda$ increase to infinity but such that $J\lambda$ converges to a constant $\lambda^\star$, say. In this limit, fluid sources arrive according to a Poisson process with rate $\lambda^\star$. (If we interpret the AMS model as the fluid counterpart of the Engset model, cf. Cooper & Heyman (1998), then Kosten's model is the counterpart of the $M/M/\infty$ queue.) Kosten (1984) extends the AMS model to a system with a superposition of independent heterogeneous on/off sources.

The superposition of on/off sources is actually an example of a birth-death modulating process. Van Doorn *et al.* (1988) study more general birth-death background processes with finite state space. As a direct consequence of the specific structure of birth-death processes, Van Doorn *et al.* (1988) can express the components of the eigenvectors as recurrence relations. Mitra (1988) allows the background process to be a reversible Markov chain, rather than strictly birth-death. Besides this, in his model the number of active sources may vary *as well as* the number of active servers that serve the buffer. Stern & Elwalid (1991) focus on a superposition of independent heterogeneous reversible background processes.

Sericola & Tuffin (1999) concentrate on a model in which the number of background states is possibly countably infinite, while the generator is uniform (i.e., all diagonal entries are bounded), and the buffer is infinite. An important requirement for their (numerically stable) method is that just one state of the background process has negative drift. The work of Virtamo & Norros (1994) is a special case of this model. Here the idle and busy periods of an M/M/1 queue modulate the drift function: while busy, fluid enters the fluid buffer at constant rate, while when idle, the fluid buffer depletes at constant rate. It also relates to the model of Van Doorn *et al.* (1988) as the modulating process is birth-death, but its state space is uncountable. Akar & Sohraby (2003) present a numerically efficient and stable algorithm to solve the two-point boundary value problem (1.23). We can also interpret the work of Cohen (1974) as an extension of the AMS model in that now the on periods may have arbitrary distribution. However, Cohen assumes that the output rate of the buffer equals the input rate when just one source is on.

Nearly all of the above mentioned fluid models assume an infinite buffer. Tucker (1988) considers the AMS model with a finite buffer. Mitra (1988) includes an analysis of both a finite and infinite buffer. Sericola (2001) extends the earlier work by Sericola & Tuffin (1999) to handle finite buffers also.

Concerning the proof of Theorem 1.4 Kulkarni refers to Mitra (1988). Interestingly, Mitra actually only proves the theorem for *reversible* Markov chains, rather than for chains with *general irreducible* generator. Mitra, in turn, mentions the paper of Sonneveld (1988) for a partial result on this problem. Finally, Sonneveld (2004), bringing the discussion to a nice conclusion, establishes the validity of Theorem 1.4. Moreover, he proves that the (algebraic) multiplicity of the eigenvector $\boldsymbol{\pi}$ is 1 and that all eigenvectors, except $\boldsymbol{\pi}$ and the eigenvector associated to the eigenvalue θ_{N_+}, have positive and negative components.

Transient Behavior In comparison to the stationary system, the analysis of the transient behavior of the process $\{W(t), C(t)\}$ received less attention. Ren & Kobayashi (1995) use double Laplace transforms (with respect to the time and buffer variable) to analyze the transient behavior of the AMS model. Tanaka *et al.* (1995) study the model of Stern & Elwalid (1991) and the case in which the distribution of the on/off periods of the sources may be phase-type, instead of exponential. They express their result as a Laplace transform with respect to the time variable. Sericola (1998) presents a numerically stable method based on recurrence relations to analyze the transient behavior of a fluid model with infinite buffer and a finite, ergodic background process.

State-dependent Drift Two primary references introduce state-dependent drift. The early paper of Elwalid & Mitra (1992) studies the case in which the buffer content controls the sending rate, thus the drift matrix, of a (finite) number of on/off sources. The control is such that when the content exceeds a certain (number of) threshold(s), traffic with too low priority cannot enter the buffer. Consequently, the drift matrix is piecewise constant. Elwalid & Mitra (1994) generalize their earlier work by considering higher-dimensional sources, i.e., sources with more than just two states.

Our discussion leading to (1.28), i.e., when the drift function is piecewise continuous, is mainly based on Kella & Stadje (2002).

Markov-modulated fluid queues with state-dependent drift relate to extensions of the classical storage process with state-dependent output. In the classical storage model, the input is a compound Poisson process, and the release rate (i.e., the rate at which the buffer depletes) is constant, see Prabhu (1980). Çinlar & Pinsky (1971) and Harrison & Resnick (1976) consider extensions of this storage process, in which the release rate is a strictly positive piecewise continuous function of the momentary buffer content.

Feedback Adan *et al.* (1998) and Scheinhardt (1998) introduced feedback fluid queues. Mandjes *et al.* (2003a, 2003b) and Scheinhardt (2001) find the stationary distribution for a class of feedback fluid queues where this dependence is piecewise constant, i.e., where the background process has a fixed generator as long as the buffer content is in between

two thresholds, or remains at one of the thresholds.

Boxma *et al.* (2005) extend the work of Kella & Stadje (2002) for an on/off source with piecewise continuous drift and transition rates and allow for unlimited buffer sizes. As a consequence, the existence of a stationary distribution requires proof, which they provide under some integrability conditions on the drift and transition functions. They also give explicit expressions for the stationary distribution when the buffer can, or cannot, become empty.

In regard of state-dependent storage processes, Bekker *et al.* (2004) allow not only the release rate but also the customer inter-arrival times to depend on the buffer content. Thus, there is feedback of information from the buffer to the source and the server. The dependence of the customer arrival rate (service rate) on the content process is comparable to the dependence of the generator matrix (drift matrix) in the fluid queueing context. As a sequel to this work, Bekker (2004) considers a similar model but with limited buffer size.

1.2 The Transmission Control Protocol

In this section we describe the workings of the Internet's Transmission Control Protocol (TCP). This description is intentionally concise; we refer to Kurose & Ross (2003), Peterson & Davie (2000), Walrand (1998), or Tanenbaum (1996), for background information on the organization and operation of the Internet in general and TCP in particular.

Section 1.2.1 summarizes TCP's responsibilities within the Internet. Part of these is to regulate the transmission rate of a source. Hence, in Section 1.2.2 we describe the algorithms used by TCP to control source transmission rates. These algorithms have been subject to several refinements over the years, with the aim of enhancing TCP's response to network congestion and packet loss. This process resulted in several, rather than one, TCP versions. As *TCP Reno* is the most popular TCP implementation in the Internet today, the focus in Section 1.2.2 is on TCP Reno. We summarize some other versions of TCP in Section 1.2.3. Finally, Section 1.2.4 discusses some functionality of routers to optimize the interaction with TCP.

1.2.1 TCP's Responsibilities

The Internet is a *packet-switched network* that provides a *best-effort service* to connections set up between a *sender* and a *receiver*. The term 'best-effort' means that the network does not guarantee to deliver a sender's packets at a receiver. In fact, during periods of overload, that is, periods during which the packet arrival rate at routers exceeds the rates of outgoing links, the network may drop packets.

Since some applications, such as file transfers, need error-free communication, Internet hosts need mechanisms to recover from packet loss and the corruption of the data carried by the packet. To establish reliable connections, hosts implement a form of *error control*. Achieving this is, by itself, not difficult, but complications arise as the probability of losing packets depends on the rate at which hosts transmit packets into the network. Thus, besides error control, senders require *congestion control*, also called *flow control*, to regulate the transmission rate. The Transmission Control Protocol (TCP) is responsible to achieve both goals at the same time. In other words, TCP should provide error and congestion control to connections.

The flow-control algorithms of TCP have to meet three criteria. First, the Internet's resources consist of *link capacity* and *buffer space*, which is located in routers. As these resources are *scarce*, it is important to use these resources *efficiently*. Second, possibly many users share these resources. Thus, to prevent some users from being blocked to the network altogether, the resources should be shared *fairly*. Third, it is well known that wild oscillations of momentary traffic load in the network adversely affect the utilization. To avoid large fluctuations in link utilization, TCP should be *stable*.

Coping with these requirements simultaneously is a complicated task. TCP achieves this (reasonably well given the circumstances) by using *feedback* from the network. This feedback consists of the network dropping packets. More precisely, when routers drop packets during overload, the network provides *information* to the sender about the level of congestion. As such, TCP uses *packet loss* as an indication of congestion. Based on this information the TCP sender adapts the rate at which it transmits packets. TCP increases the transmission rate during periods of low utilization (as then usually no packets are lost), but it decreases the rate during congestion.

We characterize this oscillatory behavior of TCP by *loss cycles* or *window cycles*. Suppose a cycle starts at a loss epoch. The sender decreases its rate until loss disappears. Then the sender increases its rate again up to the point the network drops a packet. This drop starts a new loss cycle, and so on.

We remark that the intelligence to control congestion is *distributed*: it is implemented in the hosts by means of TCP rather than in the network itself. This design principle is fundamentally different from traditional centralized telecommunication networks, such as the telephone network in which the switches control congestion by blocking, when necessary, telephony calls.

1.2.2 The Flow Control Algorithms of TCP Reno

In this section we describe the main characteristics of the flow control algorithms of TCP Reno as described by Stevens (1997) and Allman *et al.* (1998). Let us stress here that the operation of the flow algorithms depends crucially on the error control algorithms of TCP. We therefore start with summarizing the error control algorithm.

Error Control

Since packets may be dropped by the network, the receiver is supposed to return a small *acknowledgment (ack)* to the sender for each received packet[1]. When the sender does not receive an ack within some time after having sent a packet, the sender infers that the packet is lost and retransmits this (supposedly lost) packet. This procedure accomplishes error-free communication between sender and receiver.

Clearly, a certain amount of time should elapse before the sender can conclude that a packet is lost. Thus, the expiry of a timer should trigger the decision that packet loss occurred. For this purpose the sender maintains a *retransmission* timer and resets this at each packet transmission. When the retransmission timer expires before an ack arrives, a *timeout* occurs, and the sender considers the packet as lost. It is evident that the expiry time of the timer should at least be as large as the *round-trip time*. This time comprises: the time the packet needs to traverse the links from the sender to the receiver, i.e., the *propagation delay*; the time the packet spends in queue at routers, i.e., *buffering delay*; and the time the ack needs to 'travel back' from receiver to sender.

The method by which a TCP receiver informs the sender about *which* packet arrived is slightly counter-intuitive. As each TCP packet has a *sequence number*, the receiver might have included in the ack the sequence number of the just received packet. In spite of this, the ack procedure of TCP does not work in this way, but as follows. Suppose the receiver correctly received the packets $1, 2, \ldots, n-1$. When packet n arrives, it sends an ack with sequence number n. (Actually, TCP implementations send the last correctly received *byte* rather than the last correctly received packet. For ease of presentation, we explain TCP's algorithms in terms of packets.) Otherwise, when packet n is lost, but a subsequent packet, $n+1$ say, arrives, the receiver sends an ack with sequence number $n-1$, thereby acknowledging packet $n-1$ twice. These multiple acks for the same packet, in this case the packet with sequence number $n-1$, are called *duplicate* acks.

The packet-switched nature of the Internet has a subtle consequence. The order in which packets arrive at the receiver may differ from the order in which they leave the sender. When packet $n+1$ arrives immediately after packet $n-1$, and therefore *before* packet n, it is called an *out-of-order* packet. We remark that the receiver usually caches out-of-order packets. When the missing packet arrives later, the ack spawned by this missing packet acknowledges the last packet of the completed set of cached packets.

Another subtlety relates to the generation of acks. According to Braden (1989), a receiver should increase efficiency by not acknowledging every received packet, but instead, send an ack for every secondly received packet. This mechanism is known as the *delayed ack* algorithm. The receiver should bypass the delayed ack algorithm when out-of-order packets arrive, for reasons explained presently, and, instead, immediately

[1] Formally speaking, TCP end hosts do not exchange packets but segments. We do not distinguish between these two different notions.

acknowledge out-of-order packets.

We explain next the role of this error-control mechanism as part of TCP's congestion control algorithms.

Congestion Window and Flightsize

To prevent excessive loss of packets the sender should limit the rate at which it sends packets into the network. To this aim a TCP sender uses a *window* to constrain the `flightsize`, i.e., the number of packets sent but not yet acknowledged. At startup the sender transmits a window worth of packets. After each arriving ack, the window slides forward to allow the transmission of a new packet. Thus, TCP uses a *sliding window* algorithm to control its transmission rate.

The window size is subject to two constraints. First, the sender has to adapt its transmission rate to the continuously varying amount of available capacity along the path to the receiver. The state variable `cwnd`, shorthand for *congestion window*, controls this dynamic aspect of the window size. Second, the receiver may also limit the window by `rwnd`, which is the *receiver window*. Combining these two constraints, the sender's transmission rate should always satisfy:

$$\texttt{flightsize} \leq \min\{\texttt{cwnd}, \texttt{rwnd}\}. \tag{1.32}$$

Flow Control: Slow Start and Congestion Avoidance

To control the evolution of the transmission rate, i.e., `cwnd`, the sender uses a second state variable: `ssthresh`, i.e., *slow start threshold*. Initially `cwnd` is set to 1, so as to avoid sending a burst of packets into the network immediately. The initial value of `ssthresh` may be chosen arbitrarily, but it is usually considerably larger than 1. When the sender sends its first packet, `flightsize` becomes equal to 1. Since also $\texttt{cwnd} = 1$, the sender has to wait for the return of an ack, in accordance with (1.32). If no loss occurs, this ack arrives after one round-trip time. At the receipt of an ack the sender increases `cwnd` according to the update rule:

$$\texttt{cwnd} \leftarrow \begin{cases} \texttt{cwnd} + 1 & \text{if } \texttt{cwnd} < \texttt{ssthresh}, \\ \texttt{cwnd} + 1/\texttt{cwnd}, & \text{if } \texttt{cwnd} > \texttt{ssthresh}. \end{cases} \tag{1.33}$$

Since at first $\texttt{cwnd} < \texttt{ssthresh}$, the sender increases `cwnd` by one, so that it is allowed to transmit two packets. Then, after the acknowledgment of each of these two packets, `cwnd` equals four. As such, `cwnd` doubles each round-trip time, and thus the sending rate increases exponentially in time. This phase of the congestion algorithm is called *Slow Start*, as the source sends initially just one packet per round-trip time. On the other hand, when $\texttt{cwnd} > \texttt{ssthresh}$, the congestion window increases by an amount

$1/\texttt{cwnd}$ for every ack. This results in approximately linear increase of `cwnd` in time as now `cwnd` increases by (approximately) one packet every round-trip time. This phase is called *Congestion Avoidance*. Finally, when `cwnd` equals `ssthresh`, the sender may use either Slow Start or Congestion Avoidance, as the difference is small.

Error Recovery: Fast Retransmit and Fast Recovery

Observe that while no loss occurs, the source increases its sending rate as specified by (1.33). Consequently, assuming the receiver windows are so large that they do not constrain the transmission rates, there will come a point in time that the combined rate of all TCP connections using a link in the Internet exceeds this link's capacity. At first the buffer in front of the congested link begins to fill. However, it will overflow eventually, resulting in one or more packet drops.

Packets sent after the lost packets, but not discarded at the buffer, provide useful information to detect packet loss. When these packets arrive at the receiver, the receiver perceives these as out-of-order packets, and therefore responds by sending duplicate acks. Now, the sender does not know whether packet loss or a temporary reordering of packets causes these duplicate acks. Therefore it waits for three subsequent duplicate acks to conclude that a packet loss occurred. However, data should still arrive at the receiver—otherwise just one duplicate ack would have arrived. Thus, duplicate acks provide crucial information about the state of congestion of the network.

After three duplicate acks the sender enters *Fast Retransmit*. In this phase the sender retransmits what appears to be the missing packet, *without* waiting for the retransmission timer to expire. Furthermore, it updates its state variables according to the rule:

$$\texttt{ssthresh} \leftarrow \texttt{flightsize}/2 \tag{1.34a}$$

$$\texttt{cwnd} \leftarrow \texttt{ssthresh} + 3, \tag{1.34b}$$

and enters the next phase of the error recovery procedure: *Fast Recovery*.

During Fast Recovery the sender increments `cwnd` by one for each additional duplicate ack it receives. When finally the ack arrives that acknowledges the retransmitted packet, and consequently the other packets cached at the receiver, the sender *deflates the window*, i.e., it sets

$$\texttt{cwnd} \leftarrow \texttt{ssthresh},$$

and leaves Fast Recovery. Thus, the update rule (1.34a) for `ssthresh` effectively halves the rate when the sender enters Congestion Avoidance for the next loss cycle. Note that we did not discuss some of the more detailed consequences of (1.34) as pointed out by Fall & Floyd (1996); these are of less importance here.

The Ack Clock

We finally discuss a consequence of TCP's flow control algorithms pointed out by Jacobson (1988): the *ack clock*. Consider a 'tagged' connection. Packets of this connection interleave with packets of other connections at buffers. While passing a bottleneck link in the path the tagged packets spread out in time due to the service of untagged packets. After the packets leave the bottleneck, the inter-packet spacing remains approximately constant, and, so will the ack spacing. Thus, if the 'tagged' sender only transmits packets in response to an ack after the initial burst, the sender's packet spacing matches approximately the packet service time on the link that carries the highest load in the path. This phenomenon is called the *self-clocking* nature of TCP.

1.2.3 Some Other TCP Versions: Tahoe, NewReno and Sack

Above we explained the TCP Reno version. In this section we describe, shortly and with no objective of being complete, the three other TCP versions that are most relevant for our purposes: Tahoe, NewReno, and Sack.

The Tahoe version is, loosely speaking, a predecessor of TCP Reno. TCP Tahoe is nearly the same as Reno, except that a TCP Tahoe sender enters Slow Start after Fast Retransmit instead of Fast Recovery. Thus, after a loss event a Tahoe sender transmits at a considerably smaller rate than a Reno sender. Consequently, TCP Tahoe is usually less efficient than TCP Reno.

The other two versions, NewReno and Sack, intend to repair a weakness of TCP Reno. It turns out that TCP Reno sometimes cannot recover from multiple losses occurring in one window, cf. Fall & Floyd (1996) and some references therein. Consider, for instance, a case in which `cwnd` is small, e.g., five, and the first and last packet are dropped. Then the duplicate acks for packets 2, 3 and 4 trigger Fast Retransmit. After the acknowledgment of the retransmitted first packet, the sender leaves Fast Retransmit, but `flightsize` is not large enough to spawn three duplicate acks for the lost packet 5. Consequently, the sender has to wait for a timeout of the retransmission timer. In general terms, the cause of the problem is that the sender may leave Fast Recovery before the acknowledgment of all packets of the window outstanding at the moment of loss discovery.

A small change of Fast Recovery may reduce the influence of this undesirable effect, leading to TCP NewReno, cf. Floyd & Henderson (1999). These authors observe that an ack that acknowledges part of, but not the entire window outstanding at loss detection, is a signal (excluding the possibility of reordering) that more than one loss occurred in that particular window. Such acks are denoted as *partial acks*. The improvement consists of not leaving Fast Recovery after a partial ack, but instead immediately sending out the packet that is missing according to the partial ack. The sender will leave Fast Recovery

and deflate `cwnd` only after the acknowledgment of all packets of the 'inflicted' window. Thus, a sender sometimes sends out just one packet per round-trip time, but it does not have to wait as often for timeouts, which, generally, takes much longer.

Another proposal to improve TCP's behavior during multiple losses is TCP Selective Acknowledgment (SACK), see, e.g., Mathis *et al.* (1996). Basically, a TCP Sack receiver informs the sender about the sequence number of each successfully arriving packet, instead of merely the last received packet of a sequence of consecutive packets. Therefore a TCP Sack sender knows exactly which packet(s) is (are) missing so that it can just retransmit these packets, cf. Mathis *et al.* (1996).

1.2.4 Random Early Detect Buffers

Random Early Detect (RED) is not part of TCP's flow control algorithms but, instead, is implemented in routers to work *in conjunction* with TCP. The aim of implementing RED is to reduce the influence of some undesirable consequences of TCP's reaction to loss.

As an example, consider a bottleneck link used by many connections. Suppose further that the buffer in front of the link uses a *drop-tail policy*. A drop-tail buffer drops every arriving packet during a *loss epoch*, i.e., the period that it is full. Because of clustering of packets in buffers elsewhere in the network or due to particular source behavior, TCP traffic can be rather bursty. Thus, when many packets arrive for the same link in a short amount of time and the buffer is almost full, the buffer drops a considerable number of packets within one loss epoch. As a consequence, many connections may suffer from loss and therefore react *in synchrony*: they reduce their rate simultaneously after a loss epoch. As a result, the aggregate rate may become (considerably) smaller than the link rate after a loss epoch. Therefore, synchronization may adversely affect the utilization, especially when the period of underutilization lasts for a considerable number of round-trip times in succession (due to the linear increase during Congestion Avoidance).

Floyd & Jacobson (1993) propose RED as a, perhaps better, more gradual mechanism to reduce congestion. A buffer equipped with the RED algorithm drops packets of just a few, instead of many, TCP connections at the first signs of incipient congestion. Then the input rate reduces less, ideally just a bit below the link rate, and the period of underutilization lasts for a shorter amount of time.

The problem here is, of course, to choose the connections that should reduce their rates. It is, obviously, unfair to drop packets from the same connections always. To avoid this it seems best to drop packets at the buffer independent of anything else, such as the packet's address, but with some probability p equal for all packets. This randomness establishes that, generally, not the same senders suffer at every congestion period. In effect, such a drop strategy 'breaks the synchronization' as just a few, instead of many, sources reduce their rate. Moreover, sources with higher rate have proportionally higher proba-

bility to lose a packet, which should result in a more uniform sharing of the capacity.

To implement these ideas, a RED buffer with a physical capacity of B packets maintains an estimate $x \in [0, B]$ of the *average* queue length. The buffer updates the estimate x at each packet arrival according to an exponentially weighted moving average with weight $\epsilon \in (0, 1)$: if q_i is the queue length observed at the arrival of packet i then

$$x_i = (1 - \epsilon)x_{i-1} + \epsilon q_i. \tag{1.35}$$

Besides the variable x, the RED buffer has two thresholds $0 < x_{\min} < x_{\max} < B$. These thresholds regulate the drop probability function $p(x)$. The buffer drops packet i with probability $p(x_i)$, where $p(x)$ has the form

$$\text{i.e., } p(x) = \begin{cases} 0, & 0 \le x \le x_{\min} \\ \frac{x - x_{\min}}{x_{\max} - x_{\min}} p_m, & x_{\min} < x \le x_{\max} \\ 1, & x_{\max} < x, \end{cases} \tag{1.36}$$

cf. Figure 1.2. Thus, $p(x)$ is a non-decreasing function of the average queue length x. The idea behind this form for $p(x)$ is to achieve that the higher the congestion (indicated by high values of x) the more packets the buffer drops and the more sources reduce their rate.

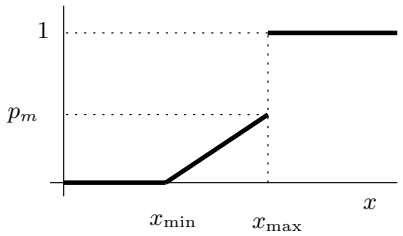

Figure 1.2: *The loss probability for the RED algorithm.*

Because of the bursty nature of the arrival stream, x seems a better estimate of the congestion level than the momentary queue length q. To see this, observe that sometimes q increases quickly for a (very) short period of time while the utilization of the link as averaged over a round-trip time is below 1. Thus, to detect congestion in a more reliable way it may be better to use some low-pass filter, such as (1.35), on the queue length dynamics. Setting $x_{\max} < B$ allows the buffer to absorb occasional bursts instead of forcing it to drop many packets.

We remark that tuning the thresholds $x_{\min}, x_{\max}$ and the weight ϵ in relation to B and the link capacity L is not particularly straightforward. It seems that an optimal choice for all network environments does not exist, see, e.g., Misra *et al.* (2000). Thus, it appears that RED does not resolve all of the above mentioned problems with drop-tail buffers.

1.3 Modeling TCP's Flow Control Mechanisms

Now that we have discussed the flow control algorithms of TCP we turn to the subject of simulating and modeling TCP's behavior.

1.3.1 Overview

TCP simulators often contain implementations of (part of) the code of TCP or even the entire protocol stack. The advantages of such simulators are clear. These simulators can handle a level of detail that comes closest to TCP as used in the 'real Internet'. Moreover, it can help reveal inconsistencies and errors in the implementation of (one of the flavors of) TCP. Finally, it offers a testing ground for mathematical models; in fact, we also use simulation for this purpose in Chapters 5 and 6 of this monograph. The simulation environment of choice for TCP is, without doubt, the network simulator ns-2. This simulator offers: implementations of the various TCP versions; buffers with different drop strategies; source models; methods to set up entire networks, and so on.

For all its merits, simulation (with ns-2) also has its drawbacks. Here we mention two of these; Floyd & Kohler (2002) discuss this issue in more detail. First, the behavior of ns-2 is deterministic when the user does not introduce 'randomness' by, for instance, using RED buffers, which rely on random number generators to drop packets. Because of this determinism the results of ns-2 are sometimes obviously wrong (a real experiment will never produce such results as always some stochasticity is present). Clearly, assessing the results of ns-2 requires an understanding of TCP behavior beyond the purely phenomenological information that simulation provides. Second, it is impossible to simulate large networks, e.g., networks consisting of (tens of) thousands of connections and routers, as the number of details that the simulation has to keep track of simply becomes too large. In conclusion, it is necessary to have TCP models that, on the one hand, provide an understanding of the most important aspects of TCP's behavior, and on the other hand, leave out less relevant details to achieve scalability.

Which details to keep and which to neglect depends partly on the *time scale* of interest. *Window-level models* focus on the time scale at which the transmission rate dynamics, i.e., the window dynamics, are of primary interest. *Flow-level models*, on the other hand, abstract away from the window dynamics, but concentrate on the time scale at which TCP sessions start, i.e., 'switch on', and stop, i.e., 'switch off'. In the next two sections we summarize some of the mathematical work carried out for either time scale.

We remark that, although some authors call the window level the 'packet level', we prefer the more descriptive term 'window-level'. Second, the division between these two levels of modeling TCP is not strict. For instance, some authors combine window-level and flow-level models.

1.3.2 Window-level Models

We first describe a model of the behavior of a generic TCP source between two loss epochs. This model is, in a sense, the starting point for the four types of window-level model we subsequently discuss. The first TCP model relates the throughput of a single connection to some properties of the underlying Internet path. A second class of models extends to situations in which two (or more) connections compete for bandwidth. These models focus on aspects of bandwidth sharing and utilization. The third class of models clarifies network issues such as bottleneck localization. The fourth class enables us to analyze transient behavior of the sources and the network.

TCP Source Modeling

Above we motivated the necessity to make some simplifying assumptions about the behavior of TCP. We discuss this now.

One assumption, made in nearly all TCP source models, is to neglect Slow Start and to concentrate on Congestion Avoidance. This simplification is based on the observation that during a single loss cycle a TCP source usually spends (much) less time in Slow Start than in Congestion Avoidance. Moreover, TCP Reno does not enter Slow Start after single losses in one window and TCP NewReno and Sack not necessarily even after multiple losses in one window. A second simplification consists of omitting the details of Fast Recovery. Padhye *et al.* (2000) provide experimental results providing further support for the validity of these assumptions.

By neglecting Slow Start we may characterize the dynamic behavior of a TCP source as *Additive-Increase/Multiplicative-Decrease* (AIMD), cf. Chiu & Jain (1989). During Congestion Avoidance the rate increases linearly in time corresponding to Additive-Increase. After each congestion signal, the source rate decreases by a factor two, corresponding to Multiplicative-Decrease. In view of this the following remark of Misra *et al.* (1999) is of interest: 'All flavors of TCP ... are successive refinements ... to implement ideal Additive-Increase/Multiplicative-Decrease behavior.'.

A second step in the modeling of TCP abstracts away from the packet-based nature of the transmission process of a TCP source. Because of self-clocking, packet spacing roughly matches the service time, so that a fluid source conveniently approximates the sender's output process. (However, Fall & Floyd (1996) mention some cases in which packet bursts *do* occur, and discuss some approaches to prevent this from happening.)

We note three further assumptions frequently made to simplify the analysis (and appear reasonable):

1. The receiver window is so large that it never constrains the congestion window;

2. The delayed ack algorithm from TCP is not switched on;

3. Acks are never dropped on the return path.

The modeling of a TCP source as a fluid source subject to Additive-Increase between loss epochs yields a straightforward analytic description. Following Lakshman & Madhow (1997), suppose the path contains just one bottleneck link. The link capacity is L and the buffer size is B. Let $W(t)$ denote the source window size at time t, dW/dt the growth rate of the window with time, dW/da the rate of growth with arriving acks, and da/dt the rate at which acks arrive at the sender. Then, during Congestion Avoidance the update rule (1.33) implies that

$$\frac{dW}{da} = \frac{1}{W}.$$

Now observe that because of self-clocking

$$\frac{da}{dt} = \begin{cases} W/T, & \text{if } q(t) = 0, \\ L, & \text{if } q(t) > 0, \end{cases} \tag{1.37}$$

where $q(t)$ is the queue length of a bottleneck buffer at time t and T is the round-trip time.

Many models do not incorporate the buffering delay B/L when it is small compared with the propagation delay T_p. Evidently, $B/L \ll T_p$ is equivalent to $B \ll LT_p$; this regime is generally called the *large bandwidth-delay limit*. In that case $q(t) \equiv 0$ so that by (1.37)

$$\frac{dW}{dt} = \frac{dW}{da}\frac{da}{dt} = \frac{1}{W}\frac{W}{T} = \frac{1}{T}. \tag{1.38}$$

The window-level models use this relation to describe the behavior of the window during loss-free periods.

Although modeling the window process between two loss event is relatively simple, it is a difficult problem to: (1) model the loss process at buffers along the path, and (2) assign the loss to the various active connections. In fact, several modeling approaches focus on (and differ about) these two points.

Finally, we mention the influence of the application layer on TCP. If an application has an infinite amount of traffic to send, the source is always 'on', or *greedy*. Otherwise, TCP sessions start and stop, for instance after the completion of a file transfer.

Relating Throughput to Path Characteristics: The Root p Law

We now summarize the work of Mathis *et al.* (1997) who establish a simple expression for the throughput of a single, greedy, AIMD source that competes with many other connections for the bandwidth of some bottleneck link in the Internet. The authors make two basic assumptions to express the throughput as a function of the round-trip time T and the perceived loss probability p. First, T is assumed to be approximately constant;

which is valid in the large bandwidth-delay limit. Second, the loss process is assumed to be periodic and *exogenous*, i.e., independent of the source rate.

To derive the result, they use the graph of the window dynamics shown in Figure 1.3. The horizontal axis represents time (with round-trip times as units) and the vertical axis represents the window size. Loss occurs when the connection reaches a maximum window $W_{\max}$, a constant to be determined shortly. Then the sender reduces its window by a factor two, in accordance with (1.34). Thus, immediately after loss the window equals $W_{\max}/2$. Now, by (1.38), W increases linearly during Congestion Avoidance from $W_{\max}/2$ to $W_{\max}$. Here a new loss occurs and a new loss cycle begins. Note that, although this figure presents a rather simplistic version of reality, it conveys at least the main aspects of the window dynamics of a single TCP source interacting with a bottleneck link.

To approximate the throughput γ, consider the area of the gray 'sawtooth' in the figure. As it takes $W_{\max}/2$ round-trip times to increase from $W_{\max}/2$ to $W_{\max}$, this area equals $(W_{\max}/2)^2(1+1/2) = 3W_{\max}^2/8$. Therefore, on average, the throughput is (in bytes per second),

$$\gamma = P\,\frac{3\,W_{\max}^2/8}{T\,W_{\max}/2} = \frac{3\,P}{4}\,\frac{W_{\max}}{T}, \tag{1.39}$$

where P is the packet size in bytes.

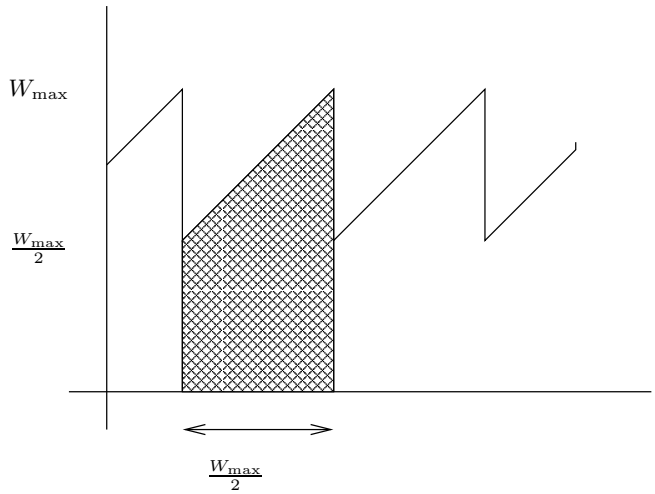

Figure 1.3: *The window size as a function of the number of round-trip times. It takes $W_{\max}/2$ round-trip times of duration T to complete one loss cycle.*

The above result still contains $W_{\max}$. To relate this to more desirable parameters, Mathis *et al.* (1997) express $W_{\max}$ in terms of the loss probability of the path. In the

setting of Figure 1.3, the loss probability p follows from observing that in one cycle one packet is lost while $3W_{\max}^2/8$ packets are sent, so, $W_{\max} = \sqrt{8/3p}$. Substituting this in (1.39) yields the celebrated *'root-p law'*:

$$\gamma = \sqrt{\frac{3}{2}}\,\frac{P}{T\sqrt{p}}. \tag{1.40}$$

Note that this expression for γ is the 'goodput', i.e., the number of transmitted bytes per second minus the number of lost bytes per second.

We close this section with mentioning some extensions of the root-p law in the literature. In the first place, the constant $\sqrt{3/2}$ depends on specific modeling assumptions. Ott *et al.* (1996) do not assume the loss epochs to occur periodically, but geometrically distributed (with a constant probability per packet), which gives $C \approx 1.31$. Second, Padhye *et al.* (2000) propose a refinement which differs primarily in that their model takes timeouts and receiver window limitations into account. Third, the independence assumption for the loss process need not always be true. Altman *et al.* (2000a) study the influence on the root-p law when packet losses are allowed to be correlated. Finally, observe that (1.40) is not appropriate to estimate the throughput for very small files. The size of such files is such that the TCP session may not leave Slow Start at all or see any loss. Cardwell *et al.* (2000) extend the work of Padhye *et al.* (2000) to capture also startup effects of TCP.

Fairness and Utilization

The root-p law (1.40) relates throughput to round-trip times and loss probabilities. It does not, however, provide insight into how contending connections share a bottleneck resource. Here we consider a simple network consisting of just two sources and one bottleneck link, to analyze how different source parameters affect capacity sharing and utilization. At the end of the section we briefly review *fairness*, a concept related to the share of the capacity each source receives, as this is of interest for its influence on the quality of service users implicitly attribute to the network.

Since now two sources share a link, it is necessary to model which source suffers from loss during congestion, as it can be either of the two, or both. Lakshman & Madhow (1997) consider the case of *synchronous loss* according to which during congestion both sources lose one or more packets. Hence, both sources react *in synchrony* to overflow by reducing their rate simultaneously. Altman *et al.* (2000b) observe that while the synchronization assumption is approximately valid for small drop-tail buffers and connections with similar round-trip times, this is not the case when RED buffers are deployed. Rather, a RED buffer uses approximately a *proportional loss model*, according to which the buffer chooses probabilistically just one connection to suffer from loss, and the probability to select a connection is proportional to its momentary transmission rate. We

now summarize expressions for the throughput for both loss models, i.e., synchronous and proportional loss, as obtained in the literature for the large bandwidth-delay limit.

First we consider two sources that react in synchrony to loss; the case analyzed by Lakshman & Madhow (1997). Clearly, during overflow the aggregate of the source rates exceeds the link capacity L. Nevertheless, the sources react within one round-trip time to loss implying that the excess cannot be large. Hence, Lakhsman and Madhow simply suppose that the sum of the rates equals the link capacity during overflow. Thus, assuming that both sources have the same packet size P, it follows that

$$\frac{W_{1,\max}}{T_1} + \frac{W_{2,\max}}{T_2} = \frac{L}{P},$$

where T_i and $W_{i,\max}$ denote for source i the round-trip time and the window size at which loss occurs. This is obviously one equation in two unknowns: $W_{1,\max}$ and $W_{2,\max}$. A second relation between $W_{1,\max}$ and $W_{2,\max}$ follows from the synchronization assumption, which implies that the sources reduce their rates at approximately the same moment and then start increasing the window according to (1.38). Suppose that a loss cycle starts at $t = t_1$ and stops at $t = t_2$. Then (1.38) gives for source 1:

$$W_{1,\max} - \frac{W_{1,\max}}{2} = \int_{W_{1,\max}/2}^{W_{1,\max}} dW = \int_{t_1}^{t_2} \frac{dW}{dt}\, dt = \frac{t_2 - t_1}{T_1},$$

from which $W_{1,\max} = 2(t_2 - t_1)/T_1$. Connection 2 has the same loss cycle so that, similarly, $W_{2,\max} = 2(t_2 - t_1)/T_2$. Thus we find that $W_{1,\max}/W_{2,\max} = T_2/T_1$. Combining the above relations gives

$$W_{1,\max} = \frac{T_1 T_2^2}{T_1^2 + T_2^2} \frac{L}{P}, \qquad W_{2,\max} = \frac{T_1^2 T_2}{T_1^2 + T_2^2} \frac{L}{P}.$$

Substituting this into (1.39) yields the throughputs,

$$\gamma_1 = \frac{3}{4} \frac{T_2^2}{T_1^2 + T_2^2} L \qquad \gamma_2 = \frac{3}{4} \frac{T_1^2}{T_1^2 + T_2^2} L. \tag{1.41}$$

An immediate consequence is that γ_1/γ_2 is inversely proportional to the *square* of the ratio of the round-trip times, i.e.,

$$\gamma_1/\gamma_2 = s^{-2}, \tag{1.42}$$

with,

$$s := T_1/T_2. \tag{1.43}$$

It is evident from (1.41) that the total utilization

$$u = \frac{\gamma_1 + \gamma_2}{L} = \frac{3}{4}. \tag{1.44}$$

Observe that this is independent of s.

The proportional loss model is much harder to analyze. However, Altman *et al.* (2002b) establish approximations which appear quite accurate. They find

$$\gamma_1 \approx \frac{1}{4}\frac{4s+3}{(s+1)^2}\,L \qquad\qquad \gamma_2 \approx \frac{s}{4}\frac{3s+4}{(s+1)^2}\,L, \tag{1.45}$$

from which

$$\frac{\gamma_1}{\gamma_2} \approx \frac{1}{s}\frac{4s+3}{3s+4} \qquad\qquad \gamma_1+\gamma_2 \approx \frac{3}{4}\frac{(s+1)^2+2s/3}{(s+1)^2}\,L > \frac{3}{4}L, \tag{1.46}$$

since $s > 0$. Clearly, the ratio of throughputs is no longer proportional to s^{-2}, as in the synchronous loss model. Rather, Altman *et al.* (2002b) remark that this is approximately equal to $s^{-0.85}$ for $s \in [0.1, 1]$. Moreover, the utilization is higher.

We now turn to the notion of fairness which relates to the share of bandwidth a connection obtains. More or less in line with Altman *et al.* (2000b) we call 'fairness' the ratio of each connection's share, i.e., γ_1/γ_2, and consider the sharing as 'fair' when the ratio equals one. Interestingly, there has been recent discussion about whether receiving equal shares of the scarce capacity under all network circumstances is actually 'fair' from an economical (or game-theoretical) point of view. For further discussion on this topic the reader might consult for instance Kelly *et al.* (1998), Kelly (1997, 2000, 2001), Massoulié & Roberts (1999).

Bottlenecks and Network Utilization

The two analytic 'fairness models' provide simple estimates for source throughputs, providing thereby a framework to interpret simulation results. However, as these models maintain the state of each source in the network, they do not scale, i.e., they do not easily extend to multiple sources or networks. Thus, when investigating issues pertaining specifically to network analysis, e.g., locating bottlenecks in a given network topology or dimensioning links used by many connections, we are in need of other approaches.

To locate bottlenecks, for instance, we may proceed as follows. First describe the network topology by a *routing matrix*, i.e., an incidence matrix of the routers used by each connection. Then, specify the capacities of the links connecting the routers and the buffer types (drop-tail, RED, and so on) and sizes. Third, use a relatively simple model for the source throughput, for instance the root-p law (1.40). Fourth, combine the source and network model and compute iteratively the load on each link.

In more specific terms this iterative procedure works as follows.

1. Start with an initial estimate for the throughput of each connection.
2. Use these estimates and the routing matrix to compute the load on each link in the network.

3. Make some (simplifying) assumptions on the packet arrival process and service discipline at each buffer. Since we have the load by the previous step we are in the position to use classical queueing models of the buffers to find the loss probability and average queue length at each buffer.

4. Attribute the loss at each buffer to the loss probability observed by each connection.

5. Now that we have the loss probability and round-trip time, including the average queueing delays, we compute with (1.40) for each connection the new estimates of the throughputs.

6. Finally, check whether the estimates of the throughputs satisfy some stop criterion. If not, return to the second step of the procedure with the new estimates for the throughputs.

It is simple to prove that for a single bottleneck buffer a unique fixed point exists to which the sequence of estimates for the loss probabilities and throughputs converges. However, not many results are available for general networks.

We now discuss some notable references which use this (or a related) approach. Altman *et al.* (2002a) consider the case of large bandwidth-delay products, i.e., negligible buffer sizes. Hence, they cannot use step 3 of the above procedure. Instead, the sum of the throughputs carried by a link cannot exceed the capacity. This imposes a number of inequalities (for each link) on the solution. As TCP connections always increase the rate up to the point of link overflow (assuming the receiver windows have no influence) necessarily some of these constraints are tight. Gibbens *et al.* (2000) and Avratchenkov *et al.* (2002) include queueing delays and model the queueing process of a drop-tail buffer as an $M/M/1/K$ queue. This yields simple expressions for the loss probability of each buffer. Bu & Towsley (2001) use a similar approach to that of Gibbens *et al.* (2000) but consider RED buffers instead of drop-tail buffers.

Another class of fixed-point methods uses detailed Markovian models of the window process of a single source. The aggregate of many such sources defines the load on the network. Then, by modeling the buffers in the network as $M/M/1/K$ queues, the loss probabilities and average queue lengths follow immediately from the load. In turn, the throughput model uses the loss probabilities and queue lengths as input. Thus, a fixed point method suffices to compute throughput, loss, and average queue length. Casetti & Meo (2000) consider a superposition of identical but statistically independent TCP Tahoe sources that send traffic into a single-node or a two-node network. Casetti & Meo (2001a, 2001b) replace TCP Tahoe by TCP Reno within their earlier framework. Ajmone Marsan *et al.* (2000) investigate the influence of synchronization by considering identical, but completely coupled, sources sharing a bottleneck link.

Garetto *et al.* (2001a) associate to each window state an $M/M/\infty$ queue in a closed queueing network. The number of jobs in each queue represents the number of identical but independent greedy TCP Tahoe sources in a certain protocol state. By mean-value analysis, cf. Harrison & Patel (1993: Section 6.4), they compute the average number of TCP sources in each protocol state (queue), from which the average aggregate throughput follows. The transition probabilities between the queues depend on the loss and round-trip times of the connections. Garetto *et al.* (2002) consider on/off sources by replacing the closed network with an open network. Alessio *et al.* (2001) consider mixtures of on/off Tahoe and New Reno connections. Garetto *et al.* (2001b) extend this work to topology aware networks: for instance, differences in routes of the connections (and hence different loss probabilities and round-trip times) are now taken into account.

The fixed-point approach can handle many connections. Moreover, it yields insight into the location of bottlenecks and the average throughput and loss, from which file transfer latencies follow. A disadvantage is its inability to provide information about transient behavior.

Transient Behavior of the Network

The models leading to (1.38), (1.41) and (1.45) extend to multiple sources and, in fact, entire networks, by using numerical methods to solve certain systems of differential algebraic equations (DAEs), which are to be derived below. With this approach we can locate bottlenecks, obtain insight into transient aspects of the network, and so on. On the other hand, in comparison to the fixed-point methods treated above, solving the large systems of DAEs is computationally much less efficient.

Brown (2000) makes a first step in the generalization of the work of Lakshman & Madhow (1997). He still concentrates on a single bottleneck link shared by multiple sources with, possibly, different round-trip times, but includes the queueing delay in the round-trip times. As we use this model in Chapters 5 and 6 we discuss it in some more detail.

Let us first consider the dynamics of the sources. Suppose that T_i is the round-trip time of source i, $1 \leq i \leq J$, when the buffer is empty. Then,

$$T_i(q(t)) := T_i + \frac{q(t)}{L} \tag{1.47a}$$

is the round-trip time of source i when the buffer content is $q(t)$ at time t. Source i maintains a window variable $W_i(t)$, supposed to be continuous, and sends fluid at rate $W_i(t)/T_i(q(t))$ into the buffer. Consequently, between consecutive loss epochs the window process evolves according to

$$dW_i(t) = \frac{dt}{T_i(q(t))}, \tag{1.47b}$$

which, as it incorporates the queue length, clearly extends (1.38). The evolution of the queue length is given by

$$\frac{dq}{dt} = \begin{cases} \max\{r(t), 0\}, & \text{if } q(t) = 0, \\ r(t), & \text{if } q(t) \in (0, B), \\ \min\{r(t), 0\}, & \text{if } q(t) = B, \end{cases} \tag{1.47c}$$

where the drift function has the form

$$r(t) := \sum_{i=1}^{J} \frac{P_i W_i(t)}{T_i(q(t))} - L, \tag{1.47d}$$

and P_i is the packet size of source i. Thus, (1.47a), (1.47b) and (1.47c) form a system of ordinary differential and algebraic equations that describes the dynamics of the (deterministic) system from loss epoch to loss epoch. However, this system does not specify the behavior *across* a loss epoch, i.e., from the start to the end of a congested period.

To compute the window values after loss, Brown (2000) uses the synchronized loss assumption. Thus, if t_1 is the first moment of loss, i.e., $q(t_1) = B$, the window right after t_1 is half of the window just before t_1 for *each source*, i.e., $W_i(t_1+) = W_i(t_1-)/2$ for $i = 1, \ldots, J$. Now Brown integrates the system (1.47) from t_1+ to the next loss moment t_2, etcetera.

This approach generalizes simply, at least conceptually, to networks. Then (1.47a) includes the queueing delays of all buffers along the path, and the sum in (1.47d) runs over all connections that use a specific link and buffer in the network. Furthermore, when source i, say, represents a *class* of n_i identical sources, it becomes relatively simple to include large numbers of sources *of the same class*.

The assumption of the loss process has decisive influence on the behavior and numerical evaluation of the system. With the synchronous loss assumption all sources in one class reduce their rate with the same amount at overflow, so that identical connections remain 'identical'. Contrary to this, with the proportional loss assumption sources react individually to loss, so that they cannot aggregate into classes. Thus, in the latter case the analysis of networks with thousands of active connections or complicated topologies results in a state space explosion.

Baccelli & Hong (2003b) follow the synchronized loss assumption of Brown (2000) to analyze large networks. Baccelli & Hong (2003a) attack the problem of handling the large system of differential equations in quite a different way. They integrate the system of ODEs from loss period to loss period. But instead of using synchronization to 'cross the loss epoch', these authors simulate a loss process, thereby assigning the loss to some, or possibly all, sources. In a sense, this approach is a mixture between simulation and analytic models based on ODEs.

Misra *et al.* (2000), following Misra *et al.* (1999), propose to incorporate the source reaction to loss directly into the equations that govern the window dynamics, thereby replacing (1.47b). Instead of (1.47b) they write

$$dW_i(t) = \frac{dt}{T_i(q(t))} - \frac{W_i(t)}{2} dM_i(t). \tag{1.48}$$

The first term of the right hand side corresponds to the Additive-Increase behavior of a source, as before. The second term implements Multiplicative-Decrease after a loss epoch. Here, $M_i(t)$ models the loss arrival process as a point process: $dM_i(t) = 1$ at the arrival of a loss and 0 elsewhere. Now Misra *et al.* (2000) take expectations at the left and right hand side of (1.47) (and of (1.35) as they consider RED buffers) and make several simplifying assumptions to obtain a (numerically) tractable system of differential equations. In Section 5.1.1 we discuss this model more thoroughly. Liu *et al.* (2003) extend this work to topology-aware networks, i.e., they incorporate the sequence in which packets traverse the routers as well as the propagation delays of the links between routers.

1.3.3 Flow-level Models

The window-level models we dealt with up to now provide insight into fairness, utilization and bottleneck localization. At coarser time scales these aspects are relatively less relevant, while the finiteness of file size and the arrival rate of files to transfer become more important.

The concept of *flow*, see, e.g., Roberts & Massoulié (2000); Ben Fredj *et al.* (2001), may be helpful to explain this point. A flow corresponds to an individual document transfer, e.g., a web page or an mp3 track. From the user's point of view, flows are *elastic* in that they do not have tight constraints about delay or throughput. Instead, the realized rate averaged over a considerable number of round-trip times is of greater importance as this determines the *latency* of the flow, which in turn represents an important aspect of the quality of service offered by the network. (We assume here that the file size is large enough; the latency for small files is dominated by TCP's Slow Start, of course.)

The main objective of flow-level models is to express the response time of a flow of given size as a function of the arrival process and the size distribution. The results may then be used to derive provisioning rules and other controls, e.g., flow admission control as advocated by Roberts & Massoulié (2000), to maintain the service level of the network above a certain level.

In certain settings processor sharing (PS) queues are appropriate models to obtain insight into the just mentioned problem. Below we first describe the processor sharing queue. Then we discuss the TCP scenarios to which PS modeling have been applied. Finally, we mention some generalizations of PS that made their appearance in TCP modeling.

Processor Sharing Models for TCP

In the M/G/1-PS queue, see for instance Kleinrock (1976) or Ross (1996), jobs arrive according to a Poisson process at a server with service capacity L. The job sizes are independent, identically distributed with arbitrary distribution, but such that the queue is stable, i.e., the load $\rho = \lambda/(\mu L) < 1$ where $1/\mu$ is the mean job size. Contrary to a first-in-first-out buffer in which the server works on *one* job at a time until finished, a processor sharing server gives an equal share of its service capacity to each job in queue. In other words, when N jobs are in queue, each job receives a service capacity equal to L/N. Further, after each arrival or departure, the server immediately reallocates its capacity over the present jobs, if any.

Closed-form expressions exist for some quantities of the M/G/1-PS queue. We mention two. First, the steady-state probability that the number of jobs in system equals n at arrival times, hence at arbitrary times (PASTA), is $\rho^n(1-\rho)$. Second, the expected waiting time S, conditional on a job's initial size x, is

$$\mathbb{E}\{S \mid x\} = \frac{x}{L(1-\rho)}. \tag{1.49}$$

Interestingly, this expression is linear in the job size x. Moreover, it is *insensitive* to the job-size distribution (or higher moments), but merely involves $\mathbb{E}\{S\} = 1/\mu$. This is significant as it shows that first-order characteristics of the network as perceived by users do not depend on the job-size distribution. We remark that no closed-form expressions are known for higher moments of the conditional waiting time.

The M/G/1-PS model typically applies to a single bottleneck link shared by arriving and leaving TCP flows. Note that we abstract away from all details of the window level, but simply assume that the TCP achieves 'ideal' sharing and utilization of the link: the available bandwidth is always efficiently, instantaneously, and equally divided among the users. Based on (1.49) we can now obtain insight into transfer times for files of given size, given the load on the link.

Although PS models of TCP are useful at flow-level, they have also some shortcomings. In the first place we know that round-trip differences among connections change the share of service each source obtains. As in the PS queue all jobs in queue obtain the same fraction of the capacity, differences in round-trip times cannot be incorporated. A second somewhat problematic property of the PS model is that a single job uses all available capacity when it is the only one in the system. It is unrealistic that a single source can fill the bottleneck when this bottleneck happens to be a gigabit link in the core of the network. The maximal throughput of a connection is necessarily limited by external constraints such as the access network or the receiver window. Third, when a job leaves a PS queue, the server immediately redistributes its capacity over the remaining jobs. This is typically not the case in TCP networks as it may take several round-trip times for sources to become aware of the free capacity left behind by a departing flow.

Moreover, it is not possible to obtain any insight into the real utilization of the link by the TCP sources. Finally, Olsén (2003) points out that the PS model is so restrictive that it does not lend itself easily to investigate the influence of new features when these become implemented in the flow control algorithms of TCP.

Extensions of Processor Sharing Models

Discriminatory Processor Sharing (DPS) queues can cope with differences in round-trip time. A DPS queue has N job classes and associates to each job class a weight g_i, $1 \leq i \leq N$. Suppose that $L_i \equiv L_i(t)$ jobs of class i are in the system at time t. Then the server assigns a fraction $g_i / \sum_{j=1}^{N} g_j L_j$ of its capacity to job i. By making an appropriate choice for the weights g_i the model can account for round-trip time differences. Although the analysis of DPS is difficult in general, closed-form expressions exist for the expected number of jobs in class i, cf. Fayolle *et al.* (1980). Bu & Towsley (2001) use DPS to compute the throughput with the (hyper-)exponential service distributions. Roberts & Massoulié (2000) show with DPS that assigning weights to realize service differentiation, for instance in accordance with a pricing scheme as developed by Gibbens & Kelly (1999), has just slight influence on long-lived flows.

Another extension of the PS queue is the so-called Generalized Processor Sharing (GPS) queue, see Cohen (1979). (The term 'generalized processor sharing' queue is not only used for Cohen's extension to processor sharing. Parekh & Gallagher (1993) use GPS for another type of queue than we do here.) This queueing model enables to include external constraints on the window size, which have, as shown by Ben Fredj *et al.* (2001), a major influence on the obtained throughput. A GPS server uses a function $r(\cdot)$ such that when n jobs are in queue, it assigns each job an amount of service $r(n)/n$ instead of L/n as in the PS queue. The definition of $r(n)$ can take into account the external constraints. A common choice is $r(n) = \min\{L, nc\}$, where c is the maximal throughput of a constrained source. Note that the constraints for all connections are necessarily identical, which is somewhat artificial from a TCP-modeling point of view.

Vranken *et al.* (2002) apply a GPS model with two priority classes to study quality-of-service differentiation on the mean sojourn time of high- and low-priority flows. Lassila *et al.* (2003) develop a model that combines the GPS queue with a fixed point method. By the fixed point method they compute the throughput for n sources, i.e., $r(n)$, numerically for various values of n. This estimate for $r(n)$ is then used in the GPS queue to compute for instance the distribution of the number of flows present. Interestingly, Lassila *et al.* (2003) thus obtain this distribution as an *endogenous* property whereas Gibbens *et al.* (2000) and Avratchenkov *et al.* (2002) have to presuppose this distribution. When sources are subject to external constraints, it may be more accurate to model the packet arrivals as being modulated by an on/off process than as a homogeneous Poisson process. In this case De Haan *et al.* (2004) use on/off fluid sources with finite buffer, as analyzed by

Tucker (1988), in the fixed-point procedure of Lassila *et al.* (2003).

1.4 Contribution & Overview of This Thesis

In the larger part of this thesis we extend some existing theory on stochastic feedback fluid queues and apply this to modeling TCP networks. In retrospect we might even say that the desire to model TCP increasingly accurately and robustly in terms of feedback fluid queues led, in part, our research. The ordering of the topics of Chapters 2–6 reflects this.

In Chapter 2 we use the feedback fluid queue discussed in Section 1.1.5 to develop a window-level model of the interaction between a single TCP source and a drop-tail buffer. The typical TCP behavior is included as follows. We let the state of the background process correspond to the source window size and define the drift function as a linear function of the window size. When the buffer is *not* full, the window size increases as a pure birth process ('Additive-Increase') with constant transition rate, whereas when the buffer *is* full, the window decreases according to a second generator that implements 'Multiplicative-Decrease'.

The model of Chapter 3 extends the single-source model to two or more heterogeneous TCP sources. This fluid model is not a 'standard' feedback fluid model in the sense of Section 1.1.5. In particular, we augment the joint window-buffer process with indicator variables to implement the synchronous loss model discussed in Section 1.3.2. Van Foreest *et al.* (2001) and Van Foreest *et al.* (2003a) provide the contents of this and the previous chapter.

In the TCP models of Chapter 2 and 3 the transition rates of the background processes are *constant and inversely proportional to the average round-trip times*. As we use only one generator for the background process while the buffer is not full, we cannot include queueing delay contributions to the round-trip times. This is an obvious shortcoming of the model when studying the influence of large buffers on network performance. Ideally, as the round-trip times are continuous functions of the queueing delays, the generators of the source processes should also depend *continuously* on the queueing delay, i.e., the buffer content. This, however, requires a much more flexible type of feedback than has been considered previously.

In Chapter 4 we generalize the stochastic feedback fluid queue of Chapters 2–3 such that the momentary behavior of the background process depends *continuously* on the actual buffer level. Loosely speaking the feedback is such that when the buffer level is y, the background process behaves 'as a Markov process' governed by generator $Q(y)$ with entries being continuous functions of y. Moreover, the drift function may also depend continuously on y. We refer to this type of fluid queue as a *continuous feedback fluid queue*, cf. Section 1.1.5. We establish an explicit solution when the number N of source

states is 2, and a numerical method when $N > 2$. Part of the contents of this chapter is to appear in Scheinhardt *et al.* (2005).

As it turns out, the numerical procedures to solve these feedback fluid models for TCP prove rather unstable, including the TCP model based on continuous feedback fluid queues. In Chapter 5 we circumvent this problem by discretizing the buffer content process and approximating the TCP fluid model by a (continuous-time) Markov chain. With this chain we study the influence of the synchronous and proportional loss model of TCP and compare the results to simulations with ns-2. Our method proves simple, yet flexible, and numerically robust. In addition to this, it extends, in principle, to networks of intermediate size, i.e., a few sources and buffers. The results of this chapter have been published by Van Foreest *et al.* (2003b).

Besides its strong points, the Markovian model of Chapter 5 is somewhat un-practical in use as the computer code to produce the generator matrix has to be implemented by hand, i.e., by the author. Besides being time consuming, it is difficult to obtain the generator for intermediately sized networks. A more suitable methodology to specify the Markov chain is provided by Stochastic Petri Nets (SPNs), cf. Ajmone Marsan *et al.* (1995). Once the SPN is implemented, the generator can be obtained automatically, thereby saving a considerable amount of work.

In Chapter 6, which is based on Van Foreest *et al.* (2004a), we use SPNs to compare performance results from our TCP model to results from the literature such as Altman *et al.* (2000b) and simulations by ns-2. Moreover, we consider fairness aspects in a *network* consisting of three sources and two buffers, and relate the results to work of Massoulié & Roberts (1999) and Lee *et al.* (2001). This chapter concludes our investigations of TCP.

Up to now we have been concerned with feedback from the buffer process to the source process, culminating in the continuous fluid queue of Chapter 4. A second type of feedback is useful in queueing *networks*. Besides informing the source about the buffer contents, the buffers can also signal 'upstream' servers to in- or decrease the service rates. The analysis of feedback networks seems exceedingly difficult. Probably one of the simplest such networks is a tandem of two $M/M/1$ queues in which the second buffer informs the first server about the queue length. In Chapter 7 we consider two variants of a two-station tandem network with blocking. In both variants the first server ceases to work when the queue length at the second station hits a 'blocking threshold'. In addition, in variant 2 the first server decreases its service rate when the second queue exceeds a 'slow-down threshold', which is smaller than the blocking level. In both variants the arrival process is Poisson and the service times at both stations are exponentially distributed. Note, however, that in case of slow-downs, server 1 works at a high rate, a slow rate, or not at all, depending on whether the second queue is below or above the slow-down threshold or at the blocking threshold, respectively. For variant 1, i.e., only blocking, we concentrate on the decay rate of the number of jobs in the first buffer and prove that for

increasing blocking thresholds the sequence of decay rates decreases monotonically and at least geometrically fast to $\max\{\rho_1, \rho_2\}$, where ρ_i is the load at server i. The methods used in the proof also allow us to clarify the asymptotic queue length distribution at the second station. Then we generalize the analysis to variant 2, i.e., slow-down and blocking, and establish similar results. The results of this chapter are based on Van Foreest *et al.* (2004b).

Chapter 2

A Feedback Fluid Model for a Single TCP Source

In the previous, introductory, chapter we discussed fluid queues and TCP as separate topics. In this chapter we combine these by developing a feedback fluid model of a single TCP source using a bottleneck buffer. The feedback from the buffer to the source is such that the model captures some of the more obvious aspects of TCP source behavior. When the buffer is not full it sends positive signals to the source to indicate that the source can increase the sending rate. During overflow the buffer sends negative signals to reduce the source rate. The TCP source model we use here is a greedy AIMD fluid source, cf. Section 1.3.2 and the fluid buffer is of a drop-tail type.

The structure of the chapter is as follows. In Section 2.1 we explain the model. Then, in Section 2.2, we derive the differential equations governing the source and buffer dynamics, and the boundary conditions. This section borrows from the work of Scheinhardt (2001) and Mandjes *et al.* (2003a), but here we additionally find a closed-form expression for the stationary distribution. The performance analysis is presented in Section 2.3.

2.1 Model

Let us start with describing the feedback fluid model. At the end of the section we relate this model to the behavior of a TCP source.

Consider a single source that transmits fluid into a bottleneck buffer served at rate L. The source state is a stochastic process $\{W(t)\} \equiv \{W(t), t \geq 0\}$ and has state space $\mathscr{W} := \{1, 2, \ldots, N\}$. The buffer content process, denoted by $\{C(t)\} \equiv \{C(t), t \geq 0\}$, is also a stochastic process. We require that $C(t) \in [0, B]$, where, importantly, $B < \infty$. The process $\{W(t)\}$ controls the source's output rate: when $W(t) = i$ the source sends

fluid at rate i times r into the buffer, where r is some positive constant. As a consequence, the buffer dynamics satisfy the differential equation, compare (1.47c),

$$\frac{dC(t)}{dt} = \begin{cases} \max\{rW(t), 0\}, & \text{if } C(t) = 0, \\ r\,W(t) - L, & \text{if } C(t) \in (0, B), \\ \min\{rW(t), 0\}, & \text{if } C(t) = B. \end{cases} \tag{2.1}$$

Thus, the drift matrix R has the form

$$R := \begin{pmatrix} r - L & & & \\ & 2r - L & & \\ & & \ddots & \\ & & & Nr - L \end{pmatrix}. \tag{2.2}$$

To avoid trivialities we suppose that

$$r < L < Nr; \tag{2.3}$$

the first inequality ensures that the buffer is not in a permanent state of overload, the second that occasionally congestion occurs.

Whereas the above equation (2.1) determines the dependence of the buffer content on the source state, the buffer controls the source by sending positive and negative feedback signals. While the buffer is not full, i.e., $C(t) < B$, the buffer sends positive signals to the source to indicate the successful transfer of data. When the source receives this feedback it increases its rate, i.e., $W(t)$ increases by one to $W(t) + 1$ (except when $W(t) = N$). When the buffer becomes congested, i.e., $C(t) = B$, it sends negative signals to the source to notify that fluid is discarded. As a response, the source decreases its rate by half, that is, when $W(t) > 1$ the state becomes $\lfloor W(t)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x. If $W(t) = 1$ the source does not decrease further.

The time intervals between two consecutive positive or negative signals are assumed to be independent, exponentially distributed random variables with parameter λ and μ, respectively. Thus, when $C(t) < B$, the source state is determined by the positive signals, leading to a generator Q of the form

$$Q := \begin{pmatrix} -\lambda & \lambda & & & \\ & -\lambda & \lambda & & \\ & & \ddots & \ddots & \\ & & & -\lambda & \lambda \\ & & & & 0 \end{pmatrix}. \tag{2.4}$$

When $C(t) = B$, the negative signals control the source behavior by the generator $\widetilde{Q}$. As an example we show $\widetilde{Q}$ for a source with 5 states:

$$\widetilde{Q} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \mu & -\mu & 0 & 0 & 0 \\ \mu & 0 & -\mu & 0 & 0 \\ 0 & \mu & 0 & -\mu & 0 \\ 0 & \mu & 0 & 0 & -\mu \end{pmatrix}. \tag{2.5}$$

The general structure of $\widetilde{Q}$ may be written in terms of Kronecker's δ, i.e., $\delta_{ij} = 1$ if $i = j$ and 0 otherwise:

$$\widetilde{Q}_{ij} := \mu \left(-\delta_{ij} + \delta_{i,2j} + \delta_{i,2j+1} + \delta_{i1}\delta_{j1}\right), \quad 1 \le i, j \le N. \tag{2.6}$$

It is perhaps of interest to provide some further illustration of the interaction between the buffer and the source. To this end, consider first the lower part of Figure 2.1. The source, with $r = 1$ and $N = 5$, increases its transmission rate every time it receives a positive signal from the buffer. The inter-arrival time between these signals is exponentially distributed with parameter λ. In fact, its behavior is governed by the generator Q. At the very moment the buffer overflows—suppose this happens in state 4—another generator $\widetilde{Q}$ becomes active. This change in generator is indicated by a switch from the lower to the upper part of the figure. The source waits in this state for an exponentially distributed time with parameter μ, and then jumps to state 2. As L is taken to be 1.5 in this example, the buffer is still in overflow when the source enters state 2, and the source has to wait for another negative feedback signal. When this is received, the source halves its state index for a second time, that is, it jumps to state 1. Now the drift is negative, and the buffer is no longer full. Consequently, the generator switches from $\widetilde{Q}$ to Q so that the buffer can start sending positive signals again.

From the above, the source and buffer behavior are clearly dependent. The source process influences the buffer content, whereas the buffer process determines the source rate by sending positive and negative feedback signals. Consequently, the source process $\{W(t)\}$ is *not* a Markov-process. However, the joint process $\{W(t), C(t), t \ge 0\}$ *is* a (multivariate) Markov process.

We now provide an interpretation of the above model in terms of TCP. First, Q and $\widetilde{Q}$ implement Additive-Increase and Multiplicative-Decrease, respectively. Second, the highest source state N can be seen as an external constraint, such as the receiver window or the speed of the access link, on the source rate. Third, $W(t) \ge 1$ so that the source always has some fluid to send, i.e., it is greedy. Fourth, the time between two consecutive positive or negative signals models the sum of the transmission, propagation and queueing delays of the packets at other routers in the network, including random delays of the operating systems at the sender and receiver. In other words, when the buffer is not full

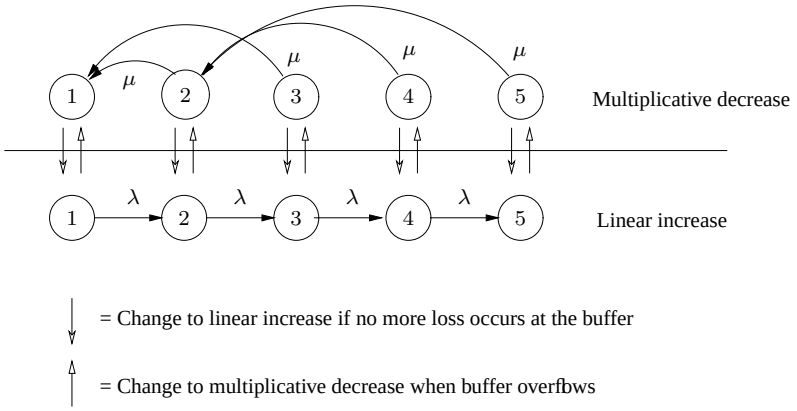

Figure 2.1: *Behavior of a source with* $N = 5$ *states,* $L = 1.5$ *and* $r = 1$.

we set

$$\lambda^{-1} := T, \tag{2.7}$$

where T is the average round-trip time when the bottleneck buffer is empty. Considering the value of μ, observe that the average round-trip time is $T + B/L$ when the buffer is full (compare (1.47a)); hence, let

$$\mu^{-1} := T + \frac{B}{L}. \tag{2.8}$$

Finally, r relates directly to the notion of packet size. To see this, notice that when the window size is W and the packet size is P (in bytes), a TCP source sends PW bytes per window, cf. (1.47d). Our fluid source sends, when $W(t) = i$, on average an amount of ir/λ fluid in between two positive signals, i.e., per round-trip time. Since $W(t)$ equals the window size, we have that $PW = Wr/\lambda$, and therefore r/λ should correspond to one packet size in bytes. Clearly, in this derivation we use the expected time between two positive signals, i.e., $1/\lambda$. However, we might as well take the time between two *negative* signals. In that case the packet size corresponds to r/μ rather than r/λ. This, perhaps somewhat unsatisfactory, conclusion is a direct consequence of modeling a packet source as a fluid source. Rather than trying to compensate for this aspect, we accept it as one of the model's idiosyncrasies. Hence, we define the packet size as

$$P := \begin{cases} r \,/\, \lambda, & \text{if } C < B, \\ r \,/\, \mu, & \text{if } C = B. \end{cases} \tag{2.9}$$

Remark 2.1. Observe that the fluid approximation presumes a separation of time scales, i.e., the inter-arrival time of packets is small compared with the time between changes in

the source rate. To see this, suppose that the window size is 10, say. Then, packets arrive ten times as often as a change in the source rate occurs.

Remark 2.2. As mentioned, Q and $\tilde{Q}$ implement Additive-Increase and Multiplicative-Decrease, respectively. Clearly, the structure of $\tilde{Q}$ is such that the source makes only *one* transition downward per round-trip time. Thus, the source should use a TCP version, such as TCP NewReno or TCP Sack, that does not frequently resort to timeouts and slow starts when multiple losses occur in one window, cf. Section 1.2.3.

2.2 Analysis

In this section we analyze the behavior of the process $\{W(t), C(t)\}$. More specifically, in Section 2.2.2 we derive the Kolmogorov forward equations. These equations reduce to a stationary system of ordinary differential equations subject to several boundary conditions. Finding these two-point boundary conditions is the topic of Section 2.2.3. In Section 2.2.4 we derive the solution. We start with introducing some notation in Section 2.2.1.

Since this model is similar to the standard fluid queue when $C(t) < B$, the derivation presented here is rather brief. We refer to Section 1.1 for details left out here.

2.2.1 Notation

The source state $W(t)$ *ascends* when $C(t) < B$ and $W(t) < N$, whereas it *descends* when $C(t) = B$ and $W(t) > 1$. To reflect this difference in behavior we split the system state space $\mathscr{S} := \mathscr{W} \times [0, B]$ into two disjoint subsets $\mathscr{T} := \mathscr{W} \times [0, B)$, and $\widetilde{\mathscr{T}} := \mathscr{W} \times \{B\}$.

On $\mathscr{T}$ we define functions

$$A_i(y,t) := \mathbb{P}\{W(t) = i, C(t) \le y\}, \quad 0 \le y < B, \tag{2.10}$$

and on $\widetilde{\mathscr{T}}$

$$D_i(t) := \mathbb{P}\{W(t) = i, C(t) = B\}.$$

Since the domain of the function $A(\cdot, t)$ is the interval $[0, B)$, we define on the boundaries:

$$A_i(B,t) := \lim_{y \uparrow B} A_i(y,t), \qquad \frac{\partial A_i(B,t)}{\partial y} := \lim_{y \uparrow B} \frac{\partial A_i(y,t)}{\partial y}.$$

Clearly, $\mathbb{P}\{W(t) = i\} = A_i(B,t) + D_i(t)$. To avoid possible confusion later in the chapter, we set $A_i(y,t) = D_i(t) = 0$ if $i \notin \mathscr{W}$ and $y \notin [0, B]$.

Since we explicitly require that $r < L < Nr$, the source rate alternates between periods with sending rates higher than L and periods with sending rate lower than L.

To distinguish between the under- and overload states, we define two subsets of $\mathscr{W}$: $\mathscr{W}_- := \{i \in \mathscr{W} \,|\, ir < L\}$ and $\mathscr{W}_+ := \{i \in \mathscr{W} \,|\, ir > L\}$. For technical reasons we prefer to choose $L/r \notin \mathscr{W}$, that is, L is not an integer multiple of r. (Consult, e.g., Mitra (1988) or Sericola & Tuffin (1999) on how to handle systems for which $L/r \in \mathscr{W}$.) Consequently, $\mathscr{W} = \mathscr{W}_+ \cup \mathscr{W}_-$ and $\mathscr{W}_- \cap \mathscr{W}_+ = \varnothing$. Furthermore we set $N_- := |\mathscr{W}_-|$, $N_+ := |\mathscr{W}_+|$, and $N := |\mathscr{W}| = N_- + N_+$.

Finally we need vectors

$$\begin{aligned}\mathbf{A}(y,t) &:= (A_1(y,t), \ldots, A_N(y,t)),\\ \mathbf{D}(t) &:= (D_1(t), \ldots, D_N(t)),\end{aligned}$$

and the restrictions of these vectors to the regions $\mathscr{W}_-$ and $\mathscr{W}_+$. Hence, dropping the arguments y and t, let

$$\begin{aligned}\mathbf{A}_- &:= (A_1, \ldots, A_{N_-}),\\ \mathbf{A}_+ &:= (A_{N_-+1}, \ldots, A_N),\end{aligned}$$

and so on.

2.2.2 Kolmogorov Forward Equations

Introduce the shorthand $r_i := ir - L$, fix $y \in (0, B)$ and take h so small that also $y - r_i h \in (0, B)$. Then we find

$$\begin{aligned}A_i(y, t+h) &= \lambda h A_{i-1}(y,t) + \\ &\qquad (1-\lambda h) A_i(y - r_i h, t) + o(h), \quad 1 < i < N,\\ A_1(y, t+h) &= (1-\lambda h) A_1(y - r_1 h, t) + o(h)\\ A_N(y, t+h) &= \lambda h A_{N-1}(y,t) + A_N(y - r_N h, t) + o(h).\end{aligned}$$

Expanding the second term on the right hand side of the first equation yields

$$A_i(y, t+h) = \lambda h A_{i-1}(y,t) + (1-\lambda h)\left[A_i(y,t) - r_i h \frac{\partial A_i(y,t)}{\partial y}\right] + o(h).$$

By collecting terms, dividing by h, and taking the limit $h \to 0$ we find in matrix form the Kolmogorov forward equations for the (row) vector $\mathbf{A}(y,t)$:

$$\frac{\partial \mathbf{A}(y,t)}{\partial t} = \mathbf{A}(y,t)Q - \frac{\partial \mathbf{A}(y,t)}{\partial y} R, \tag{2.11}$$

where Q is defined in (2.4) and R in (2.2). For completeness' sake we remark that we do not specify the behavior of $\mathbf{A}(y,t)$ on the boundary $y = 0$. The details are easy to provide, cf. (1.11–1.12).

The derivation of the differential equations for $\mathbf{D}$ is more involved. Observe that, when $i \in \mathscr{W}_+$, there is a net probability flux *into* $\widetilde{\mathscr{T}}$ from 'below', i.e., from $\mathscr{T}$ due to overflows of the buffer, and from 'above', i.e., from higher states D_{2i} and D_{2i+1} into D_i, due to multiplicative decrements of the source state. Thus, for sufficiently small h and $i \in \mathscr{W}_+$,

$$\begin{aligned} D_i(t+h) &= (1-\mu h)D_i(t) + \mu h[D_{2i}(t) + D_{2i+1}(t)] \\ &\quad + (1-\lambda h)\mathbb{P}\{W(t) = i, B - r_i h < C(t) < B\} + o(h) \\ &= D_i(t) + \mu h[D_{2i}(t) + D_{2i+1}(t) - D_i(t)] \\ &\quad + (1-\lambda h)[A_i(B,t) - A_i(B - r_i h, t)] + o(h). \end{aligned}$$

(Recall $D_i(t) \equiv 0$, if $i \notin \mathscr{W}$.) On $\mathscr{W}_-$ there is a flux from $\widetilde{\mathscr{T}}$ to $\mathscr{T}$, and the term $A_i(B - r_i h, t)$ in the above equation should be replaced by $A_i(B + r_i h, t)$ since now $r_i < 0$. Presently, in Section 2.2.3, we require that $\mathbf{D}_-(t) \equiv 0$ implying that when $i \in \mathscr{W}_-$ the fluxes from 'above' and 'below' should match. Therefore the above relation should be replaced by, for $i \in \mathscr{W}_-$,

$$\begin{aligned} \mu h(D_{2i}(t) + D_{2i+1}(t)) &= A_i(B,t) - A_i(B + r_i h, t) + o(h) \\ &= -r_i h \frac{\partial A_i(B,t)}{\partial y} + o(h). \end{aligned}$$

Now we follow the same procedure as for $\mathbf{A}$, (collect terms, etc.) to obtain the forward equations for $\mathbf{D}$:

$$\frac{d\mathbf{D}(t)}{dt} = \mathbf{D}(t)\widetilde{Q} + \frac{\partial \mathbf{A}(B,t)}{\partial y} R, \tag{2.12}$$

with $\widetilde{Q}$ as in (2.6) and $D_i(t) = 0$ if $i \notin \mathscr{W}$.

Note that $\mathbf{D}(t)$ does not appear in the derivation for $\mathbf{A}(y,t)$ leading to (2.11), whereas $\partial_y \mathbf{A}(B,t)$ does 'contribute' to $\mathbf{D}(t)$. The reason is simply that the derivation for $\mathbf{A}(y,t)$ applies to the *half open* interval $[0, B)$, so that there cannot be a contribution of boundary terms at $C = B$. In fact, to clarify further the exchange of probability flux between $\mathscr{T}$ and $\widetilde{\mathscr{T}}$ at the boundary $C = B$, we add (2.11) and (2.12) at $C = B$ to obtain

$$\frac{\partial \mathbf{A}(B,t)}{\partial t} + \frac{d\mathbf{D}(t)}{dt} = \mathbf{D}(t)\widetilde{Q} + \mathbf{A}(B,t)Q. \tag{2.13}$$

Now we see that $\mathbf{A}$ and $\mathbf{D}$ satisfy a conservation principle similar to the forward equations of a continuous-time Markov process. As (2.12) has a less familiar form than the equivalent equation (2.13), we prefer to work with the latter rather than the former.

2.2.3 The Stationary System and Boundary Conditions

In the sequel we are interested in the system in steady state, and write for brevity W and C for $W(t)$ and $C(t)$ at an arbitrary point in time. The stationary solution of (2.11)

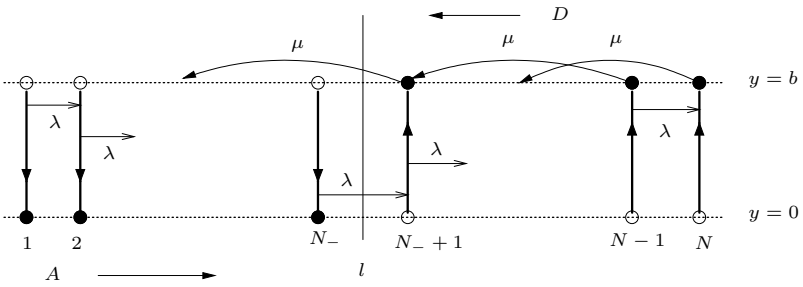

Figure 2.2: *The figure shows the state space and the boundary conditions of* $\mathbf{A}(y)$ *and* $\mathbf{D}$*. The line* $l = \{(x, y) \,|\, xr = L\}$ *splits* $\mathscr{S}$ *into* $\mathscr{W}_- \times [0, B]$ *and* $\mathscr{W}_+ \times [0, B]$*. The open circles at the points* $(i, 0)$ *for* $i \in \mathscr{W}_+$ *indicate that* $A_i(0) = 0$*. Likewise we have that* $D_i = 0$ *when* $i \in \mathscr{W}_-$ *which is shown by the open circles at* (i, B)*. Furthermore the atoms* $A_i(0) > 0$*, when* $i \in \mathscr{W}_-$*, and* $D_i > 0$*, when* $i \in \mathscr{W}_+$ *respectively, are depicted by a bullet.*

and (2.13) satisfies $\partial_t \mathbf{A}(y, t) \equiv \partial_t \mathbf{D}(t) \equiv \mathbf{0}$ allowing us to drop the dependence on t in the sequel. It follows that (2.11) reduces to a set of ordinary linear differential equations

$$\frac{d\mathbf{A}(y)}{dy} R = \mathbf{A}(y) Q, \tag{2.14a}$$

and (2.13) yields the balance equations

$$\mathbf{D}\widetilde{Q} + \mathbf{A}(B) Q = \mathbf{0}. \tag{2.14b}$$

$\mathbf{A}(y)$ and $\mathbf{D}$ have to satisfy some obvious boundary conditions. In fact, we observe that the buffer cannot be full (empty) when $W \in \mathscr{W}_-$ ($W \in \mathscr{W}_+$), whence

$$\mathbf{D}_- = \mathbf{0} \qquad \text{and} \qquad \mathbf{A}_+(0) = \mathbf{0}. \tag{2.15}$$

Figure 2.2 shows an overview of the state space $\mathscr{S}$ with the boundary conditions and the source transitions.

We now show that the number of conditions matches the number of unknowns. Concerning the unknowns, observe that $\mathbf{A}(y)$ is a solution of the N-dimensional system of differential equations (2.14a), and, hence, involves N coefficients. Second, the vector of atoms $\mathbf{D}$ has N components. On the other hand, it is evident that (2.15) yields exactly N conditions. Moreover, Lemma 2.3 below shows that the balance equations (2.14b) provide $N - 1$ conditions. Finally, the requirement that $\sum_i (A_i(B) + D_i) = 1$ fixes a scaling. Thus, the number of conditions is also equal to $2N$ so that a well-defined system remains.

Lemma 2.3. *The number of conditions implied by the equation*

$$\mathbf{A}(B)Q + \mathbf{D}\widetilde{Q} = \mathbf{0}, \tag{2.16}$$

is equal to $N - 1$.

Proof. In the first place we notice that a solution of the differential equations (2.14a) cannot be trivial when it also satisfies the condition $\sum_i (A_i(B) + D_i) = 1$. This implies

$$\mathbf{A}(B)Q \neq \mathbf{0} \qquad \text{and} \qquad \mathbf{D}\widetilde{Q} \neq \mathbf{0}.$$

Second, we write (2.16) as

$$(\mathbf{A}(B), \mathbf{D}) \begin{pmatrix} Q \\ \widetilde{Q} \end{pmatrix} = \mathbf{0},$$

i.e., we stack the entries Q and $\widetilde{Q}$ into one matrix. From (2.4) and (2.6) it follows that the columns of this matrix are linearly independent, except for the left most which is equal to (minus) the sum of the other columns. Thus the rank of this matrix is $N - 1$. ∎

2.2.4 Solving the Stationary System

Now we solve the system of differential equations (2.14a) together with the balance equations (2.14b) such that the boundary conditions (2.15) are satisfied. First we reduce (2.14a) to an N-dimensional eigenvalue problem and establish explicit expressions for the eigenvalues and corresponding left eigenvectors. The second step relates the eigensystem to the balance equations, includes the boundary conditions, and, finally, provides the solution to the entire system.

The Eigenvalues and Eigenvectors

The system of ordinary differential equations (2.14a) can be written as

$$\frac{\partial \mathbf{A}(y)}{\partial y} = \mathbf{A}(y) Q R^{-1},$$

where the inverse of R exists as, by assumption, $L \neq ir$ for any $i \in \mathscr{W}$. The eigenvalues θ_i follow immediately from the upper-triangularity of QR^{-1}: they simply form the diagonal, so that

$$\theta_i = \frac{\lambda}{L - ir}, \quad 1 \leq i < N,$$

and $\theta_N = 0$. Obviously, N_- eigenvalues are positive, and $N_+ - 1$ are negative. These eigenvalues can be given an interesting interpretation. Recall that the average round-trip time is $1/\lambda$. If the source is in state i, the increase or decrease of the buffer during one

round-trip time will be, on average, $(ir - L)/\lambda$. Observe that Theorem 1.4 does not hold here, since (1.20) is not valid with $(0, 0, \ldots, 1)$ being the left null vector of Q.

The eigenvalues are distinct so that the general solution for $\mathbf{A}$ is of the form

$$\mathbf{A}(y) = \sum_{i=1}^{N} a_i \, \mathbf{v}_i e^{\theta_i \, y},$$

where the row vectors $\mathbf{v}_i$ are the left eigenvectors of the matrix QR^{-1} associated with the eigenvalue θ_i, i.e., $\theta_i \mathbf{v}_i = \mathbf{v}_i QR^{-1}$. The solution $\mathbf{A}(y)$ can be written succinctly in matrix form. To this end, let $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_N)$ be the vector of eigenvalues,

$$\Theta(y) := \exp\left(\mathrm{diag}(\boldsymbol{\theta})\, y\right), \tag{2.17}$$

and Φ the matrix of eigenvectors, i.e.,

$$\Phi := \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_N \end{pmatrix}.$$

Now $\mathbf{A}(y)$ becomes

$$\mathbf{A}(y) = \mathbf{a}\Theta(y)\Phi, \tag{2.18}$$

where the coefficients vector $\mathbf{a}$ contains the N unknowns which were to be found in Section 2.2.3.

To determine the eigenvectors, we notice that, again because of the upper-triangularity of QR^{-1}, the j-th component of the eigenvector $\mathbf{v}_i$ should be zero when $j < i$, i.e., $\phi_{ij} = 0$ if $j < i$. The other components of ϕ_i can be found from a recursion which follows from the fact that the only non-zero entries of QR^{-1}, besides the diagonal, appear one above the diagonal. Hence, for $\mathbf{v}_i$ and $i < j < N$,

$$\phi_{ij} = \frac{1}{j-i}\frac{L - ir}{r}\phi_{i,j-1} = \frac{1}{(j-i)!}\left(\frac{L-ir}{r}\right)^{j-i}\phi_{ii}, \tag{2.19}$$

and when $i < j = N$,

$$\phi_{iN} = \frac{1}{(N-i-1)!}\left(\frac{L-ir}{r}\right)^{N-i}\frac{r}{Nr-L}\phi_{ii}.$$

The numbers ϕ_{ii} can be chosen arbitrarily as the coefficients vector $\mathbf{a}$ ensures that the boundary conditions be met. We therefore set $\phi_{ii} = 1$, $1 \leq i < N$. Furthermore, $\theta_N = 0$ so that defining $\phi_{NN} = 1$ is allowed. So, the matrix Φ is upper-triangular with 1's on the diagonal. Note that the eigenvectors are all linearly independent and, thus, span the eigenspace of our boundary value problem for all $y \in (0, B)$.

Finding $\mathbf{A}(y)$ and $\mathbf{D}$

It remains to express $\mathbf{a}$ and $\mathbf{D}$ in terms of the boundary conditions (2.15) and the balance equations (2.14b). We start with partitioning Φ according to

$$\Phi = \left(\begin{array}{c|c} \Phi^{--} & \Phi^{-+} \\ \hline \Phi^{+-} & \Phi^{++} \end{array}\right).$$

Here Φ^{--} is the $N_- \times N_-$ upper left corner of Φ, Φ^{+-} is the $N_+ \times N_-$ lower left corner, and so on. Clearly, $\Phi^{+-} \equiv 0$, since Φ is upper triangular. Furthermore Φ^{--} and Φ^{++} are invertible, as is apparent from the fact that $\Phi_{ii} = 1$ for $1 \leq i \leq N$ and, again, by upper triangularity. We partition the diagonal matrix $\Theta(y)$ of eigenvalues accordingly:

$$\Theta(y) = \begin{pmatrix} \Theta^{--}(y) & 0 \\ 0 & \Theta^{++}(y) \end{pmatrix}.$$

With these partitions we can use the condition $\mathbf{A}_+(0) = \mathbf{0}$ to express $\mathbf{a}_+$ in terms of $\mathbf{a}_-$. From (2.18), and noting that $\Theta(0)$ is the identity,

$$\begin{aligned} \mathbf{0} &= [\mathbf{a}\Theta(0)\Phi]_+ \\ &= \mathbf{a}_-\Phi^{-+} + \mathbf{a}_+\Phi^{++}, \end{aligned}$$

from which

$$\mathbf{a}_+ = -\mathbf{a}_-\Phi^{-+}(\Phi^{++})^{-1}.$$

Now we can write $\mathbf{A}(B)$ in terms of $\mathbf{a}_-$. To this end, introduce the $N_- \times N_+$ matrix

$$\Psi = \left(\Theta^{--}(B)\right)^{-1}\Phi^{-+}\left(\Phi^{++}\right)^{-1}\Theta^{++}(B),$$

so that

$$\begin{aligned} \mathbf{A}(B) &= (\mathbf{a}_-, \mathbf{a}_+)\Theta(B)\Phi \\ &= \mathbf{a}_-\Theta^{--}(B)(\Phi^{--} - \Psi\Phi^{+-}, \Phi^{-+} - \Psi\Phi^{++}) \\ &= \mathbf{a}_-\Theta^{--}(B)(\Phi^{--}, \Phi^{-+} - \Psi\Phi^{++}), \end{aligned} \tag{2.20}$$

where the last step follows from the fact that $\Phi^{+-} \equiv 0$. The advantage of introducing the matrix Ψ is that both $(\Theta^{--})^{-1}(B)$ and $\Theta^{++}(B)$ contain (very) small entries. Moreover, although $\mathbf{a}_-$ can have very small entries, e.g., 10^{-20}, the entries of the vector $\mathbf{a}_-\Theta^{--}(B)$ are in comparison roughly of order 1. The numerical analysis becomes much stabler when we compute the latter vector instead of $\mathbf{a}_-$ itself.

With the boundary condition $\mathbf{D}_- = \mathbf{0}$ and the above expression for $\mathbf{A}(B)$ we solve for $\mathbf{a}_-\Theta^{--}(B)$ and $\mathbf{D}_+$ with the balance equation $\mathbf{A}(B)Q + (\mathbf{0}, \mathbf{D}_+)\widetilde{Q} = \mathbf{0}$. This final equation is of the form

$$(\mathbf{a}_-\Theta^{--}(B), \mathbf{D}_+)M = \mathbf{0} \tag{2.21}$$

for some matrix M. The proper expressions for $\mathbf{A}(B)$ and $\mathbf{D}$ can now be found with (2.20) and the scaling

$$\sum_i (A_i(B) + D_i) = 1.$$

2.3 Results

We first define various steady-state performance measures of interest. Then we present in Section 2.3.2 analytical results for a single source with two states. For systems with more states the analytical expressions soon become unwieldy. Therefore we use numerical methods to investigate larger models in Section 2.3.3. The validation of the root p law (1.40) is the topic of Section 2.3.4.

2.3.1 Performance Measures

The source's instantaneous transmission rate at time t is equal to $r\,W(t)$. Its *average transmission rate* up to time T is therefore

$$\tau(T) := \frac{r}{T} \int_0^T W(t)dt.$$

In the limit this becomes

$$\tau := \lim_{T\to\infty} \tau(T) = r \sum_{i\in\mathscr{W}} i\,\mathbb{P}\{W = i\} = r\,\mathbb{E}\{W\}; \tag{2.22}$$

thus, the expectation is taken with respect to the distribution of $\{W, C\}$.

The *throughput*, often indicated as 'goodput', is the data volume arrived at and acknowledged by the destination during the interval $[0, T]$. Thus, the throughput is in general less than $\tau(T)$ due to packet loss. In this fluid model we suppose that all traffic sent while $C(t) < B$ arrives at the destination, whereas all traffic sent *in excess* of the link rate is lost at times when $C(t) = B$. Thus, the average throughput of the source up to time T is

$$\gamma(T) := \tau(T) - \int_0^T (rW(t) - L)\,1_{\{C(t)=B\}}dt,$$

where 1_A is the indicator function: $1_A = 1$ if the event A is true, and $1_A = 0$ otherwise. In the limit this becomes

$$\begin{aligned}\gamma &= \lim_{T\to\infty} \gamma(T) = \tau - \sum_{i\in\mathscr{W}} (ir - L)D_i = \tau - \mathbb{E}\left\{(rW - L)1_{\{C=B\}}\right\} \\ &= r \sum_{i\in\mathscr{W}} iA_i(B) + L \sum_{i\in\mathscr{W}} D_i.\end{aligned} \tag{2.23}$$

The boundary condition $\mathbf{D}_- = \mathbf{0}$ specified in (2.15) implies that $\mathbb{P}\{C = B\} = \mathbb{P}\{C = B, rW > L\}$; thus, we do not subtract negative quantities in this definition.

To facilitate a comparison of systems with different link rate L we define the (unitless) *utilization* of the link as:

$$u = \frac{\gamma}{L}.$$

Finally, the stationary distribution of the *buffer content* is given by

$$\mathbb{P}\{C \leq y\} = \sum_i A_i(y) + \sum_i D_i 1_{\{y=B\}}.$$

i.e., the *long run fraction of time* the buffer content is less than or equal to y.

2.3.2 Analytic Results for a Two-state Source

We now concentrate on a background process with only two states.

Assuming $r < L < 2r$, straightforward calculations yield the probability of finding the system in state 1:

$$\mathbb{P}\{W = 1\} = A_1(B) = \frac{\mu}{\lambda+\mu}\left[1 + \frac{\mu}{\lambda+\mu}\frac{L-r}{2r-L}\left(1 - e^{-B\theta_1}\right)\right]^{-1},$$

with $\theta_1 = \lambda/(L-r)$. The fraction of time the buffer is congested is (recall $D_1 \equiv 0$)

$$\begin{aligned} D_2 &= \frac{\lambda}{\mu} A_1(B) \\ &= \frac{\lambda}{\lambda+\mu}\left[1 + \frac{\mu}{\lambda+\mu}\frac{L-r}{2r-L}\left(1 - e^{-B\theta_1}\right)\right]^{-1}. \end{aligned}$$

For $\mathbf{A}(y)$ we obtain:

$$\begin{aligned} \mathbf{A}(y) &= (a_1, a_2)\Theta(y)\Phi \\ &= A_1(B)\left(e^{\theta_1(y-B)}, \frac{L-r}{2r-L}\left[e^{\theta_1(y-B)} - e^{-\theta_1 B}\right]\right). \end{aligned}$$

An interesting limiting case is $B = 0$. Then $A_1(B) = \mu/(\lambda+\mu)$ and $D_2 = \lambda/(\lambda+\mu)$. Clearly, this is the stationary distribution of a two-state Markov chain with exponential holding times with parameters λ and μ, respectively. This behavior is to be expected: from the boundary conditions (2.15) we have that $A_2(0) = 0$ implying that as soon as the source makes a transition from state 1 to 2, the buffer will be full.

It is seen that $A_1(B)$ and D_2 decrease monotonically as functions of B, and in the limit

$$\lim_{B\to\infty} A_1(B) = \frac{\mu}{\lambda+\mu}\left[1 + \frac{\mu}{\lambda+\mu}\cdot\frac{L-r}{2r-L}\right]^{-1}.$$

The throughput (2.23) becomes

$$\begin{aligned}\gamma &= rA_1(B) + 2rA_2(B) + L\,D_2 \\ &= r\,[A_1(B) + 2(1 - A_1(B) - D_2)] + L\,D_2 \\ &= 2r + \left((1-2r)\,L\,\frac{\lambda}{\mu} - r\right) A_1(B).\end{aligned}$$

2.3.3 Numerical Results for a Source with more than Two States

Before presenting the results we discuss some aspects of the numerical analysis itself. Replacing the analytic expressions of the entries of M, as implicitly defined in (2.21), by their numerical values introduces rounding errors. As a consequence, M is no longer singular; hence it does not have a left null vector. However, the singular values of M computed by the Singular Value Decomposition algorithm, see, e.g., Golub & van Loan (1989) or Horn & Johnson (1985), are typically in the order of one, except the last value, which is equal to 0 up to machine precision. The corresponding left singular vector is then the natural candidate solution for the left null vector of the analytic M, up to a scaling factor. The computation of the inverse of $\Theta^{--}(B)$ causes some problems as well. The entries $\Theta_i^{--}(B)$ can become very large when i is such that $|ir - L|$ is small, thereby making the numerical inversion unstable. The solution is simply to compute $\Theta(-B)$ in (2.17) and take the upper left block of this matrix.

The left panel of Figure 2.3 shows the probabilities $\pi_i = A_i(B) + D_i = \mathbb{P}\{W = i\}$ and D_i for the parameters of Table 2.1. As 12 is the lowest source state with positive net input rate, 6 is the lowest state the source jumps into after congestion; so, states $1, \ldots, 5$ simply have no influx from 'above'. Therefore $\mathbb{P}\{W \leq 5\} = 0$. Intuitively we expect that the traffic source will not enter state 20 often, and in case it does, the content $C(t)$ can only be smaller than B for a short time due to the high source rate. The fact that π_{20} is small, and only slightly greater than D_{20} supports this reasoning.

N	L	r	B	λ
20	80/7	1	20	1

Table 2.1: *System parameters*

The right panel of Figure 2.3 presents $\mathbb{P}\{C > y\} = 1 - \sum_i A_i(y)$ for three different buffer sizes, $B = 1, 10, 20$; the other parameters are as in Table 2.1. To simplify the analysis, we scaled y by dividing it by the buffer size B. Clearly, the probability that the system is empty, i.e., $\mathbb{P}\{C = 0\} = 1 - \mathbb{P}\{C > 0\}$, decreases as a function of B. Apparently, the larger the buffer, the longer the buffer stays in a congested state, and the larger $\mathbb{P}\{C = B\}$ is.

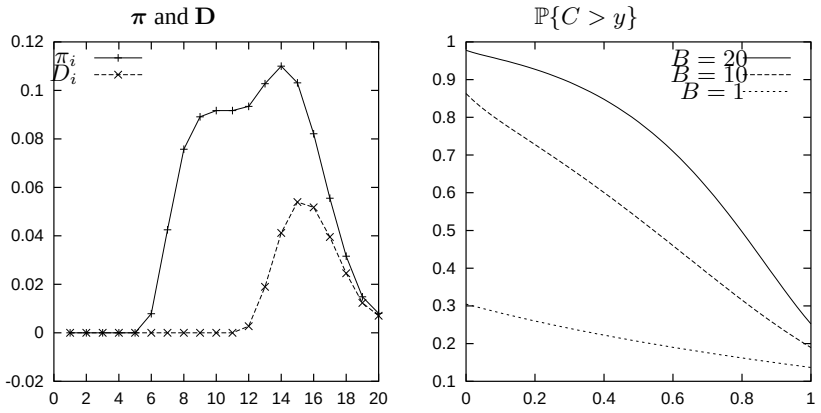

Figure 2.3: *The left panel shows the probabilities π_i and D_i. Adjacent points of the graph are connected for clarity. The right panel shows $\mathbb{P}\{C > y\}$ as a function of y/B.*

2.3.4 A Root p Law

Now we test the validity of the root p law (1.40) in the setting of our model. The root p law approximates the 'exact' throughput γ (2.23) as

$$\gamma_M = \frac{P}{T}\sqrt{\frac{3}{2p}}, \tag{2.24}$$

where p is the packet loss and T the round-trip time. The validity of this result for our model is not immediately clear for two reasons. First, the root p law assumes that the loss process is *exogenous*, i.e., the loss probability p is independent of the state of just one source. This is typically the case when many sources share a bottleneck link. In our case, however, p is not independent of the source behavior, but *endogenous*, i.e., entirely determined by the source process. Second, in (2.24) the round-trip time is assumed to be constant, whereas, in general and in our model, it is a random variable.

To apply (2.24), we have to interpret the packet size, the round-trip time and the loss in the context of the fluid model. In view of (2.9) we estimate the average of P as

$$\mathbb{E}\{P\} = \frac{r}{\lambda}\sum_i A_i(B) + \frac{r}{\mu}\sum_i D_i.$$

Reasoning analogously for the round-trip time, we get

$$\mathbb{E}\{T\} = \frac{1}{\lambda}\sum_i A_i(B) + \frac{1}{\mu}\sum_i D_i.$$

With regard to the loss probability p, Mathis *et al.* (1997) actually state that p should correspond to the number of negative (congestion) signals per acknowledged packet,

rather than the fraction of packets lost. To see this, note that whereas a TCP NewReno or Sack source generally reduces the window only once for a specific loss cycle, more than one packet per window may be actually dropped. So, to obtain an expression for p we replace the packet losses by negative signals, and use that the average amount of fluid per packet is r/λ (r/μ) if the buffer produces positive (negative) signals. To proceed, let

$$\begin{aligned}\underline{M}(T) &:= \text{fluid sent during } [0,T] \text{ while } C(t) < B,\\ \overline{M}(T) &:= \text{fluid sent during } [0,T] \text{ while } C(t) = B.\end{aligned}$$

Then, reasoning informally, we find for p:

$$\begin{aligned} p &= \lim_{T\to\infty} \frac{\text{Number of negative signals in } [0,T]}{\text{Number of packets sent in } [0,T]}\\ &= \lim_{T\to\infty} \frac{\mu \cdot \text{Total amount of time spent in congestion during } [0,T]}{\frac{\lambda}{r}\underline{M}(T) + \frac{\mu}{r}\overline{M}(T)}\\ &= \frac{\mu \sum_i D_i}{\lambda \sum_i i A_i(B) + \mu \sum_i i D_i}, \end{aligned} \tag{2.25}$$

where we use that

$$\begin{aligned} \lim_{T\to\infty} \frac{\underline{M}(T)}{T} &= \lim_{T\to\infty} \frac{r}{T} \sum_{i=1}^{N} i \int_0^T A_i(B,t)\,dt = r \sum_{i=1}^{N} i A_i(B),\\ \lim_{T\to\infty} \frac{\overline{M}(T)}{T} &= r \sum_{i=1}^{N} i D_i. \end{aligned}$$

To evaluate the approximation γ_M in the context of fluid, we plot in Figure 2.4 γ and γ_M as a function of B. Recall that p is not directly under our control (instead of being an independent quantity it is determined by system parameters such as B or λ). Hence we vary the buffer size B and compute first γ and p, and from the latter γ_M. Clearly, γ_M describes the behavior of γ quite well; qualitatively it is off by a multiplicative term. We see that, as expected, γ does not exceed L. However, γ_M, being an approximation, is not necessarily bound by this constraint. We remark that controlling the loss by changing λ rather than B turns out to give virtually the same results.

2.4 Conclusions

The feedback fluid model of a single TCP source allows us to investigate some of the intricacies of feedback systems in the presence of stochasticity. We can analyze the effects on link utilization of several relevant system parameters such as round-trip time, maximum window size (source peak rates), packet size, and buffer size. As a practical

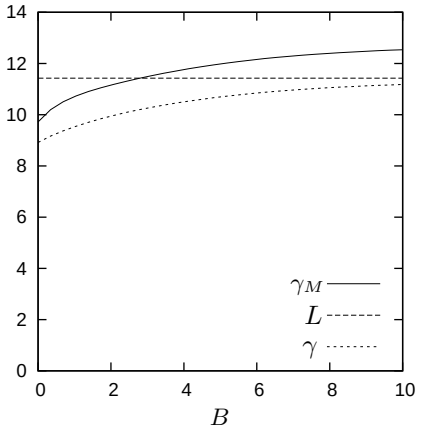

Figure 2.4: *The throughput γ and its root p approximation γ_M as functions of the buffer size B.*

case we investigate the extent to which the root p law, which Mathis *et al.* (1997) derive under the assumption of deterministic and constant round-trip times, holds when round-trip times are stochastic and the loss process is endogenous. Our analysis gives further support for validity of this law even when some of the assumptions of Mathis *et al.* (1997) are not met.

Chapter 3

A Feedback Fluid Model for Two Heterogeneous TCP Sources

In this chapter we develop a feedback fluid model for multiple TCP sources that share a single bottleneck router. In this model we can control the source parameters, such as the (stochastic) round-trip time or the maximum window size, of each individual source. This flexibility enables us to study the influence of these parameters on the throughput of each source and on the utilization of the system as a whole.

As in all multi-source models, we need to make a choice about how to distribute the loss of *fluid* during congestion over the sources. Here we model, in the parlance of the field, 'synchronized loss': all sources have to reduce their rate after a buffer overflow, cf. Section 1.3.2. Whereas this loss model is not necessary for our approach, we discuss it as it is the more difficult to analyze as compared to the 'proportional loss model'. (In this latter model just *one* source suffers from loss during congestion while the probability of losing a packet is proportional to a source's actual rate.) As a consequence of synchronized loss, the fluid model of this chapter is not a 'standard' feedback model in the sense of Section 1.1. In particular, we need to augment the joint source-buffer process with indicator variables to track which source(s) decreased their window after a congestion period. Except for this modification, the analysis is similar to the standard fluid queue discussed in Section 1.1.

In Section 3.1 we introduce the multi-source model. As the analysis becomes rather cumbersome for more than two sources we restrict the analysis, and the rest of the chapter, to the two-source case. Then, in Section 3.2, we solve the related two-point boundary value problem in steady state. We also establish a numerically efficient procedure to compute the coefficients of the solution of the differential equations. The numerical results of this model are presented in Section 3.3. By computing the throughput for each source we

obtain insight in fairness issues. We find analytic support for TCP's bias against sources with: (1) longer round-trip times; (2) smaller maximum congestion windows; (3) smaller packet sizes.

3.1 Model

In general we have J sources, labeled with index i, that send fluid into one shared buffer. The behavior of each source individually is similar to the single-source case explained in Chapter 2. However, and this is crucial in this model, each source has its own specific set of parameters $r_i, \mu_i, \lambda_i, N_i$.

The loss model we use here is 'synchronous loss', as explained in Section 1.3.2. As a direct consequence of differences in round-trip times, sources may receive positive and negative signals at different rates and at different moments in time. Therefore, after a period of congestion some sources still have to wait before they can increase their rate, while others already have started increasing their rate.

To incorporate these differences in source state, we augment the source processes with indicator variables $I_i(t)$. At times when $I_i(t) = 1$, source i is *in congestion*, that is, it should make a transition downwards as a buffer overflow occurred. When $I_i(t) = 0$, all data of source i arrives correctly at the destination and is acknowledged; hence, the source can increase its rate. Thus, the source i process is given by

$$\{W_i(t), I_i(t)\} \tag{3.1}$$

and has state space $\mathscr{W}_i \times \mathscr{I}_i = \{1, \ldots, n_i\} \times \{0, 1\}$.

Let $\{\mathbf{W}(t), \mathbf{I}(t)\} := \{W_1(t), \ldots, W_J(t), I_1(t), \ldots, I_J(t)\}$ denote the aggregate of the source processes and its state space be $\mathscr{W} \times \mathscr{I} := \Pi_{i=1}^{J} \mathscr{W}_i \times \Pi_{i=1}^{J} \mathscr{I}_i$. Then the state of the entire system can be written as $\{\mathbf{W}(t), \mathbf{I}(t), C(t)\}$. Note that whereas the process $\{\mathbf{W}(t), C(t)\}$ is *not* a Markov process, $\{\mathbf{W}(t), \mathbf{I}(t), C(t)\}$ *is* a Markov process.

Let the buffer of size $B < \infty$ be served at rate L. It is convenient to define the *net drift* function

$$r(\mathbf{n}) := \mathbf{n} \cdot \mathbf{r} - L := \sum_{i=1}^{J} n_i r_i - L, \tag{3.2}$$

where $\mathbf{n} := (n_1, \ldots, n_J)$ and $\mathbf{r} := (r_1, \ldots, r_J)$. As in (2.3), $\mathbf{r}$ should be such that

$$r(1, \ldots, 1) < 0 < r(N_1, \ldots, N_J), \tag{3.3}$$

so that the system is not trivial. Moreover, analogous to the discussion in Section 2.2.1, the vector $\mathbf{r}$ should satisfy the constraint

$$r(\mathbf{n}) \neq 0, \quad \text{for all } \mathbf{n} \in \mathscr{W}. \tag{3.4}$$

(Mitra (1988) or Sericola & Tuffin (1999) show how to handle systems in which this assumption is not made.)

The buffer content changes in accordance to, compare (1.47c),

$$\frac{dC(t)}{dt} = \begin{cases} \max\{r(\mathbf{W}(t)), 0\}, & \text{if } C(t) = 0, \\ r(\mathbf{W}(t)), & \text{if } C(t) \in (0, B), \\ \min\{r(\mathbf{W}(t)), 0\}, & \text{if } C(t) = B. \end{cases}$$

To clarify this model further, and introduce some necessary notation, let us specialize to two sources sharing the bottleneck buffer, and consider the evolution in time of the processes $\{\mathbf{W}(t)\} = \{W_1(t), W_2(t)\}$, $\{\mathbf{I}(t)\} = \{I_1(t), I_2(t)\}$, and $\{C(t)\}$. We observe that in the course of time the system is in one of four mutually exclusive 'modes', which do not necessarily occur consecutively in time:

0. $\mathbf{I}(t) = (0, 0)$, i.e., neither of the two sources is in a congested state, so both sources can increase their rate.

1. $\mathbf{I}(t) = (1, 0)$, i.e., after congestion occurred, a downward transition of source 2 removes the congestion from the buffer. Now source 2 can start increasing its rate, while source 1 still has to make a downward transition.

2. $\mathbf{I}(t) = (0, 1)$, i.e., after congestion, a transition of source 1 removes the congestion from the buffer. Source 2 still has to make a transition downward.

3. $\mathbf{I}(t) = (1, 1)$, i.e., the buffer is full so that both sources have to wait for a negative feedback signal before they decrease their rate.

For notational brevity we address the states of $\mathbf{I}(t)$ slightly differently, and define these in accordance with the enumeration immediately above, that is:

$$I(t) := \begin{cases} 0, & \text{if } \mathbf{I}(t) = (0, 0), \text{ i.e., neither source is in congestion,} \\ 1, & \text{if } \mathbf{I}(t) = (1, 0), \text{ i.e., just source 1 is in congestion,} \\ 2, & \text{if } \mathbf{I}(t) = (0, 1), \text{ i.e., just source 2 is in congestion,} \\ 3, & \text{if } \mathbf{I}(t) = (1, 1), \text{ i.e., both sources are in congestion.} \end{cases}$$

In accordance with this notation we split the state space $\mathscr{W} \times \mathscr{I} \times [0, B]$ into four subsets

$$\begin{aligned} \mathscr{T}_0 &:= \mathscr{W} \times \{0\} \times [0, B) \\ \mathscr{T}_1 &:= \mathscr{W} \times \{1\} \times [0, B) \\ \mathscr{T}_2 &:= \mathscr{W} \times \{2\} \times [0, B) \\ \widetilde{\mathscr{T}} &:= \mathscr{W} \times \{3\} \times B. \end{aligned}$$

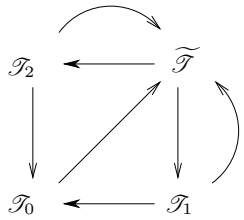

Figure 3.1: *The transitions between the four subspaces are indicated by the arrows. Note that there is no direct transition from $\widetilde{\mathscr{T}}$ to $\mathscr{T}_0$. This is a consequence of the flux conditions, to be discussed in Section 3.2.2.*

Figure 3.1 shows how these subsets communicate.

With these subsets we associate the following functions. On $\widetilde{\mathscr{T}}$ we have

$$D_{ij}(t) := \mathbb{P}\{\mathbf{W}(t) = (i,j), I(t) = 3, C(t) = B\},$$

and on $\mathscr{T}_k$, for $k \in \{0,1,2\}$ and $y < B$,

$$A_{ijk}(y,t) := \mathbb{P}\{\mathbf{W}(t) = (i,j), I(t) = k, C(t) \leq y\}.$$

Let $D_{ij}(t) = A_{ijk}(y,t) = 0$ whenever $(i,j) \notin \mathscr{W}$ or $y \notin [0,B]$. Note that from now on, we use (i,j) instead of (n_1,n_2) to denote the window states of both sources. The context will make clear whether i refers to a window state of source 1, i.e., $i \in \{1,\ldots,N_1\}$, or to a source index, i.e., $i \in \{1,2\}$.

Equation (3.3) implies that the sets of under- and overload states, i.e.,

$$\mathscr{W}_- := \{(i,j) \in \mathscr{W} | r_{ij} < 0\},$$
$$\mathscr{W}_+ := \{(i,j) \in \mathscr{W} | r_{ij} > 0\},$$

where $r_{ij} = ir_1 + jr_2 - L$, are not empty. By (3.4) we have $\mathscr{W} = \mathscr{W}_- \cup \mathscr{W}_+$, i.e., there are no $(i,j) \in \mathscr{W}$ such that $r_{ij} = 0$. As a reminder, $N := |\mathscr{W}| = |\mathscr{W}_1||\mathscr{W}_2| = N_1N_2$, $N_- := |\mathscr{W}_-|$, and $N_+ := N - N_-$. Finally, we indicate the restriction of a vector $\mathbf{x}$ to the set $\mathscr{W}_-$ by $\mathbf{x}^-$, etcetera.

3.2 Analysis

We start by deriving the Kolmogorov differential equations in Section 3.2.1, and discuss the boundary conditions in Section 3.2.2. In the subsequent Section 3.2.3 we derive the solution of the two-point boundary value problem. Finally, in Section 3.2.4 we show that in principle the analysis of the two-source case extends to the multiple-source case.

3.2.1 Kolmogorov Forward Equations

The derivation of the forward equations resembles the derivation of the single-source case of Chapter 2. However, now two sources can change their congestion state, and the communication between the states is more complicated, as shown by Figure 3.1.

Expanding $A_{ijk}(y, t+h)$ for $0 < y < B$, $1 < i < N_1$, $1 < j < N_2$, and $0 \le k \le 2$, yields:

$$\begin{aligned}
A_{ij0}(y,t+h) &= (1-(\lambda_1+\lambda_2)h)A_{ij0}(y-r_{ij}h,t) \\
&\quad + h[\lambda_1 A_{i-1,j,0}(y,t) + \lambda_2 A_{i,j-1,0}(y,t)] \\
&\quad + \mu_1 h[A_{2i,j,1}(y,t) + A_{2i+1,j,1}(y,t)] \\
&\quad + \mu_2 h[A_{i,2j,2}(y,t) + A_{i,2j+1,2}(y,t)] + o(h), \\
A_{ij1}(y,t+h) &= (1-(\mu_1+\lambda_2)h)A_{ij1}(y-r_{ij}h,t) \\
&\quad + \lambda_2 h A_{i,j-1,1}(y,t) + o(h), \\
A_{ij2}(y,t+h) &= (1-(\mu_2+\lambda_1)h)A_{ij2}(y-r_{ij}h,t) \\
&\quad + \lambda_1 h A_{i-1,j,1}(y,t) + o(h).
\end{aligned}$$

As in the single-source case, we do not discuss the details at the boundaries $i = 1$, and so on, explicitly. For $\mathbf{D}(t)$ we first consider the case $(i,j) \in \mathscr{W}_+$,

$$\begin{aligned}
D_{ij}(t+h) &= (1-(\mu_1+\mu_2)h)D_{ij}(t) \\
&\quad + \mu_1 h(D_{2i,j}(t) + D_{2i+1,j}(t)) \\
&\quad + \mu_2 h(D_{i,2j}(t) + D_{i,2j+1}(t)) \\
&\quad + (1-(\lambda_1+\lambda_2)h)\times \\
&\qquad \sum_{k=0}^{2} (A_{ijk}(B,t) - A_{ijk}(B-r_{ij}h,t)) + o(h).
\end{aligned}$$

When $(i,j) \in \mathscr{W}_-$ the same equation holds, except that $D_{ij}(t) \equiv 0$ for all $t > 0$ and the summation should be replaced by

$$\sum_{k=1}^{2} \left(A_{ijk}(B,t) - A_{ijk}(B+r_{ij}h,t)\right),$$

because now, as shown in Figure 3.1, there is a net *outflux* from $\widetilde{\mathscr{T}}$ to $\mathscr{T}_1$ and $\mathscr{T}_2$, but not to $\mathscr{T}_0$.

In the limit $h \to 0$ we get the following systems of partial differential equations: on

$\mathscr{T}_0$,

$$\begin{aligned}\frac{\partial A_{ij0}(y,t)}{\partial t} + r_{ij}\frac{\partial A_{ij0}(y,t)}{\partial y} &= \lambda_1[A_{i-1,j,0}(y,t) - A_{ij0}(y,t)] \\ &\quad + \lambda_2[A_{i,j-1,0}(y,t) - A_{ij0}(y,t)] \\ &\quad + \mu_1[A_{2i,j,1}(y,t) + A_{2i+1,j,1}(y,t)] \\ &\quad + \mu_2[A_{i,2j,2}(y,t) + A_{i,2j+1,2}(y,t)];\end{aligned} \tag{3.5a}$$

on $\mathscr{T}_1$ and $\mathscr{T}_2$,

$$\begin{aligned}\frac{\partial A_{ij1}(y,t)}{\partial t} + r_{ij}\frac{\partial A_{ij1}(y,t)}{\partial y} &= \lambda_2[A_{i,j-1,1}(y,t) - A_{ij1}(y,t)] \\ &\quad - \mu_1 A_{ij1}(y,t); \\ \frac{\partial A_{ij2}}{\partial t}(y,t) + r_{ij}\frac{\partial A_{ij2}}{\partial y}(y,t) &= \lambda_1[A_{i-1,j,2}(y,t) - A_{ij2}(y,t)] \\ &\quad - \mu_2 A_{ij2}(y,t);\end{aligned} \tag{3.5b}$$

and on $\widetilde{\mathscr{T}}$,

$$\begin{aligned}\frac{dD_{ij}(t)}{dt} &= \mu_1[D_{2i,j}(t) + D_{2i+1,j}(t) - D_{ij}(t)] \\ &\quad + \mu_2[D_{i,2j}(t) + D_{i,2j+1}(t) - D_{ij}(t)] + r_{ij}\sum_{k=0}^{2}\frac{\partial A_{ijk}(B,t)}{\partial y}.\end{aligned} \tag{3.5c}$$

The summation in (3.5c) runs from $k = 0$ to 2 for all $(i,j) \in \mathscr{W}$, instead of from $k = 1$ to 2 on $(i,j) \in \mathscr{W}_-$. This will be clarified once we derive the boundary conditions in (3.11).

3.2.2 The Stationary System and Boundary Conditions

As in the single-source case our main concern is the performance of the system in stationarity. Thus, in the sequel we consider the system to be in steady-state and let $\mathbf{W}$, $\mathbf{I}$ and C denote $\mathbf{W}(t)$, $\mathbf{I}(t)$ and $C(t)$ at an arbitrary point in time. Hence, we set $\partial_t \mathbf{A}_k = 0$ and $\partial_t \mathbf{D} = 0$, and drop the dependence on t.

We can rewrite the resulting equations conveniently with Kronecker products and sums[1], see, e.g., Lancaster & Tismenetsky (1985: Chapter 12). For this purpose associate with the *matrix* $A_k(y)$, with entries $A_{ijk}(y)$, a vector valued function $\mathbf{A}_k(y)$ by 'stacking' the rows of the matrix $A_k(y)$ into one long vector, i.e.,

$$\mathbf{A}_k := (A_{11k}, \ldots, A_{1N_2k}, A_{21k}, \ldots, A_{N_1N_2k}), \quad \text{for } k = 1, 2, 3.$$

[1]Given two matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$, the Kronecker product is written as $A \otimes B$, and Kronecker sum is $A \oplus B = A \otimes I_n + I_m \otimes B$, with $I_n(I_m)$ the identity matrix on $\mathbb{R}^n(\mathbb{R}^m)$.

Similarly, we obtain a vector $\mathbf{D}$. We also introduce matrices $Q_i, \widetilde{Q}_i$, and R_i, respectively, for $i \in \{1, 2\}$, in the form of equations (2.4), (2.2) and (2.6), with λ_i, μ_i, r_i replacing λ, μ, and r. Finally, we define the rate matrix

$$R := R_1 \oplus R_2 - LI_{N_1} \otimes I_{N_2},$$

with I_k the $k \times k$ identity matrix, and modified generators

$$\begin{aligned} \widetilde{Q}_i^D &:= -\mu_i I_{N_i}, \\ \widetilde{Q}_i^O &:= \widetilde{Q}_i - \widetilde{Q}_i^D. \end{aligned}$$

The first matrix $\widetilde{Q}_i^D$ is equal to the *diagonal* of $\widetilde{Q}_i$ apart from $(\widetilde{Q}_i^D)_{11}$, which is equal to $-\mu_i$, whereas $(\widetilde{Q}_i)_{11} = 0$. The matrix $\widetilde{Q}_i^O$ contains the *off-diagonal* non-zero elements and has zeros on its diagonal apart from $(\widetilde{Q}_i^O)_{11}$, which is μ_i.

Equations (3.5a–3.5b) can now be cast into matrix form:

$$\begin{aligned} \frac{d\mathbf{A}_0(y)}{dy} R &= \mathbf{A}_0(y)(Q_1 \oplus Q_2) + \mathbf{A}_1(y)(\widetilde{Q}_1^O \otimes I_{N_2}) \\ &\quad + \mathbf{A}_2(y)(I_{N_1} \otimes \widetilde{Q}_2^O) \\ \frac{d\mathbf{A}_1(y)}{dy} R &= \mathbf{A}_1(y)(\widetilde{Q}_1^D \oplus Q_2) \\ \frac{d\mathbf{A}_2(y)}{dy} R &= \mathbf{A}_2(y)(Q_1 \oplus \widetilde{Q}_2^D). \end{aligned} \tag{3.6}$$

We observe that $\mathbf{A}_1$ and $\mathbf{A}_2$ satisfy homogeneous differential equations similar to the single-source equation (2.14a). The differential equation for $\mathbf{A}_0$ contains two inhomogeneous terms related to $\mathbf{A}_1$ and $\mathbf{A}_2$. For (3.5c) we get,

$$\mathbf{0} = \mathbf{D}(\widetilde{Q}_1 \oplus \widetilde{Q}_2) + \sum_{k=0}^{2} \frac{d\mathbf{A}_k(B)}{dy} R. \tag{3.7}$$

Evaluating (3.6) at $y = B$, and adding this to (3.7) yields a set of balance equations:

$$\begin{aligned} \mathbf{0} &= \mathbf{A}_0(B)(Q_1 \oplus Q_2) + \mathbf{A}_1(B)(\widetilde{Q}_1 \oplus Q_2) \\ &\quad + \mathbf{A}_2(B)(Q_1 \oplus \widetilde{Q}_2) + \mathbf{D}(\widetilde{Q}_1 \oplus \widetilde{Q}_2). \end{aligned} \tag{3.8}$$

The number of unknowns is easily seen to be $4N$. Each vector $\mathbf{A}_k$, $k = 0, 1, 2$, depends on N coefficients, and $\mathbf{D}$ carries an additional N unknowns.

To solve the system of equations (3.6) we have to identify, in total, $4N - 1$ boundary conditions so that the scaling requirement

$$\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (A_{ij0}(B) + A_{ij1}(B) + A_{ij2}(B) + D_{ij}) = 1 \tag{3.9}$$

fixes the solution uniquely. We now identify these boundary conditions.

Part of the boundary conditions are analogous to the single-source boundary conditions (2.15):

$$\begin{aligned} \mathbf{A}_k^+(0) &= \mathbf{0}, \quad \text{for } k \in \{0,1,2\} \\ \mathbf{D}^- &= \mathbf{0}. \end{aligned} \tag{3.10}$$

These conditions put a restriction on the *value* of the solution at $y = 0$ and $y = B$. Apparently, we have $3N_+ + N_-$ of these conditions.

We can identify more conditions by considering the *flux* of the solutions on the boundary, i.e., conditions on the derivatives of $\mathbf{A}_k$. Clearly, when $\mathbf{W} \in \mathscr{W}_+$ the outflux of $\mathscr{T}_i$, $i \in \{0,1,2\}$, should be the influx to $\widetilde{\mathscr{T}}$. In fact, regarded in this way, (3.7) restricted to $\mathscr{W}_+$ becomes a flux condition. On $\mathscr{W}_-$ we distribute the outflux of $\widetilde{\mathscr{T}}$ over $\mathscr{T}_0, \mathscr{T}_1$ and $\mathscr{T}_2$ according to:

$$\begin{aligned} \left(\frac{d\mathbf{A}_0(B)}{dy}R\right)^- &= \mathbf{0}, \\ \left(\frac{d\mathbf{A}_1(B)}{dy}R\right)^- &= -\left(\mathbf{D}(I_{N_1} \otimes \widetilde{Q}_2)\right)^-, \\ \left(\frac{dA_{2,i}(B)}{dy}R\right)^- &= -\left(\mathbf{D}(\widetilde{Q}_1 \otimes I_{N_2})\right)^-. \end{aligned} \tag{3.11}$$

These conditions are natural in view of the current setting, i.e., modeling TCP. To see this, consider the first condition. A transition from $\widetilde{\mathscr{T}}$ to $\mathscr{T}_0$ requires that both sources make a downward jump at the same instant. Such events have zero probability. Hence the probability influx at $y = B$ to $\mathscr{T}_0$ should be 0. The second and third conditions distribute the outflux of $\widetilde{\mathscr{T}}$ on $\mathscr{W}_-$ over $\mathscr{T}_1$ and $\mathscr{T}_2$ in proportion to the downward rate of the second and first source, respectively.

Lemma 3.1. *The number of conditions implied by (3.11) and (3.8) is* $3N_- + N_+ - 1$.

Proof. It is evident that the flux conditions (3.11) only apply on $\mathscr{W}_-$; they therefore yield $3N_-$ conditions.

Let us now show that (3.8) provides $N - 1$ conditions on $\mathscr{W}$. The proof is similar to the proof of Lemma 2.3 for the single-source case. The only missing point is to show that the left null-spaces of the involved Kronecker sums are one dimensional. This follows in a straightforward manner from Lancaster & Tismenetsky (1985: Section 12.2) that the eigenvalues of the Kronecker sum $A \oplus B$ of any two matrices $A \in \mathbb{R}^{m\times m}$ with eigenvalues $\lambda_1, \ldots, \lambda_m$ and $B \in \mathbb{R}^{n\times n}$ with eigenvalues $\mu_1, \ldots, \mu_n$, are the mn numbers $\lambda_i + \mu_j$. To apply this property, let us point out that all matrices $Q_1, \widetilde{Q}_1$, and so on, in (3.8) have rank $N - 1$. (All eigenvalues not associated with the left null-spaces are negative, as is apparent from the fact that the generators are either upper or

lower tridiagonal.) Hence, the sum of any two eigenvalues is negative, except when both eigenvalues are zero. From this, and the above property of the eigenvalues of Kronecker sums, $Q_1 \oplus Q_2$ has rank $N-1$. The same reasoning applies of course to the other Kronecker sums.

Finally, as is apparent from the equivalence of (3.7) and (3.8) on the region $\mathscr{W}_-$, precisely N_- of conditions of (3.8) are automatically satisfied once the solution for $\mathbf{A}_i$ and $\mathbf{D}$ also meets (3.11). Hence, $N_+ - 1$ of the $N-1$ conditions of (3.8) remain. ∎

Clearly, we have found $3N_+ + N_-$ conditions with (3.10) and $3N_- + N_+ - 1$ conditions in the lemma above. Summarizing our findings in the table below yields

Region	$\mathscr{W}_-$	$\mathscr{W}_+$
Value conditions (3.10)	N_-	$3N_+$
Flux conditions (3.11), (3.8)	$3N_-$	$N_+ - 1$

As these conditions sum up to $4N-1$, we may conclude that the system is fully specified up to a scaling factor.

3.2.3 Solving the Stationary System

Our next task is to obtain a general solution of the equations (3.6). Whereas $\mathbf{A}_1$ and $\mathbf{A}_2$ satisfy homogeneous differential equations, the vector function $\mathbf{A}_0$ is the solution of an inhomogeneous system of linear differential equations. Before establishing the full solution for $\mathbf{A}_0$, we concentrate first on solving (3.6) for $\mathbf{A}_1$ and $\mathbf{A}_2$ and the homogeneous equation

$$\frac{d\mathbf{A}_0^{\hom}(y)}{dy} = \mathbf{A}_0^{\hom}(y)(Q_1 \oplus Q_2)R^{-1}.$$

The procedure to find the solutions for the homogeneous equations is completely analogous to the single-source case as presented in Section 2.2.4; we will here only summarize the results and fix some notation. For $i \in \{0,1,2\}$ denote by Φ_i, θ_i and $\mathbf{a}_i$, respectively, the appropriate matrix of eigenvectors, eigenvalues and vector of coefficients. Moreover, let $\Theta_i(y) := \exp(y\,\mathrm{diag}(\boldsymbol{\theta}_i))$. Now the solution can be written, for $i \in \{0,1,2\}$, as

$$\mathbf{A}_i^{\hom}(y) = \mathbf{a}_i \Theta_i(y) \Phi_i, \tag{3.12}$$

where $\mathbf{A}_1^{\hom}(y) \equiv \mathbf{A}_1(y)$ and $\mathbf{A}_2^{\hom}(y) \equiv \mathbf{A}_2(y)$.

Let us work out the particular solution for $\mathbf{A}_0$. Clearly, (3.6) is of the general form

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{x}(t)M + \mathbf{f}(t),$$

where $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{f} = (f_1, \ldots, f_n)$ and M an $n \times n$ matrix. The solution for this system is well-known, see, e.g., Lancaster & Tismenetsky (1985: Section 9.10),

$$\mathbf{x}(t) = \mathbf{x}_0 e^{Mt} + \int_0^t \mathbf{f}(s) e^{-Ms} ds\, e^{Mt}.$$

Applying this to $\mathbf{A}_0$ yields the expression

$$\mathbf{A}_0(y) = \mathbf{A}_0^{\text{hom}}(y) + \mathbf{g}_1(y) + \mathbf{g}_2(y) \tag{3.13}$$

where

$$\mathbf{g}_1(y) = \int_0^y \mathbf{A}_1(s)(\widetilde{Q}_1^O \otimes I_{N_2}) R^{-1} e^{-(Q_1 \oplus Q_2)R^{-1}s} ds\, e^{(Q_1 \oplus Q_2)R^{-1}y},$$
$$\mathbf{g}_2(y) = \int_0^y \mathbf{A}_2(s)(I_{N_1} \otimes \widetilde{Q}_2^O) R^{-1} e^{-(Q_1 \oplus Q_2)R^{-1}s} ds\, e^{(Q_1 \oplus Q_2)R^{-1}y}.$$

The integrals, which should be carried out entry-wise, can be considerably simplified. Using (3.12) and the property that $\Phi_0 \exp(s\,(Q_1 \oplus Q_2)R^{-1}) = \Theta_0(s)\Phi_0$, we obtain

$$\mathbf{A}_0(y) = (\mathbf{a}_0 + \mathbf{a}_1 M_1(y) + \mathbf{a}_2 M_2(y))\, \Theta_0(y)\Phi_0,$$

where the matrices $M_1(y)$ and $M_2(y)$ are defined as

$$M_1(y) = \int_0^y \Theta_1(s)\Phi_1(\widetilde{Q}_1^O \otimes I_{N_2}) R^{-1} \Phi_0^{-1} \Theta_0(-s) ds,$$
$$M_2(y) = \int_0^y \Theta_2(s)\Phi_2(I_{N_1} \otimes \widetilde{Q}_2^O) R^{-1} \Phi_0^{-1} \Theta_0(-s) ds.$$

Note that now, contrary to (3.13), the integration is straightforward as $\Theta_i(s)$ are diagonal matrices. The actual integration, then, yields for the ij-th component of M_1,

$$(M_1(y))_{ij} = \frac{e^{(\theta_{i1} - \theta_{j0})y} - 1}{\theta_{i1} - \theta_{j0}} \cdot \left(\Phi_1(\widetilde{Q}_1^O \otimes I_{N_2}) R^{-1} \Phi_0^{-1}\right)_{ij}, \tag{3.15}$$

with θ_{i1} (θ_{j0}) the i-th (j-th) component of $\boldsymbol{\theta_1}$ ($\boldsymbol{\theta_0}$). Similar expressions hold for the entries of $M_2(y)$. This form of the matrices $M_1(y)$ and $M_2(y)$ can be easily evaluated numerically.

We now summarize the complete solution of (3.6) in matrix form,

$$(\mathbf{A}_0(y), \mathbf{A}_1(y), \mathbf{A}_2(y)) = (\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2) \begin{pmatrix} I_N & 0 & 0 \\ M_1(y) & I_N & 0 \\ M_2(y) & 0 & I_N \end{pmatrix}$$
$$\times \begin{pmatrix} \Theta_0(y)\Phi_0 & & 0 \\ & \Theta_1(y)\Phi_1 & \\ 0 & & \Theta_2(y)\Phi_2 \end{pmatrix}.$$

The last step is to express all boundary conditions as a number of linear equations from which to we can compute $\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2$ and $\mathbf{D}$. Thus, we are to construct an expression comparable to the single-source equation (2.21). The work to be done at this point resembles Section 2.2.4 in most respects; we only point out the most important steps.

We start with the boundary conditions (3.10). Consider $\mathbf{A}_0^+(0) = \mathbf{0}$, i.e.,

$$\mathbf{0} = \mathbf{A}_0^+(0) = \mathbf{a}_0^- \Phi_0^{-+} + \mathbf{a}_0^+ \Phi_0^{++}.$$

From this we find

$$\mathbf{a}_0^+ = -\mathbf{a}_0^- \Phi_0^{-+} (\Phi_0^{++})^{-1}.$$

For $\mathbf{a}_1$ and $\mathbf{a}_2$ similar expressions hold. Clearly, $\mathbf{a}_i^+$ can be expressed in terms of $\mathbf{a}_i^-$.

At $y = B$, equations (3.8), (3.6) and (3.11) can be written as

$$\begin{aligned}
\mathbf{0} &= \mathbf{A}_0(B)(Q_1 \oplus Q_2) + \mathbf{A}_1(B)(\widetilde{Q}_1 \oplus Q_2) \\
&\quad + \mathbf{A}_2(B)(Q_1 \oplus \widetilde{Q}_2) + \mathbf{D}(\widetilde{Q}_1 \oplus \widetilde{Q}_2), \\
\mathbf{0} &= (\mathbf{A}_0(B)(Q_1 \oplus Q_2))^- + (\mathbf{A}_1(B)(\widetilde{Q}_1^O \otimes I_{N_2}))^- \\
&\quad + (\mathbf{A}_2(B)(I_{N_1} \otimes \widetilde{Q}_2^O))^-, \\
\mathbf{0} &= (\mathbf{A}_1(B)(Q_1^D \oplus Q_2))^- + (\mathbf{D}(I_{N_1} \otimes \widetilde{Q}_2))^-, \\
\mathbf{0} &= (\mathbf{A}_2(B)(Q_1 \oplus \widetilde{Q}_2^D))^- + -(\mathbf{D}(\widetilde{Q}_1 \otimes I_{N_2}))^-.
\end{aligned}$$

The condition for $\mathbf{D}$ can be easily incorporated, only $\mathbf{D}^+$ should be found, since by (3.10) we already have that $\mathbf{D}^- = \mathbf{0}$.

As in Section 2.2.4, we build matrices Ψ_i, etc. Combining all boundary conditions, we can write the final equation (after tedious but straightforward calculations) in the form

$$(\mathbf{a}_0^- \Theta_0^{--}(B), \mathbf{a}_1^- \Theta_1^{--}(B), \mathbf{a}_2^- \Theta_2^{--}(B), \mathbf{D}^+) K = \mathbf{0}, \tag{3.16}$$

where the matrix K has size $(3N_- + N_+) \times (3N_- + N_+)$. This equation has to be solved numerically, for instance by the Singular Value Decomposition algorithm. Here, as in the single-source case, solving for $\mathbf{a}_i^- \Theta_i^{--}(B)$ is numerically more robust than solving for $\mathbf{a}_i^-$ straightaway. The final result requires normalization according to (3.9).

3.2.4 The Multiple-Source Case

The multiple-source case can be handled, in principle, by similar methods as those used above in the analysis of the two-source case. However, the number of differential equations and boundary conditions grows exponentially in the number of sources. To see this, consider the J sources of Section 3.1 with state space $\mathscr{W} \times \mathscr{I}$. Is is evident that the cardinality of $\mathscr{W} \times \mathscr{I}$ equals $2^J \Pi_{i=1}^J N_i$. Due to the increasing number of dimensions, the (numerical) analysis becomes increasingly difficult. Therefore we have not studied

the multiple-source case with the fluid model developed here. In Chapters 5 and 6 we use another, more suitable, approach allowing us to extend the analysis to several sources (and networks with two buffers).

3.3 Results

In this section we focus on the bias of TCP against connections with larger round-trip times ($1/\lambda$), or smaller window growth rates (r). As mentioned in 1.3.2, this issue has been brought up in other studies, e.g., Lakshman & Madhow (1997), Floyd (1991), Kelly (2001). Besides these fairness issues we study whether competition among TCP sources affects the utilization of the entire system.

The numerical analysis is somewhat hampered by two problems. First the matrix K implicitly defined in (3.16) is ill-conditioned. Second, since the dimension of the matrix K is much larger than the one of the single-source case, its left null space is harder to compute. Consequently, the parameter ranges suitable for numerical analysis are rather small in comparison to the single-source case. Nevertheless, we can make a number of interesting observations for small-sized problems.

Before we present the results we generalize the performance measures introduced in Section 2.3.1.

3.3.1 Performance Measures

Here we define some steady state performance measures for source 1; similar definitions hold for source 2.

The expected transmission rate for source 1 is, in analogy to (2.22),

$$\tau_1 := r_1 \mathbb{E}\{W_1\} = r_1 \sum_{i,j} i\, \pi_{ij}. \tag{3.17}$$

As is apparent, we take here the expectation with respect to the stationary distribution of $\{\mathbf{W}(t), I(t), C(t)\}$, which is $\pi_{ij} := \mathbb{P}\{W_1 = i, W_2 = j\} = A_{0ij}(B) + A_{1ij}(B) + A_{2ij}(B) + D_{ij}$.

The definition of the throughput γ_1 is slightly more involved. During periods of congestion we have to distribute the excess fluid, i.e., the amount of lost fluid, over the two sources. Here we propose to do this in proportion to the source's momentary rate:

$$\begin{aligned} \gamma_1 &= \tau_1 - \mathbb{E}\left\{ r(\mathbf{W}) \frac{r_1 W_1}{r_1 W_1 + r_2 W_2} 1_{\{C=B\}} \right\} \\ &= \tau_1 - \sum_{i,j} (i r_1 + j r_2 - L) \frac{i r_1}{i r_1 + j r_2} D_{ij}. \end{aligned} \tag{3.18}$$

	Fig. 3.2	Fig. 3.3,	Fig. 3.4 Left	Fig. 3.4 Right
N_1	4,5,6,7	9	7	4
N_2	4,5,6,7	8	2–7	4–8
r_1	1	1	1	1
r_2	1	1	$N_1 r_1 / N_2$	1
T_1	0.25–1	2	1	1
T_2	1	1	1	1
L	5.87	7.87	5.87	5.87
B	1	4	1	1

Table 3.1: *Parameter settings used in Figures 3.2–3.4.*

As in Equation (2.23), $\mathbb{P}\{C = B\} = \mathbb{P}\{C = B, r(\mathbf{W}) > 0\}$ owing to the boundary condition $\mathbf{D}^- = \mathbf{0}$ in (3.10). In Chapter 6 we discuss this choice in more detail.

Finally, let $u_1 = \gamma_1 / L$ so that the total utilization becomes $u = u_1 + u_2$.

3.3.2 Fairness and Utilization

To analyze the influence of round-trip time differences on the utilization of each source we decrease the round-trip time T_1 of the first connection while keeping T_2 fixed. Figure 3.2 shows the throughput of each source along the vertical axis and $s = T_1/T_2$ is the variable set out along the horizontal axis. The parameter settings for this and the other figures are shown in Table 3.1.

From the graphs it is clear that the smaller the round-trip time of the first source, the more of the available capacity it claims. Moreover, the increase of the throughput of the first source is made at the expense of the second. Interestingly, this bias becomes more pronounced when the maximum window sizes, i.e., N_1 and N_2, increase.

The somewhat peculiar kink in the graph of u for $N_1 = N_2 = 5$ and $N_1 = N_2 = 7$ is probably due to numerical instabilities. A careful analysis of the numerical data shows that some components of (the numerical estimate for) π_{ij} are slightly negative. Simply setting these negative components to 0 does not resolve the problem completely. Besides this, the accuracy of the positive components is, by the same token, also unclear.

Figure 3.3 provides some additional insight in the phenomena observed in the previous figure. Both sources are nearly equal, only $\lambda_1 = 2\lambda_2$ and $N_1 = 9, N_2 = 8$. It is clear that π_{ij} attains its largest values around the larger (smaller) window sizes of the first (second) source. An analysis of the graphs of $A_{ij2}(B^-)$ and $A_{ij1}(B^-)$ (not included) makes plausible that the first source is mostly responsible for the congestion. It turns out that $A_{ij2}(B^-)$ is significantly larger than $A_{ij1}(B^-)$, showing that the second source spends more time waiting in a congested state than the first one. In summary, the first

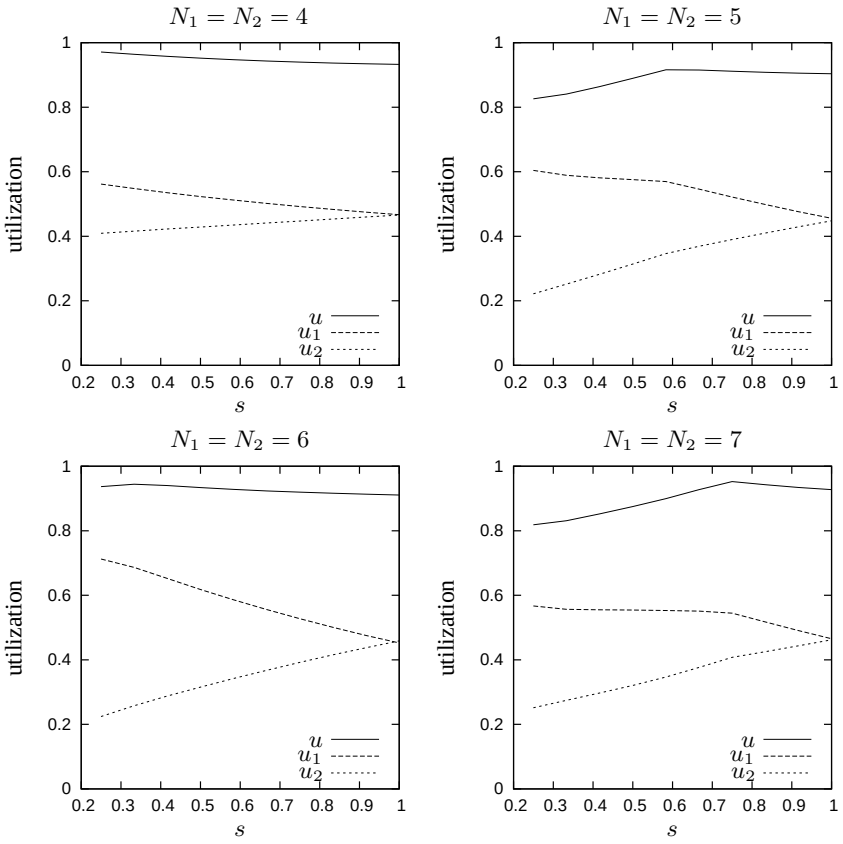

Figure 3.2: *The bias against sources with longer round-trip times. The ratio* $s = T_1/T_2$ *is set out along the horizontal axis.*

source learns sooner when congestion disappears, so that it can react quicker and build up a larger window on average.

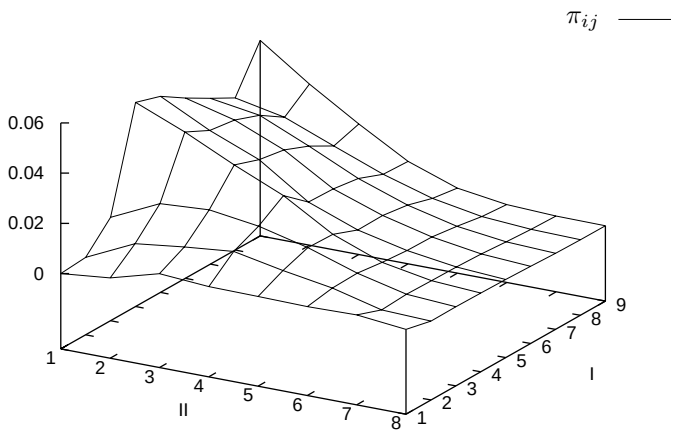

Figure 3.3: *The stationary probabilities* $\pi_{ij} = \mathbb{P}\{W_1 = i, W_2 = j\}$.

In the left panel of Figure 3.4 we study the effect of the 'aggressiveness' of Source 2 while $T_1 = T_2$. We vary N_2 from 2 to 7 but such that the peak rate $N_2 r_2$ remains fixed and equal to $N_1 r_1 = 7$ throughout. In other words, we vary r_2 and, as is clear from (2.9), thereby the packet size of source 2. The graph of the utilizations shows that, indeed, for large values of r_2 source 2 claims capacity more aggressively.

In the right panel of Figure 3.4 only N_2 changes while the other parameters remain fixed. As expected there is a bias against the source with the smaller maximum window size, i.e., source 1.

3.4 Conclusions

In this chapter we extend the feedback fluid model of TCP of the previous chapter to two, or more, sources to study link utilization and fairness as functions of system parameters. Based on the graphs of Section 3.3 the model captures quite some characteristic features of TCP as observed by, for instance, Floyd (1991). In more specific terms, the analysis carried out here provides support for the claim that bias is an intrinsic property of TCP, or more generally, AIMD congestion control algorithms. We observe bias against sources with: 1) longer round-trip times; 2) smaller window increment rates; 3) smaller maximum congestion windows (peak rates).

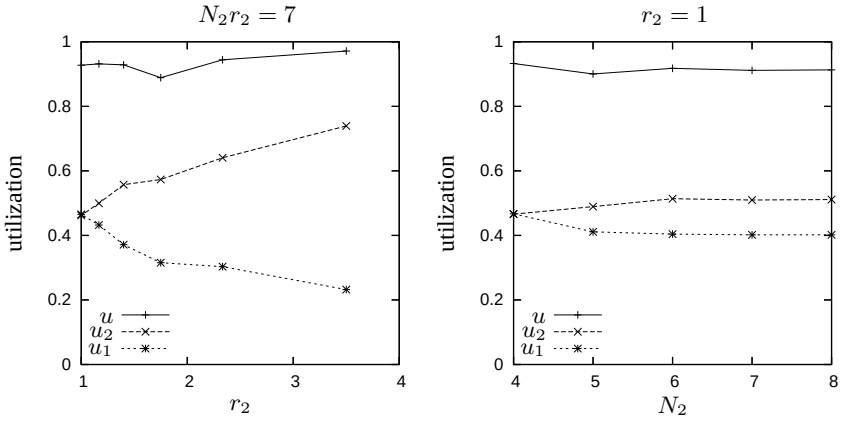

Figure 3.4: *The left panel shows the bias against non-aggressive sources while* $N_2 r_2 = \text{constant}$. *The right panel shows the bias against sources with small maximum window size.*

Although it seems fitting at this point to compare our model to simulation, we do not provide such comparison for three reasons. First, the model's intent is to provide fundamental insight into the behavior of the source and content process as functions of the system parameters. We do not expect it to produce accurate numerical output in absolute terms. Second, the numerical instabilities limit the parameter range quite severely. Third, the contribution of queueing delay to the round-trip times is incorporated rather coarsely by choosing the rates λ_i and μ_i, $i = 1, 2$, as in (2.7) and (2.8), respectively. The model of the next chapter allows us to cope with the dependence of the transition rates on the queueing delay in a better way.

Chapter 4

Fluid Queues with Continuous Feedback

The models studied in Chapters 2 and 3 are examples of the feedback queues introduced in Section 1.1.5. This type of feedback allows the infinitesimal generator to depend as a piecewise constant function on the buffer content. In this chapter we generalize the fluid model considerably in that now both the generator and the drift matrix can become *continuous* functions of the content.

Although this model is clearly interesting in its own right, we first provide some motivation in terms of modeling TCP. In the models developed in the previous two chapters the round-trip time is, by (2.7) and (2.8), a *piecewise constant* function of the buffer content. However, the round-trip time should be, by (1.47a), a *continuous* function of the buffer content. As the transition rates of the source process depend on round-trip times, these rates should also become continuous functions of the content process. The model developed in this chapter allows us to include such effects.

The structure of the chapter is as follows. We give a precise description of our model in Section 4.1 and mention some technical assumptions. In Section 4.2 we derive the Kolmogorov forward equations for the joint Markov process $\{W(t), C(t), t \geq 0\}$. We do this by carefully following the infinitesimal approach, also employed in the derivation of the standard Markov-modulated fluid models in Section 1.1, but we also pay attention to the relation with the continuity equation from physics. The details of the derivation are dealt with in Section 4.3. Section 4.4 relates the partial differential equations of Section 4.2 to a system of ordinary differential equations by taking Laplace transforms. In Section 4.5 we state the differential equations that determine the stationary distribution, including appropriate boundary conditions. In Section 4.6 we restrict the analysis to a model in which the source process has only two states. In this case it is possible to

find the stationary distribution in closed form, which we illustrate with two examples in Section 4.7. We present in Section 4.8 a numerical approach to find the stationary distribution when the source process has more than two states. Finally, in Section 4.9 we model the interaction between a TCP source and the content process as a fluid queue with continuous feedback.

4.1 Model and Preliminaries

In this section we specify the fluid model. We also state some assumptions on the functions involved and introduce some notation.

4.1.1 Model

Consider a fluid system consisting of a buffer fed by one or more fluid sources. The buffer content process $\{C(t)\} \equiv \{C(t), t \geq 0\}$ takes values in the set $[0, B]$ with B finite. The source, or background, process $\{W(t)\} \equiv \{W(t), t \geq 0\}$ has state space $\mathscr{W} = \{1, \ldots, N\}$ for some finite N. When $W(t) = i$ and $C(t) = y$, the sources transmit fluid into the buffer such that the instantaneous drift is given by some (known) function $r_i(y)$. We model the source process such that, loosely speaking, when $C(t) = y$, the process $\{W(t)\}$ behaves instantaneously as a continuous-time Markov chain with generator $Q(y)$. More precisely,

Definition 4.1. *The source process $\{W(t)\}$ is such that for all $y \in [0, B]$, $i \neq j$, and for all $t \geq 0$:*

1. $\mathbb{P}\{W(t+h) = i \,|\, W(t) = i, C(t) = y\} = 1 + Q_{ii}(y)h + o(h)$,
2. $\mathbb{P}\{W(t+h) = j \,|\, W(t) = i, C(t) = y\} = Q_{ij}(y)h + o(h)$,
3. $\mathbb{P}\{W$ *makes more than one transition in* $[t, t+h] \,|\, W(t) = i, C(t) = y\} = o(h)$.

Here the function $Q_{ij}(y), j \neq i$, is said to be the transition rate at which the source process jumps from state i to j when $C(t) = y$, and $Q_{ii}(y) = -\sum_{j \neq i} Q_{ij}(y)$.

Following the discussion on page 13 we note that, whereas $\{W(t)\}$ and $\{C(t)\}$ are not Markov processes, the joint process $\{W(t), C(t), t \geq 0\}$ *is* a Markov process. $\{W(t), C(t)\}$ is characterized by the matrix $Q(y)$ and the drift matrix $R(y) = \text{diag}(r_1(y), \ldots, r_N(y))$, and is defined on the state space $\mathscr{S} = \mathscr{W} \times [0, B]$.

Finally, define functions

$$F_i(y, t) = \mathbb{P}\{W(t) = i, C(t) \leq y\}, \quad i \in \mathscr{W},\ y \in [0, B], \tag{4.1}$$

and

$$F_i(y) = \lim_{t \to \infty} F_i(y, t), \quad i \in \mathscr{W},\ y \in [0, B].$$

Remark 4.2. The model is such that the joint process $\{W(t), C(t), t \geq 0\}$ can in principle be constructed as a piecewise-deterministic Markov process, cf. Davis (1984, 1993), similar to more traditional fluid queueing processes. For the special case of a two-state background process, Boxma *et al.* (2005) present an approach along these lines, i.e., they use the (extended) generator of the joint process. In this chapter we instead derive the Kolmogorov forward equations directly, as is often done in the literature on fluid queues, see, e.g., Kosten (1974), Anick *et al.* (1982), Kella & Stadje (2002), Mandjes *et al.* (2003a).

4.1.2 Assumptions

In the sequel we need some assumptions on $Q_{ij}(y)$ and $r_i(y)$, which we collect here for ease of reference. Let $C_k(X)$ denote the space of k-times continuously differentiable functions on the set X. We assume the following.

1. $Q_{ij}(y) \in C[0, B]$.
2. $r_i(y) \in C_1(0, B)$ and finite on $[0, B]$.
3. $\min_i \inf_{y\in(0,B)} |r_i(y)| > 0$, i.e., the functions r_i are strictly bounded away from 0 on $(0, B)$.
4. $r_i(y) < 0 < r_j(y)$ for at least one i, one j, and one (and hence all) $y \in (0, B)$.

The continuity assumptions in 1 and 2 may be weakened; we refer to Remark 4.5 to see how models with discontinuous $Q(y)$ and/or $R(y)$ can be analyzed. Due to the assumptions on $r_i(y)$ we can unambiguously define two disjoint subsets of $\mathscr{W}$: the set of *up-states* $\mathscr{W}_+ = \{i \in \mathscr{W} \,|\, r_i(y) > 0\}$, and the set of *down-states* $\mathscr{W}_- = \{i \in \mathscr{W} \,|\, r_i(y) < 0\}$, when $y \in (0, B)$. Let $N_- = |\mathscr{W}_-|$ and $N_+ = |\mathscr{W}_+|$. Clearly, by Assumption 3 we have that $\mathscr{W} = \mathscr{W}_+ \cup \mathscr{W}_-$. Assumption 4 ensures that both subsets are non-empty, thereby avoiding trivial models.

Because the boundaries at 0 and B act as impenetrable barriers for the content process, we have to assume

5. $r_i(0) = 0$ if $i \in \mathscr{W}_-$ and $r_j(B) = 0$ if $j \in \mathscr{W}_+$.
6. $r_i(0) = r_i(0+) > 0$ if $i \in \mathscr{W}_+$ and $r_j(B) = r_j(B-) < 0$ if $j \in \mathscr{W}_-$.

Here, and in the sequel, we use shorthands like $r_i(0+) = \lim_{y\downarrow 0} r_i(y)$ and $r_i(B-) = \lim_{y\uparrow B} r_i(y)$. Note that by items 5 and 6 we take the point of view as expressed in (1.24–1.25) and the discussion there.

Our next concern is the irreducibility of the process. Because it seems difficult to find necessary conditions for the process to be irreducible, we will only give a sufficiency

condition. Notice that by Assumptions 2 and 3 the integrals

$$\int_x^y \frac{du}{r_j(u)}, \quad j \in \mathscr{W}_+ \qquad \text{and} \qquad \int_y^x \frac{du}{-r_j(u)}, \quad j \in \mathscr{W}_-$$

are finite for all $x, y \in [0, B]$. These integrals represent the time it takes for the joint process to move from (j, x) to (j, y) without making jumps in between. As a consequence, and in combination with the other assumptions it is not too difficult to see that a sufficient (but not necessary) condition for irreducibility is given by

7a. For all $i \in \mathscr{W}_-$ ($i \in \mathscr{W}_+$) there is some $j \in \mathscr{W}_+$ ($j \in \mathscr{W}_-$) such that $Q_{ji}(B) > 0$ ($Q_{ji}(0) > 0$);

b. The matrix $\tilde{Q}$ with entries $\tilde{Q}_{ij} = 1_{\{i \in \mathscr{W}_+\}} Q_{ij}(B) + 1_{\{i \in \mathscr{W}_-\}} Q_{ij}(0)$ is irreducible.

Assumption 7a ensures that any downstate (upstate) can be reached from above (below). Suppose now that we would like to show that it is possible to reach state (j, y) from (i, x) with $i, j \in \mathscr{W}_-$ and $y \in (0, B)$. Clearly, Assumption 7a implies that a state (k, B) exists with $k \in \mathscr{W}_+$ from which the process can jump to (j, B) (followed by a drift from (j, B) to (j, y)), while assumption 7b ensures that it is possible to reach state (k, B) from $(i, 0)$ (and hence from (i, x)). Similar arguments for the cases where i and/or j are not in $\mathscr{W}_-$ establish that assumptions 7a and 7b indeed imply that any state (j, y) can be reached with positive probability from any starting state. For models that do no satisfy Assumption 7, we note that the results here remain valid as long as the process under study is irreducible or has a single absorbing set. In many instances this is not difficult to verify.

With regard to the functions $F_i(y, t)$ defined in (4.1), we observe that is it possible to represent these as $F_i(y, t) = A_i(y, t) + D_i(y, t)$, where $A_i(y, t)$ is an absolutely continuous function of y for all t and $D_i(y, t)$ is a jump function of y. (Contrary to the previous two chapters, the symbols 'A_i' and 'D_i' are here mnemonics for 'absolutely continuous' and 'discrete', rather than 'ascending' and 'descending', respectively.) We write for the atoms of $F_i(y, t)$ at $y = 0$ and $y = B$: $D_i(0, t) = F_i(0, t)$ and $D_i(B, t) = F_i(B, t) - F_i(B-, t)$. It is clear that when $D_i(y, 0)$ does not contain any jumps for $y \in [0, B]$, $D_i(y, t)$ will also not contain jumps for $y \in (0, B)$ and $t \geq 0$. Hence the densities $f_i(y, t) = \partial_y F_i(y, t)$ in that case exist for $y \in (0, B)$. In Section 4.2 we actually need the stronger assumption

8. $F_i(y, t) \in C_2((0, B) \times [0, \infty)), \quad i \in \mathscr{W}$,

because this implies the continuity of $\partial_t f_i(y, t)$ and $\partial_y f_i(y, t)$ for $t \geq 0$ and $\partial_y \partial_t F_i = \partial_t \partial_y F_i$. Note that Assumption 8 is not of the same type as the previous ones; in fact it is unclear what precise conditions on $R(y)$ and $Q(y)$ make sure this is satisfied. We remark that by using the 'generator approach' of Davis (1993), this problematic point is avoided altogether.

Remark 4.3. Boxma *et al.* (2005) establish conditions for the stability of the fluid queue for a two-state background process and an *unlimited* buffer size. The stability conditions are formulated in terms of the convergence of certain integrals involving the density functions. As these are available in closed-form for a system with a two-state source, the evaluation is, in principle, possible. In our model, however, the background process has more than two states, in general. Regrettably, no explicit solutions for the densities are known for such higher dimensional systems, so that we have been unable to find sufficiency conditions to guarantee stability. Hence, we restrict the analysis to finite buffer sizes.

4.2 Kolmogorov Forward Equations

In this section we derive the Kolmogorov forward equations for the process $\{W(t), C(t)\}$ by two different methods. The first is, in a sense, the standard method in which the probabilities $F_i(y, t+h)$ are expressed in terms of $F_i(y,t)$ for small h, cf. Section 1.1.2. In other words, here we fix an event and express its probability in terms of the distribution functions involved, both at time t and $t+h$, and finally equate these. The other method, which we describe in Section 4.2.4, is based on an interpretation of the forward equation in physical terms resulting in a continuity equation, see e.g., Keilson (1965). The derivation now depends on fixing a (measurable) subset of the state space, rather than an event, and considering the in- and outflow of probability mass for this set. We believe that this method is of independent interest as it has a natural interpretation and it is a quick method leading us directly to the correct equations.

4.2.1 Derivation of the Forward Equations

First we summarize the derivation of the forward equations for the situation in which Q, but not necessarily R, is constant as a function of the buffer content, see also Kella & Stadje (2002). This derivation is completely analogous to that of the standard case in which R and Q are fixed, cf. Section 1.1.1. Then we focus on the case in which the entries of which $Q(y)$ are non-constant functions of y. Although at first sight it may seem obvious that a similar system of forward equations results, this is in fact not the case. By a more careful analysis we obtain the correct result.

4.2.2 Constant Q

When Q is a constant matrix the usual approach to derive the forward equations, cf. Section 1.1, is to express $F_i(y, t+h)$ in terms of $F_i(y,t)$ for sufficiently small h and

$y \in (0, B)$, i.e.,

$$F_i(y, t+h) = (1 + hQ_{ii})F_i(y - r_i(y)h, t) + \sum_{j \neq i} hQ_{ji}F_j(y, t) + o(h). \tag{4.2}$$

By Assumption 8 we may write $F_i(y - r_i(y)h, t) = F_i(y, t) - hr_i(y)\partial_y F_i(y, t) + o(h)$. Rearranging terms, dividing by h, and letting h approach zero, yields

$$\frac{\partial}{\partial t}F_i(y, t) - r_i(y)\frac{\partial}{\partial y}F_i(y, t) + \sum_j F_j(y, t)Q_{ji}. \tag{4.3}$$

This is precisely the result obtained by Kella & Stadje (2002). Notice that (4.3) has the same form as the corresponding result for traditional Markov-modulated fluid queues, where both R and Q are constant matrices, cf. (1.8), only with a function $r_i(y)$ replacing the constant r_i. To complete the analysis we should also provide boundary conditions. However, as they are not relevant for the sequel, we continue with the case of primary interest here.

4.2.3 Variable Q

Focusing on the interval $(0, B)$, it may be tempting to assume that in case the coefficients of Q are non-constant functions of y, the partial differential equation (4.3) can be adapted simply by replacing Q_{ji} by $Q_{ji}(y)$, just as (4.3) is found from the equations for the standard Markov-modulated fluid model by replacing the constants r_i by $r_i(y)$. This is *not* the case, however, since $Q_{ji}(y)$ are transition rate functions for the source process, *provided* $C(t) = y$, while the event $\{W(t) = j, C(t) \leq y\}$ — the probability of which is given by $F_j(y, t)$ — does not at all imply that $C(t) = y$. This problem can be circumvented by considering *densities* rather than distribution functions. Thus, we write $d_y F_j(y, t) \equiv \mathbb{P}\{W(t) = j, C(t) \in dy\} = f_j(y, t)dy$, the last equality being valid if $y \in (0, B)$, and find the following extension of (4.2) for $y \in (0, B)$:

$$\begin{aligned} F_i(y, t+h) = &\int_0^{y - r_i(y)h} (1 + hQ_{ii}(x))\, d_x F_i(x, t) \\ &+ h \sum_{j \neq i} \int_0^y Q_{ji}(x)\, d_x F_j(x, t) + o(h). \end{aligned} \tag{4.4}$$

Again using $F_i(y - r_i(y)h, t) = F_i(y, t) - hr_i(y)\partial_y F_i(y, t) + o(h)$ and the fact that $\int_{y - r_i(y)h}^y hQ_{ii}(x)\, d_x F_i(x, t) = o(h)$, we find

$$F_i(y, t+h) = F_i(y, t) - hr_i(y)\frac{\partial}{\partial y}F_i(y, t) + h \sum_j \int_0^y Q_{ji}(x)\, d_x F_j(x, t) + o(h). \tag{4.5}$$

By subtracting $F_i(y,t)$ from both sides, dividing by h and taking the limit $h \to 0$, we find

$$\frac{\partial}{\partial t}F_i(y,t) = -r_i(y)\frac{\partial}{\partial y}F_i(y,t) + \int_0^y \sum_j Q_{ji}(x)\, d_x F_j(x,t). \tag{4.6}$$

This is the correct generalization of (4.3) at the interior of $[0,B]$. Notice that we were somewhat careless in the derivation above; in particular we did not prove assertion (4.4). We present a precise derivation of (4.5) in Section 4.3, with proper attention to $o(h)$ details.

We now provide the forward equations at the boundaries $y = 0$ and $y = B$. The equation for $y = 0$ follows easily by letting $y \downarrow 0$ in (4.6). Taking this limit yields

$$\frac{\partial}{\partial t}D_i(0,t) = -f_i(0+,t)r_i(0+) + \sum_j D_j(0,t)Q_{ji}(0). \tag{4.7}$$

To obtain the equation at $y = B$ we first write down the forward equations for the process $\{W(t)\}$:

$$\frac{\partial}{\partial t}F_i(B,t) = \int_0^B \sum_j Q_{ji}(x)\, d_x F_j(x,t), \tag{4.8}$$

which can be obtained by similar methods as those used in Section 4.3 to derive (4.6). Next, we take the limit $y \uparrow B$ in (4.6),

$$\frac{\partial}{\partial t}F_i(B-,t) = -f_i(B-,t)r_i(B-) + \int_0^{B-} \sum_j Q_{ji}(x)\, d_x F_j(x,t)$$

and subtract this from (4.8) to find

$$\frac{\partial}{\partial t}D_i(B,t) = f_i(B-,t)r_i(B-) + \sum_j D_j(B,t)Q_{ji}(B). \tag{4.9}$$

Finally, with respect to the boundary conditions it is clear, on physical grounds, that for $t > 0$ we must have

$$\mathbb{P}\{W(t) \in \mathscr{W}_-, C(t) = B\} = \mathbb{P}\{W(t) \in \mathscr{W}_+, C(t) = 0\} = 0.$$

With these boundary conditions, equations (4.6), (4.7) and (4.9) fully specify the stochastic behavior of the process (together with initial conditions).

We prefer to state our main result as a partial differential equation for densities instead of the integro-differential equation (4.6). Thus, we differentiate with respect to y, using Assumptions 2 and 8 from Section 4.1.2, while up to now we only needed $r_i(y) \in C(0,B)$ and $F_i(y,t) \in C_1((0,B)\times[0,\infty))$. The result is straightforward:

$$\frac{\partial}{\partial t}f_i(y,t) = -\frac{\partial}{\partial y}(f_i(y,t)r_i(y)) + \sum_j f_j(y,t)Q_{ji}(y). \tag{4.10}$$

Note that by differentiating (4.3) with respect to y and replacing Q_{ji} by $Q_{ji}(y)$ we obtain precisely this equation. Thus, although in (4.3) we could not simply replace Q_{ji} with $Q_{ji}(y)$, we apparently can in the equations for densities. This may not come as a surprise, in view of the discussion leading to equation (4.4).

The following theorem summarizes the results in vector form, where the row vector $\mathbf{f}(y,t)$ (respectively $\mathbf{D}(y,t)$) has components $f_i(y,t)$ (respectively $D_i(y,t)$), $i = 1,\ldots,N$.

Theorem 4.4. *Under the assumptions of Section 4.1.2, the Kolmogorov forward equations for the joint process $\{W(t),C(t)\}$ are, in row vector form,*

$$\frac{\partial}{\partial t}\mathbf{f}(y,t) = -\frac{\partial}{\partial y}(\mathbf{f}(y,t)R(y)) + \mathbf{f}(y,t)Q(y) \tag{4.11a}$$

$$\frac{d}{dt}\mathbf{D}(0,t) = -\mathbf{f}(0+,t)R(0+) + \mathbf{D}(0,t)Q(0) \tag{4.11b}$$

$$\frac{d}{dt}\mathbf{D}(B,t) = \mathbf{f}(B-,t)R(B-) + \mathbf{D}(B,t)Q(B). \tag{4.11c}$$

The boundary conditions to be satisfied for $t > 0$ are

$$D_i(0,t) = D_j(B,t) \equiv 0, \quad \text{if } i \in \mathscr{W}_+, j \in \mathscr{W}_-. \tag{4.12}$$

Remark 4.5. Our analysis extends easily to the case in which $Q(y)$ and $R(y)$ depend piecewise continuously on the buffer content y by combining the ideas presented here with those of Mandjes *et al.* (2003a). Considering thresholds $0 = B_0 < B_1 < \ldots < B_K = B$ in the buffer as in (1.26), we obtain a system of differential equations as in (4.11a) for each interval (B_k, B_{k+1}), while the equations (4.11b) and (4.11c) will be supplemented with similar equations for $\mathbf{D}(B_k,t)$.

4.2.4 Different Interpretation of the Forward Equations

In this section we relate the forward equations for $f_i(y)$, i.e., (4.10), to *continuity equations*. These are well-known relations in, e.g., the hydrodynamics and diffusion literature, expressing a conservation law in differential form, see e.g., Feynman (1970: Volume II, 13.1) or, for a probabilistic setting, Keilson (1965: Section II.3).

In the physics context the continuity equation in one dimension is given by

$$\frac{\partial}{\partial t}\rho(x,t) = -\frac{\partial}{\partial x}(\rho(x,t)v(x)), \tag{4.13}$$

where $\rho(x,t)$ is the density function of some conserved quantity, e.g., mass or electric charge, and $v(x)$ is the velocity at which this quantity is moving. Note that the partial derivative with respect to position operates on the *product* ρv, just as in (4.10). We now re-derive (4.10) from this point of view.

The dynamics of *any* conserved quantity is governed by a conservation law (admittedly, this is tautological). In general such a law states, in words,

$$\text{'Rate of Change'} = \text{'Influx'} - \text{'Outflux'} + \text{'Source terms'} - \text{'Sink terms'}.$$

Note that this principle is slightly more general than the equation for the density ρ of (4.13) as it also incorporates contributions from sources and sinks. Let us interpret this equation in terms of the probability mass assigned to the interval $(a, y]$ such that $0 < a \leq y < B$. As such, we treat probability mass as the conserved quantity of interest and discuss each term of this equation separately.

Consider first the left hand side. The rate of change of $\mathbb{P}\{W(t) = i, C(t) \in (a, y]\}$ is given by

$$\frac{d}{dt}(F_i(y,t) - F_i(a,t)),$$

or, in integral form,

$$\frac{d}{dt}\int_a^y f_i(x,t)dx = \int_a^y \frac{\partial}{\partial t} f_i(x,t)dx,$$

where the second equality holds as, by assumption, $\partial_t f_i$ is continuous in y and t.

Now turn to the right hand side of the conservation law and consider an $i \in \mathscr{W}_+$, so that $r_i(y) > 0$ for $y \in (0, B)$. Then the rate at which probability mass flows out of the interval $(a, y]$ at the boundary y, i.e., the *probability outflux* at y, is $f_i(y,t)r_i(y)$. The *influx* at a is seen to be $f_i(a,t)r_i(a)$. Thus the *net* outflux from the interval $(a, y]$ is

$$f_i(y,t)r_i(y) - f_i(a,t)r_i(a).$$

When $i \in \mathscr{W}_-$ this expression also gives the net outflux, because now the outflux is given by $-f_i(a,t)r_i(a)$, while $-f_i(y,t)r_i(y)$ is the influx. Further, we interpret the first term in the expression

$$\int_a^y \sum_{j \neq i} f_j(x,t)Q_{ji}(x)dx - \int_a^y f_i(x,t)|Q_{ii}(x)|dx$$

as a source term of probability mass, and the second as a sink term.

So now, by the 'conservation of probability mass', the above combines into

$$\begin{aligned}\int_a^y \frac{\partial}{\partial t} f_i(x,t)dx = &-f_i(y,t)r_i(y) + f_i(a,t)r_i(a) \\ &+ \int_a^y \sum_{j \neq i} f_j(x,t)Q_{ji}(x)dx - \int_a^y f_i(x,t)|Q_{ii}(x)|dx.\end{aligned}$$

As $(a, y]$ is an arbitrary interval, and $\partial_t f_i, \partial_y f_i$ and $\partial_y r_i$ are by assumption continuous we obtain the one-dimensional continuity equation (4.10) by differentiating with respect to y. Similar reasoning at the boundaries 0 and B immediately gives (4.7) and (4.9).

4.3 Proof of Theorem 4.4

In the previous section we expressed the dynamics of $F_i(y, t+h)$ for $h > 0$ sufficiently small and $y \in (0, B)$ in terms of the expansion

$$F_i(y,t+h) = \int_0^{y-r_i(y)h} (1+hQ_{ii}(x))\, d_x F_i(x,t) + h \sum_{j\neq i} \int_0^y Q_{ji}(x)\, d_x F_j(x,t) + o(h).$$

It is not immediately obvious that this expansion is indeed correct. For instance, the upper limit of integration y in the second term suggests that we assume that the content level remains constant during $[t, t+h]$, while this is certainly *not* the case. Compensating for this effect is difficult since the knowledge that $W(t) = j, W(t+h) = i$, and $C(t) = y$ is not sufficient to determine $C(t+h)$. To do that, we also need the epoch $\tau \in [t, t+h]$ at which the source makes its transition from state j to i, which is random. Similarly, we notice that the upper limits of both integrals should incorporate some $o(h)$ term. In the following we prove that when the assumptions of Section 4.1.2 are satisfied the influence of such subtleties can be absorbed in the term $o(h)$. In fact we prove (4.5), from which (4.6) then easily follows.

Lemma 4.6. *Under the assumptions of Theorem 4.4, the expansion (4.5), i.e.,*

$$F_i(y,t+h) = F_i(y,t) - hr_i(y)\frac{\partial}{\partial y}F_i(y,t) + h\sum_j \int_0^y Q_{ji}(x)\, d_x F_j(x,t) + o(h)$$

is valid for any $i \in \mathscr{W}$.

Proof. The proof consists of four steps. First we state three differential equations that bound all possible paths of $\{C(s), s \in [t, t+h]\}$. Then we define a family of transition functions that allow us in the third step to express the functions $F_i(y, t+h)$ in terms of some integrals, each involving the functions $F_j(x, t)$. In the last step we rewrite these and thereby prove the lemma. In the sequel consider $i \in \mathscr{W}$ fixed.

1. We introduce three differential equations aiming to relate events at time $t+h$, e.g., $\{C(t+h) \leq y\}$, to events at time t. To that end we define for $y \in (0, B)$,

$$\underline{r}(y) = \min(r_1(y), \ldots, r_N(y)), \qquad \overline{r}(y) = \max(r_1(y), \ldots, r_N(y)).$$

Clearly, $\underline{r}(y)$ and $\overline{r}(y)$ are continuous and finite functions on $(0, B)$ by the continuity and finiteness of the functions $r_i(y), 1 \leq i \leq N$, and satisfy $\underline{r}(y) < 0 < \overline{r}(y)$. Let a prime denote differentiation with respect to s. Then the three problems of interest are as follows,

$$\begin{aligned} y'(s) &= r_i(y(s)), & y(t+h) &= y, \\ \underline{y}'(s) &= \underline{r}(\underline{y}(s)), & \underline{y}(t+h) &= y, \\ \overline{y}'(s) &= \overline{r}(\overline{y}(s)), & \overline{y}(t+h) &= y, \end{aligned}$$

where $y \in (0, B)$ is some fixed *terminal* value. Notice that these terminal value problems are well defined on $s \in [t, t+h]$, provided that h is so small that $\underline{y}(t), \overline{y}(t) \in (0, B)$ if $y \in (0, B)$. The solutions to these terminal value problems are unique. In the remainder we will be particularly interested in these solutions evaluated at t, for which we have

$$\overline{y}(t) \leq y(t) \leq \underline{y}(t)$$

2. Let J_0, J_1 and J_2 denote the events that the source process makes respectively $0, 1,$ or more than 1 transitions in the interval $[t, t+h]$. Then we can define the following transition functions:

$$P_n(i, y, t+h \,|\, j, x, t) = \mathbb{P}\{W(t+h) = i, C(t+h) \leq y, J_n \,|\, W(t) = j, C(t) = x\}.$$

From our definition of $\overline{y}(t), y(t)$ and $\underline{y}(t)$ and Definition 4.1 it can be seen that for h sufficiently small we actually have

$$\begin{aligned}
P_0(i, y, t+h \,|\, j, x, t) &= \begin{cases} 1 + hQ_{ii}(x) + o(h) & \text{if } j = i \text{ and } x \leq y(t) \\ 0 & \text{otherwise} \end{cases} \\
P_1(i, y, t+h \,|\, j, x, t) &= \begin{cases} hQ_{ji}(x) + o(h) & \text{if } j \neq i \text{ and } x \leq \overline{y}(t) \\ 0 & \text{if } j \neq i \text{ and } x \geq \underline{y}(t), \text{ or if } j = i \end{cases} \\
P_1(i, y, t+h \,|\, j, x, t) &\leq hQ_{ji}(x) + o(h) \qquad \text{if } j \neq i \text{ and } \overline{y}(t) \leq x \leq \underline{y}(t), \qquad (*) \\
P_2(i, y, t+h \,|\, j, x, t) &= o(h).
\end{aligned}$$

The inequality in $(*)$ is intentional: when the process starts in (j, x) with $\overline{y}(t) \leq x \leq \underline{y}(t)$, it may 'escape' the set $(i, \underline{y}(t))$ during $[t, t+h]$.

3. By conditioning on $\{W(t) = j, C(t) \leq x\}$ we obtain

$$F_i(y, t+h) = \mathbb{P}\{W(t+h) = i, C(t+h) \leq y\} = I_0 + I_1 + I_2, \tag{4.14}$$

where

$$I_n = \sum_{j \in \mathcal{W}} \int_0^B P_n(i, y, t+h \,|\, j, x, t)\, d_x F_j(x, t).$$

We consider the individual integrals I_0, I_1, and I_2, consecutively.

4. With the expression above for $P_0(i, y, t+h \,|\, j, x, t)$, the integral I_0 becomes

$$I_0 = \int_0^{y(t)} (1 + hQ_{ii}(x))\, d_x F_i(x, t) + o(h),$$

where we used dominated convergence to establish that $\int_0^{y(t)} o(h)\, d_x F_i(x,t) = o(h)$. We rewrite this expression further by: (1) assuming that $F_i \in C_1((0,B)\times[0,\infty))$, which is implied by, but weaker than, Assumption 8; (2) using the Taylor expansion of $y(t)$ around $t+h$,

$$y(t) = y(t+h) - h\frac{d}{dy}y(t+h) + o(h) = y - hr_i(y) + o(h).$$

The result becomes

$$\begin{aligned} I_0 &= F_i(y - hr_i(y) + o(h), t) + \int_0^{y+(y(t)-y)} hQ_{ii}(x)\, d_x F_i(x,t) + o(h) \\ &= F_i(y,t) - hr_i(y)\frac{\partial}{\partial y}F_i(y,t) + \int_0^{y} hQ_{ii}(x)\, d_x F_i(x,t) \\ &\quad + \int_y^{y(t)} hQ_{ii}(x)\, d_x F_i(x,t) + o(h). \end{aligned} \tag{4.15}$$

The second integral is $o(h)$, and can therefore be absorbed in the $o(h)$ term.

For the second integral I_1 we can derive that

$$\begin{aligned} -\sum_{j\neq i}\int_{\overline{y}(t)}^{y} hQ_{ji}(x)\, d_x F_j(x,t) + o(h) &\leq I_1 - \sum_{j\neq i}\int_0^{y} hQ_{ji}(x)\, d_x F_j(x,t) \\ &\leq \sum_{j\neq i}\int_y^{\underline{y}(t)} hQ_{ji}(x)\, d_x F_j(x,t) + o(h). \end{aligned}$$

Since the left-hand side and right-hand side are both $o(h)$, we find

$$I_1 = \sum_{j\neq i}\int_0^{y} hQ_{ji}(x)\, d_x F_j(x,t) + o(h). \tag{4.16}$$

Finally, it is clear that $I_2 = o(h)$. Combining this with (4.14), (4.15) and (4.16) yields the desired result. ∎

4.4 Transient Behavior

Assuming for a moment that all assumptions in Section 4.1 are satisfied (including assumption 8) we take Laplace transforms with respect to time so that the partial differential equations (4.11a) are transformed into ordinary differential equations, and the ordinary differential equations (4.11b) and (4.11c) are turned into algebraic equations. This is of interest as there exist efficient numerical procedures to solve such systems of (ordinary) differential and algebraic equations for fixed $s > 0$, where s is the Laplace transform

variable. Moreover, once such solutions are known for various well-chosen values of s, the transient solution $f_i(y,t)$ can be obtained by standard numerical Laplace inversion, see e.g., Abate *et al.* (1999).

Thus, let us define for fixed $s > 0$ the Laplace transforms with respect to time:

$$\tilde{f}_i(y) = \int_0^\infty e^{-st} f_i(y,t)\, dt,$$

and analogously $\tilde{D}_i(0)$ and $\tilde{D}_i(B)$; note that we suppress the dependence on s in the transforms. Taking the transform of (4.11) is straightforward, so that we obtain in vector form,

$$-\mathbf{f}(y,0) + s\,\tilde{\mathbf{f}}(y) = -\frac{d}{dy}\left(\tilde{\mathbf{f}}(y)R(y)\right) + \tilde{\mathbf{f}}(y)Q(y) \tag{4.17a}$$

$$-\mathbf{D}(0,0) + s\,\tilde{\mathbf{D}}(0) = -\tilde{\mathbf{f}}(0+)R(0+) + \tilde{\mathbf{D}}(0)Q(0) \tag{4.17b}$$

$$-\mathbf{D}(B,0) + s\,\tilde{\mathbf{D}}(B) = \quad \tilde{\mathbf{f}}(B-)R(B-) + \tilde{\mathbf{D}}(B)Q(B). \tag{4.17c}$$

For fixed s, the solution of this system involves $3N$ constants, namely the $2N$ constants in $\tilde{\mathbf{D}}(0)$ and $\tilde{\mathbf{D}}(B)$, and N coefficients corresponding to the fundamental set of solutions of (4.17a). To check whether this system is well-defined we reason as follows. The transform of (4.12) yields N_+ conditions at $y = 0$ and N_- at $y = B$. Let I be the identity, and note that $R(y)$ and $sI - Q(y)$ are invertible for all y (due to assumption 4 and the fact that the eigenvalues of $sI - Q$ have positive real part). Then it is clear that (4.17b) imposes N_+ constraints on the coefficients vector and relates $\tilde{D}_i(0), i \in \mathscr{W}_-$, to the remaining N_- degrees of freedom of $\tilde{f}(y)$. These N_- degrees of freedom are, in geometric terms, 'propagated' by (4.17a) from the level $y = 0$ to $y = B$. There the fact that $D_i(B) = 0, i \in \mathscr{W}_-$, provides the missing conditions so that the N_- degrees of freedom of the coefficients vector are removed.

So far, we have worked with (Laplace transforms of) the density functions $f_i(y,t)$, complemented with the atoms $D_i(0,t)$ and $D_i(B,t)$. We could also, as Kella & Stadje (2002), derive equations for the Laplace transforms of the distribution functions, i.e., for $\tilde{F}_i(y) = \int_0^\infty e^{-st} F_i(y,t)\, dt$. It is not difficult to find from (4.6) that for $0 \le y < B$,

$$-\mathbf{F}(y,0) + s\tilde{\mathbf{F}}(y) = -\frac{d\tilde{\mathbf{F}}(y)}{dy}R(y) + \int_0^y d_x\tilde{\mathbf{F}}(x)Q(x), \tag{4.18}$$

while for $y = B$ the first term on the right-hand side vanishes, compare (4.8). The notation in the integral is similar to that in Section 4.2.3, only generalized to matrix notation, so that, e.g., the lower limit of the integral yields the term $\tilde{\mathbf{D}}(0)Q(0)$. By differentiation and taking some appropriate limits we can obtain the more appealing equations (4.17) from (4.18). However, for (4.6) and hence (4.18) to hold, we only need to assume the initial distribution to have no atoms, i.e., $F_i(y,0) \in C_1[0,B]$, instead of the more stringent Assumption 8 in Section 4.1.2. Moreover, we can find any initial distribution as a

weak limit of a sequence of distributions in $C_1([0,B])$, so that (4.18) must in fact hold for any initial distribution, possibly including atoms. This broader applicability can also be carried over to (4.17), since initial atoms at the boundaries are explicitly taken into account, while atoms in the open interval $(0,B)$ can be included by means of Dirac delta functions. Thus we arrive at the following theorem.

Theorem 4.7. *Under assumptions 1–7 in Section 4.1.2, $F_i(y,0) \in C_1[0,B]$, and for any initial distribution, the Laplace transforms $\tilde{f}_i(y)$, $\tilde{D}_i(0)$ and $\tilde{D}_i(B)$ satisfy* (4.17)*.*

Let us now specialize to the case with a two-state source, i.e., $\mathscr{W} = \{1,2\}$, with

$$Q(y) = \begin{pmatrix} -\lambda(y) & \lambda(y) \\ \mu(y) & -\mu(y) \end{pmatrix}.$$

Considering just the differential equation (4.17a), we find

$$-f_1(y,0) + s\tilde{f}_1(y) + (r_1(y)\tilde{f}_1(y))' = -\lambda(y)\tilde{f}_1(y) + \mu(y)\tilde{f}_2(y) \tag{4.19a}$$

$$-f_2(y,0) + s\tilde{f}_2(y) + (r_2(y)\tilde{f}_2(y))' = \quad \lambda(y)\tilde{f}_1(y) - \mu(y)\tilde{f}_2(y). \tag{4.19b}$$

Now express $\tilde{f}_2$ by means of (4.19a) in terms of $\tilde{f}_1$ and $\tilde{f}_1'$, differentiate this with respect to y so that $\tilde{f}_2'$ can be written in terms of $\tilde{f}_1$, $\tilde{f}_1'$ and $\tilde{f}_1''$, and put these relations in (4.19b). The result is, after some algebra, the following inhomogeneous second order differential equation for $\tilde{f}_1$:

$$\begin{aligned} &\tilde{f}_1'' + \left(\frac{s + r_2' + \mu}{r_2} - \frac{\mu'}{\mu} + \frac{s + 2r_1' + \lambda}{r_1} \right) \tilde{f}_1' \\ &\quad + \left(\frac{s + r_1' + \lambda}{r_1} \left(\frac{s + r_2' + \mu}{r_2} - \frac{\mu'}{\mu} \right) + \frac{\lambda' + r_1''}{r_1} - \frac{\lambda\mu}{r_1 r_2} \right) \tilde{f}_1 \\ &= \left(\frac{s + r_2' + \mu}{r_2} - \frac{\mu'}{\mu} \right) \frac{f_1(y,0)}{r_1} + \frac{f_1'(y,0)}{r_1} + \frac{\mu f_2(y,0)}{r_1 r_2}, \end{aligned} \tag{4.20}$$

where we suppress the dependence of λ, μ, r_1 and r_2 on y for conciseness. If we take the drift functions of the form $r_i(y) = \alpha_i y + \beta_i$, and take λ and μ to be independent of y, the corresponding homogeneous equation can be rewritten as the differential equation for the hyper-geometric function, see e.g. Lanczos (1997: pp. 349–351). Indeed, precisely this is done by Kella & Stadje (2002), where the transient behavior of this system is found explicitly.

4.5 Stationary Behavior

First we will show that a stationary distribution actually exists. The approach we follow is of interest in its own right, and may be described best as the uniformization of the joint

process $\{W(t), C(t)\}$. It is similar to the approach of Kella & Stadje (2002), where the background process is uniformized. Although in our context the processes $\{W(t)\}$ and $\{C(t)\}$ are not Markov processes, it turns out that we can still follow the same line of reasoning.

We start by choosing a constant $\lambda < \infty$ such that

$$\lambda \geq \sup_{(i,y)\in \mathscr{W}\times[0,B]} |Q_{ii}(y)|,$$

which is possible by Assumption 1. Then we define for $y \in [0, B]$ the following functions:

$$p_{ij}(y) = \begin{cases} Q_{ij}(y)/\lambda, & \text{if } i \neq j, \\ 1 + Q_{ii}(y)/\lambda, & \text{if } i = j. \end{cases}$$

Although we cannot consider the background process as a discrete-time Markov chain embedded at (i.e., just before) the points of increase of an independent Poisson process with intensity λ, we *can* do so for the joint process $\{W(t), C(t)\}$. To derive the transition kernel of the resulting joint discrete-time process, suppose first that $j \in \mathscr{W}_+$. Then,

$$\begin{aligned} P_{i,x}(j, (y, B]) &= \begin{cases} p_{ij}(x), & \text{if } 0 \leq y \leq x, \\ p_{ij}(x) \exp\left(-\lambda \int_x^y \frac{du}{r_j(u)}\right), & \text{if } x \leq y < B. \end{cases} \\ P_{i,x}(j, \{B\}) &= p_{ij}(x) \exp\left(-\lambda \int_x^B \frac{du}{r_j(u)}\right), \end{aligned} \tag{4.21a}$$

When $j \in \mathscr{W}_-$ (recall $r_j < 0$ if $j \in \mathscr{W}_-$),

$$\begin{aligned} P_{i,x}(j, [0, y)) &= \begin{cases} p_{ij}(x), & \text{if } x \leq y < B, \\ p_{ij}(x) \exp\left(-\lambda \int_x^y \frac{du}{r_j(u)}\right), & \text{if } 0 < y \leq x, \end{cases} \\ P_{i,x}(j, \{0\}) &= p_{ij}(x) \exp\left(-\lambda \int_x^0 \frac{du}{r_j(u)}\right). \end{aligned} \tag{4.21b}$$

The proof of the following proposition shows that the uniformized process has the same jump-behavior as the source process specified in Definition 4.1, and may actually serve to better understand this definition.

Proposition 4.8. *The jump behavior of the uniformized process above is in agreement with Definition 4.1.*

Proof. By conditioning on the number of jumps of the uniformizing Poisson process we

find that

$$\begin{aligned}
&\mathbb{P}\{W(t+h) = i \,|\, W(t) = i, C(t) = x\} \\
&= \sum_k \exp(-\lambda h) \frac{(\lambda h)^k}{k!} \times \\
&\qquad \mathbb{P}\{W(t+h) = i \,|\, W(t) = i, C(t) = x,\ k \text{ jumps in } [t, t+h]\} \\
&= (1 - \lambda h + o(h))\,(1 + p_{ii}(x+\epsilon)\lambda h + o(h)) \\
&= 1 + Q_{ii}(x+\epsilon)h + o(h) \\
&= 1 + Q_{ii}(x)h + o(h),
\end{aligned}$$

since $|\epsilon| < h \sup_{(i,y)\in\mathscr{W}\times[0,B]} |r_i(y)|$, and similarly for $j \neq i$,

$$\begin{aligned}
&\mathbb{P}\{W(t+h) = j \,|\, W(t) = i, C(t) = x\} \\
&= \sum_k \exp(-\lambda h) \frac{(\lambda h)^k}{k!} \times \\
&\qquad \mathbb{P}\{W(t+h) = j \,|\, W(t) = i, C(t) = x,\ k \text{ jumps in } [t, t+h]\} \\
&= (1 - \lambda h + o(h))\,(0 + p_{ij}(x+\epsilon)\lambda h + o(h)) \\
&= Q_{ij}(x+\epsilon)h + o(h) \\
&= Q_{ij}(x)h + o(h).
\end{aligned}$$

Finally, the process also satisfies the third part of Definition 4.1, since the number of jumps by $\{W(t)\}$ in $[t, t+h]$, is bounded from above by the number of jumps of the uniformizing Poisson process in $[t, t+h]$, which is $o(h)$. ∎

We next prove the existence of a stationary distribution for the discrete-time process.

Lemma 4.9. *The discrete Markov chain governed by the Markov transition kernel P defined by (4.21a) and (4.21b) is (strong) Feller. Therefore, by Meyn & Tweedie (1993: Theorem 12.0.1), there exists at least one invariant, i.e., stationary, distribution for this Markov chain.*

Proof. The transition kernel P acts on (measurable) functions on $\mathscr{S}$ as an operator defined by

$$Th_i(x) = \sum_j \int_0^B P_{i,x}(j, dy) h_j(y).$$

According to Meyn & Tweedie (1993: Theorem 6.1.1) we have to show that for T to be strong Feller, it maps all bounded measurable functions h (on $\mathscr{S}$) to continuous functions. If we interpret the functions h_i as the coordinate functions of a bounded function h, it is clear that it suffices to show that $Th_i(x)$ is continuous for any bounded measurable h_i.

From (4.21),

$$\begin{aligned} Th_i(x) &= \sum_j \int_0^B P_{i,x}(j, dy) h_j(y) \\ &= \sum_{j \in \mathscr{W}_-} p_{ij}(x) \exp\left(-\lambda \int_x^0 \frac{du}{r_j(u)}\right) h_j(0) \\ &\quad - \sum_{j \in \mathscr{W}_-} p_{ij}(x) \int_0^x \exp\left(-\lambda \int_x^y \frac{du}{r_j(u)}\right) \frac{\lambda}{r_j(y)} h_j(y) dy \\ &\quad - \sum_{j \in \mathscr{W}_+} p_{ij}(x) \int_x^B \exp\left(-\lambda \int_x^y \frac{du}{r_j(u)}\right) \frac{\lambda}{r_j(y)} h_j(y) dy \\ &\quad + \sum_{j \in \mathscr{W}_+} p_{ij}(x) \exp\left(-\lambda \int_x^B \frac{du}{r_j(u)}\right) h_j(B). \end{aligned}$$

Due to the continuity of p_{ij}, r_j and the boundedness of h_j this expression is continuous in x. Finally, regarding the tightness condition of Meyn & Tweedie (1993: Theorem 12.0.1) we remark that our state space $\mathscr{S}$ is already compact. ∎

Remark 4.10. This proof does not easily extend to discontinuous generators $Q(y)$ as then the function $p_{ij}(y)$ is not continuous.

Finally, using PASTA, it follows that the stationary distribution of the continuous-time process $\{W(t), C(t)\}$ exists and is the same as the stationary distribution of the discrete-time process. From (4.11) it is now clear that the following theorem holds.

Theorem 4.11. *Under Assumptions 1–7 in Section 4.1.2, a stationary distribution for the process $\{W(t), C(t)\}$ exists. It satisfies the following system of (ordinary) differential and algebraic equations,*

$$\frac{d}{dy}(\mathbf{f}(y)R(y)) = \mathbf{f}(y)Q(y) \tag{4.22a}$$

$$\mathbf{f}(0+)R(0+) = \mathbf{D}(0)Q(0) \tag{4.22b}$$

$$-\mathbf{f}(B-)R(B-) = \mathbf{D}(B)Q(B), \tag{4.22c}$$

with boundary conditions

$$D_i(0) = D_j(B) = 0, \quad \text{if } i \in \mathscr{W}_+, j \in \mathscr{W}_-, \tag{4.22d}$$

and normalization condition

$$\sum_j F_j(B) = \sum_j D_j(0) + \sum_j \int_0^B f_j(x) dx + \sum_j D_j(B) = 1. \tag{4.22e}$$

Let us check that the number of equations suffices to make the system complete. The number of unknowns is $3N$: the N coefficients appearing in the general solution of (4.22a), and $2N$ constants in the form of $D_i(0)$ and $D_i(B)$. The required number of conditions should then also equal $3N$. It is evident that (4.22d) and (4.22e) together contain $N+1$ conditions. However, the number of conditions provided by (4.22b) and (4.22c) is not immediately obvious as the rank of $Q(y)$ is less than N for all y (and in particular for $y=0$ and $y=B$). In particular, in the presence of (4.22d) we may replace both $Q(0)$ and $Q(B)$ in the equations (4.22b–4.22c) by the matrix $\tilde{Q}$, defined in Assumption 7b. Since we assumed this matrix to be irreducible, its rank is $N-1$. So, formally speaking, it might seem that (4.22b) and (4.22c) provide only $2N-2$ conditions, and that one condition is lacking. Interestingly, this sought-after condition lies hidden in (4.22a), (4.22b) and (4.22c) and is stated in the following lemma.

Lemma 4.12. *Any solution of (4.22) satisfies the condition*

$$\sum_i f_i(y) r_i(y) = 0, \quad y \in (0,B).$$

To see this, note that the row sums of $Q(y)$ equal 0 for all $y \in [0,B]$. Hence, taking this sum in (4.22a) and then integrating yields that $\sum_i f_i(y) r_i(y) = C$ for all $y \in (0,B)$ and some constant C. From (4.22b) and (4.22c) it immediately follows that $C=0$.

We end this section by mentioning that the statement in Lemma 4.12 also has a probabilistic interpretation, in the form of a level crossing argument which has also been employed by, e.g., Bekker *et al.* (2004). The argument is based on the fact that, in stationarity, the number of times that the buffer level moves up through some level y, must in the long run balance the number of times it moves down through the same level y. It is not difficult to see that this reasoning also leads to $\sum_i f_i(y) r_i(y) = 0$.

4.6 Explicit Solution for the Stationary Two-State System

The goal of this section is to find a closed-form expression for the solution of (4.22) when $\mathscr{W} = \{1,2\}$. The first step is concerned with finding a fundamental solution that satisfies Lemma 4.12. In the second step we find the constants involved. For the sake of clarity we summarize the results of each step in a lemma, and state the final result in a theorem.

Let us start by writing down (4.22a) in full detail for the present case. Here

$$R(y) = \text{diag}(r_1(y), r_2(y)),$$

where, without loss of generality, $r_1(y) < 0 < r_2(y), y \in (0,B)$, and

$$Q(y) = \begin{pmatrix} -\lambda(y) & \lambda(y) \\ \mu(y) & -\mu(y) \end{pmatrix}.$$

Hence, the system of differential equations (4.22a) becomes

$$f_1'(y)r_1(y) + f_1(y)r_1'(y) = -\lambda(y)f_1(y) + \mu(y)f_2(y) \tag{4.23a}$$
$$f_2'(y)r_2(y) + f_2(y)r_2'(y) = \quad \lambda(y)f_1(y) - \mu(y)f_2(y). \tag{4.23b}$$

Lemma 4.13. *Any positive solution of the system (4.23) that satisfies Lemma 4.12 is given by*

$$\mathbf{f}(y) = ae^{-g(y)}\left(-\frac{1}{r_1(y)}, \frac{1}{r_2(y)}\right), \tag{4.24}$$

where a is a positive constant, and

$$g(y) = \int_0^y \left(\frac{\lambda(x)}{r_1(x)} + \frac{\mu(x)}{r_2(x)}\right) dx. \tag{4.25}$$

Proof. When the meaning is clear we suppress in the proofs the functional dependence of $r_1(y), f_1(y)$, etc., on y, i.e., we write $r_1 = r_1(y), f_1 = f_1(y)$, etc.

For ease of notation, we prefer to analyze

$$h_i = f_i r_i,$$

which is equivalent to analyzing f_i as $|r_i(y)| \geq \epsilon$ for some $\epsilon > 0$ and for all $y \in (0, B)$. By substitution in (4.23) we obtain the equivalent problem

$$h_1' = -\frac{\lambda}{r_1}h_1 + \frac{\mu}{r_2}h_2, \tag{4.26a}$$
$$h_2' = \quad \frac{\lambda}{r_1}h_1 - \frac{\mu}{r_2}h_2. \tag{4.26b}$$

By Lemma 4.12 we have that $h_1 = -h_2$. Therefore (4.26a) becomes

$$h_1' = -\left(\frac{\lambda}{r_1} + \frac{\mu}{r_2}\right)h_1.$$

Its solution is seen to be of the form

$$h_1(y) = -ae^{-g(y)}, \tag{4.27}$$

where $g(y)$ is given by (4.25) and a is some constant. Finally, writing $f_1 = h_1/r_1$ and $f_2 = -h_1/r_2$ we find the fundamental solution (4.24), which is positive if $a > 0$. ∎

Lemma 4.14. *The flux relations (4.22b) and (4.22c) together with the boundary conditions (4.22d) and the normalization (4.22e) imply that*

$$\begin{aligned} a &= \lambda(0)D_1(0) \\ D_1(0) &= \left[1 + \lambda(0)\int_0^B e^{-g(x)}\left(\frac{-1}{r_1(x)} + \frac{1}{r_2(x)}\right)dx + e^{-g(B)}\frac{\lambda(0)}{\mu(B)}\right]^{-1} \\ D_2(B) &= e^{-g(B)}\frac{\lambda(0)}{\mu(B)}D_1(0). \end{aligned} \tag{4.28}$$

Proof. Combining (4.22b) with the boundary condition $D_2(0) = 0$ yields

$$(f_1(0+)r_1(0+), f_2(0+)r_2(0+)) = (-\lambda(0)D_1(0), \lambda(0)D_1(0)).$$

With (4.24) this becomes

$$a(-1, 1) = (-\lambda(0)D_1(0), \lambda(0)D_1(0)),$$

implying that

$$a = \lambda(0)D_1(0).$$

At the boundary $y = B$ we find from (4.22c) and $D_1(B) = 0$ that

$$-f_1(B-)r_1(B-) = ae^{-g(B)} = \mu(B)D_2(B).$$

Hence,

$$D_2(B) = e^{-g(B)}\frac{\lambda(0)}{\mu(B)}D_1(0).$$

The last step is to find $D_1(0)$. This follows from (4.22e), i.e.,

$$D_1(0) + \int_0^B (f_1(y) + f_2(y))dy + D_2(B) = 1.$$

∎

Theorem 4.15. *When $N = 2$ the solution of (4.22a) satisfying (4.22b–4.22e) is given by*

$$\mathbf{f}(y) = \lambda(0)D_1(0)e^{-g(y)}\left(-\frac{1}{r_1(y)}, \frac{1}{r_2(y)}\right), \tag{4.29}$$

where $D_1(0)$ and $D_2(B)$ are given by (4.28)*, and $g(y)$ is defined by* (4.25)*. In terms of the distribution function we have for $y \in [0, B]$*

$$\begin{aligned}\mathbf{F}(y) = &\left(D_1(0)\,,\, D_2(B)1_{\{y=B\}}\right)\\ &+ \lambda(0)D_1(0)\int_0^y e^{-g(x)}\left(\frac{-1}{r_1(x)}, \frac{1}{r_2(x)}\right)dx.\end{aligned} \tag{4.30}$$

Remark 4.16. The systems (4.19) and (4.23) have a similar structure. However, the seemingly innocuous presence of $s\tilde{f}_1(y)$ and $s\tilde{f}_1(y)$ complicates the solvability of (4.19). To see this, note that the reasoning leading to Lemma 4.12 yields for equation (4.19) of the transient case

$$s\tilde{f}_1(y) - f_1(y, 0) + (r_1(y)\tilde{f}_1(y))' + s\tilde{f}_2(y) - f_2(y, 0) + (r_2(y)\tilde{f}_2(y))' = 0.$$

Based on this, it is clear that now we cannot conclude $(r_1(y)\tilde{f}_1(y))' + (r_2(y)\tilde{f}_2(y))' = 0$, and thus, we cannot establish Lemma 4.12 for the transient case.

Remark 4.17. Finding an explicit solution when the source process has more than two states seems to be exceedingly difficult. To see this, suppose that $N = 3$. Lemma 4.12 enables us to reduce the number of differential equations in (4.22a) by one, leaving a two dimensional homogeneous linear system of differential equations with variable coefficients. For such systems, once one fundamental solution is known, the second (linearly independent) solution can be found by integration, see e.g. Lanczos (1997: p. 367). However, there is no standard theory on how to find the first solution.

Actually, for the two-dimensional system (4.23) we used the method indicated by Lanczos (1997) to compute the second solution, based on the first solution given in (4.24). This second solution turns out not to satisfy Lemma 4.12, as expected.

Remark 4.18. When λ and μ do not depend on y, say $\lambda(y) \equiv \lambda$ and $\mu(y) \equiv \mu$ as considered by Kella & Stadje (2002), it is not difficult to see that

$$\lambda \int_0^y e^{-g(x)} \frac{-1}{r_1(x)} dx - \mu \int_0^y e^{-g(x)} \frac{1}{r_2(x)} dx = e^{-g(y)} - 1,$$

so that by taking $y = B$ in the above, equation (4.28) yields

$$D_1(0) = \frac{\mu}{\lambda + \mu} \left[e^{-g(B)} + \int_0^B e^{-g(x)} \frac{\mu}{r_2(x)} dx \right]^{-1}.$$

After some algebra we can actually write (4.30) in the form

$$F_1(y) = \frac{\mu}{\lambda + \mu} \frac{e^{-g(y)} + H(y)}{e^{-g(B)} + H(B)},$$
$$F_2(y) = \frac{\lambda}{\lambda + \mu} \frac{1_{\{y=B\}} e^{-g(B)} + H(y)}{e^{-g(B)} + H(B)},$$

where

$$H(y) = \int_0^y e^{-g(x)} \frac{\mu}{r_2(x)} dx.$$

This coincides with Kella & Stadje (2002: Section 4).

4.7 Examples

To illustrate our results, we present two examples, each of which has $\mathscr{W} = \{1, 2\}$. In the first example, presented in Section 4.7.1, the rate at which the source turns on depends on the current buffer content. Second, the example of in Section 4.7.2 makes clear that the solution obtained for $B < \infty$ can sometimes be extended by taking the limit $B \to \infty$.

4.7.1 Discouraged Two-State Source

Suppose that the rate at which the source process changes states from state 1 to 2 depends on the content level: the higher the level, the less willing, or more 'discouraged', the source is to make a transition from state 1 to 2. One simple model for this behavior is to take

$$\lambda(y) = \lambda_0 \left(1 - \frac{y}{B}\right), \tag{4.31}$$

with $\lambda_0 > 0$. We assume $\mu(y) \equiv \mu$, and $r_1(y) \equiv -r_2(y) \equiv -r$, where μ and r are some positive constants. By Theorem 4.15

$$\mathbf{f}(y) = \frac{\lambda_0 D_1(0)}{r} e^{-g(y)} \left(1\,,\,1\right),$$

where $g(y)$ can be found from (4.25) to be

$$\begin{aligned} g(y) &= \frac{1}{r}\int_0^y \left(\mu - \lambda_0\left(1 - \frac{x}{B}\right)\right) dx \\ &= \frac{\mu - \lambda_0}{r} y + \frac{\lambda_0}{2rB} y^2. \end{aligned}$$

Hence we obtain for $\mathbf{F}(y)$, $0 \le y \le B$,

$$\begin{aligned} \mathbf{F}(y) = &\left(D_1(0)\,,\, D_2(B) 1_{\{y=B\}}\right) \\ &+ \frac{\lambda_0 D_1(0)}{r} \int_0^y \exp\left(-\frac{\mu - \lambda_0}{r} x - \frac{\lambda_0}{2rB} x^2\right) dx \left(1\,,\,1\right), \end{aligned}$$

where $D_2(B) = D_1(0) e^{-g(B)} \lambda_0/\mu$ and $D_1(0)$ follows from normalization. Notice that $\mathbf{F}(y)$ can also be expressed as

$$\begin{aligned} \mathbf{F}(y) = &\left(D_1(0)\,,\, D_2(B) 1_{\{y=B\}}\right) \\ &+ \frac{\lambda_0 D_1(0)}{r} \sqrt{2\pi}\,\sigma \exp\left(\mu^2/2\sigma^2\right) \left(\Phi_{\mu,\sigma^2}(y) - \Phi_{\mu,\sigma^2}(0)\right) \left(1\,,\,1\right), \end{aligned}$$

where Φ_{μ,σ^2} is the distribution function of a normal random variable with mean $\mu = B(1 - \mu/\lambda_0)$ and variance $\sigma^2 = rB/\lambda_0$.

4.7.2 Infinite-buffer Systems

In the entire analysis so far we assumed that $B < \infty$. The reason is that it seems difficult to find conditions for non-trivial $Q(y)$ and $R(y)$ such that a limiting distribution for the process $\{W(t), C(t), t \ge 0\}$ exists. In fact, let the vector $\pi(y)$ be the corresponding stationary distribution for $Q(y)$, i.e., the vector that satisfies $\pi(y)Q(y) = 0$ and $\sum_i \pi_i(y) = 1$. Then the following constitutes a plausible stability condition: for all

y larger than some $y_0 < \infty$, we must have $\sum \pi_i(y) r_i(y) < 0$. Obviously, this is not a necessary condition, since we can adapt the matrix function Q in some stable model such that above each level y_0 there are some (very small) regions of the buffer space where the condition does not hold, while the adapted model is still stable. However, this proposal is also not sufficient as we have a counterexample for a simple two-state model. Let us first compute the solution in case $B < \infty$, and then consider the limit $B \to \infty$. Note that this procedure seems reasonable when normalization is possible in this limit.

To define our model, set for $\mathscr{W} = \{1, 2\}$

$$\begin{aligned} \lambda(y) &= \frac{\lambda_0}{1+y} & \mu(y) &= \frac{\mu_0}{1+y} \\ r_1(y) &= -1 & r_2(y) &= 1, \end{aligned}$$

where λ_0 and μ_0 are positive constants. Now (4.25) yields

$$g(y) = \int_0^y \left(\frac{-\lambda_0}{1+x} + \frac{\mu_0}{1+x} \right) dx \; = \; (\mu_0 - \lambda_0) \log(1+y),$$

so that (4.29) becomes

$$\mathbf{f}(y) = D_1(0) \lambda_0 \left(\frac{1}{1+y} \right)^{\mu_0 - \lambda_0} \; \Big(1 \, , \, 1 \Big).$$

After some algebra we find

$$D_1^{-1}(0) = \frac{\mu_0 + \lambda_0 - 1}{\mu_0 - \lambda_0 - 1} - \frac{\lambda_0}{\mu_0} \frac{\mu_0 + \lambda_0 + 1}{\mu_0 - \lambda_0 - 1} \left(\frac{1}{1+B} \right)^{\mu_0 - \lambda_0 - 1}.$$

Clearly, when $\mu_0 > \lambda_0 + 1$, $D_1(0)$ has a finite limit when $B \to \infty$. In that case,

$$\mathbf{F}(y) = \left(\frac{\mu_0 - 1}{\mu_0 + \lambda_0 - 1}, \frac{\lambda_0}{\mu_0 + \lambda_0 - 1} \right) - \frac{\lambda_0}{\mu_0 + \lambda_0 - 1} \left(\frac{1}{1+y} \right)^{\mu_0 - \lambda_0 - 1} \; \Big(1 \, , \, 1 \Big).$$

For this particular model our previously mentioned condition $\sum \pi_i(y) r_i(y) < 0$ leads to $\mu_0 > \lambda_0$, while we just established that the correct condition must be $\mu_0 > \lambda_0 + 1$. One other aspect worth mentioning for this example is that when $\lambda_0 < \mu_0 < \lambda_0 + 1$ the normalization breaks down due to the fact that the integrals of the densities f_1 and f_2 become infinite for $B \to \infty$, and not because the value $D_2(B)$ does not approach zero (which in fact it does). Hence, the condition that $D_2(B) \to 0$ as $B \to \infty$ is a necessary, but not a sufficient condition for proper normalization, and hence for the stability of the infinite-buffer model.

As we have been unable to find any elegant stability conditions for the infinite buffer model, we do not include any theoretical results for this case, although the limiting procedure described above works well in many cases. As a matter of fact, we can also consider

the limit $B \to \infty$ for the example of Section 4.7.1. Here we expect that the influence of the factor $1 - y/B$ in (4.31) disappears. Indeed this is the case, resulting in the familiar expression for a fluid model without feedback,

$$\mathbf{F}(y) = \left(\frac{\mu}{\lambda_0 + \mu}, \frac{\lambda_0}{\lambda_0 + \mu} \right) - \frac{\lambda_0}{\lambda_0 + \mu} e^{\frac{\lambda_0 - \mu}{r} y} \Big(1 \, , \, 1 \Big).$$

4.8 Numerical Method

For the case when $N > 2$, explicit results cannot be expected (see Remark 4.17), and we wish to resort to numerical methods to obtain the stationary distribution. Because our problem (4.22) is not an initial boundary value problem, but a two-point boundary value problem, this is not entirely trivial. We discuss first a simple method to solve such problems, and then demonstrate its use by solving an extension of Example 4.7.1.

4.8.1 The Method

We start by rewriting the differential equation (4.22a) as $\mathbf{g}'(y) = \mathbf{g}(y)R^{-1}(y)Q(y)$, where $\mathbf{g}(y) = \mathbf{f}(y)R(y)$. With the fundamental matrix G, i.e., the matrix that satisfies

$$G'(y) = G(y)R^{-1}(y)Q(y), \tag{4.32}$$

we write the solution as $\mathbf{g}(y) = \mathbf{a}G(y)$, where $\mathbf{a}$ is a row vector to be determined based on the boundary conditions. In the following we will choose the matrix $G(y)$ such that $G(0) = I$, so that in fact $\mathbf{g}(y) = \mathbf{g}(0)G(y)$. Thus, the first step is to solve (4.32) numerically with $G(0) = I$, which can be done by standard methods.

The next step is to use the expression $\mathbf{g}(B) = \mathbf{g}(0)G(B)$ in the flux equations (4.22b) and (4.22c). In the present setting these become

$$\begin{aligned} \mathbf{D}(0)Q(0) = \quad \mathbf{f}(0+)R(0+) \; = \quad \mathbf{g}(0+), \\ \mathbf{D}(B)Q(B) = -\mathbf{f}(B-)R(B-) = -\mathbf{g}(B-) = -\mathbf{g}(0)G(B), \end{aligned}$$

so that substituting the first into the second yields $\mathbf{D}(B)Q(B) = -\mathbf{D}(0)Q(0)G(B)$, or

$$\mathbf{D}(0)Q(0) + \mathbf{D}(B)Q(B)G^{-1}(B) = \mathbf{0}. \tag{4.33}$$

To express the boundary conditions (4.22d) efficiently in matrix form we need projection operators I_- and I_+. The first operator is the projection on $\mathscr{W}_-$, i.e.,

$$(I_-)_{ij} = \begin{cases} 1 & \text{if } i = j \in \mathscr{W}_-, \\ 0 & \text{else.} \end{cases}$$

The second operator $I_+ = I - I_-$ evidently projects on $\mathscr{W}_+$. We are now ready to define the row vectors

$$\begin{aligned} \mathbf{D}_-(0) &= \mathbf{D}(0)I_- & \mathbf{D}_+(0) &= \mathbf{D}(0)I_+ \\ \mathbf{D}_-(B) &= \mathbf{D}(B)I_- & \mathbf{D}_+(B) &= \mathbf{D}(B)I_+, \end{aligned}$$

so that we can include the boundary conditions $\mathbf{D}_+(0) = \mathbf{0}$ and $\mathbf{D}_-(B) = \mathbf{0}$ in (4.33) to obtain

$$\mathbf{D}_-(0)Q(0) + \mathbf{D}_+(B)Q(B)G^{-1}(B) = \mathbf{0}.$$

To solve $\mathbf{D}_-(0)$ and $\mathbf{D}_+(B)$ from the above equation, we rewrite it by using the property of projection operators, $I_- = I_-^2$ and $I_+ = I_+^2$. Defining the ancillary vector $\mathbf{v} = \mathbf{D}_-(0) + \mathbf{D}_+(B)$, for which we also have

$$\mathbf{D}_-(0) = \mathbf{v}I_- \qquad \mathbf{D}_+(B) = \mathbf{v}I_+, \tag{4.34}$$

the problem is to solve $\mathbf{v}$ from

$$\mathbf{v}[I_-Q(0) + I_+Q(B)G^{-1}(B)] = \mathbf{0}.$$

This can be done, up to normalization, by the Singular Value Decomposition, see for instance Golub & van Loan (1989). Once $\mathbf{v}$ is known it is immediate, by (4.34), to find $\mathbf{D}_-(0)$ and $\mathbf{D}_+(B)$, up to normalization.

The last steps are to integrate the differential equation $\mathbf{g}'(y) = \mathbf{g}(y)R^{-1}(y)Q(y)$ with initial condition $\mathbf{g}(0) = \mathbf{D}(0)Q(0)$, to compute $\mathbf{f}(y) = \mathbf{g}(y)R^{-1}(y)$ and to normalize according to (4.22e).

Remark 4.19. Formally, the matrix $G(B)$ is invertible as it is a fundamental set of solutions, see e.g. Petrovski (1966) for a proof. However, *numerically* the state of affairs may be less agreeable as the problem (4.32) is principally ill-conditioned. Nevertheless, for relatively small buffer sizes and a small number of source states the above method can be successful.

4.8.2 Three Discouraged Sources

We use the numerical method above to compare a model with three independent identical discouraged sources—compare the source model of Example 4.7.1—to a model without feedback, namely the model of Anick *et al.* (1982) applied to three sources, with a finite buffer. The Q-matrices of interest in these models are denoted by $Q(y)$ and Q_{AMS},

respectively, and are given by

$$Q(y) = \begin{pmatrix} -3\lambda(y) & 3\lambda(y) & 0 & 0 \\ \mu & -2\lambda(y)-\mu & 2\lambda(y) & 0 \\ 0 & 2\mu & -\lambda(y)-2\mu & \lambda(y) \\ 0 & 0 & 3\mu & -3\mu \end{pmatrix}$$

$$Q_{\text{AMS}} = Q(0) = \begin{pmatrix} -3\lambda_0 & 3\lambda_0 & 0 & 0 \\ \mu & -2\lambda_0-\mu & 2\lambda_0 & 0 \\ 0 & 2\mu & -\lambda_0-2\mu & \lambda_0 \\ 0 & 0 & 3\mu & -3\mu \end{pmatrix},$$

where $\lambda(y) = \lambda_0(1 - y/B)$, as in Example 4.7.1. The drift matrices are equal for both models:

$$R = \begin{pmatrix} -L & & & 0 \\ & r-L & & \\ & & 2r-L & \\ 0 & & & 3r-L \end{pmatrix}.$$

To produce Figure 4.1 we set $\lambda_0 = 0.5$, $\mu = 1$, $r = 1$, $L = 0.5$ and $B = 2$. We conclude from the panels of the figure that the atoms at $y = B$ ($y = 0$) are larger (smaller) in the setting without feedback, as could be expected.

4.9 A Fluid Model of a TCP Source

The setting of the present chapter allows us to improve the TCP fluid model of Chapter 2 by taking into account the influence of buffering delay on the round-trip times.

Recall that in this model one TCP source with window size $W \in \{1, \dots, N\}$ sends fluid into a buffer of size B served by a link with constant capacity L. Instead of the generators defined by (2.4) and (2.6) in which the transition rates λ and μ of (2.7) and (2.8), respectively, are constant, we now use content-dependent rates in accordance with (1.47a). Thus, we take

$$\lambda^{-1}(y) = T(y) \equiv T + \frac{y}{L},$$

where T is the mean propagation time and y/L the queueing delay.

Now, when $y < B$ let $Q(y)$ be as Q in (2.4) but replace λ by $\lambda(y)$, and let $Q(B)$ be as $\widetilde{Q}$ in (2.6) but replace μ by $\lambda(B)$. Hence, the generator is not continuous, but piecewise continuous. Furthermore, we take the drift matrix as in (2.2) with the function

$$r(y) = \frac{P}{T(y)}, \tag{4.35}$$

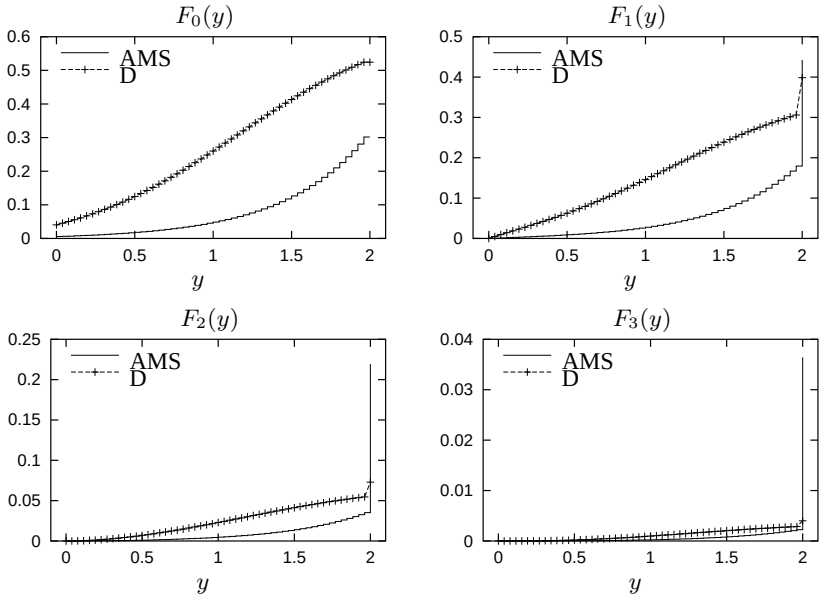

Figure 4.1: *The distribution functions for the AMS model and the discouraged users model (D). Note the different scales on the vertical axes.*

replacing the constant r. Thus, $r(y)$ corresponds to the (fluid) transmission rate required to transmit one packet of size P during one round-trip time, so that if the window size at time t is n, the source rate equals $nr(y) = nP/T(y)$, in accordance with (1.47d).

It remains to ensure that our assumption 3 on the drift functions is met, i.e., the model parameters P, T, N, L and B should be such that $|nr(y) - L| \geq \epsilon > 0$ for all $(n, y) \in \mathscr{S}$. As in Figure 2.2 define the set $l = \{(x, y) \in \mathbb{R}^2 \,|\, xr(y) = L\}$. This set should not intersect the system state space $\mathscr{S}$. Clearly, in Figure 2.2 l is a vertical line so that it will never have a point in common with $\mathscr{S}$ when L/r is not equal to an integer. Here, because of (4.35), l is still a line, but no longer parallel to the vertical axis. Hence, when the system parameters have the 'wrong' value, l will cut (at least) one of the sets $i \times [0, B]$.

While we now have a fairly complete TCP fluid model, we do not further investigate the consequences of this model. The first reason is that the models to be introduced in the next two chapters are even more flexible and easier to handle numerically; we prefer to compare these models to simulation. Second, the numerical evaluation of this model is possible for only a rather limited parameter range because of the problems mentioned in the previous paragraph, i.e., l intersecting $\mathscr{S}$.

4.10 Conclusions

Fluid queues with continuous feedback have many useful applications, e.g., dams subject to maintenance or manufacturing networks. On the other hand, the model is still not as versatile as we desire for the purpose of modeling TCP. For instance, it would be interesting to extend the TCP model of Section 4.9 such that we can analyse multiple TCP sources sharing two buffers. Whereas this appears possible in principle, it will be troublesome to carry out in practice. In the next two chapters we approach the problem of modeling TCP in another way. We simply discretize the system state space $\mathscr{S}$ and model the behavior of the source(s) and the buffer(s) as a continuous-time Markov *chain* with suitable infinitesimal generator.

Chapter 5

A Discretized Fluid Model for Asymmetric TCP Sources

The goal of this chapter is to develop a stochastic, Markovian model of the interaction between TCP sources and a bottleneck link to study fairness and utilization. This model is as flexible as the one developed by Misra *et al.* (2000) but preserves the stochasticity of the source-buffer interaction allowing us to derive *probability distributions* of the source and buffer processes, rather than merely *expectations* as in the deterministic model of Misra *et al.* (2000). Moreover, our stochastic model does not need some of their mathematical simplifications. With respect to the dependence of the throughput on parameters as packet size, round trip time, and buffer size the results of our model are consistent with those of earlier models, e.g., Brown (2000), and therefore not reported here. A drawback of our model is its lack of scalability to large numbers of sources and routers. Therefore we restrict the numerical analysis to two sources sharing a fluid drop-tail buffer. In Chapter 6 we generalize this class of models to networks of intermediate size, that is, networks consisting of a few connections and routers.

In Section 5.1 we develop the stochastic model of the source-buffer interaction, and present some of the results, among which a comparison to simulations with ns-2, in Section 5.2. Section 5.3 concludes.

5.1 Model

In Section 5.1.1 we discuss the TCP fluid model of Misra *et al.* (2000) in considerable more detail than in Section 1.3.2, and point out a few weaknesses. By trying to bypass these we introduce in Section 5.1.2 a Markov chain model of the source-buffer interaction. In Section 5.2.1 we define performance measures for the sources in terms of the

steady-state distribution of the Markov chain.

5.1.1 Discussion of a Deterministic TCP Fluid Model

Here we apply the model of Misra *et al.* (2000) to a network of J greedy TCP sources that share a (fluid) drop-tail buffer of size B served by a link with capacity L. Then we discuss some consequences of the simplifications they introduce to obtain a numerically tractable model. (Although, strictly speaking, the model of Misra *et al.* (2000) applies to a RED buffer, it is simple to reduce this to a drop-tail buffer by setting $x_{\min} = x_{\max} = B$ in (1.36) and $\epsilon = 1$ in (1.35).)

Let us recall the dynamics of the source and buffer process as modeled by (1.47a), (1.47c) and (1.48). Suppose that T_i is the round-trip time of source i, $1 \leq i \leq J$, when the buffer is empty. Then,

$$T_i(q(t)) = T_i + \frac{q(t)}{L} \tag{5.1}$$

is the round-trip time of source i when the buffer content is $q(t)$ at time t. Source i maintains a window variable $W_i(t)$, supposed to be continuous, and sends fluid at rate $W_i(t)/T_i(q(t))$ into the buffer. Let

$$r(t) = \sum_{i=1}^{m} \frac{W_i(t)}{T_i(q(t))} - L.$$

Then the evolution of the queue length is given by, cf. (1.21),

$$\frac{dq}{dt} = \begin{cases} \max\{r(t), 0\}, & \text{if } q(t) = 0, \\ r(t), & \text{if } q(t) \in (0, B), \\ \min\{r(t), 0\}, & \text{if } q(t) = B. \end{cases} \tag{5.2}$$

Finally, Misra *et al.* (2000) model the window dynamics as

$$dW_i(t) = \frac{dt}{T_i(q(t))} - \frac{W_i(t)}{2} dM_i(t). \tag{5.3}$$

The first term of the right hand side corresponds to the Additive-Increase behavior of a source. The second term implements Multiplicative-Decrease at a loss epoch. Here, $M_i(t)$ models the loss arrivals as a point process, so that $dM_i(t) = 1$ at the arrival of a loss and 0 elsewhere.

At this stage Misra *et al.* (2000) take expectations of the left and right hand sides of (5.3). To simplify the analysis they assume that

$$\mathbb{E}\{W_i(t)dM_i(t)\} = \mathbb{E}\{W_i(t)\}\,\mathbb{E}\{dM_i(t)\} = \mathbb{E}\{W_i(t)\}\,\lambda_i(t)dt, \tag{5.4}$$

where $\lambda_i(t)$ is the time-varying rate of the loss process. Then, they argue that

$$\lambda_i(t) = 1_{\{\bar{q}(t-\tau_i(t))>B\}} \frac{\bar{W}_i(t-\tau_i(t))}{T_i(\bar{q}(t-\tau_i(t)))}, \tag{5.5}$$

where $\bar{X} = \mathbb{E}\{X\}$ for the random variables involved and $\tau_i(t) = T_i(\bar{q}(t))$ is the feedback delay, i.e., the difference in time between the moment loss occurs and the moment source i takes notice of this loss.

With these approximations the window dynamics specified by (5.3) reduce to the following differential equation for $\bar{W}_i$:

$$\frac{d\bar{W}_i}{dt} = \frac{1}{T_i(\bar{q}(t))} - \frac{\bar{W}_i(t)}{2} \frac{\bar{W}_i(t-\tau_i(t))}{T_i(\bar{q}(t-\tau_i(t)))} 1\{\bar{q}(t-\tau_i(t)) > B\}, \tag{5.6}$$

Misra et al. solve the system of differential-algebraic equations (5.1), (5.2) and (5.6) numerically and obtain information about the expected transient behavior of, for instance, the queue.

The above model is very flexible indeed: it allows to study the effects of source heterogeneity; it extends nicely to large networks; and it includes the influence of the queue on the round-trip times so that the sub-linearity of the source transmission rate process is captured, cf. Altman *et al.* (2000a). Nevertheless we see some fundamental points in which the model might be improved.

In the first place, by taking expectations, much probabilistic information is lost. Consequently, obtaining expressions for, say, the variance of the throughputs is problematic. As a further consequence of merely considering averages it is difficult to study the influence of on/off behavior of sources. Consider, for instance, a file transfer. The probability that a file ends within t seconds, say, changes as a function of the sending rate of a source. To model this correctly, it is necessary to keep track of a source's momentary transmission rate rather than its average rate.

Second, their approximation (5.4) implies that $W_i(t)$ and $dM_i(t)$ are uncorrelated (not independent as they write). This is strange as in (5.5) $\lambda_i(t)\,dt = \mathbb{E}\{dM_i(t)\}$ is a function of (the expectation of) $\{W_i(t)\}$. Another technical point is that in the derivation of (5.4) Misra et al. approximate $\mathbb{E}\{f(X)\}$ by $f(\mathbb{E}\{X\})$, where f is some function and X some random variable. This is not entirely correct, as Misra *et al.* (2000) also point out.

The third problem is due to the feedback delay, see (5.6), as the content $\bar{q}$ may now exceed the buffer size B. In the case of drop-tail buffers this is clearly impossible. This point is less problematic when B corresponds to a buffer threshold, which Misra *et al.* (2000) consider, instead of the buffer size itself.

The last problem relates to a somewhat technical aspect of the analysis. Misra *et al.* (2000) do not mention how to *compute* the expectations that are taken. (In other words, the probability space is not provided.)

Our model does not suffer from these problems. More specifically, concerning the first point our stochastic model maintains a notion of the momentary window size and buffer content so that the momentary (fluid) transmission rate is known. About the second point, loosely speaking, we *first* solve the system and *then* take expectations, whereas Misra *et al.* (2000) take expectations first and then solve the system. Reversing this ordering is not a mere technicality; we do not have to resort to the same type of simplifications as made to arrive at (5.6). We circumvent the third objection by implementing feedback delay in a somewhat different manner, which we explain below. Finally, concerning the last comment, we also take expectations, but with respect to the stationary distribution of a Markov chain, so that no problems about the interpretation remain. We remark, once again, that as a consequence of our approach our model does not scale well to networks of sizes studied by Misra *et al.* (2000).

5.1.2 A Stochastic TCP Model

In the stochastic model to be introduced now, we aim to avoid the drawbacks of the deterministic model discussed in Section 5.1.1. The central idea is to discretize the source and queue processes and model the joint source-buffer process as a continuous-time Markov chain. The resulting Markov chain shares all of the modeling assumptions and many of the features of the model of Misra *et al.* (2000); in fact most of (5.1), (5.2) and (5.3) carries over. However, we do not take expectations, i.e., we do not arrive at (5.6), but compute the steady-state probabilities of the Markov chain by means of the infinitesimal generator matrix. As we have this matrix at our disposal, we can study transient properties as well.

As an aside, we refer to Adan & Resing (2000) who apply a similar discretization approach to facilitate the study of a two-level traffic shaper, a model analyzed in Kroese & Scheinhardt (2001) by means of Laplace transforms.

To begin, we modify the (continuous) content process described by (5.2) to a corresponding *discrete* process $\{C(t)\} \equiv \{C(t), t \geq 0\}$ with state space $\mathscr{K} = \{0, 1, \ldots, K\}$. The window process $\{W_i(t)\} \equiv \{W_i(t), t \geq 0\}$ of source i is a discrete process with state space $\mathscr{W}_i = \{1, 2, \ldots, N_i\}$. Here N_i denotes the maximum window of source i. The joint process $\{\mathbf{W}(t), C(t)\}$ has state space

$$\mathscr{S} = \left(\Pi_{i=1}^{J} \mathscr{W}_i\right) \times \mathscr{K}.$$

When $C(t) = k$, $k \in \mathscr{K}$, the corresponding buffer content equals kB/K. Consequently, the round-trip time for source i becomes, compare (5.1):

$$T_i(k) = T_i + \frac{k}{K}\frac{B}{L}. \tag{5.7}$$

Given the round-trip time, when $W_i(t) = n_i$ the source sends traffic at rate $n_i\, P_i/T_i(k)$.

We abbreviate this to $n_i\, r_i(k)$ where

$$r_i(k) = \frac{P_i}{T_i(k)}, \tag{5.8}$$

cf. (2.9). Clearly, the source peak rate is $r_i(0)N_i$.

Observe that if the buffer content were a continuous variable, the drift at time t would be, cf. (5.2),

$$\mathbf{r}(k) \cdot \mathbf{n} - L \equiv \sum_{i=1}^{J} r_i(k)\, n_i - L, \tag{5.9}$$

where $\mathbf{r}(k) = (r_1(k), \ldots, r_J(k))$ and $\mathbf{n} = (n_1, \ldots, n_J)$.

We model the joint process $\{\mathbf{W}(t), C(t)\}$ as a continuous-time Markov chain and use the differential behavior of (5.2) and (5.3) to specify the generator Q of the process. To obtain the transition rates, we introduce the following notation:

$$\mathbf{n}^i = (n_1, \ldots, n_i + 1, \ldots, n_J)$$
$$\mathbf{n}_i = (n_1, \ldots, \lfloor n_i/2 \rfloor, \ldots, n_J),$$

where $\lfloor x \rfloor$ is the integer part of x when $x > 1$. Further, let

$$\beta(\mathbf{n}, k) = \frac{K}{B}(\mathbf{r}(k) \cdot \mathbf{n} - L) \tag{5.10}$$

be the transition rate at which in- and decrements of the buffer level process occur. Observe that this is K/B times the drift (5.9) of the buffer. The constant K appears in the numerator to ensure that the average time required to drain a full buffer, given that $\mathbf{W}(t)$ does not change state, is approximately independent of K. (Loosely stated, when K is large, the average time at one buffer level should be short.) Finally, let the rate at which the window size in- or decreases be given by

$$\lambda_i(k)) = \frac{1}{T_i(k)}. \tag{5.11}$$

Now, writing $\mathbf{x} > \mathbf{y}$ to mean that $x_i > y_i$ for all $1 \le i \le J$, we define the elements of Q as:

$$q_{\mathbf{n}k;\mathbf{n},k+1} = \beta(\mathbf{n}, k), \qquad \text{if } 0 \le k < K \text{ and } \mathbf{n} \cdot \mathbf{r}(k) > L, \tag{5.12a}$$
$$q_{\mathbf{n}k;\mathbf{n},k-1} = -\beta(\mathbf{n}, k), \qquad \text{if } 0 < k \le K \text{ and } \mathbf{n} \cdot \mathbf{r}(k) < L, \tag{5.12b}$$
$$q_{\mathbf{n}k;\mathbf{n}^i k} = \lambda_i(k), \qquad \text{if } 0 \le k < K \text{ and } n_i < N_i, \tag{5.12c}$$
$$q_{\mathbf{n}K;\mathbf{n}_i K} = \frac{n_i r_i(K)}{\sum_j n_j r_j(K)} \lambda_i(K), \qquad \text{if } n_i > 1, \tag{5.12d}$$
$$q_{\mathbf{n}k;\mathbf{m}l} = 0, \qquad \text{elsewhere.} \tag{5.12e}$$

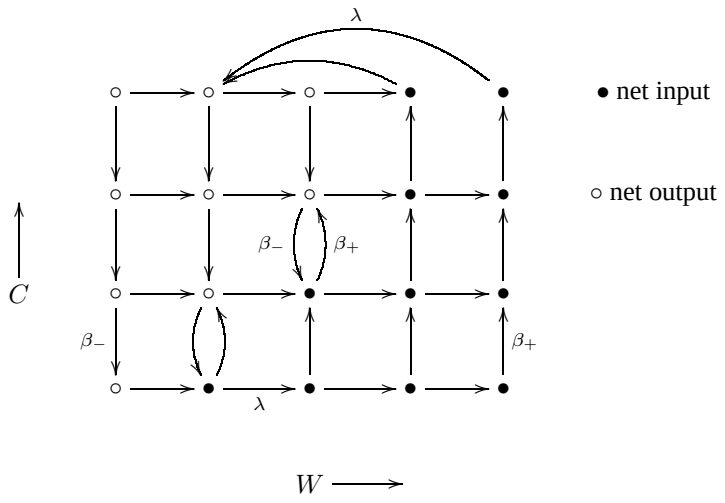

Figure 5.1: *Graphical representation of Q for a model with $N = 5$ and $K = 4$. Here we use the shorthands $\beta_- = -\beta$ and $\beta_+ = \beta$, respectively, where β is as in* (5.10)*.*

Equation (5.12a) and (5.12b) are the discrete analogs of (5.2). Equation (5.12c) corresponds to the Additive-Increase part of (5.6), and (5.12d) implements Multiplicative-Decrease with *proportional loss*. (Recall that in the synchronous loss model, when a loss event occurs all connections simultaneously reduce their window size by a factor two. In the proportional model just one source suffers and the probability of losing a packet is proportional to the source's window size.) We remark that the condition $n_i > 1$ implies that a source cannot make any downward transition while in state 1. Note furthermore that on average the source spends an amount $T_i(k)$ in state i, given $C(t) = k$. In this way our model incorporates feedback delay. Figure 5.1 shows a graphical example for Q in which one source with $N = 5$ uses a buffer with $K = 4$.

Note also that the source makes one downward transition 'per congestion signal'. Thus, as explained below (2.9), the source models TCP NewReno or TCP Sack.

Clearly, a buffer cannot be full (empty) when the drift is negative (positive). Hence, to ensure that the sets

$$\{(\mathbf{n}, k) \in \mathscr{S} \,|\, \mathbf{n} \cdot \mathbf{r}(K) < L, k = K\} \text{ and } \{(\mathbf{n}, k) \in \mathscr{S} \,|\, \mathbf{n} \cdot \mathbf{r}(0) > L, k = 0\} \quad (5.13)$$

have zero probability in the steady state limit, we modify some entries in Q so that no transitions into these undesired states exist (in the limit $t \to \infty$, states without influx have zero probability). This modified generator is used in the sequel of the chapter.

We assume that the parameter values are chosen such that, whereas part of the state space $\mathscr{S}$ of $\{\mathbf{W}(t), C(t)\}$ may be transient, the complement forms a closed (possibly

proper) subset of $\mathscr{S}$. We further assume that this subset contains more than one point, and at least one point such that $C = K$. Then the distribution of $\{\mathbf{W}(t), C(t)\}$ converges for $t \to \infty$ to a non-trivial distribution $\boldsymbol{\pi} = \pi(\mathbf{n}, k)$ satisfying $\boldsymbol{\pi}Q = 0$. Let the random variables $\{\mathbf{W}, C\}$ have distribution $\boldsymbol{\pi}$, i.e.,

$$\mathbb{P}\{\mathbf{W} = \mathbf{n}, C = k\} = \pi(\mathbf{n}, k).$$

Whereas the Markov chain with generator Q specified by (5.12) implements proportional loss, the TCP model with synchronous loss is slightly different. To capture synchronized loss we augment, as in Section 3.1, the process $\{\mathbf{W}(t), C(t)\}$ with indicator variables $\mathbf{I}(t)$ and study the process $\{\mathbf{W}(t), \mathbf{I}(t), C(t)\}$. The indicator variable I_i reflects the 'congestion state' of source i. When $I_i(t) = 0$, source i is allowed to increase its sending rate, while when $I_i(t) = 1$, source i is recovering from a packet loss. Once a source has reduced its sending rate after a loss, its congestion variable changes to 0 again. To implement synchronized loss, we set $I_i(t) = 1$ for all source $i = 1, \ldots, J$ when $C(t)$ becomes equal to K. Since, in general, the round-trip times of the sources are different, the epochs at which I_i changes from state 1 to 0 will be different.

We close this section by comparing the deterministic model of Section 5.1.1 to the stochastic model of this section. The stochastic model overcomes the drawbacks mentioned in Section 5.1.1. As such it can track the source states more accurately: it includes stochasticity and does not need approximations such as (5.4). However, due to state space explosion it does not scale to large networks. To study these cases the deterministic model is more suited, although the question remains how well the deterministic model captures source behavior, as it is not completely clear to what extent the approximations leading to (5.6) are valid. A second point of criticism of the stochastic model might relate to the exponentiality of the times between consecutive transmissions. This is not a fundamental problem, as reducing the variance by implementing a three stage Erlang distribution, for instance, is, at least in principle, simple for the present model.

5.2 Results

With the above Markovian model we investigate three aspects of TCP. First, we explore the validity of the synchronized and proportional loss model as introduced in Section 1.3.2. We implement both loss models and compare, in Section 5.2.3, the results to those obtained with ns-2 simulations. The second aspect, which we consider in Section 5.2.4, concerns an evaluation of how well the root p law (1.40) performs for the Markovian model. Third, the fact that the model is stochastic allows us, in Section 5.2.5, to include the influence of the application layer on TCP such as a source switching on and off with rates depending on think-time, file size and momentary source transmission rate, respectively.

Let us start with defining the relevant performance measures in the next section and present the network set-up in Section 5.2.2.

5.2.1 Performance Measures

We define three steady state performance measures. The average transmission rate of source i is analogous to (3.17):

$$\tau_i = \mathbb{E}\{r(C) \cdot \mathbf{W}\} \tag{5.14}$$

where the expectation is computed with respect to $\boldsymbol{\pi}$. The definition of the throughput γ_i is based on the assumption that during a loss event a source loses fluid in proportion to its rate, cf. (3.18). Therefore,

$$\begin{aligned} \gamma_i &= \tau_i - \mathbb{E}\left\{ (\mathbf{r}(C) \cdot \mathbf{W} - L) \, \frac{r_i(C) W_i}{\mathbf{r}(C) \cdot \mathbf{W}} 1_{\{C=K\}} \right\} \\ &= \tau_i - \sum_{\mathbf{n}} (\mathbf{r}(K) \cdot \mathbf{n} - L) \frac{r_i(K) n_i}{\mathbf{r}(K) \cdot \mathbf{n}} \pi(\mathbf{n}, K). \end{aligned} \tag{5.15}$$

Note that as $\pi(\mathbf{n}, K) = 0$ if $\mathbf{n} \cdot \mathbf{r}(K) < L$ (by the boundary conditions on $\boldsymbol{\pi}$) the states of the left set of (5.13) do not contribute to the loss.

The above definition is not consistent with the proportional loss model in the following sense. According to this loss model *just one* source observes loss during a congestion epoch, whereas in the throughput definition (5.15) *both* sources lose a fraction of their traffic proportional to their sending rate during congestion. The influence of this inconsistency is relatively small as mostly the source that suffers from loss will be the one with the highest transmission rate, and consequently, with the largest loss fraction. Moreover, the time spent in congestion is relatively small. Removing this inelegant aspect is an onerous task in the present setting as the corrections have to be implemented by hand in an appropriate generator matrix. In Section 6.2.1 we slightly change the involved processes and use a more suitable method to obtain the related generator matrix. With this other approach it is much easier to reduce the influence of this inconsistency.

Finally, we define the utilization for source i as $u_i = \frac{\gamma_i}{L}$.

5.2.2 Network Configuration and Parameters

Figure 5.2 shows the network configuration and the parameters used for both the model and the simulation. Two greedy TCP NewReno sources, S_1 and S_2, communicate with destinations D_1 and D_2 respectively, via router R_1. The receiver windows are large enough to not constrain the windows. We vary the propagation delay d_1 of the link connecting S_1 and R in 10 steps from 40 ms to 120 ms so that $T_1 = 2(d_1 + 10)$ ms. For source 2 we fix the round-trip time to $T_2 = 2(120 + 10) = 260$ ms.

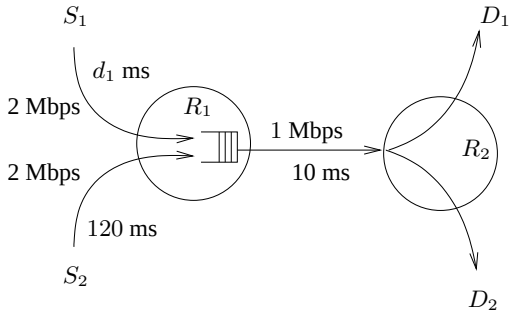

Figure 5.2: *The network configuration and some parameters.*

Concerning the model, the choice of N_i, r_i, L, and B requires some coordination. First, the ratio B/L determines the maximal buffering delay. So, given L, we set B such that $B/L = 100\text{kb}/(1\text{Mb/s}) = 100\text{ms}$. Second, the number $\alpha_i = N_i r_i/L$ is the maximal fraction of the link capacity that can be filled by source i. Given α_i and L, the source granularity is controlled by N_i (from which $r_i = \alpha_i L/N_i$ follows). Finally, in the simulation each source can congest the link, so that $\alpha_1, \alpha_2 \geq 1$. The values $N_1 = N_2 = 26$, $r_1 = r_2 = 1$, $L = 25.7$, and $B = 2.57$ satisfy the above constraints. (We take $N_i = 26$ as we want rather fine-grained sources. The choice $r_1 = r_2 = 1$ fixes then, more or less, L and B.)

It remains to choose the value of the 'grid' parameter K in the model. It is plausible that for $K \to \infty$, the functions $\mathbb{P}\{W = i, C/K \leq y\}$ converge to functions in which the content process has a continuous state space such as in (2.10). Here we take K such that the buffer discretization works well, in the sense that the performance measures do not change much when we increase K any further. It turns out that $K = 5$, a relatively small number, is already large enough for the parameter ranges we investigate in this chapter.

In the simulation the parameters besides those specified in Figure 5.2 are presented in Table 5.1. For motivation behind the RED parameters we refer to Altman *et al.* (2000b).

Observe that in the simulation we use a RED buffer whereas the buffer in the model is of a drop-tail type. On the face of it, this is inconsistent with the simulated network. As a motivation for using RED in ns-2 we follow an argument of Altman *et al.* (2000b). It is commonly seen in simulations with two sources sharing one drop-tail buffer that sometimes one and sometimes both sources lose packets during a congested period. Thus, at least in simulations, bursts at the packet level determine which source(s) lose(s) traffic in case a drop-tail buffer overflows. However, such rapid fluctuations at the packet level are absent in the context of fluid sources. Thus, a fluid source never perceives a 'true' drop-tail buffer. As such, comparing fluid models to simulations with drop-tail buffers will not be appropriate. As RED is a queue management technique that can effectively absorb

these rapid queue-length fluctuations, it is apt to use RED buffers in the simulations even when the modeled fluid buffer is a drop-tail buffer.

$$x_{\min} = 22 \text{ packets} \quad p_m = 0.1 \quad P = 576 \text{ B}$$
$$x_{\max} = 27 \text{ packets} \quad \epsilon = 0.002$$

Table 5.1: *Some parameters used in the simulation with ns-2. The choice for* $x_{\min}$ *corresponds to a buffering delay of* 100 *ms.*

We remark here that as a consequence of using RED in the simulations, packets will be dropped with a probability proportional to the sending rate of a source. Thus, our proportional loss model seems the more appropriate to compare against the simulations.

5.2.3 Comparison of Proportional and Synchronized Loss

The loss process of a buffer is complex to characterize, especially when multiple sources share a buffer. Subtle effects at packet level, for instance due to the ack-clock mechanism, decide whether a (and which) packet is dropped at a buffer. In this section we investigate the validity of the synchronized and proportional loss model by considering two sources that share one bottleneck buffer as in Figure 5.2. We implement both loss models in the framework of Section 5.1.2 and compare the theoretical throughputs to results obtained from a simulation with two TCP NewReno sources in ns-2.

Figure 5.3 contains two panels presenting the results of the (numerical) investigations. The left (right) panel shows the results of the proportional (synchronized) loss model and simulation. In both panels, $s = T_1/T_2$ runs along the horizontal axis. Along the vertical axis we set out the utilizations u_1 and u_2 obtained by the model. The corresponding simulation results are indicated by $\tilde{u}_1$, $\tilde{u}_2$.

We see from the left panel (proportional loss) that the model correctly predicts the trend in the bias toward longer connections. However, it underestimates the influence of delay on fairness. In the right panel (synchronous loss) we observe the same bias but the capacity is shared less fairly than in the simulation. Thus, the two loss models are at either extreme of the (simulated) reality.

One explanation for these observations could be as follows. In the proportional loss model just one source loses traffic during periods with congestion, whereas in the synchronous model both sources always lose packets. However, in the simulation sometimes just *one* suffers from loss, and sometimes *both*. Hence, the loss process of a RED buffer is neither strictly proportional nor strictly synchronized, so that the utilization obtained by simulation should be in between the results of the two models, which is the case. We therefore infer that a more detailed model of the loss process might yield better resemblance to the simulation results. Hence, the claim at the end of Section 5.2.2 is not

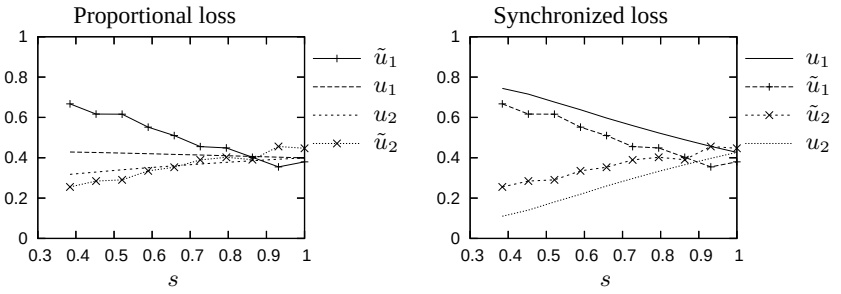

Figure 5.3: *A comparison of the utilizations as a function of $s = T_1/T_2$ obtained by the ns-2 simulator and the model. The results from the model (simulation) are labeled by u_1 and u_2 ($\tilde{u}_1$ and $\tilde{u}_2$).*

supported by the results.

In the sequel of the chapter we use the synchronized loss model, but, as the above indicates, we might as well use the proportional loss model. The fairness results obtained from the model will therefore show stronger bias with respect to differences in round-trip times than those obtained from simulations with ns-2.

5.2.4 A Root p Law

It is interesting to check the validity of the root p formula (1.40) in the context of this model, analogous to Section 2.3.4. The form of the root p law for source 1 is according to Mathis *et al.* (1997)

$$\gamma_{M,1} = \frac{P_1}{\mathbb{E}\{T_1\}}\sqrt{\frac{3}{2p_1}}. \tag{5.16}$$

Similar expressions hold for source 2.

Clearly, to compute (5.16) for the model we need expressions for the packet size P_1 and the loss probability p_1. The first follows from (5.8), hence,

$$P_1 = r_1 T_1(0).$$

An expression for p_1 follows from the observation of Mathis *et al.* (1997) that p_1 should not be the packet loss rate itself, but rather the number of negative (congestion) signals per acknowledged packet. Assuming that the loss is small, we take

$$p_1 = \lim_{t\to\infty} \frac{\text{Number of negative signals sent in } [0,t]}{\text{Number of transmitted packets in } [0,t]}.$$

Now, the number of negative signals sent during $[0,t]$ is the same as the time the source spent in congestion during $[0,t]$ divided by the average time in a congested state. Since

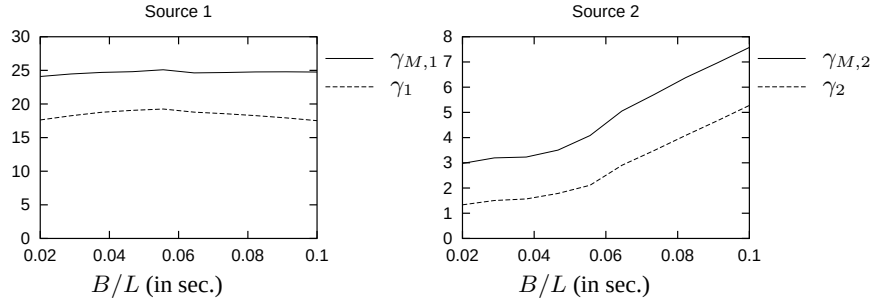

Figure 5.4: *The throughputs of both sources as a function of buffering delay. (The parameters are chosen as for Figure 5.3 except that $T_1 = 100$ ms, $T_2 = 520$ ms, and N_1, N_2 such that each source can fill $3/4$ of the link.)*

the latter is $T_1(K)$, we obtain, cf. (2.25),

$$\begin{aligned} p_1 &= \lim_{t\to\infty} \frac{\text{Time spent in congestion in } [0,t]/T_1(K)}{\text{Fluid transmitted in } [0,t]/P_1} \\ &= \frac{P_1}{T_1(K)} \frac{\text{Fraction of time in congestion}}{\text{Fluid transmission rate}} \\ &= \frac{r_1 T_1(0)}{T_1(K)} \frac{\mathbb{P}\{I_1 = 1\}}{\tau_1}. \end{aligned} \tag{5.17}$$

Note that p_i, $i = 1, 2$, is an endogenous variable of the model; as such, it cannot be directly controlled. To change its value, we vary B instead, and compute p_i and the other performance measures as a function of B/L. Note furthermore that, owing to the dependence of the drift function on the buffer content, the notion of packet size is here less cumbersome than (2.9).

We plot the results of a numerical evaluation of the above in Figure 5.4. Clearly, the 'root-p-throughputs' $\gamma_{M,i}$ overestimate the model throughputs γ_i by about a factor 1.5 for source 1 and a factor 2 for source 2, but they capture the trends quite accurately.

5.2.5 On/Off Behavior

So far we have assumed that the sources are greedy. This assumption, however, is obviously never satisfied in reality. Rather, a source usually switches off after the delivery of a file, and switches on after some time when a new request arrives. In this section we explore the influence of on/off behavior on fairness.

To implement this type of behavior for source 2 (source 1 is still greedy) we allow it to switch off by adding an element 0 to $\mathscr{W}_2$, representing the off state, and transitions from any state $W_2 > 0$ directly to state $W_2 = 0$. Suppose then that source 2 sends files

with exponentially distributed size of average size S. The rate λ_{off} at which source 2 switches off clearly depends on the source's sending speed. When $W_2(t) = n_2$ and $C(t) = k$ we set $\lambda_{\text{off}} = n_2 r_2(k)/S$. To see that this is correct for all states with $n_2 > 0$, assume first that $W_2(t)$ and $C(t)$ do not change before the file has been transferred. Then the expected on-time $\lambda_{\text{off}}^{-1}$ equals S divided by the momentary transmission rate $n_2 r_2(k)$ of source 2. Suppose, now, that the transmission rate of source 2 increases during the transfer. We may, by the memoryless property of the exponential file size distribution, just restart the transfer process of the file at the moment the source makes the transition. The expected file size, then, is again S. Once the transfer is completed, source 2 switches on again after an exponentially distributed time with rate λ_{on}, and starts in Congestion Avoidance with window size 1.

Figure 5.5 shows the effect of finite file sizes. The variable $\tilde{u}_1$ ($\tilde{u}_2$) represents the utilization of source 1 (source 2) *given* that source 2 is on, whereas $u_i = \gamma_i/L, i = 1, 2$ is the unconditional utilization. When $\lambda_{\text{on}} = 1$ sec^{-1}, corresponding to the right panel, source 2 switches on relatively quickly, i.e., within a few round trip times ($T_2 = 260$ ms). Hence, source 1 does not have much time to benefit from off-periods of source 2. The reason source 1 is better off is that when source 2 switches off it always restarts with a window of size 1. Loosely stated, source 2 suffers from the fact that it does not constantly participate in the competition for bandwidth, and, once it switches on, it starts from rather unfavorable conditions. When the file size increases, the bias becomes less. We also see that the overall utilization, $u = u_1 + u_2$, decreases in the presence of fierce competition for bandwidth.

In the left panel, $\lambda_{\text{on}} = 0.1$ sec^{-1}, so that source 2 is off for longer periods. The utilization u_1 (u_2) is in this case higher (lower) as compared to the right panel, since source 2 is less active. However, comparing the left and right panel, we see that the conditional utilizations are nearly the same. Clearly, the amount of capacity that source 2 can claim while it is on, hardly depends on λ_{on}, as is to be expected.

5.3 Summary and Conclusions

We developed a Markovian model of the interaction between AIMD sources and a drop-tail fluid buffer. The source and buffer behaviors are strictly stochastic. Due to its flexibility the model enables us to obtain qualitative insight in the influence of various source and network parameters on long-term properties such as source throughput, link utilization and fairness. With respect to the dependence of the throughput on parameters as packet size, round trip time, and buffer size the results of our model are consistent with those of earlier models, e.g., Brown (2000), and therefore not reported here.

We applied the model to three specific cases. (1) We implemented two popular loss mechanisms, viz. proportional loss and synchronized loss, and compared each to a simu-

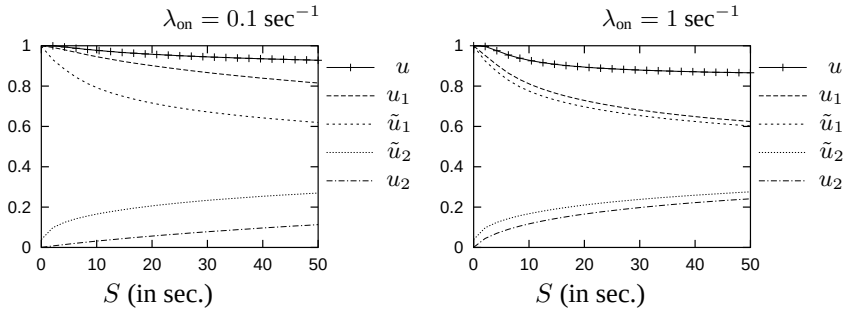

Figure 5.5: *Utilizations when source 2 is an on/off source, whereas source 1 is greedy. (The parameters are the same as for Figure 5.4.)*

lation in ns-2. Is is shown that the loss process is somewhere in between the proportional and synchronized loss model. (2) We show that the root p law also holds in our context, where the loss process is endogenous instead of exogenous. (3) From the investigation of the influence of source on/off behavior on fairness we conclude that a source that switches off often is not denied access to the link, but is capable of claiming its fair share quite quickly when it switches on, provided file-sizes are not (too) small.

One problematic aspect of the present approach is that we have to implement the generator ourselves in computer code. Although conceptually simple, carrying out this task is difficult as it is error-prone. It would be better to specify the source and buffer behavior in a more human-friendly manner, and let the computer carry out the task of deriving the generator. One such approach is developed in the next chapter.

Chapter 6

SPN Models for Networks with Asymmetric TCP Sources

The model developed in Chapter 5 for two TCP sources interacting with a buffer is simple, yet flexible, and in principle extendable to *intermediate* networks, that is, a few sources and buffers. It is formulated in terms of a continuous-time Markov chain and enables us to study various stochastic aspects of this interaction. However, its use is somewhat limited in practice as a human being, i.e., the author, has to program the code that produces the infinitesimal generator of the Markov chain. This process is error-prone and rather time consuming. Moreover, it becomes increasingly difficult to correctly implement extensions to multiple sources or multiple buffers. Thus, we cannot easily apply this method when we are interested in the performance of networks of intermediate size. To make this additional step we need a different methodology to specify the Markov chain and obtain the generator. One framework we found particularly suitable to extend the method of Chapter 5 is provided by Stochastic Petri Nets (SPNs), see e.g., Ajmone Marsan *et al.* (1995).

An SPN is a (graphical) formalism to describe systems which exhibit complicated dynamics. It incorporates a notion of state and a set of rules describing the allowed state changes, thereby capturing static and dynamic characteristics of complex systems such as communication systems. The fact that the SPN is a graphical representation of (a model of) a system contributes to the understanding of (the dynamics of) the system. Moreover, computer tools such as the Stochastic Petri Net Package (SPNP) of Ciardo *et al.* (1989) exist that automatically map an SPN to an underlying Markov chain and generate the corresponding infinitesimal generator[1]. Using this generator, SPNP can

[1]To obtain the performance results in this chapter we made extensive use of the software package SPNP version 4. We thank Kishor S. Trivedi of Duke University for making this package available.

compute stationary and transient performance measures formulated as expected reward functions on the SPN. Clearly, tools as SPNP handle the more cumbersome aspects of the performance analysis of complicated systems so that the user can concentrate on the aspects related to *modeling* and *design*.

In the current chapter we apply SPNs to study the interaction between multiple TCP sources and buffers in intermediately-sized networks. This approach allows us to express various performance measures of interest, such as packet loss probability, the throughput, file transfer latency, and so on.

With SPNs we can generalize the TCP models of Altman *et al.* (2000b, 2002b) and Chapter 5 considerably in that we can handle large buffers and networks with more than just one router. Especially the last aspect seems difficult to incorporate in the setting of the work of Altman et al. We refer to Section 5.1.1 for further motivation for our model.

We provide the necessary background about SPNs in Section 6.1. Then, in Section 6.2, we specify an SPN of two TCP sources that share a buffer and implement the synchronous and proportional loss models. In Section 6.3, we extensively compare the results to theoretical results provided by Altman *et al.* (2000b, 2002b), Lakshman & Madhow (1997), and simulation results obtained with ns-2. In Section 6.4 we first present some further possible extensions of the source model. Then we consider a network consisting of three sources and two buffers. The implementation of the throughput formulas in this SPN are 'topology aware' in that they respect the order in which packets of a TCP connection traverse the buffers. Hence, effects such as shaping at up-stream buffers are taken into account. We compare the sharing of link capacity to the minimum-potential-delay fairness scheme as defined by Massoulié & Roberts (1999) and which, according to Lee *et al.* (2001), is the most appropriate for TCP.

6.1 Some Concepts of Stochastic Petri Nets

In this section we introduce the concepts of stochastic Petri nets that are relevant to this chapter. We refer to Figure 6.1 as an example.

An SPN consists of a set of *places* and a set of *transitions*. These two sets are connected via *directed* arcs as a bipartite graph: places (drawn as circles) connect only to transitions, whereas transitions (drawn as bars) connect only to places. A directed arc from a place (transition) to a transition (place) is called an *input* (*output*) arc to (from) a transition. Places can contain tokens indicated as a number of black dots or an integer in the place. If a place contains at least one token, we say that it is *marked*. The distribution of the tokens over the places represents the state of the net and is called the *marking*. When *all* input places, i.e., all places connected to the input arcs of a transition, are marked the transition is *enabled*. Once enabled, the transition can *fire*, thereby removing tokens from its input places and adding tokens to its output places. Thus, a firing

nearly always changes the marking. These firings are to occur immediately (contrary to timed transitions to be introduced below) and atomically, i.e., other transitions cannot fire before the action of the firing transition is completed. Note that during firing the number of tokens in the Petri net is *not* necessarily 'conserved'. it may happen that the transition removes (adds) more tokens than it adds (removes).

Starting from an *initial* marking M_0, the *reachability set* $\mathscr{M}$ is the set of all different markings M reachable by any succession of enabled transitions starting from M_0.

Besides the input and output arcs just mentioned, a Petri net can contain *inhibitor* arcs, to be drawn as an arc from a place to a transition with as arrowhead a small circle. If the place connected to an inhibitor arc is marked, the related transition is disabled.

An important property of input, output, and inhibitor arcs is their *multiplicity*. A *multiple* input (output) arc removes (adds) a number of tokens according to its multiplicity from (to) a place, provided it is enabled. Note that the transition is only enabled if the number of tokens at each input place is larger than or equal to the multiplicity of the corresponding input arc. A multiple inhibitor arc becomes effective as soon as the place contains a number of tokens at least as large as the inhibitor's multiplicity. Besides multiple arcs we need *variable* in- and output arcs. The multiplicity of these arcs may depend on the actual marking of the net. Thus, the multiplicity of variable arcs is generally not constant. Variable arcs are shown as directed arcs with a 'zigzag':  .

Sometimes it is desirable to incorporate probabilistic behavior in the net. One mechanism for this is a *random switch*; the other mechanism, related to time, will be discussed presently. Such a switch consists of a set of immediate transitions which are all simultaneously enabled by the same marking. A set of weights is adjoined to the random switch. The probability that a certain transition of the random switch fires is proportional to its weight. Such weights are allowed to depend on the marking at the moment just before firing.

The arc types we have discussed above permit us to specify various types of conditions to enable or disable transitions. However, sometimes it is rather cumbersome to specify complicated conditions in the SPN with places and arcs. To avoid such awkward complications we can use *guards*. A guard is a marking-dependent enabling function attached to a transition. If the condition of the guard is satisfied, the transition is enabled; otherwise the transition is disabled. Thus, with guards quite complex marking dependent conditions can be imposed on the dynamics of the net. In this chapter we draw a guard as a box with a dashed boundary containing a (shorthand of a) condition. (This graphical representation of a guard is not standard in the literature.)

Up to now the transitions discussed above are *immediate*: if a transition is enabled, and chosen when it is an element of a random switch, it fires immediately. We can introduce the concept of time in the Petri net with *timed* transitions, which are drawn as open rectangles. Such a transition fires, if enabled, after an exponentially distributed amount of time. A useful feature is that the transition rates of such transitions are allowed

to depend on the marking.

Once we have specified the SPN the computation of performance measures is relatively straightforward. Under some mild boundedness conditions, it is possible to automatically map the SPN to a continuous-time Markov chain $\{M(t), t \geq 0\}$ with infinitesimal generator Q and initial probability vector representing the initial marking of the SPN. The size of the chain equals the cardinality $|\mathscr{M}|$ of the reachability set $\mathscr{M}$. If $\{M(t)\}$ is irreducible, the stationary distribution $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{|\mathscr{M}|})$ exists and does not depend on the initial marking. The vector $\boldsymbol{\pi}$ satisfies $\boldsymbol{\pi}Q = 0$, $\sum_i \pi_i = 1$, and can be computed by Gauss-Seidel iteration, or other, more advanced, numerical procedures, cf. Stewart (1994).

The performance measures of interest for the stationary limit M of $\{M(t)\}$ can then be expressed in terms of a reward rate function $r : \mathscr{M} \to \mathbb{R}$ which associates with every state $m \in \mathscr{M}$ a real-valued reward rate $r(m)$. The expected reward in steady-state is then given as

$$\mathbb{E}\{r(M)\} = \sum_{m \in \mathscr{M}} r(m)\pi_m. \tag{6.1}$$

For more information regarding SPNs consult, e.g., Ajmone Marsan *et al.* (1995). For details concerning SPNP, see Ciardo *et al.* (1994).

6.2 An SPN for Two TCP Sources and One Buffer

In this section we model two TCP sources sharing one buffer with proportional loss as a stochastic Petri Net. Section 6.2.1 presents the details of this SPN. The resulting TCP model is similar but *not* identical to the one of Section 5.1.2; we indicate the differences at the end the section. In Section 6.2.2 we use the steady state probabilities of the underlying Markov chain to express some performance measures as reward functions on the SPN. To obtain insight into the time required to compute these probabilities we discuss some computational issues in Section 6.2.3. Finally, in Section 6.2.4 we show that it is nearly trivial to modify the SPN with proportional loss to an SPN that implements synchronized loss.

6.2.1 The Proportional Loss Model

Here we model the behavior of two TCP sources and a buffer, i.e., processes $\mathbf{W}(t) = (W_1(t), W_2(t))$, and $C(t)$, subject to proportional loss with the SPN shown in Figure 6.1. The SPN contains three 'subnets' indicated by dashed boxes around several places and transitions. The subnets S_1 and S_2 represent the sources whereas the subnet B represents the buffer. We describe these subnets first, then we focus on the dynamics of the complete SPN.

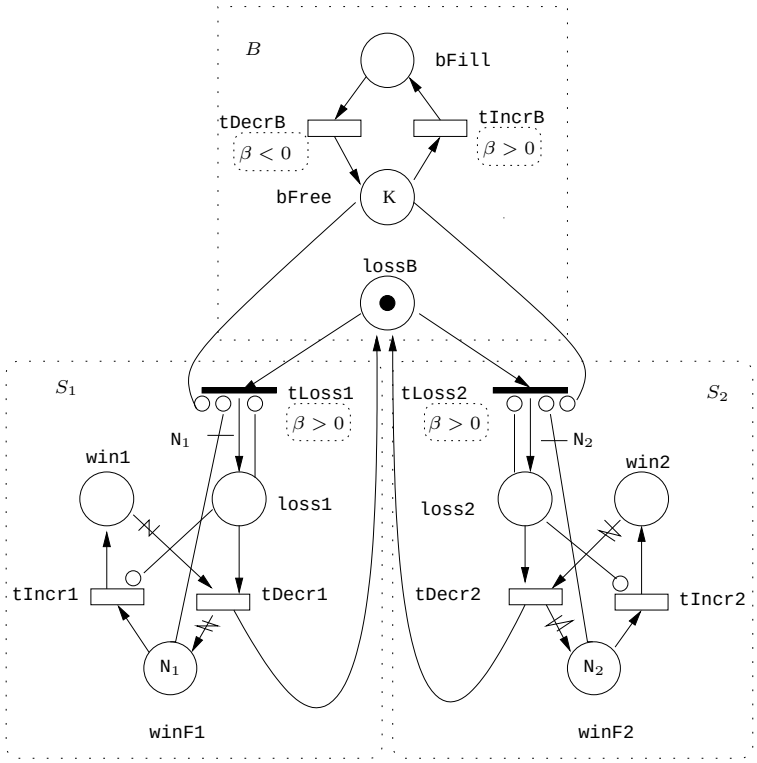

Figure 6.1: *A Petri Net model of two TCP sources sharing a buffer with the loss proportional scheme. We indicate guards with dashed boxes around strings, such as* $\beta < 0$ *appearing immediately below* `tDecrB`*.*

The Subnets

Subnet S_1 contains three places: `win1`, `winF1` and `loss1`; one immediate transition: `tLoss1`; and two timed transitions: `tIncr1` and `tDecr1`. Subnet S_2 is, except for the naming, identical; as such the rest of the discussion applies equally well to source 2. The state of source 1 is given by the markings of `win1`, `winF1` and `loss1`, respectively. Here, the number of tokens in `win1`, i.e., #`win1`, denotes the momentary congestion window of source 1. The marking of `winF1` denotes how much further the window can increase. Initially, #`winF1` $= N_1$, so that at all times during the evolution of the SPN it holds that #`win1` + #`winF1` $= N_1$. The loss state of source 1 is indicated by #`loss1`. When #`loss1` $= 0$ the source is allowed to increase #`win1`, while when #`loss1` $= 1$ the source has to reduce #`win1` by a factor 2.

The buffer subnet contains three places: `bFill`, `bFree`, and `lossB`; and two timed transitions: `tDecrB` and `tIncrB`. The marking of the place `bFill` is the fill level of the buffer. The place `bFree` has initially K tokens. Its marking corresponds to the free space, i.e., the maximum buffer level K minus the fill level $\#\texttt{bFill}$; thus, $\#\texttt{bFill} + \#\texttt{bFree} = K$. Finally, when $\#\texttt{lossB} = 0$ ($\#\texttt{lossB} = 1$) the buffer is (not) congested.

Observe that, for instance, the processes $\{W_1(t), t \geq 0\}$ and $\{\#\texttt{win1}, t \geq 0\}$ are identical processes. For notational brevity and consistency with previous chapters we use in the sequel $W_1(t)$ instead of $\#\texttt{win1}$ to denote the marking of `win1`, etcetera. Moreover, when no confusion arises, we also occasionally drop the dependency on t of the processes $W_1(t)$, etcetera, to save space. Table 6.1 shows the relation between the stochastic processes and the markings of the other places in the SPN. Here $I_i(t)$ represents the same indicator process as defined in (3.1).

$\#\texttt{win1} = W_1(t)$	$\#\texttt{win2} = W_2(t)$	$\#\texttt{bFill} = C(t)$
$\#\texttt{winF1} = N_1 - W_1(t)$	$\#\texttt{winF2} = N_2 - W_2(t)$	$\#\texttt{bFree} = K - C(t)$
$\#\texttt{loss1} = I_1(t)$	$\#\texttt{loss2} = I_2(t)$.	

Table 6.1: *The correspondence between the stochastic processes* $\{\mathbf{W}(t), \mathbf{I}(t), C(t)\}$ *and the markings of places in Figure 6.1.*

The *marking-dependent* firing rates and guards associated with all transitions are summarized in Table 6.2. Here $T_i(k)$ is given by (5.7), whereas the function

$$\beta(\mathbf{n}, k) = \frac{K}{B}(\mathbf{r}(k) \cdot \mathbf{n} - L) \tag{6.2}$$

denotes the transition rate at which in- and decrements of the buffer level process occur, where the marking of the SPN is such that $\mathbf{W}(t) = \mathbf{n} = (n_1, n_2)$ and $C(t) = k$, cf. (5.10). Finally, recall that $\lambda_i(k) = T_i^{-1}(k)$, cf. (5.11).

From Initial State to Congestion (Congestion Avoidance)

The initial marking of the SPN is as shown in Figure 6.1. Sources 1 and 2 are 'off' and not in a loss state; the buffer is empty and in possession of the loss token. In the initial state only `tIncr1` and `tIncr2` are enabled and fire at rate $1/T_1$ and $1/T_2$, respectively. Each firing increases W_1 or W_2 by one, which clearly models the Additive-Increase phase of TCP. Note that on average source i spends an amount $T_i(k)$ in state n_i, given $C = k$. In this way the SPN incorporates feedback delay.

Transition	Rate	Guard
`tIncrB`	$\beta(\mathbf{n}, k)$	$\beta(\mathbf{n}, k) > 0$
`tDecrB`	$-\beta(\mathbf{n}, k)$	$\beta(\mathbf{n}, k) < 0$
`tLoss1`	∞	$\beta(\mathbf{n}, k) > 0$
`tLoss2`	∞	$\beta(\mathbf{n}, k) > 0$
`tIncr1`	$\lambda_1(k)$	—
`tDecr1`	$\lambda_1(k)$	—
`tIncr2`	$\lambda_2(k)$	—
`tDecr2`	$\lambda_2(k)$	—

Table 6.2: *Rate functions and guards for the transitions in Figure 6.1.*

As W_1 and W_2 increase, the scaled net input rate (6.2) increases as well. After several firings of `tIncr1` and `tIncr2`, W_1 and W_2 are so large that $\beta(\mathbf{W}, 0)$ becomes positive. This will set the guard at `tIncrB` to true, so that `tIncrB` becomes enabled. Each firing of `tIncrB` increments C by one. After K firings of `tIncrB`, the buffer is completely filled, i.e., $C = K$. When this happens, the inhibitor arcs from `bFree` to `tLoss1` and `tLoss2` are now no longer active, so that the random switch consisting of the immediate transitions `tLoss1` and `tLoss2` becomes enabled.

Suppose `tLoss1` fires first so that source 1 receives the loss token. As such, the loss token represents the congestion signal that the buffer sends to a source. Clearly, in this case the inhibitor from `loss1` to `tIncr1` will prevent further increments of the window of source 1. Note that, as source 1 receives the loss token, `loss2` does not become marked (unlike the synchronous model, to be discussed shortly), and consequently, `tIncr2` can still fire.

It is evident that when source i is inactive, i.e., when $W_i = 0$, it cannot suffer from loss. To prevent the loss token from being sent to a quiet source a multiple inhibitor arc connects `winF1` (`winF2`) to `tLoss1` (`tLoss2`) with multiplicity N_1 (N_2). The multiplicity is indicated in Figure 6.1 at the inhibitor arc.

The Proportional Loss Model

The buffer uses a proportional loss model, according to which the buffer chooses only one connection to suffer from loss during overload. The probability to select a particular connection is proportional to its momentary transmission rate. We implement this behavior by means of the random switch consisting of `tLoss1` and `tLoss2`.

The marking-dependent weights of the random switch are chosen such that `tLoss1` fires with probability p_1 whereas `tLoss2` fires with probability $p_2 = 1 - p_1$. Table 6.3 shows the values of p_1 when $r_1(K)W_1 > L$ and $r_2(K)W_2 > L$, etcetera. The motiva-

tion behind this loss model is based on the insight that if, for instance, $r_1(K)W_1 > L$ and $r_2(K)W_2 \leq L$, connection 1 certainly loses traffic. Thus, in this case connection 1 should surely receive the loss token. Due to the proportional loss model just one loss token is available, so that connection 2 cannot receive a loss token. Therefore, in this case, $p_1 = 1$ and $p_2 = 0$. When $r_1(K)W_1 \leq L$ and $r_2(K) \leq L$ (but $\mathbf{r}(K) \cdot \mathbf{W} > L$) both sources can be hit by a loss with a probability proportional to their sending rates. In the (very) rare case that $r_1(K)W_1 \geq L$ and $r_2(K)W_2 \geq L$ both sources should reduce their rate. However, as `lossB` contains just one token, it cannot simultaneously send both sources a loss token. Therefore, we again take the loss probabilities proportional to the sending rates. We emphasize that the influence of this inconsistency will be small in nearly all relevant parameter settings.

	$r_1(K)W_1 \leq L$	$r_1(K)W_1 > L$
$r_2(K)W_2 \leq L$	$p_1 = \frac{r_1(K)W_1}{\mathbf{r}(K)\cdot\mathbf{W}}$	$p_1 = 1$
$r_2(K)W_2 > L$	$p_1 = 0$	$p_1 = \frac{r_1(K)W_1}{\mathbf{r}(K)\cdot\mathbf{W}}$

Table 6.3: *Firing probability p_1 of `tLoss1`.*

Removing the Congestion (Multiplicative Decrease)

When $I_1(t) = 1$ the timed, variable transition `tDecr1` is enabled. Once it fires, it moves the loss token from `loss1` to `lossB`, and, to reflect the Multiplicative Decrease, removes half of the tokens from `win1` and adds these to `winF1`. Specifically, the multiplicity $m_{\texttt{win1}}$ of the variable arc between `win1` and `tDecr1` is

$$m_{\texttt{win1}} = \begin{cases} \lfloor (W_1 + 1)/2 \rfloor & \text{when } W_1 > 1, \\ 0 & \text{when } W_1 = 1. \end{cases} \tag{6.3}$$

(A transition from, e.g., $W_1 = 5$ to $W_1 = 2$ requires to remove 3 tokens from `win1`. Compare also the transition (5.12d) in the model of Chapter 5.)

If, with the new marking, still $\beta(\mathbf{W}, K) > 0$ either `tLoss1` or `tLoss2` will immediately fire again. After a sufficient number of multiplicative decrements of W_1 and W_2 the net input rate becomes negative. When this is the case, firings of `tDecrB` decrement the buffer content. As a second consequence of $\beta(\mathbf{W}, C) < 0$ the guards at `tLoss1` and `tLoss2` prevent the loss token from being passed on to either of the sources. Thus, the source windows cannot decrease further.

We have not yet discussed two arcs: the inhibitors from `loss1` and `loss2` to `tLoss1` and `tLoss2` respectively. Their role will be clarified in the synchronous loss model presented in Section 6.2.4. In the proportional model they have no function, but they do not influence the performance measures in any way either.

Differences with the Model of Chapter 5

The TCP model as implemented in the SPN is similar but not identical to the TCP model described in Section 5.1.2. We enumerate the differences now; for clarity we use the self-describing names "discretized model" and "SPN model".

The first difference relates to the time the sources stay in a congested state. In the discretized model this time is *exponentially* distributed. This follows as a state (n_1, n_2, K) such that $\mathbf{r}(K) \cdot \mathbf{n} > L$ has an outward transition to $(\lfloor n_1/2 \rfloor, n_2, K)$. The rate out of the state (n_1, n_2, K) is therefore, according to (5.12d),

$$q_{n_1,n_2,K;\lfloor n_1/2 \rfloor,n_2,K} + q_{n_1,n_2,K;n_1,\lfloor n_2/2 \rfloor,K} = \frac{n_1 r_1(K)\lambda_1(K) + n_2 r_2(K)\lambda_2(K)}{n_1 r_1(K) + n_2 r_2(K)}.$$

In the SPN model the time in a congested state is *hyper-exponentially* distributed. To see this, observe that once the Markov chain enters a congested state, a random switch chooses with probability p_i, $i = 1, 2$, which source receives the loss token, and the holding time of the loss token in place `loss1` or `loss2` is exponentially distributed with rate $\lambda_1(K)$ or $\lambda_2(K)$, respectively.

The second difference pertains to the boundary conditions of the (underlying) Markov chains. In the discretized model we ensure that sets indicated by (5.13) have zero probability by removing any transition into these sets. (In effect, we implement transitions $q_{n_1,n_2,K;\lfloor n_1/2 \rfloor,n_2,K-1}$ rather than $q_{n_1,n_2,K;\lfloor n_1/2 \rfloor,n_2,K}$ if $\beta(\lfloor n_1/2 \rfloor, n_2, K) < 0$ to enforce the boundary conditions.) In the SPN model we only remove a token from `bFill` once `tDecr1`, or `tDecr2`, fires. The main reason for this 'shortcut' is to keep the Petri net relatively simple. The influence of this difference is small. (In the preparation of obtaining the results for Chapter 5 we also studied a model with transitions of the type $q_{n_1,n_2,K;\lfloor n_1/2 \rfloor,n_2,K}$ and compared the results to those presented in Chapter 5. We found only minor differences in the values of the performance measures.)

6.2.2 Performance Measures

We now express three performance measures as rewards functions of the form (6.1) on the net: a connection's expected transmission rate, its throughput and the utilization of the link.

1. The expected transmission rate for connection i is easy:

$$\tau_i = \mathbb{E}\left\{r_i(C)W_i\right\}.$$

2. The throughput is not as simple to specify in exact terms in the present setting; we can concentrate on the fluid that *enters* the buffer, and the fluid that *leaves* the buffer. We discuss a proposal related to each possibility and compare these numerically in Section 6.3.

2a. For the first proposal we consider the fluid that *enters* the buffer. While the buffer is not full it accepts fluid, but when it is full it drops the excess fluid. We assign this excess to the connection that receives the loss token. There is only excess traffic when $C = K$ and $\beta(\mathbf{W}, C) > 0$. As these conditions imply, and are implied by, the condition that either $I_1 = 1$ or $I_2 = 1$ (due to the proportional loss assumption), we define

$$\gamma_i^{\text{in}} = \tau_i - \mathbb{E}\left\{(\mathbf{r}(C) \cdot \mathbf{W} - L)1_{\{I_i=1\}}\right\}. \tag{6.4}$$

This definition assigns *all* excess fluid $e = \mathbf{r}(C) \cdot \mathbf{W} - L$ to *one source*: the source that receives the loss token. As a consequence we need to verify whether indeed $r_i(K)W_i > e$ when source i receives the loss token, for otherwise we subtract too much in the above. Now, since, for instance, the condition $r_1(C)W_1 < L$ is equivalent to $r_2(C)W_2 > e$, this problem does not occur in three of the four possibilities shown in Table 6.3. Only when both source rates exceed L we may subtract too much. As this problematic case has (very) small probability, we neglect its consequences, as we did before, in Section 6.2.1.

Observe that in (6.4) we use the indicator $1_{\{I_i=1\}}$ whereas in (5.15) we use $1_{\{C=K\}}$. The reason for this difference in now easy to explain. In the current model we have a loss indicator variable I_i for both loss models, whereas in Section 5.1.2 this indicator is only a process variable of the model with synchronous loss.

2b. The other possibility is to consider the fluid that *leaves* the buffer. When the buffer is empty the departure rate at time t is equal to the arrival rate. When at time t the buffer contains $k > 0$ units of fluid the departure rate of source i at time t equals the service capacity L times the fraction of traffic of source i that arrived at time $t - kB/(KL)$. Since the Markov process $\{\mathbf{W}(t), \mathbf{I}(t), C(t), t \geq 0\}$ does not maintain the history of the source states as supplementary variables, the source rates at time $t - kB/(KL)$ are (principally) unknown. Hence we cannot incorporate the effect of buffering delay on the throughput. We therefore neglect the influence of the delay and approximate the output process by the arrival process. To see that this approximation is acceptable we reason as follows. Observe that the round-trip times of all sources include the buffering delays along the route. Hence, it always takes less time to refresh the buffer content than it takes for a source to change its rate. Consequently, while the buffer content is refreshed the input rates are nearly constant. We conclude that neglecting the delay only shifts the output process backward in time, but does not substantially change its shape or the ratio of fluid of the sources. By the above we arrive at

$$\gamma_i^{\text{out}} = \mathbb{E}\left\{r_i(C)W_i 1_{\{C=0\}} + L\frac{r_i(C)W_i}{\mathbf{r}(C) \cdot \mathbf{W}} 1_{\{C>0\}}\right\}. \tag{6.5}$$

3. We define utilizations as

$$u_i = \frac{\gamma_i}{L}, \quad i = 1, 2 \qquad\qquad u = \frac{\gamma_1 + \gamma_2}{L} = u_1 + u_2.$$

With respect to the existence of the stationary distribution $\boldsymbol{\pi}$, which is needed in the computation of the expectations above, we remark that the Markov chain associated with the proportional SPN may be reducible, depending on the choice of parameters. This has, however, no consequence for the existence of $\boldsymbol{\pi}$. Whereas some states may belong to transient classes, the other states form *one* recurrent class implying that a unique stationary distribution still exists.

6.2.3 Computational Issues

It is of interest to estimate the size $|\mathscr{M}|$ of the Markov chain as this gives insight into the time required to solve for the stationary distribution. We have not been able to find an accurate but simple expression for $|\mathscr{M}|$, mainly because the number of recurrent states depends critically on the values of the system parameters and the presence of guards. Hence, given the sizes of $\mathscr{W}_1$, $\mathscr{W}_2$, and $\mathscr{K}$, and that the loss token can reside in in three places (i.e., `lossB`, `tLoss1`, and `tLoss2`), we conclude that

$$|\mathscr{M}| \leq 3(N_1+1)(N_2+1)(K+1) = O((N_1+1)(N_2+1)(K+1)). \tag{6.6}$$

This estimate is an upper bound as the guards in the SPN may considerably reduce the number of transitions, that is, not all markings counted in this formula represent reachable states. It is evident from Figure 6.1 that the SPN contains only eight transitions thereby bounding the number of non-zero entries per row in the generator also by eight. Consequently, the generator is sparse.

Observe that the model possesses some scaling freedom in the parameters P_i, $i = 1, 2$, L and B. To remove this freedom assume henceforth that source 2 is the distant one, i.e., $T_2 \geq T_1$, and set $r_2(0) = 1$. Moreover, assume that the packet sizes are equal, i.e., $P_1 = P_2$, which has as a consequence that, cf. (5.8),

$$r_1(k) = \frac{P_2}{T_1(k)} = \frac{r_2(0)T_2}{T_1(k)} = \frac{T_2}{T_1(k)}, \quad (\text{as } r_2(0) = 1).$$

Next, suppose that the sending rates are not constrained by the receiver windows. Thus, each source can congest the link. This is achieved by setting $N_i \geq \lceil L/r_i(K) \rceil = \lceil LT_i(K)/T_2 \rceil$, where $\lceil x \rceil$ is the smallest integer larger than x. Thus, we see that L determines the source granularity: a small (large) value of L means that a few (many) source transitions are needed for buffer overflow. It is intuitively clear that choosing N_i much larger than $\lceil LT_i(K)/T_2 \rceil$ hardly affects the value of the performance measures, as such 'high' source states are visited only relatively seldom. From numerical evaluations we conclude that choosing N_i equal to $1.1\lceil LT_i(K)/T_2 \rceil$ is large enough in our setting. Finally, in the (numerical) analysis we wish to specify the buffering delay d_B instead of the buffer size B itself; therefore let $B = d_B L$. As a result the parameters T_1, T_2, L and d_B now fully characterize the system.

Clearly, from the computational point of view it is of interest to choose N_1, N_2 and K as small as possible without significantly affecting the overall results, i.e., the outcome of the performance measures. The following provides some insight into how small N_1, etcetera, can be reasonably chosen. As a first step we notice that as source 2 is the distant source, it follows that

$$N_2 = 1.1\lceil LT_2(K)/T_2\rceil \geq 1.1\lceil LT_1(K)/T_2\rceil = N_1.$$

However, the probability that $W_2 > N_1$ is small, given the bias of TCP against long connections. Therefore, we can safely set $N_2 = N_1$. Second, as in Section 5.1.2 it turns out that $K = 5$ is already large enough for the parameter ranges we investigate in this chapter, and setting K to larger values makes practically no difference. We can even set $K = 1$ in the large bandwidth-delay product regime. In this regime it takes an AIMD source much more time to 'fill the pipe' than to fill the buffer. Thus, approximately, the buffer is either empty or congested so that $C = 0$ or $C = 1$. We remark once again that the numerical results to be presented below provide support for all these approximations.

6.2.4 The Synchronous Loss Model

Here we show the changes required to modify the SPN with proportional loss to an SPN with synchronized loss.

First, to signal both sources about congestion requires *two* loss tokens to be initially present at `lossB`. Second, in the new setting the transitions `tLoss1` and `tLoss2` will no longer form a random switch. Instead, each fires with probability 1, if enabled.

Now suppose again that once `bFree` becomes empty, `tLoss1` is the first to fire. This results in one of the loss tokens to move to `loss1`. The inhibitor from `loss1` to `tLoss1` prevents this transition to fire again. Consequently, the immediate transition `tLoss2` fires so that both sources receive a loss token at the same instant. While `loss1` (`loss2`) is marked source 1 (source 2) cannot receive a loss token which becomes available after a firing of `tDecr2` (`tDecr1`) due to the inhibitor arcs from `loss1` (`loss2`) to `tLoss1` (`tLoss2`).

As both sources receive a loss token, both sources can take their share of the excess fluid arriving during congestion. Therefore the throughput as defined in (6.4) should become

$$\gamma_i^{\text{in}} = \tau_i - \mathbb{E}\left\{(\mathbf{r}(C)\cdot\mathbf{W} - L)\frac{r_i(C)W_i}{\mathbf{r}(C)\cdot\mathbf{W}}1_{\{C=K,\beta(W_1,W_2,C)>0\}}\right\}. \qquad (6.7)$$

The condition expressed by the indicator function is here more difficult that in (6.4). In the discussion in Section 6.2.1 on the differences between the 'discretized model' and the 'SPN' model, we pointed out that the SPN does not implement boundary conditions such that $\mathbb{P}\{C = K, \beta(W_1, W_2, C) < 0\} = 0$. Hence, the events $\{C = K\}$ and

$\{\beta(W_1, W_2, C) > 0\}$ are not the same (up to null-events). Moreover, in the synchronous loss model $I_i = 1$ does not imply $\beta(W_1, W_2, C) > 0$ or $C = K$.

Definition (6.5) does not need any modification.

Finally, about the size of the state space we remark that the number of possibilities for the loss tokens is 4 rather than 3. Thus, the bound in (6.6) should be multiplied with $4/3$ to obtain an upper bound for the size of the reachability set of the SPN with synchronized loss.

6.3 A Comparison with Analytic Models and ns-2

In this section we compare the numerical results of the SPN to other analytic work and simulation with ns-2. We specify the investigated scenarios in Section 6.3.1 and present the results in Section 6.3.2.

6.3.1 Scenarios

Figure 5.2 shows the network we used for the numerical analysis and ns-2. Two greedy TCP sources, S_1 and S_2, communicate with destinations D_1 and D_2 via a router with buffer size $B = d_B L$. The receiver windows are assumed large enough to not constrain the congestion windows. Table 6.4 shows the values of L and d_B for the four investigated scenarios. For each scenario we vary the propagation delay d_1 of the link connecting S_1 and R_1 in 10 steps from 40 ms to 240 ms. The delay between S_2 and R_1 is here taken as 250 ms instead of 120 ms, which is the value shown in Figure 5.2. Moreover, the link rate is 1.5 Mb/s whereas it is 1 Mb/s in the figure. The parameters for the SPN follow from Section 6.2.3.

Scenario	L	d_B (ms)	K
'25s'	25.7	16	1
'80s'	80.7	16	1
'25l'	25.7	160	5
'80l'	80.7	160	5

Table 6.4: *SPN parameter values. We use mnemonics such as '25s' to denote the scenario in which the link rate is 25.7 and the maximum buffering delay d_B is 'small'. A buffering delay of 16 ms corresponds to a buffer size of 5 packets in the ns-2 simulation.*

In the simulations with ns-2 we use a RED buffer and consider a small and a large buffer case. (Consult Section 5.2.2 for a discussion on using a RED buffer for the simulation and a drop-tail buffer for the model.) In the former (latter) the buffer's total size

is 20 (200) packets. The RED parameters in (1.35–1.36) are taken as follows. The minimum threshold $x_{\min}$ is 5 (50) packets, the maximum threshold $x_{\max}$ is 10 (55) packets, the maximum drop probability $p_m = 0.1$ and the weight $\epsilon = 0.002$. The packet size is, including IP header, 576 Bytes. We note that the RED parameter values for the small buffer are also identical to the values chosen by Altman *et al.* (2000b).

6.3.2 Results

Lakshman & Madhow (1997) derive that for synchronous loss and a small buffer the ratio of the throughputs $\gamma_1/\gamma_2 \approx s^{-\alpha}$, where $s = T_1/T_2$ is the ratio of propagation delays and $\alpha = 2$, cf. (1.42). Altman *et al.* (2000b) present a model with proportional loss in which $\gamma_1/\gamma_2 = s^{-\alpha}$ with $\alpha \approx 0.85$, cf. (1.46). As our model allows to analyze both loss models we investigate what values for α the model will give in either case. Moreover we analyze the influence of buffering delay. The left and right panel of all figures of this section show the results for the small and large buffer case, respectively.

In Figure 6.2 we plot the throughput ratio computed by the model with proportional loss as a function of s for the scenarios of Table 6.4. We compare the differences between the 'input ratio' $\gamma_1^{\text{in}}/\gamma_2^{\text{in}}$ obtained by (6.4) and the 'output ratio' $\gamma_1^{\text{out}}/\gamma_2^{\text{out}}$ by (6.5). We also plot $s^{-\alpha}$ for several values of α, and the analytic estimate (1.46) of Altman *et al.* (2002b).

The results for the small buffer case show that for relatively coarse-grained sources the input and output ratios are different. The input ratio is too high, as compared to the function $s^{-0.85}$, whereas the output ratio is too low. By increasing L the two ratios seem to converge to, approximately, $s^{-0.87}$, which is close to the result of Altman *et al.* (2000b). Note that, as observed in Altman *et al.* (2002b), (1.46) approximates $s^{-0.85}$ very well. Clearly, the graphs of the output ratios lie below the graphs of the input ratios, implying that the output ratios are fairer than the input ratios. We are unable to provide intuition why this is the case. Observe also that the sharing of the link becomes fairer when the buffer size increases (α drops to approximately 0.65), which is in accordance with intuition.

We explain the influence of the choice for L on these two ratios by considering the overload states. Suppose first that just source 1 is in state 26 when congestion occurs. When $L = 25.7$ source 1 needs 13 round-trip times after a loss to fill the link by itself again, whereas when $L = 80.7$, and source 1 is in state 81 it needs 41 round-trip times. Thus, applying this insight to the two source model, the fraction of time spent in congestion, i.e., $C = K$ and $\beta > 0$, is smaller when $L = 80.7$ than when $L = 25.7$ (using the scaling of the other parameters as explained in Section 6.2.3). As the computations of γ_i^{out} and γ_i^{in} mainly differ when $C = K$, and the fraction of time in congestion is less when $L = 80.7$ as compared to $L = 25.7$, the difference between γ_i^{out} and γ_i^{in} is smaller when $L = 80.7$.

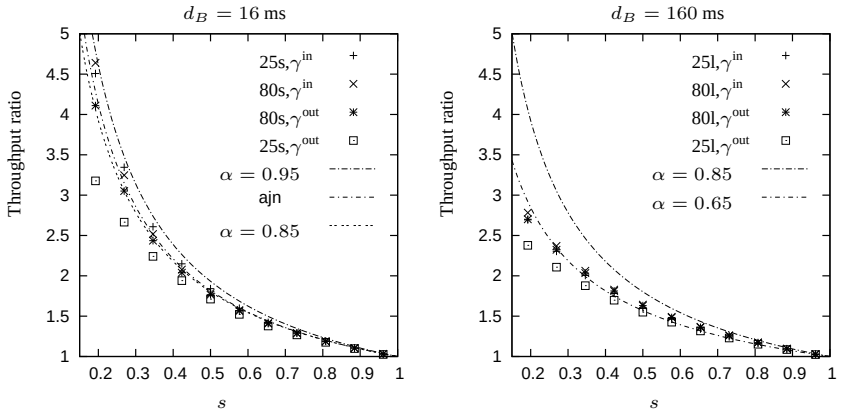

Figure 6.2: *Ratio of throughputs as a function of $s = T_1/T_2$ for the proportional loss model. The left (right) panel shows the ratio when $d_b = 16$ ms ($d_b = 160$ ms). The label '25s,γ^{in}' refers to the input-related throughput* (6.4) *computed for Scenario 25s, etcetera; the label 'ajn' refers to* (1.46).

In Figure 6.3 we plot similar results but now for the synchronous loss model. There is hardly any difference between the input and output ratios. For small buffers we see that the ratios according to our model behave like $s^{-2.2}$ instead of s^{-2} as obtained by Lakshman & Madhow (1997). When the buffer size increases, the power decreases to a value smaller than 2, in line with the results of Lakshman & Madhow (1997).

In Figure 6.4 we compare the utilization $(\gamma_1^{\text{out}} + \gamma_2^{\text{out}})/L$ as computed by our model to a simulation of two NewReno sources and two TCP Sack sources, and theoretical results of Lakshman & Madhow (1997) and Altman *et al.* (2002b). In the synchronous loss model Lakshman & Madhow (1997) estimate the utilization as $3/4$ independent of the ratio s, cf. (1.44). For the proportional model Altman *et al.* (2002b) provide in their Equation (23) the approximation

$$\frac{\gamma_1}{L} \approx \frac{2s^2-1}{2(s^2-1)}\frac{1}{s+1} - \frac{1}{4(s^2-1)} = \frac{1}{4}\frac{4s+3}{(s+1)^2}. \tag{6.8}$$

Combining this with (1.46) yields a similar expression for

$$\gamma_2 \approx \frac{s}{4}\frac{3s+4}{(s+1)^2}\,L.$$

(We restate (1.45)–(1.46) here for ease of comparison.)

We see from the graphs that most models overestimate the utilizations in comparison to simulation. Our model, contrary to (1.45), correctly captures the trend that the

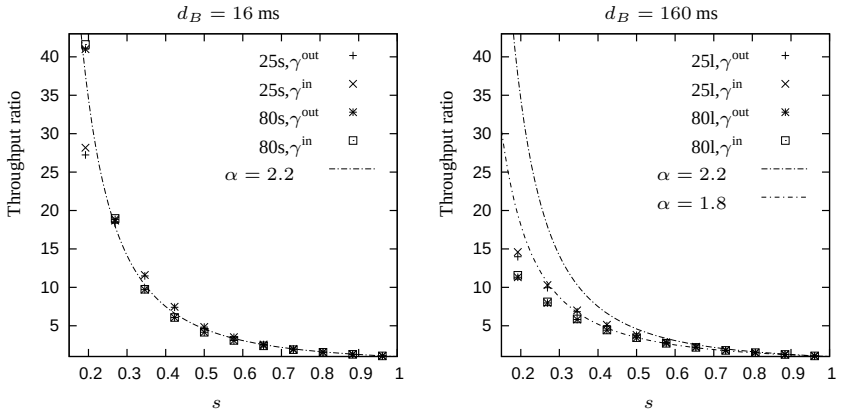

Figure 6.3: *Ratio of throughputs for the synchronous loss model as a function of* $s = T_1/T_2$*. The labeling is as in Figure 6.2.*

utilization decreases as a function of s. In the right panel, showing the results for large buffers, we do not include the results of Altman *et al.* (2002b) and Lakshman & Madhow (1997) as these only apply to small buffers. Interestingly, in line with an observation of Altman *et al.* (2000b), the utilization for proportional loss is higher than the utilization for synchronized loss. Finally, we conclude that the theoretical models are too optimistic about link utilization in all cases. (The results of the TCP NewReno simulation in the right panel are a bit odd. This behavior did not disappear by slight changes of the parameters of the RED buffer. We did not investigate large changes as this would introduce considerable differences between the model and the simulation.)

Figure 6.5 shows the normalized throughput of the first connection $\gamma_1^{\text{out}}/(\gamma_1^{\text{out}} + \gamma_2^{\text{out}})$ in comparison to (6.8) and the simulations; the results for the second connection follow immediately, as $\gamma_2^{\text{out}} = 1 - \gamma_1^{\text{out}}/(\gamma_1^{\text{out}} + \gamma_2^{\text{out}})$. Clearly, the theoretical ratios are in nearly perfect agreement. Moreover, for the small buffer case the proportional loss models are 'too fair', whereas the synchronous models are 'too unfair', which is in line with the findings in Chapter 5.

6.4 Extensions

The extensions presented in this section provide further support for the versatility of applying SPNs to modeling TCP. We start with two relatively simple extensions of the model of Section 6.2. The first allows sources to switch on ('downloading') and off ('thinking'); the second is such that multiple sources can share the bottleneck link. Then

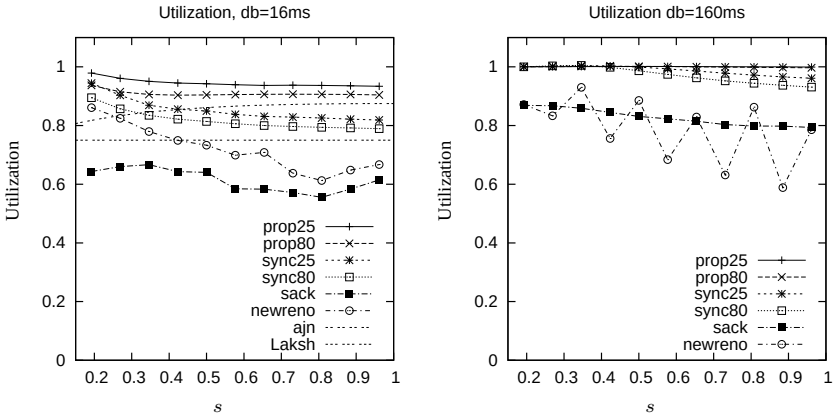

Figure 6.4: *The utilization as a function of the ratio of propagation delays. The label 'ajn' refers to (6.8), whereas 'Laksh' labels the line $3/4$, cf. (1.44). The label 'prop25' in the left (right) panel refers to scenario 25s (25l) of the proportional model, etcetera.*

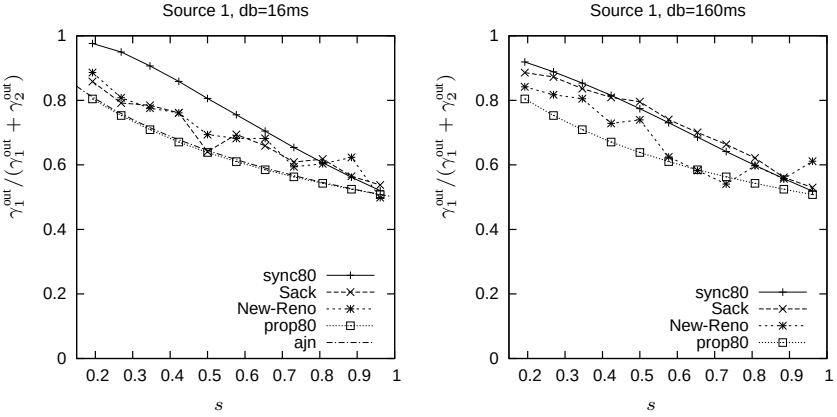

Figure 6.5: *The normalized throughput as a function of s.*

we specify and analyze a network consisting of three sources and two links. Especially this last model appears difficult to tackle 'by hand'.

6.4.1 On/Off Behavior and Multiple Sources

Figure 6.6 specifies a source that can switch on and off. The extension of the source subnet consists of adding two transitions `tOn` and `tOff` that fire at rate λ_{on} and λ_{off}. The probability that the on-time exceeds x is $\exp(-\lambda_{\text{on}}x)$. We take the file sizes as exponentially distributed with average size $\mathbb{E}\{\text{file size}\}$. Consequently, if $C = k$ and $W = n$, the rate at which the source switches off is given as $\lambda_{\text{off}} = r(k)n/\mathbb{E}\{\text{file size})\}$, in accordance with the reasoning in Section 5.2.5.

Suppose the source is off. Then, clearly, all its window tokens should be positioned in `winF` and the marking W of `win` should equal 0. The inhibitor from `winF` to `tIncr` with multiplicity N disables `tIncr` as long as $W = 0$. Thus, the only possibility to move a token from `winF` to `win` is the transition `tOn`. As soon as `win` contains one token, the inhibitor to `tOn` disables this transition. The source switches off when `tOff` removes all tokens at `win` via the variable arc from `win` and adds these tokens to `winF`. Observe that as a consequence of (6.3) the multiple arc between `win` and `tDecr` never removes all tokens from `win`. Hence the decrements due to Multiple Decrease do not switch off the source. When the source is in a loss state, i.e., `loss` is marked, it cannot finish a file transfer. The inhibitor from `loss` to `tOff` prevents this. Note that this implementation of on/off behavior does not come at the cost of extra places. Hence, the set of markings $\mathscr{M}$ does not increase.

We refer the reader to Section 5.2.5 for an analysis of the influence of on/off behavior on the sharing and utilization of the link.

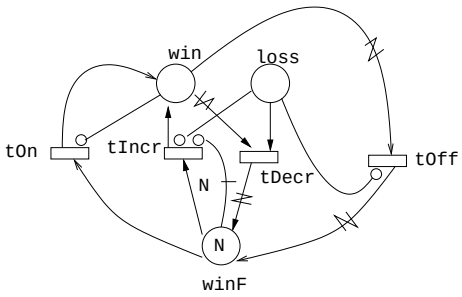

Figure 6.6: *An on/off source. (We do not draw the arcs that connect the source subnet to the buffer subnet.)*

Extending the SPN of Section 6.2 to incorporate more than two sources is quite simple. SPNP, cf. Ciardo *et al.* (1994), supports arrays of places, transitions, and so on.

The window size W_i of source i corresponds then to the value of the i-th element of the 'window' array, etcetera. The size of the arrays equals the number of sources, which can be controlled, clearly, by a single variable. For the synchronous loss model the number of loss tokens initially present at `lossB` should, of course, equal the number of sources. When the number of sources J is larger than 2 it is even possible to put an initial number of tokens in `lossB` somewhere in between 1 and J. (We did not investigate such scenarios.)

6.4.2 Three TCP Sources Sharing Two Buffers

In this section we extend the model to a network consisting of three sources and two buffers in a configuration as shown in Figure 6.7. We explain the SPN in which the buffers use a proportional loss scheme, define the performance measures and present some results.

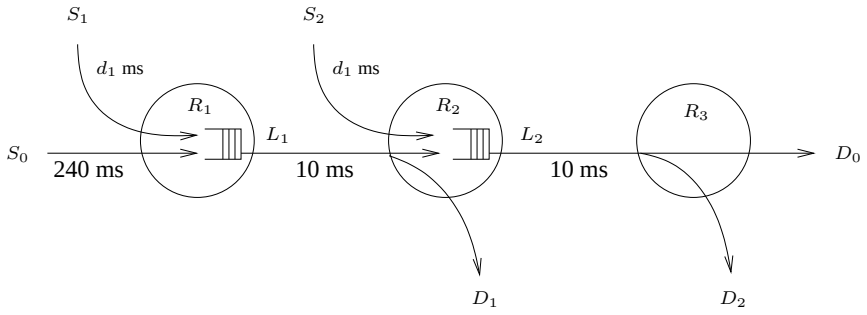

Figure 6.7: *A network of three sources sharing the links between routers R_1, R_2, and R_3. Router R_1 (R_2) contains the first (second) shared buffer in front of link L_1 (L_2). Router R_3 splits the traffic of connections 0 and 2 without further buffering*

The Model

The SPN for the network is shown in Figure 6.8. The subnets for sources 1 and 2 and the buffers B_1 and B_2 are identical to their counterparts of Section 6.2.1. Source 0, as shown by the middle, lower subnet in Figure 6.8, is different in that its connection uses both buffers. We elaborate on this now. To avoid tedious repetition we do not formally introduce variables such as L_1, B_1, etcetera, when the meaning is obvious.

The average round-trip time of source 0 is

$$T_0(\mathbf{k}) = T_0 + \frac{k_1}{K_1}\frac{B_1}{L_1} + \frac{k_2}{K_2}\frac{B_2}{L_2},$$

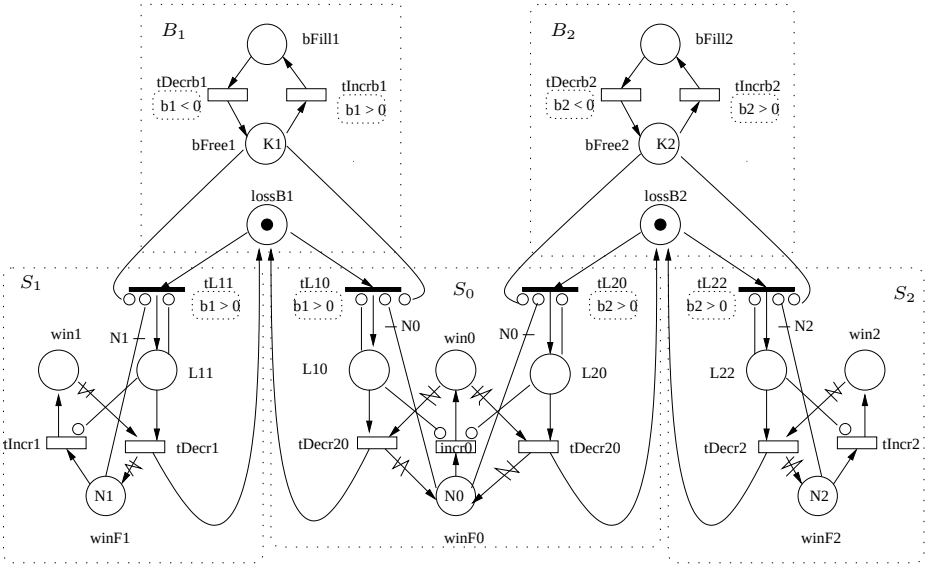

Figure 6.8: *TCP source 0 uses buffers 1 and 2, whereas source 1 (2) uses buffer 1 (2). Note that in Figure 6.1 we use rather descriptive names for the transitions. In the present case we turn to shorter, but less descriptive, names for presentational reasons.*

where we write $\mathbf{k} = (k_1, k_2)$. Thus, the analog of (5.8) becomes

$$r_0(\mathbf{k}) = \frac{P_0}{T_0(\mathbf{k})}.$$

With respect to the loss model we see that source 0 can receive a loss token from both buffers. As a consequence source 0 can have both loss tokens in possession simultaneously. As such it suffers from loss twice within one round-trip time, i.e., within one window of data, and reduces its rate twice accordingly. This is inconsistent with the behavior of TCP NewReno or TCP Sack which mostly decrease only once even when more than one packet of a window of data are lost. We contend that this undesirable side effect has small influence. First, for this event to happen, congested periods of both buffers have to overlap. Second, source 0 should receive the loss token of both buffers. As source 0 is usually sending at a lower rate than source 1 and 2, both conditions will not often be satisfied simultaneously.

The functions β_1 and β_2 should also be adapted to the network environment. Clearly,

$$\beta_1(\mathbf{n}, \mathbf{k}) = \frac{K_1}{B_1} \left(r_0(\mathbf{k}) n_0 + r_1(k_1) n_1 - L_1 \right).$$

To obtain a similar expression for β_2 we should account for the fact that the first buffer shapes the output process of source 0. We approximate the output rate of source 0 at the first buffer, δ_0 say, similarly to (6.5):

$$\delta_0(\mathbf{n}, \mathbf{k}) = \begin{cases} r_0(\mathbf{k}) n_0, & \text{if } k_1 = 0, \\ L_1 \frac{r_0(\mathbf{k}) n_0}{r_0(\mathbf{k}) n_0 + r_1(k_1) n_1}, & \text{if } k_1 > 0. \end{cases}$$

Now we can define β_2 as

$$\beta_2(\mathbf{n}, \mathbf{k}) = \frac{K_2}{B_2} \left(\delta_0(\mathbf{n}, \mathbf{k}) + r_2(k_2) n_2 - L_2 \right).$$

Performance Measures

For reasons of consistency with the definition of δ_0 above, we use the output-related definitions of throughput as in (6.5). Thus, omitting the superscript 'out',

$$\gamma_0(\mathbf{n}, \mathbf{k}) = \begin{cases} \delta_0, & \text{if } k_2 = 0, \\ \frac{\delta_0}{\delta_0 + \Delta_2} L_2, & \text{if } k_2 > 0, \end{cases}$$

$$\gamma_1(\mathbf{n}, \mathbf{k}) = \begin{cases} \Delta_1, & \text{if } k_1 = 0, \\ \frac{\Delta_1}{\Delta_0 + \Delta_1} L_1, & \text{if } k_1 > 0, \end{cases}$$

$$\gamma_2(\mathbf{n}, \mathbf{k}) = \begin{cases} \Delta_2, & \text{if } k_2 = 0, \\ \frac{\Delta_2}{\delta_0 + \Delta_2} L_2, & \text{if } k_2 > 0, \end{cases}$$

where

$$\Delta_0 \equiv r_0(\mathbf{k})n_0, \qquad \Delta_1 \equiv r_1(k_1)n_1,$$
$$\Delta_2 \equiv r_2(k_2)n_2, \qquad \delta_0 \equiv \delta_0(\mathbf{n}, \mathbf{k}).$$

With this we set

$$\gamma_i = \mathbb{E}\{\gamma_i(\mathbf{W}, \mathbf{C})\}, \quad i = 0, 1, 2, \tag{6.9}$$

where $\mathbf{C} = (C_1, C_2)$. Finally, the utilizations for the first and second link become, respectively,

$$u_1 = \frac{\gamma_0 + \gamma_1}{L_1}, \qquad u_2 = \frac{\gamma_0 + \gamma_2}{L_2}.$$

Results

Similar to Figure 6.2 we study the influence of the ratio of the round-trip times on fairness. We simultaneously vary the propagation delay d_1 of the links connecting sources 1 and 2 to the routers R_1 and R_2, respectively, from 40 ms to 250 ms in ten steps. The parameters of the second router are identical to those of the first, i.e., $L_2 = L_1 = 25.7$ and $d_{B_2} = d_{B_1} = 16$ ms, as in Scenario 1 of Table 6.4.

The left panel of Figure 6.9 shows the ratios of the throughputs γ_1/γ_0 and γ_2/γ_0 as functions of $s = T_1/T_0 = T_2/T_0$. We see that the throughputs of source 1 and 2 are nearly the same. This is to be expected when the fraction of lost traffic at the first buffer is small. Indeed, in that case the rate of the 'thinned connection 0', i.e., the traffic of connection 0 minus the loss incurred at the first buffer, is nearly the same as the transmission rate of source 0. Hence, connection 1 and connection 2 have to compete with approximately the same connection. The fact that γ_2 is just slightly larger than γ_1 shows, in accordance with the above, that the rate of the thinned connection 0 is a bit smaller than its initial rate. The right panel shows the graphs of the scaled throughputs γ_0/L_1, γ_1/L_1, and the utilization u_1. As the difference between γ_1 and γ_2 is small, the results for the second router are nearly identical, hence, not shown. Clearly, the overall utilization u_1 decreases when the 'competition between connections 0 and 1 increases'.

When $T_0 = T_1 = T_2$ we can compare γ_1/γ_0 to some theoretical fairness results for networks as derived by Massoulié & Roberts (1999). Lee *et al.* (2001) claim that, in the terminology of Massoulié & Roberts (1999), the bandwidth sharing obtained by TCP in networks results in minimum-potential-delay fairness. When we apply these results to the network shown in Figure 6.7 we obtain that, theoretically, $\gamma_1/\gamma_0 = \sqrt{2}$. Our model, on the other hand, gives $\gamma_1/\gamma_0 = 13.81/9.21 = 1.5$, which is quite near to $\sqrt{2}$.

In the left panel of Figure 6.9 we also plot the function $s \to \sqrt{2}s^{-0.85}$ as reference. Interestingly, this shows considerable agreement to the numerical results. It seems that the power of s is dictated by the loss model whereas the pre-factor is determined by the

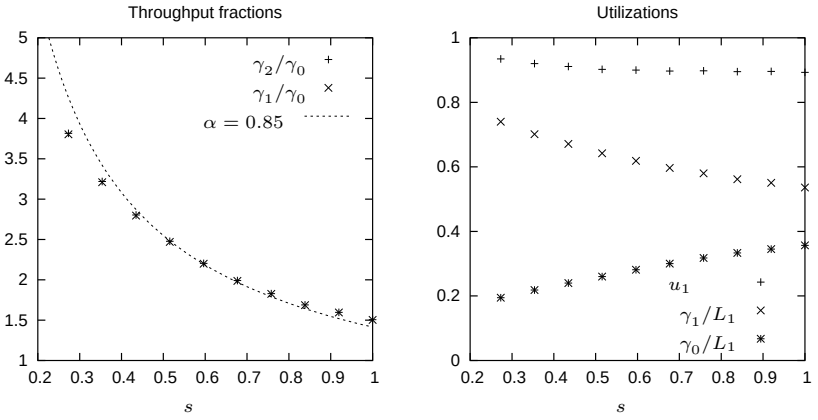

Figure 6.9: *The throughput ratios for both buffers and utilization for the first buffer as functions of* $s = T_1/T_0 = T_2/T_0$.

topology. Further research is needed to see whether, and if so, which conditions, this phenomenon holds.

6.5 Summary and Recommendations

We use stochastic Petri nets to specify, in a versatile way, Markovian models of TCP NewReno or Sack (more specifically, AIMD) sources that share one or two buffers. The first model contains two connections competing for a single bottleneck link and buffer. The second model describes one connection traversing two consecutive buffers, while each buffer receives additional side traffic from other TCP connections. We also show that the first model can be simply extended to more than two sources, and present a modification of the source model to include on/off behavior.

The methodology is flexible, extendable, and enables to obtain qualitative insight into the influence of various source and network parameters on transient and long-term properties such as source throughput, link utilization and fairness. With respect to parameters as packet size, round-trip time, and buffer size, the results of our model are consistent with those of Chapter 5 and therefore not reported here.

In the first model (two sources, one buffer) we implement two popular assumptions about the loss process at the buffer, viz. proportional loss and synchronized loss. We validate the Markovian models for either loss process by extensively comparing it, on the one hand, to the theory developed by Altman *et al.* (2000b), Altman *et al.* (2002b) and Lakshman & Madhow (1997), and, on the other hand, to simulation by ns-2. The

models provide results that are consistent with the theoretical results or improve these in that better resemblance is found to simulation results obtained by ns-2.

The second model (three sources, two buffers) shows that when the round-trip times of all connections are equal, the computed 'fairness' is approximately minimum-potential-delay fair as defined by Massoulié & Roberts (1999). It would be interesting to investigate the type of fairness in case buffer sizes are not small or when the round-trip times differ. To the best of our knowledge, the approach based on SPNs is one of the few theoretical approaches that enables such quantitative analysis. Massoulié & Roberts (1999) consider the influence of round-trip time differences, but their window-control model is non-adaptive contrary to our source model.

We like to mention that specifying the Markovian models with SPNs, so that the generator of Markov chain and the performance measures are computed automatically, has some noteworthy advantages over implementing the generator by hand as is done in Chapter 5. The implementation of the SPNs is straightforward and less error-prone. Moreover, the automatically generated Markov chains usually need fewer states, and can therefore be solved more efficiently. Finally, it is easy to include complex behavior of the application layer or modify aspects of TCP in the SPN. In summary, we feel that using SPNs shifts the burden of the work from simple but awkward programming to the pleasant task of designing a Petri net that behaves according to a set of pre-specified rules.

As a possible extension, it would be interesting to implement an intermediate level in the buffer such that when the queue exceeds this level, the buffer starts sending negative feedback signals to the source. In the context of Kelly (2000); Gibbens & Kelly (1999); Kelly *et al.* (1998) we might interpret these negative feedback signals as *charging* signals. This functionality might increase system utilization by two effects. The first is that the source may have reduced its rate before packets are dropped. Consequently, there will be fewer lost packets. Second, when the level is set quite a bit lower than the size of the buffer, the control loop becomes shorter, so that a source can adapt more promptly to over- and under-load of the link. One of the points of interest is to explore the gain in utilization when such thresholds are used. The overall effect is, however, not clear as the fraction of time the buffer is empty may also increase when using thresholds.

Chapter 7

A Tandem Queue with Server Slow-down and Blocking

In the previous chapters we have been concerned with stochastic fluid queues with feedback. It is also of interest to study the influence of such feedback on the behavior of classical queueing networks, i.e., networks that serve discrete jobs, rather than fluid. In this chapter we consider *congestion-dependent feedback of information* (not jobs) from downstream stations to upstream stations. Specifically, we analyze the consequences of feedback on the queue-length distribution in a two-station tandem queueing network in which the second station informs the first server to change its service rate depending on the queue length in the second station.

As this chapter is not concerned with feedback fluid queues, we start with a separate introduction. At the end of this introduction we present the structure of the rest of the chapter.

7.1 Introduction

The tandem queue we study here resembles a two-station Jackson tandem queue in which jobs arrive according to a Poisson process with rate λ at the first station and require at the first and second station exponentially distributed service times with mean $1/\mu_1$ and $1/\mu_2$, respectively. Thus, the load on the first and second server is $\rho_1 := \lambda/\mu_1$ and $\rho_2 := \lambda/\mu_2$, respectively. However, we allow the second station to inform the first station about the number of jobs in queue. Immediately after the second station contains n jobs, it signals the first server to stop processing any job in service. We assume that the feedback signal from the second station to the first is not delayed. When the queue length in the second station becomes less than n, the first server may resume service

again. Clearly, this reaction of the first station to a signal of the second leads to blocking at *blocking threshold* n.

Due to the presence of feedback, the stationary joint distribution π_{ij} that the number of jobs in the first and second station is i and j, respectively, does not have a product-form, so that finding a closed-form expression for π_{ij} is difficult. We therefore concentrate on its (asymptotically) dominant structure and consider the *decay rate* of the number of jobs in the first buffer. This quantity, also known as the caudal characteristic, cf. Neuts (1986), gives insight into the probability of the first queue reaching high levels. Clearly, such events occur now more often because of blocking; on the other hand, the second buffer is protected from overflow.

It is simple to bound the decay rate by viewing the two queues in tandem as one black box at which jobs arrive at rate λ. The slower server in the black box evidently dominates the total number of jobs in the system, wherever these jobs may reside in the 'box'. Thus, the decay rate of the number of jobs must be bounded from below by $\rho := \max\{\rho_1, \rho_2\}$. By 'opening the black box' we see that, as the second buffer is finite, necessarily the first queue is large when the system contains many jobs. Hence the decay rate of the number of jobs in the first station lies somewhere in the interval $(\rho, 1]$, a result also obtained by Grassman & Drekic (2000). However, in this paper we rigorously show that the decay rate as a function of the blocking threshold decreases monotonically and at least geometrically fast to ρ.

As a second topic of interest we estimate the ratio $\pi_{i,j+1}/\pi_{ij}$ when $i \gg 1$, i.e., the ratio of the probability that the number of jobs in the second queue is $j+1$ to the probability that this number is j, while the first queue is large. Thus, our approach also reveals the asymptotic probabilistic structure of the number of jobs in the second station.

By reasoning heuristically we can find a guess for $\pi_{i,j+1}/\pi_{i,j}$ when server 1 is the bottleneck, i.e., $\mu_1 < \mu_2$. First, evidently, the decay rate of the first queue should be at least ρ_1. Moreover, as server 2 works at a higher rate than server 1, presumably the queue length in the second buffer is, mostly, small. Hence, the presence of blocking should have only minor effect. In analogy with the Jackson tandem queue we therefore infer that $\pi_{i,j+1}/\pi_{i,j} \approx \lambda/\mu_2$, i.e., the decay rate of the second queue in the Jackson tandem network.

Consider now the opposite case: the second server is the bottleneck, i.e., $\mu_1 > \mu_2$. The decay rate of the first queue has to be at least ρ_2. It is therefore likely that the second buffer is mostly full, leading to the blocking of server 1. However, making an educated guess in this case about the probability distribution for the queue length at the second station proves difficult.

As a third topic we study a more complicated type of feedback then just blocking. Now, when the number of jobs at station 2 is in excess of some threshold m (which should be smaller than the blocking threshold n to be effective), server 1 *slows down*, i.e., it reduces its service rate to $\tilde{\mu}_1$, where $0 < \tilde{\mu}_1 < \mu_1$. Thus, depending on the queue

length in station 2, server 1 works at a high rate μ_1, a low rate $\tilde{\mu}_1$, or not at all. In the sequel we distinguish both types of feedback queue by calling the first the *network with blocking* and the second the *network with slow-down and blocking*. The analysis of such queueing networks with service slow-downs have interesting applications in the domain of manufacturing, but also in the design of Ethernet networks with feedback. For the network with slow-down and blocking we can establish analogous results as obtained for the network with blocking. The asymptotic distribution of the number in the second queue turns out to be of particular interest in this case.

Our focus on the asymptotic behavior of π_{ij} has two reasons. First, the resulting expressions are in closed form, contrary to the numerical methods, to be discussed presently, available in the literature. Second, given the rapid convergence of the sequence of networks with blocking when the blocking threshold n increases, the asympotic system provides considerable insight into the form of π_{ij} even when the blocking threshold is not (very) large, e.g., $n \geq 10$.

Tandem queues with blocking (but *without slow-down*) received considerable attention over the years. Konheim & Reiser (1976, 1978) take z-transforms of the balance equations satisfied by π_{ij} and study the properties of the resulting generating function to establish a stability condition and devise an algorithm to compute π_{ij}. The derivation of the stability condition for this and related models is simplified by Latouche & Neuts (1980) by using the methods of Quasi-Birth-Death (QBD) processes. Grassman & Drekic (2000) derive, also by using QBDs, a more efficient numerical procedure to compute π_{ij}. They restrict the eigenvalues to a set of (non-overlapping) intervals. After locating the eigenvalues in the bounding intervals, they derive a recursion to obtain the associated eigenvectors. Finally, a suitable linear combination of the eigenvectors should solve the boundary conditions for π_{0j}. Interestingly, by using the bounding intervals derived by Grassman & Drekic (2000) for the eigenvalues, our approach extends straightforwardly to a method to compute π_{ij} with the same algorithmic complexity as their's. In a final remark Grassman & Drekic (2000) mention the idea of slow-down, however, they do not analyze the consequences in detail. Kroese *et al.* (2004) also consider a two-station tandem queue with blocking. However, now the rate of the arrival process is set to zero when the first station contains n jobs. The second buffer is assumed infinitely large. For this system the authors compute the decay rate of the number of jobs in the second buffer. They also consider the limiting regime in which $n \to \infty$. Leskelä (2004) studies a two-station tandem network with feedback, but now station 2, rather than server 1, provides feedback to the arrival process to change service as a function of the length of the second queue. He establishes a stability criterion for the system with unlimited first *and* second buffer.

The chapter has the following structure. In Section 7.2 we specify the network with blocking in detail and interpret it as a QBD process. Next, we show in Section 7.3 that for fixed blocking level n, the decay rate x_n of the number of jobs in the ***first*** buffer lies in

the interval $(\rho, 1)$, where $\rho = \max\{\rho_1, \rho_2\}$, a result also derived by Grassman & Drekic (2000). Then, in Section 7.4, we prove that $x_n \downarrow \rho$ geometrically fast when $n \to \infty$. As a consequence, the bottleneck server determines the decay rate, however large the blocking level. As a second topic in Section 7.4 we study the structure of the distribution of the number of jobs in the second buffer. In Section 7.5 we consider similar topics for the tandem queue with threshold and blocking.

7.2 Model and Preliminaries

We now present the model and write it as a QBD process. We then discuss by which method we obtain the decay rate in Section 7.3.

Jobs arrive according to a Poisson process with rate λ and require exponentially distributed service with rate μ_1 and μ_2 at the first and second station, respectively. We assume throughout this paper that $\mu_1 \neq \mu_2$. After service completion at the first station, jobs move on to the second. Once service is completed there also, jobs leave the network. Let $X_i^{(n)}(t)$ denote the number of jobs at station i, $i = 1, 2$, at time t (including the job in service). When $X_2^{(n)}(t)$ is equal to the blocking threshold n, the first server blocks, i.e., its service rate becomes zero. Right after the departure of the job in service at the second station, the first server resumes service (if a job is present there, of course). It is clear that the joint process $\{X_1^{(n)}(t), X_2^{(n)}(t)\} \equiv \{X_1^{(n)}(t), X_2^{(n)}(t), t \geq 0\}$ is a (continuous-time) Markov chain. The state space of this process is $\mathscr{X}^{(n)} = \{(i, j) \,|\, i = 0, 1, 2, \ldots; j = 0, 1, \ldots, n\}$. We present the state transition diagram of $\{X_1^{(n)}(t), X_2^{(n)}(t)\}$ in Figure 7.1. Finally, let

$$\rho_1 := \lambda/\mu_1, \ \rho_2 := \lambda/\mu_2, \text{ and } \rho := \max\{\rho_1, \rho_2\}, \tag{7.1}$$

i.e., ρ is the load at the slowest server.

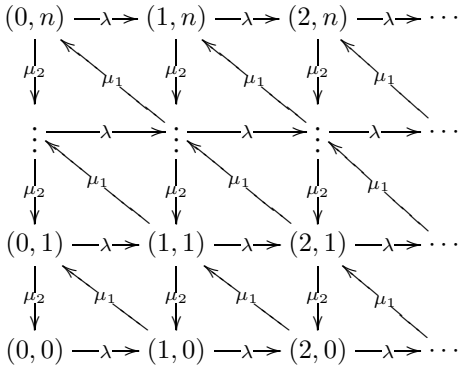

Figure 7.1: *State space and transition rates of the truncated tandem queue.*

The Markov process $\{X_1^{(n)}(t), X_2^{(n)}(t)\}$ can be interpreted as a continuous-time QBD process. We identify some common subsets of $\mathscr{X}^{(n)}$ associated specifically to the QBD structure. *Level* i contains all states $(i,j) \in \mathscr{X}^{(n)}$ with i constant. *Phase* j contains the states (i,j) with j constant. Thus, the levels contain the 'vertical' sets of states in Figure 7.1, whereas the phases contain the 'horizontal' sets of states.

To facilitate the presentation we prefer to concentrate on the aperiodic discrete-time Markov chain $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ obtained by uniformizing $\{X_1^{(n)}(t), X_2^{(n)}(t)\}$ at rate

$$a := \lambda + \mu_1 + \mu_2.$$

This procedure allows us to refer directly to a number of results in the literature which we otherwise have to reformulate for the continuous-time model. Evidently, by PASTA, the results we derive for $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ apply also to $\{X_1^{(n)}(t), X_2^{(n)}(t)\}$.

Writing

$$p := \frac{\lambda}{a}, \qquad q := \frac{\mu_1}{a}, \qquad r := \frac{\mu_2}{a}$$

for the transition probabilities associated to λ, μ_1 and μ_2, the matrix of transition probabilities of the QBD $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ is of the form

$$P^{(n)} = \begin{pmatrix} B^{(n)} & A_0^{(n)} & & & \\ A_2^{(n)} & A_1^{(n)} & A_0^{(n)} & & \\ & \ddots & \ddots & \ddots & \end{pmatrix}. \tag{7.2a}$$

Here the $(n+1) \times (n+1)$ matrices in $P^{(n)}$ are given by

$$B^{(n)} = \begin{pmatrix} q+r & & & & \\ r & & q & & \\ & \ddots & & \ddots & \\ & & & r & q \end{pmatrix}, \tag{7.2b}$$

$$A_0^{(n)} = p\, I^{(n)}, \tag{7.2c}$$

$$A_1^{(n)} = \begin{pmatrix} r & & & & \\ r & 0 & & & \\ & \ddots & \ddots & & \\ & & r & 0 & \\ & & & r & q \end{pmatrix}, \tag{7.2d}$$

and

$$A_2^{(n)} = \begin{pmatrix} 0 & q & & \\ & \ddots & \ddots & \\ & & 0 & q \\ & & & 0 \end{pmatrix}, \tag{7.2e}$$

and $I^{(n)}$ is the $(n+1) \times (n+1)$ identity matrix.

Provided a certain stability criterion to be addressed in Lemma 7.1 below is satisfied, an irreducible QBD chain is positive recurrent. Consequently, its stationary probability vector exists. Let us henceforth consider the system in steady state, and write for brevity $X_i^{(n)}$, $i = 1, 2$, for $X_{i,k}^{(n)}$ at an arbitrary point in time. Furthermore, let $\pi_{ij}^{(n)} = \mathbb{P}\{X_1^{(n)} = i, X_2^{(n)} = j\}$, i.e., the steady-state probability that the number of jobs in the first and second station is i and j respectively.

It can be shown that the stationary probability vector $\boldsymbol{\pi}^{(n)}$ can be appropriately partitioned as

$$\boldsymbol{\pi}^{(n)} = \left(\boldsymbol{\pi}_0^{(n)}, \boldsymbol{\pi}_0^{(n)} R^{(n)}, \boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^2, \ldots, \right), \tag{7.3}$$

where $\boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^i = \left(\pi_{i0}^{(n)}, \pi_{i1}^{(n)}, \ldots, \pi_{in}^{(n)}\right)$ and $R^{(n)}$ is the minimal nonnegative solution of the equation

$$A_0^{(n)} + R^{(n)} A_1^{(n)} + \left(R^{(n)}\right)^2 A_2^{(n)} = R^{(n)}. \tag{7.4}$$

For our case $R^{(n)}$ has to be computed numerically, for instance with the algorithms derived by Latouche & Ramaswami (1999).

Rather than computing $R^{(n)}$ directly, Neuts (1986) associates two interesting (probabilistic) quantities to $R^{(n)}$. He starts by observing that when $R^{(n)}$ is irreducible, it satisfies

$$\left(R^{(n)}\right)^i = (x_n)^i \left(\mathbf{u}^{(n)}\right)' \cdot \mathbf{v}^{(n)} + o\left((x_n)^i\right), \quad \text{as } i \to \infty, \tag{7.5}$$

where $\mathbf{v}^{(n)} = (v_0^{(n)}, \ldots, v_n^{(n)})$ and $\mathbf{u}^{(n)}$ are strictly positive left and right eigenvectors of $R^{(n)}$ associated to its largest eigenvalue $x_n \in (0, 1)$. (The prime denotes the transpose of a vector.) The first quantity of interest is

$$\lim_{i \to \infty} \frac{\boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^{i+1} \mathbf{e}}{\boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^i \mathbf{e}} = x_n, \tag{7.6}$$

where $\mathbf{e}$ is the (column) vector consisting of ones. This says that the ratio of the expected time spent at a high level $i+1$ to that spent at level i is approximately equal to x_n. In

other words, the largest eigenvalue x_n of $R^{(n)}$ is the *geometric decay rate*, which is also known as the caudal characteristic, cf Neuts (1986), of the QBD process. Second,

$$\lim_{i\to\infty} \frac{(\boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^i)_j}{\boldsymbol{\pi}_0^{(n)} \left(R^{(n)}\right)^i \mathbf{e}} = v_j^{(n)}, \tag{7.7}$$

which is to say that (in stationary state) the probability that the chain is in phase j conditional on being in level i, is approximately equal to $v_j^{(n)}$ for large i.

In the sequel we are concerned with determining the probabilistic structure of the number of jobs in the first and second station in steady state for high levels. Specifically, in Section 7.3 we derive some properties of the largest eigenvalue x_n of $R^{(n)}$. We show, first, that $\rho < x_n < 1$ for any $n < \infty$ and, second, that the sequence $\{x_n\}_n$ converges to ρ when $n \to \infty$. In Section 7.4 we are concerned with determining the associated left eigenvector $\mathbf{v}^{(n)}$ in the limit $i \to \infty$ and $n \to \infty$; the order of the limits is important. The results of this section provide insight into the ratio of subsequent components of $\mathbf{v}^{(n)}$ which, in turn, reveals information about $\pi_{i,j+1}^{(n)}/\pi_{ij}^{(n)}$.

It remains to discuss the stability condition of the chain $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$, which we henceforth assume satisfied. The proof of the next lemma involves the matrix $A^{(n)}(x)$, which is also useful for later purposes,

$$\begin{aligned} A^{(n)}(x) &= A_0^{(n)} + xA_1^{(n)} + x^2 A_2^{(n)} \\ &= \begin{pmatrix} p+rx & qx^2 & & \\ rx & p & qx^2 & \\ & \ddots & \ddots & \ddots \\ & & rx & p+qx \end{pmatrix}, \quad \text{for } x \in [0,1]. \end{aligned} \tag{7.8}$$

Lemma 7.1. *The chain $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ is positive recurrent if and only if*

$$\frac{\lambda}{\mu_1\mu_2} \frac{{\mu_1}^{n+1} - {\mu_2}^{n+1}}{{\mu_1}^n - {\mu_2}^n} < 1. \tag{7.9}$$

This condition is equivalent to:

$$n > N(\rho_1, \rho_2) := \frac{\log(1-\rho_1) - \log(1-\rho_2)}{\log\rho_2 - \log\rho_1}. \tag{7.10}$$

Proof. It is simple to see that the QBD $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ is irreducible and that the number of phases is finite. Moreover, the stochastic matrix $A^{(n)}(1)$ is irreducible. These properties allow us to apply Latouche & Ramaswami (1999: Theorem 7.2.3). This theorem states that the QBD is positive recurrent iff $\boldsymbol{\alpha}A_0^{(n)}\mathbf{e} < \boldsymbol{\alpha}A_2^{(n)}\mathbf{e}$, where $\boldsymbol{\alpha}$ is the stationary probability vector of $A^{(n)}(1)$. Clearly, $A^{(n)}(1)$ is the stochastic matrix of a simple

birth-death process. Hence, the desired solution vector $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_n)$ is given by $\alpha_i = \alpha_0 \beta^i$, $0 \leq i \leq n$, where $\beta = \mu_1/\mu_2$ and

$$\alpha_0 = \left(\sum_{i=0}^{n} \beta^i \right)^{-1} = \frac{1-\beta}{1-\beta^{n+1}}.$$

The condition $\boldsymbol{\alpha} A_0^{(n)} \mathbf{e} < \boldsymbol{\alpha} A_2^{(n)} \mathbf{e}$ becomes $\lambda < \mu_1 \sum_{i=0}^{n-1} \alpha_i$, which leads to (7.9). Equation (7.10) follows from (7.9) after some calculations. ∎

7.3 The Geometric Decay Rate

In this section we prove that for fixed blocking threshold n the decay rate x_n lies in the open interval $(\rho, 1)$. To achieve this, we use the following result stated in Latouche & Ramaswami (1999: Section 9.1)

Theorem 7.2. *The decay rate x_n is the unique solution in $(0,1)$ of the equation*

$$x = \xi^{(n)}(x), \tag{7.11}$$

where $\xi^{(n)}(x)$ is the spectral radius of $A^{(n)}(x)$.

We apply this as follows. Since $A^{(n)}(x)$ is irreducible and nonnegative for $x > 0$, it follows from the theorem of Perron-Frobenius that the spectral radius $\xi^{(n)}(x)$ is also the largest (and simple) eigenvalue of $A^{(n)}(x)$. Suppose now that we can find an $(n+1)$-dimensional row vector $\mathbf{v}^{(n)} > 0$, i.e., each component $v_j^{(n)}$ of $\mathbf{v}^{(n)}$ is strictly positive, and $x > 0$ such that

$$\mathbf{v}^{(n)} A^{(n)}(x) = \mathbf{v}^{(n)} x. \tag{7.12}$$

Then by the theorem of Perron-Frobenius, x necessarily solves the equation $x = \xi^{(n)}(x)$, and $\mathbf{v}^{(n)}$ is the left Perron-Frobenius vector associated to x. In Section 7.3.1 we use this formula to efficiently combine $\xi^{(n)}(x)$ and the components of the Perron-Frobenius eigenvector into one numerical sequence. As these numbers can be written as a recurrence relation, we explore the properties of this recurrence in Section 7.3.2. Finally, in Section 7.3.3, by combining and exploiting these properties in various ways we can characterize the decay rate x_n.

In this section the blocking threshold n is fixed; hence, when no confusion arises, we mostly suppress the dependence on n here. However, we always write x_n for the decay rate and $\xi^{(n)}(x)$ for the spectral radius.

Remark 7.3. The approach below is entirely analytic. It is, perhaps, somewhat unsatisfactory that we do not use probabilistic arguments in the analysis. However, this approach enables us to explore *networks with slow-down and blocking*, which seems much

more complicated to handle probabilistically. In a sense, with the recurrence relations of Section 7.3.2 we can analyze the structure of the probability distribution above the slow-down threshold on the same footing as below the slow-down threshold.

7.3.1 A Consequence of the Perron-Frobenius Theorem

Let us interpret (7.12) as a constraint on x and $\mathbf{v}$ and work out its implications. Thus, assuming that (7.12) is true and expanding with (7.8) we find that $x > 0$ and $\mathbf{v} > 0$ should satisfy

$$x = p + rx + \frac{rxv_1}{v_0}, \tag{7.13a}$$

$$x = \frac{qx^2 v_{j-1}}{v_j} + p + \frac{rxv_{j+1}}{v_j}, \quad 1 \le j < n, \tag{7.13b}$$

$$x = \frac{qx^2 v_{n-1}}{v_n} + p + qx. \tag{7.13c}$$

From the first relation we see that for given x and v_0, the value of v_1 follows. But then, the second relation provides $v_2, \ldots, v_n$. Since we are free to choose the norm of $\mathbf{v}$, we can set, arbitrarily, $v_0 \equiv 1$. As a consequence, the first and second relation completely fix $\mathbf{v}$ once x is given. The third relation forms a necessary condition on x such that x and $\mathbf{v}$ indeed form an eigenvalue and eigenvector pair of $A(x)$. In other words, whereas the simultaneous validity of the first and second relation above leaves x free, the third relation fixes it.

To clarify the structure of (7.13) and the dependence on x somewhat further, we define the following sequence of functions of x:

$$\chi_0(x) := \mu_1 x^2, \tag{7.14a}$$

$$\chi_j(x) := arx\frac{v_j}{v_{j-1}} = \mu_2 x\frac{v_j}{v_{j-1}}, \qquad 1 \le j \le n, \tag{7.14b}$$

$$\chi_{n+1}(x) := ax - \lambda - \frac{\mu_1\mu_2 x^3}{\chi_n(x)}; \tag{7.14c}$$

recall that $r = \mu_2/a$. We define $\chi_0(x)$ and $\chi_{n+1}(x)$ for notational convenience, although they do not relate immediately to $\mathbf{v}$ by (7.14b). Now, multiply the left and right hand sides of (7.13) by $a = \lambda + \mu_1 + \mu_2$ and rearrange, to obtain, respectively,

$$\chi_1(x) = ax - \lambda - \frac{\mu_1\mu_2 x^3}{\chi_0(x)} = (\lambda + \mu_1)x - \lambda, \tag{7.15a}$$

$$\chi_j(x) = ax - \lambda - \frac{\mu_1\mu_2 x^3}{\chi_{j-1}(x)}, \qquad 2 \le j \le n+1, \tag{7.15b}$$

$$\chi_{n+1}(x) = \mu_1 x. \tag{7.15c}$$

From the above we conclude the following.

Theorem 7.4. *Let $x \in (0,1)$ be such that the sequence $\{\chi_j(x)\}_{0\le j\le n+1}$ satisfies (7.15) and each element $\chi_j(x) > 0$. Then x is the unique solution of $\xi^{(n)}(x) = x$, i.e., x equals the geometric decay rate x_n of the tandem queue with blocking at threshold n.*

Proof. When x satisfies the hypothesis, the validity of (7.12) follows by constructing $\mathbf{v}$ according to (7.14b). Regarding the positivity of $\mathbf{v}$, which we do not require in the definition (7.14) of $\chi_j(x)$, the conditions $x > 0$ and $\chi_j > 0$ imply that v_j and v_{j-1} have the same sign. Hence, as all $\chi_j > 0$, it is straightforward to construct $\mathbf{v} > 0$. ∎

Remark 7.5. It is apparent from (7.15) that the desired x can be expressed as a root of a rational function. However, this insight might not provide the easiest method to characterize the decay rate. With the approach below we can achieve our goals with elementary methods. Hence, we do not try to bound the decay rate by locating or bounding the root(s) of rational functions.

Our search for the decay rate x_n motivates a study of the structure of the sequence $\{\chi_j(x)\}_{0\le j\le n+1}$.

7.3.2 A Useful Recursion

Clearly, (7.15) shows that the elements of $\{\chi_j(x)\}_{0\le j\le n+1}$ satisfy a recurrence relation. Let

$$T : \eta \mapsto ax - \lambda - \frac{\mu_1\mu_2 x^3}{\eta}. \tag{7.16}$$

Then we can write

$$\chi_{j+1}(x) = T\left(\chi_j(x)\right), \qquad \text{for } 0 \le j \le n. \tag{7.17}$$

It turns out that T is the key to understanding the structure of $\{\chi_j\}$, and thereby to obtaining the decay rate.

The mapping $\eta \to T(\eta)$ is a hyperbolic linear fractional transformation, see, e.g., Needham (2000). It is infinitely differentiable everywhere except in the origin, and it has an inverse

$$T^{(-1)} : \eta \mapsto \frac{\mu_1\mu_2 x^3}{ax - \lambda - \eta}.$$

The equation $\eta = T(\eta)$ reveals that T has two fixed points: η_+ and η_-. These points are the solutions of the quadratic (in η) equation $\eta^2 - (ax-\lambda)\eta + \mu_1\mu_2 x^3 = 0$ so that

$$\eta_\pm = \frac{ax-\lambda}{2} \pm \frac{1}{2}\sqrt{(ax-\lambda)^2 - 4\mu_1\mu_2 x^3}. \tag{7.18}$$

Below we show that only real-valued $\eta_\pm$ are of importance for our purposes. Hence, it suffices to take x such that the discriminant

$$D(x) = (ax - \lambda)^2 - 4\mu_1\mu_2 x^3 > 0. \tag{7.19}$$

The behavior of the sequence of iterates $\ldots, T^{(-1)}(\eta), T^{(0)}(\eta) := \eta, T^{(1)}(\eta), \ldots$ for $\eta \in (\eta_-, \eta_+)$ is also of interest. The next lemma formalizes what might be anticipated from Figure 7.2.

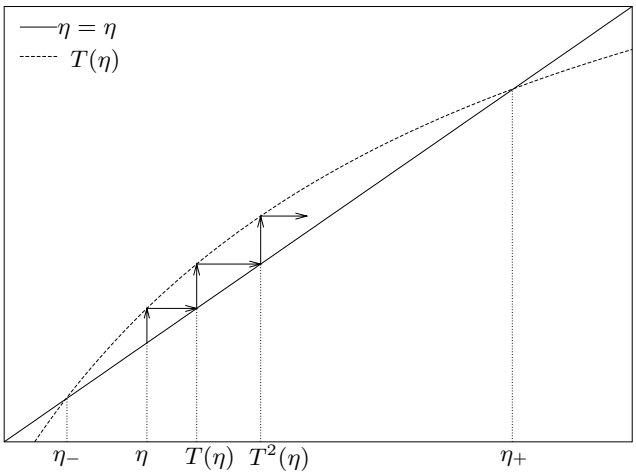

Figure 7.2: *Some properties of the mapping $\eta \to T(\eta)$. The variable η is set out along the horizontal axis. The solid line refers to the line $\eta = \eta$.*

Lemma 7.6. *If x such that $D(x) > 0$ and $\eta \in (\eta_-, \eta_+)$,*

$$\eta_- = \inf_{i>1}\left\{T^{(-i)}(\eta)\right\} < T^{(-1)}(\eta) < \eta < T(\eta) < \sup_{j>1}\left\{T^{(j)}(\eta)\right\} = \eta_+,$$

$$\eta_+ - T^{(j)}(\eta) < \left(\frac{\eta_-}{\eta}\right)^j (\eta_+ - \eta), \qquad j > 0,$$

$$T^{(-i)}(\eta) - \eta_- < \left(\frac{\eta}{\eta_+}\right)^i (\eta - \eta_-), \qquad i > 0.$$

Proof. First, from (7.18),

$$\begin{aligned} \eta_+ + \eta_- &= ax - \lambda, \\ \eta_-\eta_+ &= \mu_1\mu_2 x^3. \end{aligned}$$

Now, as $\eta \in (\eta_-, \eta_+)$, it follows that

$$\begin{aligned}\eta_+ - T(\eta) &= \eta_+ - (ax - \lambda) + \frac{\mu_1\mu_2 x^3}{\eta} \\ &= -\eta_- + \frac{\eta_-\eta_+}{\eta} = \frac{\eta_-}{\eta}(\eta_+ - \eta),\end{aligned}$$

Clearly, η_-/η and $\eta_+ - \eta$ are positive, which implies $\eta_+ > T(\eta)$. Moreover, $\eta_-/\eta < 1$ so that $\eta_+ - T(\eta) < \eta_+ - \eta$. Therefore, for all $\eta \in (\eta_-, \eta_+)$ we have $\eta_- < \eta < T(\eta) < \eta_+$. Concerning the convergence rate to η_+, note that

$$\begin{aligned}\eta_+ - T^{(2)}(\eta) &= \frac{\eta_-}{T(\eta)}(\eta_+ - T(\eta)) \\ &= \frac{\eta_-^2}{T(\eta)\eta}(\eta_+ - \eta) < \left(\frac{\eta_-}{\eta}\right)^2 (\eta_+ - \eta).\end{aligned}$$

By induction, $T^{(j)}(\eta) \to \eta_+$ geometrically fast.

By similar computations we obtain

$$T^{(-1)}(\eta) - \eta_- = \frac{T^{(-1)}(\eta)}{\eta_+}(\eta - \eta_-) > 0.$$

So, $T^{(-1)}(\eta) \in (\eta_-, \eta_+)$ whenever $\eta \in (\eta_-, \eta_+)$. Moreover,

$$T^{(-i)}(\eta) - \eta_- < (\eta/\eta_+)^i(\eta - \eta_-).$$

∎

7.3.3 Bounding the Geometric Decay Rate

The properties of the mapping T help to further characterize the sequence $\{\chi_j\}_{0\le j\le n+1}$. By appropriately combining the material we assemble here it follows that $x_n \in (\rho, 1)$.

We start with pointing out an interesting, and perhaps unexpected, relation between the stability of the QBD chain $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ and the sequence $\{\chi_j\}_{0\le j\le n+1}$.

Lemma 7.7. *The stability condition* (7.9) *on the Markov chain* $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ *is satisfied if and only if*

$$\mu_1 > \chi'_{n+1}(1).$$

Proof. First of all, the differentiability of T implies (by the chain rule) that $\chi_{n+1}(x)$ has a derivative. Next, from (7.15) it is immediate that $\chi_j(1) = \mu_1$ for all $j = 0, \ldots, n+1$. Hence, from (7.15) and writing $\beta = \mu_1/\mu_2$ as in the proof of Lemma 7.1, we find by induction

$$\chi'_j(1) = (\lambda + \mu_1)\frac{1 - \beta^{-j}}{1 - \beta^{-1}} - 2\mu_2\frac{1 - \beta^{-j+1}}{1 - \beta^{-1}}, \quad 1 \le j \le n+1.$$

The condition $\chi'_{n+1}(1) < \mu_1$ is therefore equivalent to

$$\lambda \frac{1-\beta^{-(n+1)}}{1-\beta^{-1}} + \mu_1 \beta^{-1} \frac{1-\beta^{-n}}{1-\beta^{-1}} - 2\mu_2 \frac{1-\beta^{-n}}{1-\beta^{-1}} < 0.$$

After a bit of algebra we see that this condition is precisely (7.9). ∎

Let us now concentrate on the fixed points η_+ and η_- of T. From their definition (7.18) it can be seen that they are actually functions of x. To provide further intuition about these functions, we plot in Figure 7.3 their graphs together with $\chi_2(x)$ and $\chi_3(x)$.

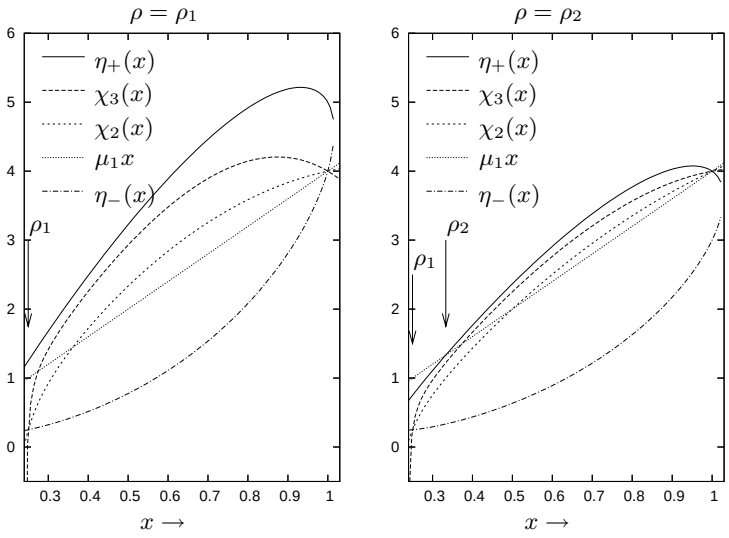

Figure 7.3: *Plots of the functions $\chi_2(x)$, $\chi_3(x)$, and $\eta_\pm(x)$. In the left panel $\lambda = 1, \mu_1 = 4, \mu_2 = 5$, while in the right $\lambda = 1, \mu_1 = 4, \mu_2 = 3$.*

Lemma 7.8. *First, the functions $x \to \eta_\pm(x)$ are real valued and positive on $[\rho, 1]$. Second,*

$$\eta_-(x) < \chi_0(x) = \mu_1 x^2 < \eta_+(x), \quad \text{if } x \in (\rho, 1). \tag{7.20}$$

Third,

$$\chi_0(\rho_1) = \lambda\rho_1 = \eta_-(\rho_1), \tag{7.21a}$$

$$\chi_0(\rho_2) \in (\eta_-(\rho_2), \eta_+(\rho_2)) = (\lambda\rho_2, \mu_1\rho_2), \quad \text{if } \rho = \rho_2, \tag{7.21b}$$

$$\chi_0(1) = \mu_1 = \begin{cases} \eta_-(1), & \text{if } \rho = \rho_1, \\ \eta_+(1), & \text{if } \rho = \rho_2. \end{cases} \tag{7.21c}$$

Proof. For the first claim we focus on the discriminant $D(x) = (ax - \lambda)^2 - 4\mu_1\mu_2 x^3$ in the definition of $\eta_\pm(x)$. Clearly, $D(x)$, being a cubic polynomial, can have at most three real roots: ξ_1, ξ_2, and ξ_3, say. By simple computations we see that $D(0) > 0$, $D(\lambda/a) < 0$, $D(\rho_2) > 0$, $D(\rho_1) > 0$, $D(1) \geq 0$, and $\lim_{x\to\infty} D(x) = -\infty$. It follows that $0 < \xi_1 < \lambda/a < \xi_2 < \min\{\rho_1, \rho_2\} \leq \max\{\rho_1, \rho_2\} < 1 \leq \xi_3$. So, on $[\rho, 1]$ the discriminant $D(x)$ is positive, and $\eta_\pm(x)$ are real valued. It is now simple to check that $\eta_\pm(x) > 0$ for $x \in [\rho, 1]$.

To prove the second claim, rewrite the inequality $\eta_-(x) < \mu_1 x^2 < \eta_+(x)$ to

$$(2\mu_1 x^2 - (ax - \lambda))^2 \leq (ax - \lambda)^2 - 4\mu_1\mu_2 x^3.$$

After some algebra and using the positivity of x we find the above to be equivalent to $\lambda(1 - x) < \mu_1\, x(1 - x)$. This is clearly true for all $x \in (\rho_1, 1)$ and, hence, for all $x \in (\rho, 1)$.

Verifying the third claim is simple. ∎

With the above observations it is straightforward to apply Lemma 7.6 to the functions $\chi_j(x)$, $0 \leq j \leq n + 1$. For later purposes we formulate this intermediate result in somewhat greater generality than is necessary for the moment. The generalization consists of extending $\{\chi_j(x)\}_{0\leq j\leq n+1}$ to a doubly infinite sequence $\{\chi_j(x)\}_{j\in\mathbb{Z}}$ by continuing in (7.17) the iterative operation of T and $T^{(-1)}$ beyond χ_{n+1} and χ_0, respectively. Thus, define for $j \geq 1$,

$$\chi_j(x) := T^{(j)}\left(\chi_0(x)\right) = T\left(T^{(j-1)}\left(\chi_0(x)\right)\right) = T\left(\chi_{j-1}(x)\right),$$
$$\chi_{-j}(x) := T^{(-j)}\left(\chi_0(x)\right) = T^{(-1)}\left(T^{(-j+1)}\left(\chi_0(x)\right)\right) = T^{(-1)}\left(\chi_{-j+1}(x)\right).$$

This extension allows us to state the following.

Lemma 7.9. *Whenever $x \in (\rho, 1)$,*

$$\begin{aligned}\eta_-(x) &< \ldots < \chi_{-i}(x) < \ldots \\ &< \chi_0(x) < \chi_1(x) < \ldots < \chi_{n+1}(x) < \ldots \\ &< \chi_j(x) < \ldots < \eta_+(x),\end{aligned} \tag{7.22}$$

for $i > 0$ and $j > n+1$. Moreover, $\chi_{-i}(x) \to \eta_-(x)$ and $\chi_j(x) \to \eta_+(x)$ geometrically fast for $i, j \to \infty$.

Proof. As, by Lemma 7.8, $x \in (\rho, 1)$ implies that $\chi_0(x) \in (\eta_-(x), \eta_+(x))$, we can use $\chi_0(x)$ as the 'starting point' for (the iterates of) T and $T^{(-1)}$ and apply Lemma 7.6. ∎

As a last intermediate result we consider the concavity of the sequence of functions $\chi_j(x)$, $2 \leq j \leq n+1$, and $\eta_+(x)$. Proving that $\eta_+(x)$ is concave is not immediate as the discriminant (7.19) need *not* be concave on $(\rho, 1)$.

Lemma 7.10. *The functions $\chi_j(x)$, $2 \le j \le n+1$, and $\eta_+(x)$ are strictly concave on $(\rho, 1)$. The function $\eta_-(x)$ is strictly convex on $(\rho, 1)$.*

Proof. We assert by induction that $\chi_j''(x) < 0$ for all $x \in (\rho, 1)$ and $j \ge 2$. First, $\chi_1(x) = (\lambda + \mu_1)x - \lambda$ is concave. Now, for $j \ge 2$, we have by (7.15),

$$\frac{\chi_j''(x)}{\mu_1 \mu_2} = \frac{x}{\chi_{j-1}(x)} \left(-6 + 6\,\frac{x\,\chi_{j-1}'(x)}{\chi_{j-1}(x)} - 2\left(\frac{x\,\chi_{j-1}'(x)}{\chi_{j-1}(x)}\right)^2 + \frac{x^2 \chi_{j-1}''(x)}{\chi_{j-1}(x)} \right).$$

Let $y(x) = x\chi_{j-1}'(x)/\chi_{j-1}(x)$ and write the first three terms within the brackets as the parabola $-6 + 6y - 2y^2$. It is simple to see that, as both roots are not real, this parabola is negative for all y. The fourth term in the expression above cannot be positive as $\chi_{j-1}(x) > 0$ for $x \in (\rho, 1)$ and $\chi_{j-1}''(x) \le 0$, by the induction hypothesis. Hence, $\chi_j''(x) < 0$.

Now, for any $x, y \in [\rho, 1]$, and $\alpha \in (0, 1)$ take the limit $j \to \infty$ of both sides of

$$\chi_j(\alpha x + (1-\alpha)y) > \alpha \chi_j(x) + (1-\alpha)\chi_j(y),$$

and conclude that $\eta_+(x)$ is also strictly concave. Finally, since $\eta_-(x) = ax - \lambda - \eta_+(x)$, it follows that $\eta_-(x)$ is strictly convex. ∎

By now we have identified all required intermediate results so that we can bound x_n from below.

Theorem 7.11. *Suppose the system is stable. Then, the decay rate x_n lies in the interval $(\rho, 1)$, where $\rho \equiv \max(\rho_1, \rho_2)$.*

Proof. We prove that the conditions of Theorem 7.4 are satisfied. Regarding the positivity of the numbers $\chi_j(x)$ for $x \in (\rho, 1)$ we have by Lemma 7.9 that $\chi_j(x) > \eta_-(x) > 0$ for $j = 0, \ldots, n+1$. It remains to prove that the function $\chi_{n+1}(x)$ intersects the line $\mu_1 x$ somewhere in the interval $(\rho, 1)$. First, from (7.21a) $\chi_{n+1}(\rho_1) = \eta_-(\rho_1) = \lambda\rho_1 < \mu_1\rho_1$. Also, when $\rho = \rho_2$, $\chi_0(\rho_2) \in (\eta_-(\rho_2), \eta_+(\rho_2))$, which implies by (7.22) that $\chi_{n+1}(\rho_2) < \eta_+(\rho_2) = \mu_1\rho_2$. Hence, $\chi_{n+1}(\rho) < \mu_1\rho$. On the other hand, $\chi_{n+1}(1) = \mu_1$ and $\chi_{n+1}'(1) < \mu_1$, by Lemma 7.7. Consequently, the concavity of $\chi_{n+1}(\cdot)$ implies there exists a unique $x \in (\rho, 1)$ such that $\chi_{n+1}(x) = \mu_1 x$. ∎

As a direct by-product of the above proof and the uniqueness of the solution of $\mu_1 x = \chi_{n+1}(x)$ in $(0, 1)$ we obtain

Corollary 7.12. *$\chi_{n+1}(x) < \mu_1 x$ for all $x \in (\rho, x_n)$ and $\chi_{n+1}(x) > \mu_1 x$ for all $x \in (x_n, 1)$.*

Remark 7.13. This corollary shows that we can find x_n numerically by the method of bisection. Take the first estimate $x_{n,1}$ of x_n as $(\rho+1)/2$. Compute $\chi_j(x_{n,1})$ for $j=0,\ldots,n+1$. If $\chi_{n+1}(x_{n,1}) > \mu_1 x_{n,1}$ then $x_{n,1}$ must be too large by the corollary, whereas if $\chi_{n+1}(x_{n,1}) < \mu_1 x_{n,1}$, the estimate $x_{n,1}$ must be too small. Based on this result we can compute the next estimate $x_{n,2}$, and so on. Clearly, the sequence $\{x_{n,m}\}_{m\geq 1}$ converges to x_n.

At this point the computation of x_1 is very simple indeed. The equation $\chi_2(x) = \mu_1 x$ reduces to

$$\frac{(x-1)\,\left(\mu_2\,\mu_1\,x^2 - \lambda\,a\,x + \lambda^2\right)}{\chi_1(x)} = 0.$$

Since $x_1 \in (\rho, 1)$ we conclude that

$$x_1 = \frac{\lambda\,a}{2\,\mu_1\,\mu_2}\left(1 + \sqrt{1 - \frac{4\,\mu_1\,\mu_2}{a^2}}\right). \tag{7.23}$$

7.4 Raising the Blocking Threshold

It may seem that the network with blocking resembles the two-station tandem Jackson network more and more when the blocking threshold n increases. For instance, writing g_n for the left hand side of (7.9), the stability condition $\lim_{n\to\infty} g_n < 1$ is equivalent to the conditions $\rho_1 < 1$ when $\mu_1 < \mu_2$, and $\rho_2 < 1$ when $\mu_1 > \mu_2$. Thus we arrive at the condition $\rho < 1$, which is the stability criterion familiar from the two-station tandem Jackson network.

However, whereas the stability condition resembles more and more the stability criterion of the Jackson network for $n \to \infty$, the decay rate behaves *differently* than, perhaps, expected in the limit $n \to \infty$ for at least certain parameter regimes. In Section 7.4.1 below we prove that the sequence of decay rates x_n for increasing blocking threshold converges (from above) to ρ, rather than ρ_1. We also bound the rate of convergence of the sequence $\{x_n\}_n$ to its limit point.

A second topic of interest is to explore the probabilistic structure in the direction of the phases for some given level $i \gg 1$. In other words, we would like to find an approximation for the probability $\pi_{ij}^{(n)} = \mathbb{P}\{X_1^{(n)} = i, X_2^{(n)} = j\}$ as a function of j when i is large. (We again use the superscript n to label the variables of interest, since in this section the dependence on the blocking threshold n plays a central role. Let us denote by a superscript ∞ the random variables, etc., related to the Jackson tandem queue, as then the size of the second buffer is unlimited.) Now for the tandem Jackson network, being a product-form network, it is well-known that the ratio

$$\frac{\mathbb{P}\{X_1^{(\infty)} = i, X_2^{(\infty)} = j+1\}}{\mathbb{P}\{X_1^{(\infty)} = i, X_2^{(\infty)} = j\}} = \rho_2, \quad \text{for all } (i,j) \in \mathscr{X}^{(\infty)}. \tag{7.24}$$

As we shall see in Section 7.4.2, when n is large but finite this ratio need not always be close to ρ_2.

7.4.1 The Limiting Geometric Decay Rate

In this section we study the limiting behavior of the sequence of decay rates $\{x_n\}$ when the blocking threshold n increases to ∞. This result proves the claim obtained by the heuristic reasoning of the Introduction.

First, however, we have to settle an issue concerning the stability of the Markov chains $\{X_{1,k}^{(n)}, X_{2,k}^{(n)}\}$ when n is allowed to change. It follows from Lemma 7.1 that for given ρ_1 and ρ_2, the blocking threshold n should be larger than $N := N(\rho_1, \rho_2)$. Then Theorem 7.11 implies that for each $n > N$ the equation $\xi^{(n)}(x) = x$ has a solution x_n. Let us therefore combine these decay rates into the sequence $\{x_n\}_{n>N}$.

Theorem 7.14. *The sequence $\{x_n\}_{n>N}$ decreases monotonically to ρ and its elements satisfy the bounds*

$$0 < x_n - \rho < \begin{cases} \alpha_1^n \, \beta_1 \, \gamma_1, & \text{if } \rho = \rho_1, \\ \alpha_2^n \, \beta_2 \, \gamma_2, & \text{if } \rho = \rho_2, \end{cases} \tag{7.25}$$

where the constants

$$\alpha_1 := \max_{x \in [\rho_1, 1]} \left\{ \frac{\mu_1 \, x}{\eta_+(x)} \right\}, \qquad \alpha_2 := \max_{x \in [\rho_2, 1]} \left\{ \frac{\eta_-(x)}{\chi_1(x)} \right\},$$

$$\beta_1 := \max_{x \in [\rho_1, 1]} \{\mu_1 \, x - \eta_-(x)\}, \qquad \beta_2 := \max_{x \in [\rho_2, 1]} \{\eta_+(x) - \chi_1(x)\},$$

$$\gamma_1 := \left(\lambda + \mu_1 - \frac{\eta_-(x_1) - \eta_-(\rho_1)}{x_1 - \rho_1} \right)^{-1}, \quad \gamma_2 := \left(\frac{\eta_+(x_1) - \eta_+(\rho_2)}{x_1 - \rho_2} - \mu_1 \right)^{-1},$$

are positive, $\alpha_i < 1$, $i = 1, 2$, and x_1 is given by (7.23).

The maxima involved do not occur at the boundaries of the intervals but in the interiors, as is clear from Figure 7.3 for a concrete case. The form of the solutions obtained by taking the derivative with respect to x are cumbersome; we chose not to display these here.

Proof. We first show that $\{x_n\}$ is decreasing, that is, $x_n \notin [x_m, 1)$ whenever $n > m$. By (7.22) we see that $\chi_{j+1}(x) > \chi_j(x)$ for all $j \geq 0$ and $x \in (\rho, 1)$. Combining this with Corollary 7.12 for $x \in (x_m, 1)$ and noting that $\chi_{m+1}(x_m) = \mu_1 x_m$ we conclude that for $x \in [x_m, 1)$

$$\chi_{n+1}(x) > \chi_{m+1}(x) \geq \mu_1 x.$$

As no $x \in [x_m, 1]$ can solve the equation $\chi_{n+1}(x) = \mu_1 x$, it must be that $x_n < x_m$.

With regard to the convergence of $\{x_n\}$ to ρ, we consider first the case $\rho = \rho_2$. Let $\delta_n := x_n - \rho_2$, which is positive for all $n > N$. From Lemma 7.6,

$$\left(\frac{\eta_-(x_n)}{\chi_1(x_n)}\right)^n (\eta_+(x_n) - \chi_1(x_n)) > \eta_+(x_n) - \chi_{n+1}(x_n). \tag{7.26}$$

As $\eta_+(\cdot)$ is strictly concave on $(\rho_2, 1)$ and $x_n < x_1 < 1$ (for $n > 1$) we can bound $\eta_+(x_n)$ by

$$\eta_+(x_n) > \eta_+(\rho_2) + \frac{\eta_+(x_1) - \eta_+(\rho_2)}{x_1 - \rho_2}\,\delta_n. \tag{7.27}$$

Therefore, using $\eta_+(\rho_2) = \mu_1\rho_2$ and $\chi_{n+1}(x_n) = \mu_1\, x_n = \mu_1(\rho_2 + \delta_n)$, the right hand side of (7.26) satisfies,

$$\eta_+(x_n) - \chi_{n+1}(x_n) > \left(\frac{\eta_+(x_1) - \eta_+(\rho_2)}{x_1 - \rho_2} - \mu_1\right)\delta_n,$$

from which (7.25) follows.

For $\rho = \rho_1$, let $\delta_n = x_n - \rho_1 > 0$. Clearly, as $\eta_-(\cdot)$ is convex,

$$\eta_-(x_n) < \eta_-(\rho_1) + \frac{\eta_-(x_1) - \eta_-(\rho_1)}{x_1 - \rho_1}\delta_n \tag{7.28}$$

Therefore, by Lemma 7.6 and using that $\chi_1(x_n) = (\lambda + \mu_1)(\rho_1 + \delta_n) - \lambda$ and $\eta_-(\rho_1) = \lambda\,\rho_1$, we obtain

$$\begin{aligned}\left(\frac{\chi_{n+1}(x_n)}{\eta_+(x_n)}\right)^n (\chi_{n+1}(x_n) - \eta_-(x_n)) &> \chi_1(x_n) - \eta_-(x_n)\\ &> \left(\lambda + \mu_1 - \frac{\eta_-(x_1) - \eta_-(\rho_1)}{x_1 - \rho_1}\right)\delta_n.\end{aligned}$$

Moreover,

$$\frac{\chi_{n+1}(x_n)}{\eta_+(x_n)} = \frac{\mu_1\, x_n}{\eta_+(x_n)} \le \max_{x\in[\rho_1,1]}\left\{\frac{\mu_1\, x}{\eta_+(x)}\right\},$$

and, likewise,

$$\chi_{n+1}(x_n) - \eta_-(x_n) \le \max_{x\in[\rho_1,1]}\left\{\mu_1\, x - \eta_-(x)\right\}.$$

The positivity of the constants, except γ_1 and γ_2, as well as the fact that $\alpha_i < 1$, follows from Lemma 7.9. For γ_2, observe that

$$\frac{\eta_+(x_1) - \eta_+(\rho_2)}{x_1 - \rho_2} - \mu_1 = \frac{\eta_+(x_1) - \eta_+(\rho_2)}{x_1 - \rho_2} - \frac{\eta_+(1) - \eta_+(\rho_2)}{1 - \rho_2} > 0,$$

since η_+ is strictly concave and $\rho_2 < x_1 < 1$. Similar reasoning applies to γ_1. ∎

7.4.2 The 'Rate of Decay' in the Phase Direction

Up to now we have been concerned with the decay rate x_n in the direction of the levels and found that $x_n \downarrow \rho$ for $n \to \infty$. It is also of interest to analyze the probabilistic structure in the direction of the phases. The most convenient notion to consider in the present setting is

$$\lim_{i\to\infty} \frac{\pi^{(n)}_{i,j+1}}{\pi^{(n)}_{i,j}} = \frac{v^{(n)}_{j+1}}{v^{(n)}_{j}}, \tag{7.29}$$

see (7.7). This says that the ratio of the probability that the chain is in phase $j+1$ to the probability that the chain is in phase j, while the chain is in some high level i, is approximately equal to $v^{(n)}_{j+1}/v^{(n)}_{j}$. It follows from (7.14b) that this ratio is also proportional to the element $\chi_{j+1}(x_n)$ of the sequence $\{\chi_j(x_n)\}_{1\le j\le n}$.

To gain some insight into the effect of increasing the blocking threshold on the values of $\{\chi_j(x_n)\}_{1\le j\le n}$, we plot in Figure 7.4 the graphs of the sequences $\{\chi_j(x_5)\}_{1\le j\le 5}$, $\{\chi_j(x_{10})\}_{1\le j\le 10}$, and $\{\chi_j(x_{20})\}_{1\le j\le 20}$ for $\rho=\rho_2$ and $\rho=\rho_1$, respectively. (To obtain x_5, x_{10} and x_{20} we follow the procedure specified in Remark 7.13.) These graphs suggest that most of the elements of $\{\chi_j(x_n)\}_{1\le j\le n}$ are close to $\eta_-(x_n)$ or $\eta_+(x_n)$ when $\rho=\rho_1$ or $\rho=\rho_2$.

Theorem 7.15. *When the chain $\{X^{(n)}_{1,k}, X^{(n)}_{2,k}\}$ is stable, i.e., $n > N$,*

(i)

$$\lim_{i\to\infty} \frac{\pi^{(n)}_{i,j+1}}{\pi^{(n)}_{i,j}} = \frac{v^{(n)}_{j+1}}{v^{(n)}_{j}} = \frac{\chi_{j+1}(x_n)}{\mu_2\, x_n}. \tag{7.30a}$$

(ii) If, in addition, $\rho=\rho_1$,

$$\left|\frac{v^{(n)}_{j+1}}{v^{(n)}_{j}} - \frac{\lambda}{\mu_2}\right| < \frac{\alpha_1^{n-j}\,\beta_1}{\mu_2\,\rho_1} + \frac{\alpha_1^{n}\,\beta_1\,\gamma_1}{\mu_2\,\rho_1}(\gamma_1+\mu_1), \tag{7.30b}$$

or, if $\rho=\rho_2$,

$$\left|\frac{v^{(n)}_{j+1}}{v^{(n)}_{j}} - \frac{\mu_1}{\mu_2}\right| < \frac{\alpha_2^{j}\,\beta_2}{\lambda} + \frac{\alpha_2^{n}\,\beta_2\,\gamma_2}{\rho_2}\frac{1-\rho_2}{1-\rho_1}, \tag{7.30c}$$

where the constants $\alpha_i, \beta_i, \gamma_i$, are as defined in Theorem 7.14.

Proof. Statement (i) is immediate from (7.14b) and (7.29).

For (ii) we first prove the result for $\rho=\rho_2$. Observe that by the triangle inequality

and the inequality $\mu_2\, x_n > \lambda$,

$$\begin{aligned}\left|\frac{v_{j+1}^{(n)}}{v_j^{(n)}} - \frac{\mu_1}{\mu_2}\right| &= \left|\frac{\chi_{j+1}(x_n)}{\mu_2\, x_n} - \frac{\eta_+(\rho_2)}{\mu_2\, \rho_2}\right| \\ &= \frac{|\rho_2\, \chi_{j+1}(x_n) - x_n\, \eta_+(\rho_2)|}{\mu_2\, x_n\, \rho_2} \\ &< \frac{|\chi_{j+1}(x_n) - \eta_+(x_n)|}{\lambda} + \frac{|\rho_2\, \eta_+(x_n) - x_n\, \eta_+(\rho_2)|}{\lambda\, \rho_2}.\end{aligned} \tag{7.31}$$

Clearly, by Lemma 7.6

$$0 < \eta_+(x_n) - \chi_{j+1}(x_n) < \left(\frac{\eta_-(x_n)}{\chi_1(x_n)}\right)^j (\eta_+(x_n) - \chi_1(x_n))\,.$$

For the second term, we observe that as η_+ is concave, $\eta_+(x_n) > x_n\, \mu_1$ from which $\rho_2\, \eta_+(x_n) > x_n\, \mu_1\, \rho_2 = x_n\, \eta_+(\rho_2)$, and

$$\eta_+(x_n) < \eta_+(\rho_2) + \delta_n\, \eta_+'(\rho_2) = \eta_+(\rho_2) + \delta_n\, \frac{\mu_1 + \mu_2 - 2\lambda}{1 - \rho_1}. \tag{7.32}$$

Hence, after some calculations,

$$0 < \rho_2\, \eta_+(x_n) - x_n\, \eta_+(\rho_2) < \big(\rho_2\, \eta_+'(\rho_2) - \eta_+(\rho_2)\big)\, \delta_n = \lambda \frac{1 - \rho_2}{1 - \rho_1}\, \delta_n.$$

The rest follows immediately from Theorem 7.14.

When $\rho = \rho_1$, so that $x_n > \rho_1$, consider

$$\begin{aligned}\left|\frac{v_{j+1}^{(n)}}{v_j^{(n)}} - \frac{\lambda}{\mu_2}\right| &= \left|\frac{\chi_{j+1}(x_n)}{\mu_2\, x_n} - \frac{\eta_-(\rho_1)}{\mu_2 \rho_1}\right| \\ &< \frac{|\chi_{j+1}(x_n) - \eta_-(x_n)|}{\mu_2\, \rho_1} + \frac{|\rho_1\, \eta_-(x_n) - x_n\, \eta_-(\rho_1)|}{\mu_2\, \rho_1^2}.\end{aligned}$$

Now we use the estimates

$$\chi_{j+1}(x_n) - \eta_-(x_n) < \left(\frac{\chi_{n+1}(x_n)}{\eta_+(x_n)}\right)^{n-j} (\chi_{n+1}(x_n) - \eta_-(x_n))$$

and (7.28) for the final result. ∎

Remark 7.16. Observe that when server 1 is the bottleneck, $\pi_{i,j+1}^{(n)}/\pi_{i,j}^{(n)} \approx \rho_2$, which is natural in view of the Jackson tandem queue. In the other case, $\pi_{i,j+1}^{(n)}/\pi_{i,j}^{(n)} \approx \mu_1/\mu_2$. This number is larger than one, which is to be expected as the second buffer is mostly full in this regime leading to blocking of the first server.

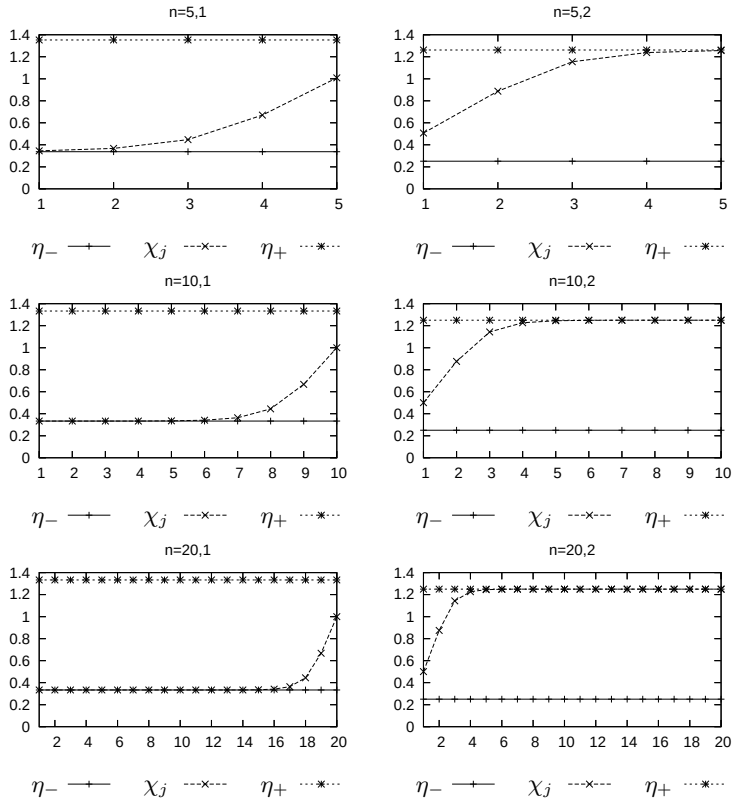

Figure 7.4: *Graphs of the sequence* $\{\chi_j(x_n)\}_{1\le j\le n}$ *for* $n = 5$, 10, *and* 20. *At the left* $\rho = \rho_1$ *(*$\lambda = 1$, $\mu_1 = 3$ *and* $\mu_2 = 4$*), whereas at the right* $\rho = \rho_2$ *(*$\lambda = 1$, $\mu_1 = 5$ *and* $\mu_2 = 4$*). The phase* j *increases along the* x*-axis; the value of* $\chi_j(x_n)$ *is set out along the* y*-axis. For clarity we connect subsequent terms* $\chi_j(x_n)$ *by lines.*

7.5 The Tandem Queue with Slow-down and Blocking

Consider now a network in which the second server signals the first to slow down, i.e., to work at rate $\tilde{\mu}_1 < \mu_1$ instead of at rate μ_1, when the second station contains m or more jobs, where, of course, $m < n$. Figure 7.5 shows the state transition diagram of the resulting queueing process.

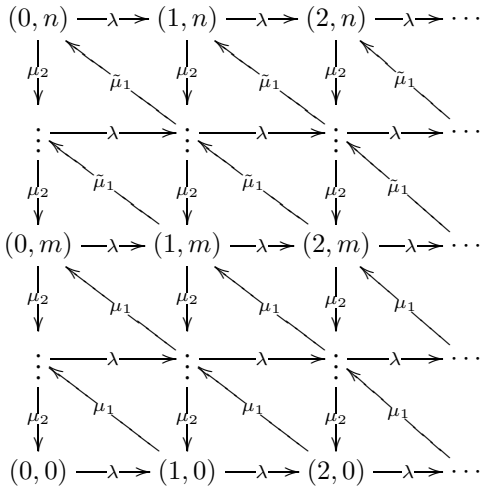

Figure 7.5: *State space and transition rates of the two-station tandem queue with slow-down and blocking. Note that above phase m server 1 works at rate $\tilde{\mu}_1$ rather than at μ_1.*

In this section we assume the following ordering of parameters:

$$\lambda < \mu_2 < \tilde{\mu}_1 < \mu_1, \text{ or, equivalently, } \rho_1 < \tilde{\rho}_1 < \rho_2 < 1, \tag{7.33}$$

where $\tilde{\rho}_1 := \lambda/\mu_1$. Observe that as a consequence, $\rho = \rho_2$ in this section. Henceforth we do no longer use ρ, but always ρ_2. With this ordering we generalize Theorem 7.11 the present case and restate Theorems 7.14 and 7.15 in somewhat weaker form. The methods of proof are similar to those of Sections 7.3 and 7.4. Due to these similarities we only show the main steps to arrive at the results stated here. (The details may sometimes be slightly more involved algebraically, but are seldom more complicated conceptually.)

Remark 7.17. It would, of course, be interesting to consider other orderings of the system parameters such as, for instance, $0 < \tilde{\mu}_1 < \lambda < \mu_2 < \mu_1$. However, Lemma 7.19 below does not immediately carry over to these cases as its proof depends crucially on the ordering (7.33). We conjecture, based on numerical experiments, that similar results

can be obtained for all cases. Thus, 'case checking', i.e., proving each step of the line of reasoning below for every possible ordering of parameters (provided the chain is stable), seems a possible method to obtain stronger results. However, this approach is, admittedly, not elegant, neither might it reveal much of the structure of the problem. It remains for further research to find the general underlying principle; here we concentrate on the ordering specified in (7.33).

Since $\tilde{\mu}_1 < \mu_1$ we can again uniformize the related continuous-time Markov chain $\{X_1^{(n,m)}(t), X_2^{(n,m)}(t)\}$ at rate $a = \lambda + \mu_1 + \mu_2$ to obtain an aperiodic discrete-time QBD chain $\{X_{1,k}^{(n,m)}, X_{2,k}^{(n,m)}\}$. The matrix of transition probabilities $P^{(n,m)}$ has the same form as $P^{(n)}$ in (7.2), but whereas $B^{(n,m)} = B^{(n)}$ and $A_0^{(n,m)} = A_0^{(n)}$, the other matrices become, with $\tilde{q} = \tilde{\mu}_1/a$,

$$A_1^{(n,m)} = \begin{pmatrix} r & & & & & & & & \\ r & 0 & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & r & 0 & & & & & \\ & & & r & q-\tilde{q} & & & & \\ & & & & \ddots & \ddots & & & \\ & & & & & r & q-\tilde{q} & & \\ & & & & & & r & q \end{pmatrix}, \tag{7.34}$$

and

$$A_2^{(n,m)} = \begin{pmatrix} 0 & q & & & & & & \\ & \ddots & \ddots & & & & & \\ & & 0 & q & & & & \\ & & & 0 & \tilde{q} & & & \\ & & & & \ddots & \ddots & & \\ & & & & & 0 & \tilde{q} & \\ & & & & & & 0 & \end{pmatrix}, \tag{7.35}$$

where at the m-th row the changes occur. Finally, let $A^{(n,m)}(x) := A_0^{(n,m)} + xA_1^{(n,m)} + x^2A_2^{(n,m)}$.

Concerning the stability of the chain we follow the approach of Lemma 7.1 to derive a necessary and sufficient stability condition. In accordance with our expectations for a system with the ordering (7.33), this condition reduces to $\lambda < \mu_2$ when $n \to \infty$.

Lemma 7.18. *Let $\beta = \mu_1/\mu_2$ and $\tilde{\beta} = \tilde{\mu}_1/\mu_2$. The two-station tandem network with*

slow-down threshold m and blocking at $n \geq m$ is positive recurrent if and only if

$$\lambda < \frac{\mu_1 (1-\beta^m)(1-\tilde{\beta}) + \tilde{\mu}_1 \beta^m (1-\beta)(1-\tilde{\beta}^{n-m})}{(1-\beta^m)(1-\tilde{\beta}) + \beta^m (1-\beta)(1-\tilde{\beta}^{n-m+1})}. \tag{7.36}$$

Proof. Following the proof of Lemma 7.1, the normalized solution of $\boldsymbol{\alpha} A^{(n,m)}(1) = \boldsymbol{\alpha}$ has the form,

$$\alpha_i = \begin{cases} \alpha_0 \beta^i, & \text{if } i \leq m-1, \\ \alpha_0 \beta^m \tilde{\beta}^{i-m}, & \text{if } m \leq i \leq n, \end{cases}$$

and

$$\alpha_0^{-1} = \frac{1-\beta^m}{1-\beta} + \beta^m \frac{1-\tilde{\beta}^{n+1-m}}{1-\tilde{\beta}}.$$

The inequality $\boldsymbol{\alpha} A_0^{(n,m)} \mathbf{e} < \boldsymbol{\alpha} A_2^{(n,m)} \mathbf{e}$ becomes

$$\begin{aligned} \lambda &< \alpha_0 \left(\mu_1 \sum_{i=0}^{m-1} \beta^i + \tilde{\mu}_1 \beta^m \sum_{i=m}^{n-1} \tilde{\beta}^{i-m} \right) \\ &= \alpha_0 \left(\mu_1 \frac{1-\beta^m}{1-\beta} + \tilde{\mu}_1 \beta^m \frac{1-\tilde{\beta}^{n-m}}{1-\tilde{\beta}} \right). \end{aligned}$$

∎

The next step is to rewrite the equation

$$\mathbf{v}^{(n,m)} A^{(n,m)}(x) = \mathbf{v}^{(n,m)} x, \tag{7.37}$$

and derive a sequence $\{\chi_j(x)\}_{1 \leq j \leq n}$ in terms of mappings similar to T defined in (7.16). With this aim, let $\chi_j(x) = \mu_2 \, x \, v_j / v_{j-1}$ as in (7.14b). However, contrary to (7.15) we now need *three*, rather than one, mappings to cast (7.37) into a sequence $\{\chi_j(x)\}_{1 \leq j \leq n}$:

$$\begin{aligned} T &: \eta \mapsto ax - \lambda - \frac{\mu_1 \mu_2 x^3}{\eta}, \quad \text{(as in (7.16))}, \\ S &: \eta \mapsto \tilde{a}x - \lambda - \frac{\mu_1 \mu_2 x^3}{\eta}, \\ \tilde{T} &: \eta \mapsto \tilde{a}x - \lambda - \frac{\tilde{\mu}_1 \mu_2 x^3}{\eta}, \end{aligned} \tag{7.38}$$

where $\tilde{a} = \lambda + \tilde{\mu}_1 + \mu_2$. With these mappings, (7.37) implies that

$$\chi_j(x) := \begin{cases} \mu_1 x^2, & \text{if } j = 0, \\ T\left(\chi_{j-1}(x)\right), & \text{if } 1 \leq j \leq m, \\ S\left(\chi_m(x)\right), & \text{if } j = m+1, \\ \tilde{T}\left(\chi_{j-1}(x)\right), & \text{if } m+2 \leq j \leq n+1. \end{cases}$$

Loosely speaking, S moves $\chi_m(x)$ across the slow-down threshold at m to the iterate $\chi_{m+1}(x)$ on which $\tilde{T}$ can start operating. The condition on x of the last row of the equation $\mathbf{v}^{(n,m)}A^{(n,m)}(x) = \mathbf{v}^{(n,m)}x$ is,

$$\chi_{n+1}(x) = \tilde{\mu}_1 x, \tag{7.39}$$

rather than $\chi_{n+1}(x) = \mu_1 x$ as in (7.15c).

Theorem 7.4 carries over immediately. Thus, if we can find $x \in (0,1)$ such that each element of the sequence $\{\chi_j(x)\}_{0\le j\le n+1}$ is positive and $\chi_{n+1}(x) = \tilde{\mu}_1 x$, then x is the decay rate we are searching for.

To establish that the elements of $\{\chi_j(x)\}_{0\le j\le n+1}$ are positive we would like to apply Lemma 7.9. Supposing that $\chi_0(x) \in (\eta_-(x), \eta_+(x))$, it follows that the elements of $\{\chi_j(x)\}_{0\le j\le m}$ all lie in the interval $(\eta_-(x), \eta_+(x))$, hence are positive. However, it is not immediately obvious that $S(\chi_m(x))$ lies somewhere in between the fixed points $\tilde{\eta}_-(x)$ and $\tilde{\eta}_+(x)$ (regarded as functions of x) of $\tilde{T}$. Now realize that $\chi_0(x) < \chi_m(x) < \eta_+(x)$, and therefore by (7.38), that $S(\chi_0(x)) < S(\chi_m(x)) < S(\eta_+(x))$. Below we prove that $\tilde{\eta}_-(x) < S(\chi_0(x))$ and $S(\eta_+(x)) \le \tilde{\eta}_+(x)$ so that S maps any element in $(\chi_0(x), \eta_+(x))$, and *in particular* $\chi_m(x)$, into the interval $(\tilde{\eta}_-(x), \tilde{\eta}_+(x))$. Therefore, Lemma 7.9, which applies to equally well to $\tilde{T}$ due to the ordering (7.33), ensures that also the elements of $\{\chi_j(x)\}_{m+2\le j\le n+1}$ lie within the interval $(\tilde{\eta}_-(x), \tilde{\eta}_+(x))$. Finally, due to the ordering (7.33) Lemma 7.8 implies that $\tilde{\eta}_-(x) > 0$ for $x \in [\rho_2, 1]$, thereby guaranteeing the positivity of all elements of the sequence $\{\chi_j(x)\}_{0\le j\le n+1}$ for $x \in [\rho_2, 1]$.

Lemma 7.19. *For all $x \in (\rho_2, 1)$:*

$$\tilde{\eta}_-(x) < S(\chi_0(x)) \quad \textit{and} \quad S(\eta_+(x)) \le \tilde{\eta}_+(x). \tag{7.40}$$

Proof. Let us start with proving the first inequality. As $\lambda < \mu_2 < \tilde{\mu}_1$ it follows from Lemma 7.8 that $\tilde{\eta}_-(x) < \tilde{\mu}_1 x^2$. Hence, $\mu_1 \tilde{\eta}_-(x)/\tilde{\mu}_1 < \mu_1 x^2 = \chi_0(x)$. Applying S to both sides and noting that $S(\mu_1 \tilde{\eta}_-(x)/\tilde{\mu}_1) = \tilde{T}(\tilde{\eta}_-(x)) = \tilde{\eta}_-(x)$ gives the result.

Concerning the second inequality in (7.40) observe that this is equivalent to

$$\eta_+(x) + (\tilde{\mu}_1 - \mu_1)x = S(\eta_+(x)) \le \tilde{\eta}_+(x). \tag{7.41}$$

Clearly, in case $\tilde{\mu}_1 = \mu_1$, the left hand side and the right hand side are equal. Next, if the derivative with respect to $\tilde{\mu}_1$ of the left hand side of (7.41) is larger than the derivative of the right hand side then, as $\tilde{\mu}_1 < \mu_1$, the inequality must hold.

Thus, we like to show that when $x \in (\rho_2, 1)$,

$$x > \frac{\partial \tilde{\eta}_+(x)}{\partial \tilde{\mu}_1} = \frac{x}{2} + \frac{1}{2}\frac{(\tilde{a}x - \lambda)x - 2\mu_2 x^3}{\sqrt{(\tilde{a}x - \lambda)^2 - 4\tilde{\mu}_1\mu_2 x^3}}.$$

Rewrite this to

$$\sqrt{(\tilde{a}x-\lambda)^2-4\tilde{\mu}_1\mu_2x^3} > \tilde{a}x-\lambda-2\mu_2x^2.$$

This inequality is implied by

$$(\tilde{a}x-\lambda)^2-4\tilde{\mu}_1\mu_2x^3 > (\tilde{a}x-\lambda)^2-4\mu_2x^2(\tilde{a}x-\lambda)+4\mu_2^2\,x^4,$$

which in turn reduces to

$$\lambda(x-1) > \mu_2x(x-1).$$

This is true since $x \in (\rho_2, 1)$. ∎

As counterpart of Theorems 7.11 and 7.14 we obtain the following.

Theorem 7.20. *If $\rho_1 < \tilde{\rho}_1 < \rho_2 < 1$ and the blocking threshold n and slow-down threshold $m \leq n$ are such that the chain $\{X_{1,k}^{(n,m)}, X_{2,k}^{(n,m)}\}$ is stable, the sequence $\{x_{n,m}\}_n$ decreases monotonically to ρ_2 for m fixed.*

Proof. The positivity of the elements of $\{\chi_j(x_{n,m}\}_{1\leq j\leq n+1}$ is settled by the discussion leading to Lemma 7.19.

To prove that there exists a unique $x \in (\rho_2, 1)$ such that $\chi_{n+1}(x) = \tilde{\mu}_1 x$, we reason an in the proof of Theorem 7.11. Observe that: (i) $\chi_0(\rho_2) < \eta_+(\rho_2) \Rightarrow \chi_j(\rho_2) < \tilde{\eta}_+(\rho_2) = \tilde{\mu}_1\rho_2$ for all $j > m$; (ii) $\chi_{n+1}(1) = \tilde{\mu}_1$; (iii) Condition (7.36) is equivalent to $\chi'_{n+1}(1) < \tilde{\mu}_1$; (iv) $\chi''_{n+1}(x) < 0$, i.e., $\chi_{n+1}(x)$ is strictly concave, for $x \in (\rho_2, 1)$.

By similar reasoning as in the first part of Theorem 7.14 it can be seen that $\{x_{n,m}\}$ decreases monotonically. Finally, pertaining to the convergence to ρ_2, the sequence $\{x_{n,m}\}$, being bounded and decreasing, has a unique limit point ζ in $\mathbb{R}$. Suppose that $\zeta > \rho_2$. Then, since, $\tilde{\eta}_+(\zeta) > \tilde{\mu}_1\zeta$ and $\lim_{j\to\infty}\chi_j(x) = \tilde{\eta}_+(x)$ for all $x \in (\rho_2, 1)$, there exists $M > 0$ such that for all $j > M$, $\chi_j(\zeta) > \tilde{\mu}_1\zeta$. On the other hand, we derived above that $\chi_j(\rho_2) < \tilde{\mu}_1\rho_2$ for $j > m$. The continuity of $\chi_j(x)$ implies that there exists $x_{j-1} \in (\rho_2, \zeta)$ such that $\chi_j(x_{j-1}) = \tilde{\mu}_1\, x_{j-1}$. This contradicts $\zeta > \rho_2$. ∎

It proves difficult to bound the rate of convergence of the sequence of decay rates $\{x_{n,m}\}$, which thereby prevents us from generalizing (7.25) to the present case. As a result, we also cannot carry over Theorem 7.15. However, we can achieve the following slightly weaker result in which we appropriately scale the slow-down threshold m as a function of the blocking threshold n.

Theorem 7.21. *Let the slow-down threshold m scale as $m(n) = \alpha n$ for a fixed $\alpha \in (0,1)$ and write $\pi^{(n,m)}(i,j)$ for $\pi_{i,j}^{(n,m)}$. Then,*

$$\lim_{n\to\infty}\lim_{i\to\infty}\frac{\pi^{(n,m)}(i,\lfloor yn\rfloor)}{\pi^{(n,m)}(i,\lfloor yn\rfloor-1)} = \begin{cases} \frac{\eta_+(\rho_2)}{\mu_2\rho_2} = \frac{\mu_1}{\mu_2}, & \text{if } y \in (0,\alpha], \\ \\ \frac{\tilde{\eta}_+(\rho_2)}{\mu_2\rho_2} = \frac{\tilde{\mu}_1}{\mu_2}, & \text{if } y \in (\alpha,1), \end{cases} \tag{7.42}$$

where $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x.

In Theorem 7.15 we could bound this ratio for any *fixed* phase j, $j \leq n$, for $n \to \infty$. Here we scale the phase $j(n)$ as a function of n. In fact, the proof below makes clear that we establish the point-wise limit of the functions $\chi_{j(n)}(x_{n,m})/\mu_2\, x_{n,m}$ for $n \to \infty$ rather than for j fixed.

Proof. Recall

$$\lim_{i\to\infty} \frac{\pi^{(n,m)}(i, \lfloor yn \rfloor)}{\pi^{(n,m)}(i, \lfloor yn \rfloor - 1)} = \frac{v^{(n,m)}(\lfloor yn \rfloor)}{v^{(n,m)}(\lfloor yn \rfloor - 1)} = \frac{\chi_{\lfloor yn \rfloor}(x_{n,m})}{\mu_2\, x_{n,m}},$$

and concentrate on the right hand side.

First, let $y \in (0, \alpha]$. Clearly, it follows from Theorem 7.20 that $x_{n,m} \to \rho_2$ for $n \to \infty$, and therefore, by applying Theorem 7.15, $\chi_{\lfloor yn \rfloor}(x_{n,m}) \to \eta_+(\rho_2)$. In particular, $\chi_{\lfloor \alpha n \rfloor}(x_{n,m}) \to \eta_+(\rho_2)$ so that, by (7.38),

$$\lim_{n\to\infty} S\left(\chi_{\lfloor \alpha n \rfloor}(x_{n,m})\right) = \tilde{a}\rho_2 - \lambda - \frac{\mu_1\mu_2\rho_2^3}{\eta_+(\rho_2)} = \tilde{\mu}_1\, \rho_2 = \tilde{\eta}_+(\rho_2).$$

Now let $y \in (\alpha, 1)$. As $S\left(\chi_{\lfloor \alpha n \rfloor}(x_{n,m})\right) < \chi_{\lfloor yn \rfloor}(x_{n,m}) < \tilde{\eta}_+(x_{n,m})$, and the left and right hand side converge to $\tilde{\eta}_+(\rho_2)$ for $n \to \infty$, the functions $\chi_{\lfloor yn \rfloor}(x_{n,m})$ have the same limit. ∎

In terms of the Perron-Frobenius vector $\mathbf{v}^{(n,m)}$ of $R^{(n,m)}$ this results means the following, cf. (7.30a),

$$\frac{v_j^{(n,m)}}{v_{j-1}^{(n,m)}} \approx \begin{cases} \mu_1/\mu_2 & \text{if } j < m(n) \\ \tilde{\mu}_1/\mu_2 & \text{if } j \geq m(n). \end{cases}$$

Thus, a 'kink' appears in the graph of ratio of the consecutive components of $\mathbf{v}^{(n,m)}$.

Remark 7.22. The approach to obtain the geometric decay rate and the 'decay rate in the direction of the phases' generalizes of course to any number of slow-down thresholds when the adapted rates $\tilde{\mu}_1, \tilde{\tilde{\mu}}_1, \ldots$, form a decreasing sequence bounded below by μ_2.

Bibliography

ABATE, J., G.L. CHOUDHURY, & W. WHITT. 1999. An introduction to numerical transform inversion and its application to probability models. In *Comp. Prob.*, ed. by W. Grassman, chapter 1, pages 257–828. Kluwer.

ADAN, I.J.B.F., E.A. VAN DOORN, J.A.C. RESING, & W.R.W. SCHEINHARDT. 1998. Analysis of a single-server queue interacting with a fluid reservoir. *Queueing Systems* 29:313–336.

——, & J.A.C. RESING. 2000. A two-level traffic shaper for an on-off source. *Performance Evaluation* 42:279–298.

AJMONE MARSAN, M., G. BALBO, G. CONTE, S. DONATELLI, & G. FRANCESCHINIS. 1995. *Modelling with Generalized Stochastic Petri Nets*. John Wiley & Sons.

——, C. CASETTI, R. GAETA, & M. MEO. 2000. Performance analysis of TCP connections sharing a congested internet link. *Performance Evaluation* 42(2-3):109–127.

AKAR, N., & K. SOHRABY. 2003. Algorithmic solution of finite Markov fluid queues. In *Proc. of ITC 18*, pages 621–630.

ALESSIO, E., M. GARETTO, R. LO CIGNO, M. MEO, & M. AJMONE MARSAN. 2001. Analytical estimation of the completion times of mixed NewReno and Tahoe TCP connections over single and multiple bottlenecks. In *Proc. of Globecom*.

ALLMAN, M., V. PAXSON, & W. STEVENS. 1998. RFC 2581: TCP congestion control. Technical report, IETF.

ALTMAN, E., K. AVRATCHENKOV, & C. BARAKAT. 2000a. A stochastic model of TCP/IP with stationary ergodic random losses. *ACM SIGCOMM Computer Communication Review* 30(4):231–242.

——, ——, & ——. 2002a. TCP network calculus: The case of large delay-bandwidth product. In *Proc. of IEEE INFOCOM*.

——, C. BARAKAT, E. LABORDE, P. BROWN, & D. COLLANGE. 2000b. Fairness analysis of TCP/IP. In *Proc. of IEEE Conference on Decision and Control*.

——, T. JIMENEZ, & R. NÚÑEZ-QUEIJA. 2002b. Analysis of two competing TCP/IP connections. *Performance Evaluation* 49:43–55.

ANICK, D., D. MITRA, & M.M. SONDHI. 1982. Stochastic theory of a data-handling system with multiple sources. *Bell Sys. Tech. J.* 61(8):1871–1894.

ASMUSSEN, S. 2003. *Applied Probability and Queues*. Wiley, 2nd edition.

AVRATCHENKOV, K., U. AYESTA, E. ALTMAN, P. NAIN, & C. BARAKAT. 2002. The effect of router buffer size on the TCP performance. In *LONIIS workshop on Telecommunication Networks and Teletraffic Theory*.

BACCELLI, F., & D. HONG. 2003a. Flow level simulation of large IP networks. In *Proc. of IEEE INFOCOM*.

——, & ——. 2003b. Interaction of TCP flows as billiards. In *Proc. of IEEE INFOCOM*.

BEKKER, R. 2004. Finite buffer queues with workload-dependent service and arrival rates. Technical Report SPOR-Report 2004-01, Einhoven University of Technology.

——, S.C. BORST, O.J. BOXMA, & O. KELLA. 2004. Queues with workload-dependent arrival and service rates. *Queueing Systems* 46(3/4):537–556.

BEN FREDJ, S., T. BONALD, A. PROUTIÈRE, G. RÉGNIÉ, & J.W. ROBERTS. 2001. Statistical bandwidth sharing: A study of congestion at flow level. *ACM SIGCOMM Computer Communications Review* 31(4):111–122.

BOXMA, O., H. KASPI, O. KELLA, & D. PERRY. 2005. On/off storage systems with state dependent input, output and switching rates. *Probability in the Engineering and Informational Sciences* 19(1):1–14.

BRADEN, R. 1989. RFC 1122: requirements for internet hosts – communication layers. Technical report, IETF.

BROWN, P. 2000. Resource sharing of TCP connections with different roundtrip times. In *Proc. of IEEE INFOCOM*, pages 1734–1741.

BU, T., & D. TOWSLEY. 2001. Fixed point approximation for TCP behavior in an AQM network. *ACM SIGMETRICS Performance Evaluation Review* 29(1):216–225.

CARDWELL, N., S. SAVAGE, & T. ANDERSON. 2000. Modeling TCP latency. In *Proc. of IEEE INFOCOM*, pages 1742–1751.

CASETTI, C., & M. MEO. 2000. A new approach to model the stationary behavior of TCP connections. In *Proc. of IEEE INFOCOM*, pages 367–375.

——, & ——. 2001a. An analytical framework for the performance evaluation of TCP Reno connections. *Computer Networks* 37(5):669–682.

——, & ——. 2001b. Modeling the stationary behavior of TCP Reno connections. In *QOS-IP*, pages 141–156.

CHIU, D.H., & R. JAIN. 1989. Analysis of the increase and decrease algorithms of congestion avoidance in computer networks. *Computer Networks and ISDN Systems* 17:1–14.

CIARDO, G., G. M. FRICKS, J.K. MUPPALA, & K.S. TRIVEDI, 1994. *SPNP Users Manual*, 4th edition.

——, G. MUPPALLA, & K. TRIVEDI. 1989. SPNP: Stochastic Petri Net Package. In *3rd Int. Workshop on Petri Nets and Performance Models (PNPM'89)*, pages 142–151. IEEE Comp. Soc. Press.

ÇINLAR, E., & M. PINSKY. 1971. A stochastic integral in storage theory. *Z. Wahr. verw. Geb.* 17:227–240.

COHEN, J.W. 1974. Superimposed renewal processes and storage with gradual input. *Stochastic Processes Appl.* 2:31–58.

—— 1979. The multiple phase service network with generalized processor sharing. *Acta Informatica* 12:245–284.

COOPER, R.B., & D.P. HEYMAN. 1998. *Encyclopedia of Telecommunications*, volume 16, chapter Teletraffic Theory and Engineering, pages 453–483. Marcel Dekker, Inc.

DAVIS, M.H.A. 1984. Piecewise-deterministic Markov processes: a general class of non-diffusion stochastic models. *J. Royal Statist. Soc. (B)* 46:353–388.

—— 1993. *Markov Models and Optimization.* Chapman and Hall.

DOORN, E.A. VAN, A.A. JAGERS, & J.S.J. DE WIT. 1988. A fluid reservoir regulated by a birth-death process. *Stochastic Models* 4(3):457–472.

ELWALID, A.I., & D. MITRA. 1992. Fluid models for the analysis and design of statistical multiplexing with loss priorities on mutiple classes of bursty traffic. In *Proc. of IEEE INFOCOM*, pages 415–425.

——, & ——. 1994. Statistical multiplexing with loss priorities in rate-based congestion control of high-speed networks. *IEEE Transactions on Communications* 42(11):2989–3002.

FALL, K., & S. FLOYD. 1996. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. *ACM SIGCOMM Computer Communication Review* 26(3):5–21.

FAYOLLE, G., I. MITRANI, & R. IASNOGORODSKI. 1980. Sharing a processor among many jobs. *Journal of the ACM* 27(3):519–532.

FELLER, W. 1968. *An Introduction to Probability Theory and Its Applications*, volume 1. John Wiley & Sons.

FEYNMAN, R.P. 1970. *Feynman Lectures On Physics*. Addison-Wesley.

FLOYD, S. 1991. Connections with multiple congested gateways in packet-switched networks Part1: One-way traffic. Technical report, Lawrence Berkeley Laboratory.

——, & T. HENDERSON. 1999. RFC 2582: The NewReno modification to TCP's Fast Recovery algorithm. Technical report, IETF.

——, & V. JACOBSON. 1993. Random Early Detection gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking* 1(4):397–413.

——, & E. KOHLER. 2002. Internet research needs better models. *ACM SIGCOMM Computer Communication Review* 33(1):29 –34.

FOREEST, N.D. VAN, B.R. HAVERKORT, M.R.H. MANDJES, & W.R.W. SCHEINHARDT. 2004a. Versatile Markovian models for networks with asymmetric TCP sources. *Submitted.* .

——, M.R.H. MANDJES, & W.R.W. SCHEINHARDT. 2001. Performance analysis of heterogeneous interacting TCP sources. Memorandum 1607, Faculty of Mathematical Sciences, University of Twente, Enschede, The Netherlands.

——, ——, & ——. 2003a. Analysis of a feedback fluid model for heterogeneous TCP sources. *Stochastic Models* 19(3):299–324.

——, ——, & ——. 2003b. A versatile model for asymmetric TCP sources. In *Proc. of ITC 18*, pages 631–640.

——, M.R.H. MANDJES, J.C.W. VAN OMMEREN, & W.R.W. SCHEINHARDT. 2004b. A tandem network with server slow-down and blocking. *Submitted* .

GARETTO, M., R. LO CIGNO, M. MEO, E. ALESSIO, & M. AJMONE MARSAN. 2002. Modeling short-lived TCP connections with open multiclass queuing networks. In *7th International Workshop on Protocols For High-Speed Networks (PfHSN)*, pages 100–116.

——, R. LO CIGNO, M. MEO, & M. AJMONE MARSAN. 2001a. A detailed and accurate closed queueing network model of many interacting TCP flows. In *Proc. of IEEE INFOCOM*, pages 1706–1715.

——, ——, ——, & ——. 2001b. Queuing network models for the performance analysis of multibottleneck IP networks loaded by short-lived TCP connections. Technical Report DE/RLC/2001-5, Politecnico di Torino.

GIBBENS, R.J., & F.P. KELLY. 1999. Resource pricing and the evolution of congestion control. *Automatica* 35:1969–1985.

——, S.K. SARGOOD, C. VAN EIJL, F.P. KELLY, H. AZMOODEH, R.N. MACFADYEN, & N.W. MACFADYEN. 2000. Fixed-point models for the end-to-end performance analysis of IP networks. In *ITC Specialist Seminar: IP Traffic Measurement, Modeling and Management*, volume 13.

GOLUB, G. H., & C. VAN LOAN. 1989. *Matrix Computations*. The John Hopkins University Press, 2nd edition.

GRASSMAN, W. K., & S. DREKIC. 2000. An analytical solution for a tandem queue with blocking. *Queueing Systems* 36:221–235.

GRIMMETT, G., & D. STIRZAKER. 2001. *Probability and Random Processes*. Oxford University Press, 3rd edition.

HAAN, R. DE, J.L. VAN DEN BERG, R.E. KOOIJ, R.D. VAN DER MEI, & A.P. ZWART. 2004. Enhancement of an integrated packet/flow model for TCP performance. Preprint.

HARRISON, J. M., & S. I. RESNICK. 1976. The stationary distribution and first exit probabilities of a storage process with general release rule. *Mathematics of Operations Research* 1(4):347–358.

HARRISON, P. G., & N. M. PATEL. 1993. *Performance Modelling of Communication Networks and Computer Architectures*. Addison-Wesley.

HORN, R. A., & C. A. JOHNSON. 1985. *Matrix Analysis*. Cambridge University Press.

JACOBSON, V. 1988. Congestion Avoidance and Control. In *ACM SIGCOMM Computer Communication Review*, volume 25, pages 314–329.

KEILSON, J. 1965. *Green's Function Methods in Probability Theory*. Charles Griffin & Company Ltd.

KELLA, O., & W. STADJE. 2002. Exact results for a fluid model with state-dependent flow rates. *Probability in the Engineering and Informational Sciences* 16(4):389–402.

KELLY, F.P. 1997. Charging and rate control for elastic traffic. *European Transactions on Telecommunications* 8:33–37.

—— 2000. Models for a self-managed Internet. *Philosophical Transactions of the Royal Society* A358:2335–2348.

—— 2001. Mathematical modelling of the Internet. In *Mathematics Unlimited - 2001 and Beyond*, ed. by B. Engquist & W. Schmid. Springer-Verlag.

——, A.K. MAULLOO, & D.K.H. TAN. 1998. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* 49:237–252.

KLEINROCK, L. 1975. *Queueing Systems*, volume I. John Wiley & Sons.

—— 1976. *Queueing Systems*, volume II. John Wiley & Sons.

KONHEIM, A.G., & M. REISER. 1976. A queuing model with finite waiting room and blocking. *Journal of the ACM* 23(2):328–341.

——, & ——. 1978. Finite capacity queuing systems with applications in computer modeling. *SIAM Journal of Computing* 7(2):210–229.

KOSTEN, L. 1974. Stochastic theory of a multi-entry buffer, part 1. *Delft Progress Report, Series F* 1:10–18.

—— 1984. Stochastic theory of data handling systems, with groups of multiple sources. In *Performance of Computer-communication Systems*, ed. by H. Rudin & W. Bux, pages 321–331. Elsevier Science Publishers B.V.

KROESE, D.P., & W.R.W. SCHEINHARDT. 2001. Joint distributions for interacting fluid queues. *Queueing Systems* 37:99–139.

——, ——, & P.G. TAYLOR. 2004. Spectral properties of the tandem Jackson network, seen as a quasi-birth-and-death process. *To appear in Annals of Applied Probability* .

KULKARNI, V. G. 1997. Fluid models for single buffer systems. In *Frontiers in Queueing. Models and Applications in Science and Engineering*, ed. by J.H. Dshalalow, pages 321–338. CRC Press.

KUROSE, J.F., & K.W. ROSS. 2003. *Computer Networking*. Addison Wesley, 2nd edition.

LAKSHMAN, T. V., & U. MADHOW. 1997. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking* 5(3):336–350.

LANCASTER, P., & M. TISMENETSKY. 1985. *The Theory of Matrices with Applications*. Academic Press, 2nd edition.

LANCZOS, C. 1997. *Linear Differential Equations*. Dover Publ., Inc.

LASSILA, P., J.L. VAN DEN BERG, M.R.H. MANDJES, & R.E. KOOIJ. 2003. An integrated packet/flow model for TCP performance analysis. In *Proc. of ITC 18*, volume 18, pages 651–660.

LATOUCHE, G., & M.F. NEUTS. 1980. Efficient algorithmic solutions to exponential tandem queues with blocking. *SIAM Journal of Algebraic and Discrete Methods* 1(1):93–106.

——, & V. RAMASWAMI. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM.

LEE, K.W., T.E. KIM, & V. BHARGHAVAN. 2001. A comparison of end-to-end congestion control algorithms: the case of AIMD and AIPD. In *Proc. of Globecom*.

LESKELÄ, L. 2004. Stabilization of an overloaded queueing network using measurement-based admission control. Technical Report A470, Department of Engineering, Physics and Mathematics, Helsinki University of Technology.

LIU, Y., F. LO PRESTI, V. MISRA, D. TOWSLEY, & Y. GU. 2003. Fluid models and solutions for large-scale IP networks. *ACM SIGMETRICS Performance Evaluation Review* 31(4):91–101.

MANDJES, M.R.H., D. MITRA, & W.R.W. SCHEINHARDT. 2003a. Models of network access using feedback fluid queues. *Queueing Systems* 44:365–398.

——, ——, & ——. 2003b. A simple model of network access: feedback adaptation of rates and admission control. *Computer Networks* 41:489–504.

MASSOULIÉ, L., & J.W. ROBERTS. 1999. Bandwidth sharing: objectives and algorithms. In *Proc. of IEEE INFOCOM*, pages 1395–1403.

MATHIS, M., J. MAHDAVI, S. FLOYD, & A. ROMANOW. 1996. RFC 2018: TCP selective acknowledgment options. Technical report, IETF.

——, J. SEMSKE, J. MAHDAVI, & T. OTT. 1997. The macroscopic behaviour of the TCP congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review* 27(3):67–82.

MEYN, S.P., & R.L. TWEEDIE. 1993. *Markov Chains and Stochastic Stability*. Springer Verlag.

MISRA, V., W. GONG, & D. F. TOWSLEY. 1999. Stochastic differential equation modeling and analysis of TCP-windowsize behavior. In *Performance 1999 (Istanbul)*.

——, ——, & ——. 2000. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. *ACM SIGCOMM Computer Communication Review* 30(4):151–160.

MITRA, D. 1988. Stochastic theory of a fluid model of producers and consumers coupled by a buffer. *Adv. Appl. Prob.* 20:646–676.

NEEDHAM, T. 2000. *Visual Complex Analysis*. Oxford University Press Inc.

NEUTS, M.F. 1986. The caudal characteristic curve of queues. *Advances in Applied Probability* 18:221–254.

NS-2. Available at: `http://www.isi.edu/nsnam/ns/`.

OLSÉN, J. 2003. *Stochastic Modeling and Simulation of the TCP Protocol*. Uppsala, Sweden: Uppsala University dissertation.

OTT, T., J. KEMPERMAN, & M. MATHIS. 1996. The stationary behavior of ideal TCP congestion avoidance. Available at: ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps.

PADHYE, J., V. FIROIU, D. TOWSLEY, & J. KUROSE. 2000. Modeling TCP throughput: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking* 8(2):133–145.

PAREKH, A. K., & R. G. GALLAGHER. 1993. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking* 1(3):344–357.

PETERSON, L.L., & B.S. DAVIE. 2000. *Computer Networks*. Morgan Kaufman Publ., 2nd edition.

PETROVSKI, I.G. 1966. *Ordinary Differential Equations*. Dover Publ., Inc.

PRABHU, N. U. 1980. *Stochastic Storage Processes*. Springer Verlag.

PRYCKER, M. DE. 1995. *Asynchronous Transfer Mode, Solution for Broadband ISDN*. Prentice Hall.

REN, Q., & H. KOBAYASHI. 1995. Transient solutions for the buffer behavior in statistical multiplexing. *Performance Evaluation* 23:65–87.

ROBERTS, J.W., & L. MASSOULIÉ. 2000. Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems* 15:185–201.

——, U. MOCCI, & J. VIRTAMO (eds.) 1996. *Broadband Network Traffic*, volume 1155 of *Lecture Notes in Computer Science*. Springer Verlag.

ROSS, S.M. 1993. *Introduction to Probability Models*. Academic Press, 5th edition.

—— 1996. *Stochastic Processes*. John Wiley & Sons, 2nd edition.

SCHEINHARDT, W.R.W. 1998. *Markov-Modulated and Feedback Fluid Queues*. Enschede, The Netherlands: Faculty of Mathematical Sciences, University of Twente dissertation.

—— 2001. Analysis of feedback fluid queues. In *14th ITC specialists seminar on access networks and systems*, pages 215–220.

——, N.D. VAN FOREEST, & M.R.H. MANDJES. 2005. Continuous feedback fluid queues. *To appear in Operations Research Letters* .

SCHWARTZ, M. 1996. *Broadband Integrated Networks*. Prentice Hall.

SERICOLA, B. 1998. Transient analysis of stochastic fluid models. *Performance Evaluation* 32:245–263.

—— 2001. A finite buffer fluid queue driven by a Markovian queue. *Queueing Systems* 38:213–220.

——, & B. TUFFIN. 1999. A fluid queue driven by a Markovian queue. *Queueing Systems* 31:253–264.

SHIRYAEV, A.N. 1996. *Probability*. Springer, 2nd edition.

SONNEVELD, P. 1988. Some properties of the generalized eigenvalue problem $Mx = \lambda(\Gamma - cI)x$, where M is the infinitesimal generator of a Markov process, and Γ is a real diagonal matrix. Preliminary report, Delft University of Technology, Delft.

—— 2004. Some properties of the generalized eigenvalue problem $Mx = \lambda(\Gamma - cI)x$, where M is the infinitesimal generator of a Markov process, and Γ is a real diagonal matrix. ISSN 1389-6520, Delft University of Technology, Delft.

STERN, T., & A. ELWALID. 1991. Analysis of separable Markov-modulated models for information handling systems. *Adv. Appl. Prob.* 23:105–129.

STEVENS, W. 1997. RFC 2001: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. Technical report, IETF.

STEWART, W. J. 1994. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press.

TANAKA, T., O. HASHIDA, & Y. TAKAHASHI. 1995. Transient analysis of fluid model for ATM statistical multiplexer. *Performance Evaluation* 23:145–162.

TANENBAUM, A. 1996. *Computer Networks*. Prentice Hall.

TUCKER, R.C.F. 1988. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Transactions on Communications* 36(4):479–483.

VIRTAMO, J., & I. NORROS. 1994. Fluid queue driven by an M/M/1 queue. *Queueing Systems* 16:373–386.

VRANKEN, R., R.D. VAN DER MEI, R.E. KOOIJ, & J.L. VAN DEN BERG. 2002. Performance of TCP with multiple priority classes. In *International Seminar Telecommunication Networks and Teletraffic Theory*, pages 78–87.

WALRAND, J. 1998. *Communication Networks: A First Course*. McGraw-Hill, 2nd edition.

Symbols Index

Samenvatting

Dit proefschrift beschrijft uitbreidingen van de theorie van wachtrijmodellen met terugkoppeling en gebruikt een aantal van deze modellen voor een prestatie-analyse van het Transmission Control Protocol, een flow-control protocol dat veel gebruikt wordt in het Internet.

De wachtrijen waaraan we de meeste aandacht besteden zijn 'fluid queues'. Hierbij sturen één of meerdere bronnen met steeds wisselende snelheid vloeistof in een buffer die zelf met een constante snelheid leegloopt. De standaard fluid queue karakteriseert het brongedrag als een continue-tijd Markovketen met een daarmee geassocieerde generatormatrix. De netto instroomsnelheid van vloeistof is een functie van de brontoestand; deze snelheden vatten we samen in een driftmatrix. Als er sprake is van terugkoppeling zullen de elementen van de generator- en driftmatrix van de bufferinhoud afhangen hetwelk resulteert in een 'feedback fluid queue'. In de bekende feedback fluid modellen is deze afhankelijkheid discreet van aard: de generator- en driftmatrix zijn constant zolang de inhoud zich tussen twee (vaste) drempels bevindt. Zodra echter de bufferinhoud een drempel passeert, kunnen zowel de generatormatrix als de driftmatrix veranderen.

In hoofdstuk 2 gebruiken we een feedback fluid queue met een enkele drempel als een beschrijving van een TCP-bron die verkeer over een link met een eindige buffer stuurt. Zolang de buffer niet vol is, zendt de buffer signalen naar de TCP-bron om diens zendsnelheid te verhogen. Echter, als congestie optreedt, ontvangt de bron negatieve terugkoppeling met het doel de instroomsnelheid te verlagen. Het blijkt dat de bekende wortel-p wet—een relatie tussen de throughput van een TCP-verbinding, de pakket-verlieskans en de round-trip-tijd—ook in ons model geldig is.

Daarna, in hoofdstuk 3, onderzoeken we een model met twee of meer TCP-bronnen. In dit model is het essentieel het gedrag van de bronnen vast te leggen nadat verlies optreedt tengevolge van het overlopen van de buffer. Hier implementeren we 'synchronous loss' (alle bronnen halveren hun transmissiesnelheid na congestie), in plaats van 'proportional loss' (slechts één van de bronnen halveert na congestie). De keuze voor synchronous loss noodzaakt ons het bronproces uit te breiden met een extra stochastische variabele. Voor een model met twee bronnen beschouwen we de benutting van de link

en de fairness, dat wil zeggen, de fractie van de beschikbare capaciteit die ieder van de bronnen krijgt.

Het modelleren van TCP met een fluid queue met één drempel is niet erg nauwkeurig als de buffer groot is. De reden is dat de frequentie waarmee terugkoppelsignalen de bron bereiken een functie is van de round-trip-tijd, en die hangt in werkelijkheid op een continue wijze van de bufferinhoud af. In een accurater TCP model moeten daarom de (elementen van de) generator- en driftmatrix continue functies van de bufferinhoud zijn. Voor een fluid queue met continue terugkoppeling en eindige buffergrootte stellen we in hoofdstuk 4 de Kolmogorov-vergelijkingen op. Vervolgens bewijzen we dat een stationaire verdeling bestaat en vinden een gesloten uitdrukking voor deze verdeling voor modellen waarin de bron twee toestanden heeft.

Hoewel een TCP-model op basis van een fluid queue met continue feedback nauwkeuriger is, blijkt het numeriek oplossen van een dergelijk model moeilijk. In hoofdstuk 5 omzeilen we dit probleem door het buffergedrag te modelleren als een discreet, in plaats van continu, proces. Hierdoor kunnen we het gehele systeem als een Markovketen weergeven. Nu vergelijken we de beide modellen ten aanzien van de reactie van de bronnen op congestie, t.w., synchronous en proportional loss, met simulaties uitgevoerd met de netwerk simulator ns-2. Het blijkt dat de resultaten van de gesimuleerde werkelijkheid tussen die van de beide verliesmodellen in vallen.

Ook dit Markovmodel kent zijn beperkingen in de zin dat de implementatie van meerdere TCP verbindingen over meerdere links ingewikkeld is. Door het systeem als een Petrinet te beschrijven, verkrijgen we de generatormatrix met behulp van standaardprogrammatuur. Deze procedure, die we uitvoeren in hoofdstuk 6, vereenvoudigt de gehele implementatie sterk. De numerieke evaluatie met behulp van dit TCP-model levert een nieuw inzicht in de verdeling van bandbreedte ingeval een TCP-verbinding over twee links concurreert met zijverkeer bestaande uit twee andere TCP-verbindingen met verschillende round-trip-tijden.

Tenslotte, in hoofdstuk 7, onderzoeken we een tandem-netwerk van twee exponentiële servers met Poisson-aankomsten. Nu is de terugkoppeling zodanig dat zolang de bufferinhoud in het tweede station zekere drempelwaarden overstijgt, de eerste server langzamer gaat werken, of zelfs stopt. Omdat de stationaire verdeling van dit model geen produktvorm heeft, concentreren we ons op de (asymptotische) structuur van deze verdeling. We verkrijgen hierin inzicht met behulp van matrix geometrische methoden.

Summary

This dissertation expands the theory of feedback queueing systems and applies a number of these models to a performance analysis of the Transmission Control Protocol, a flow control protocol commonly used in the Internet.

In the first six chapters we are concerned with so-called fluid queues. In such queueing models one or more sources send fluid at varying rates into a buffer which depletes with constant rate. In the standard fluid queue the source(s) behave according to a continuous-time Markov chain with an associated generator matrix. The input rates of fluid depend on the source state; we assemble these rates into a drift matrix. In the presence of feedback, the elements of the generator and drift matrix are allowed to depend on the buffer content. The feedback fluid queues considered in the literature have discrete feedback, meaning that, while the content is in between two (fixed) thresholds, the generator and drift matrix are constant. However, when the content crosses a threshold, both the generator matrix and the drift matrix can change.

In Chapter 2 we use a feedback fluid queue with a single threshold to model a TCP connection which uses a single link and finite buffer. While the buffer is not congested, it sends feedback signals to the source to increase its transmission rate, whereas if the buffer is full, it informs the source to reduce the rate. We show that the root-p law—a relation between the source throughput, the packet loss probability, and the round-trip time—also holds for our model.

Then, in Chapter 3, we extend this single-source model such that it can incorporate multiple TCP connections. As a consequence, it becomes essential to model the reaction of the sources to loss occurring at buffer overflow. In the setting of this chapter we implement 'synchronous loss' (all sources reduce their rate after congestion), rather than 'proportional loss' (just one of the sources reduces its rate). This implies that we need to augment each source process with an extra random variable. We investigate, for a two-source system, the link utilization and fairness, that is, the fraction of link capacity each source receives.

It turns out that the TCP model with merely one threshold is quite inaccurate when the buffer size is large. One reason is that the frequency at which the buffer sends signals to

the source depends on the round-trip time, which, in turn, is a continuous function of the buffer content. Thus, in a more accurate TCP model the generator and drift matrix should be continuous functions of the buffer content. For such continuous feedback fluid queues we derive in Chapter 4 the Kolmogorov equations, prove that a stationary distribution exists, and obtain a closed form expression for this distribution in case the source process has only two states.

Whereas the TCP model based on continuous feedback fluid queues is more accurate, the numerical evaluation of this model proves problematic. In Chapter 5 we resolve this by proposing to model the buffer process as a discrete, rather than as a continuous process; this allows us in effect to represent the entire system as a single Markov chain. We compare the synchronous and proportional loss models to simulations with the network simulator ns-2. Interestingly, the results of the simulated reality lie within those of the two loss models.

Still, this approach has its limitations: it is difficult to build by hand a generator matrix that represents multiple TCP sources which also share multiple links. By considering such systems as Petri nets, standard software can produce the generator matrix instead, thereby simplifying the implementation of more complicated networks considerably. The numerical evaluation carried out in Chapter 6 gives some new insights into the sharing of bandwidth when a single TCP connection competes on two links with cross traffic from two other TCP connections with different round-trip times.

Finally, in Chapter 7 we investigate a two-station tandem network in which jobs arrive according to a Poisson process and require exponential service at the stations. Now the feedback is such that when the queue length at the second station crosses a certain threshold, the first server reduces its service rate or stops altogether. As the stationary distribution of this system does not have a product form, we concentrate on its (asymptotic) structure. By using matrix analytic methods we obtain considerable insight into this.

Dankwoord

De afgelopen vier jaar heb ik mij mogen verheugen in de samenwerking met mijn promotores Michel Mandjes en Werner Scheinhardt. Ik ben hen zeer erkentelijk voor hun betrokkenheid bij mijn onderneming ‘de “s” weg te poetsen’. Aan iedere fase van de totstandkoming van dit boekje—het genereren van ideeën, het uitwerken ervan, hun onverdroten houding ten aanzien van het verschaffen van (detail) commentaar—hebben zij veel bijgedragen. Wellicht het grootste compliment dat ik kan geven is dat ik hen ‘beleefd aanbeveel’ bij andere doctorandi die promovendi op het gebied van toegepaste kansrekening willen worden. Dat er velen mogen volgen! Daarnaast, ik heb veel genoegen beleefd aan onze talloze, wijdlopige gesprekken over allerhande onderwerpen volkomen bezijden de inhoud van het onderzoek.

Ik dank Jan-kees van Ommeren voor de vele discussies en zijn toegankelijkheid. Ik bewonder zijn vermogen vragen te stellen die erg voor de hand lijken te liggen, maar pas *nadat* hij ze opgebracht heeft, zelden *ervoor*.

Henk Zijm bedank ik graag voor de mogelijkheid die hij mij bood te promoveren bij de vakgroep SOR.

Ik dank mijn collegas voor de (lunch)gesprekken en hun bereidheid mijn inhoudelijke vragen te beantwoorden.

About the Author

Nicky van Foreest finished grammar school at the Stedelijk Gymnasium Arnhem in 1987. Then he studied Theoretical Physics at Utrecht University until 1993. After having worked for KPN Research, Fortis Investments, and the Bell Labs department of Lucent Technologies in Enschede, he started a Ph.D. project at the University of Twente, Enschede. His research focused mainly on performance analysis of stochastic feedback fluid models and applications to the Transmission Control Protocol of the Internet. He is married to Sheila Timp and has three sons: Jorden, Lucas, and Pieter.