Queueing models for bandwidth-sharing disciplines

Lieshout, P.M.D.

Publication date
2008
Document Version
Final published version

Queueing Models for Bandwidth-Sharing Disciplines

Queueing Models for Bandwidth-Sharing Disciplines

Pascal Lieshout

Pascal Lieshout

Queueing Models for Bandwidth-Sharing Disciplines

Queueing Models for Bandwidth-Sharing Disciplines /
Pascal Merijn Daniël Lieshout, 2008
Proefschrift Universiteit van Amsterdam

Gedrukt door Ponsen & Looijen B.V.
ISBN 978-90-6464-270-8

Thomas Stieltjes Institute
for Mathematics

BRICKS

Part of this research has been funded by the Dutch BSIK/BRICKS project

Queueing Models for Bandwidth-Sharing Disciplines

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus prof. dr. D.C. van den Boom
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in de Agnietenkapel
op vrijdag 5 september 2008, te 12:00 uur

door

Pascal Merijn Daniël Lieshout

geboren te Amsterdam

Promotiecommissie:

Promotores:	prof. dr. M.R.H. Mandjes
	prof. dr. ir. S.C. Borst
Overige leden:	prof. dr. R.J. Boucherie
	prof. dr. C.A.J. Klaassen
	dr. R. Núñez Queija
	dr. A.A.N. Ridder
	dr. P.J.C. Spreij
	prof. dr. ir. J. van der Wal

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Dankwoord

Dit proefschrift is het resultaat van het door mij verrichte onderzoek in de periode november 2004 tot juli 2008. Ik heb deze periode ervaren als een ontzettend leuke tijd waarin ik veel heb geleerd. Hoewel alleen mijn naam op de omslag staat, had ik het niet kunnen schrijven zonder de hulp van velen.

Allereerst wil ik graag mijn promotores Michel Mandjes en Sem Borst bedanken voor de begeleiding. Ik heb onze samenwerking altijd als erg plezierig ervaren. Daarnaast wil ik mijn collega's van PNA2 van het CWI bedanken. Mede dankzij jullie kijk ik terug op een fijne periode bij het CWI. Ook wil ik mijn collega's van het Korteweg-de Vries Instituut voor Wiskunde (UvA) bedanken.

Mijn familie en vrienden wil ik bedanken voor alle steun van de afgelopen jaren. Ook al begrepen jullie waarschijnlijk nooit veel van wat ik vertelde over mijn onderzoek, toch hadden jullie een feilloos vertrouwen in mij. Als laatste wil ik graag Diana bedanken voor haar onvoorwaardelijke liefde en steun.

Amsterdam, juli 2008
Pascal Lieshout

Contents

Chapter 1

Introduction

Current communication networks are evolving towards *integrated-services* networks, implying that they are expected to support a wide range of heterogeneous services, including data, video, and voice-applications, but also more demanding multimedia applications, such as gaming, remote surgery, video-conferencing, etc. In order to provide proper service to each application, it is important that the available service capacity is shared among the various traffic classes in a suitable manner.

In this monograph we analyze mathematical models for *bandwidth sharing* in such multi-service networks. In particular, we focus on i) explicit scheduling in network links, and ii) bandwidth sharing as a consequence of the end-to-end rate control by end-users. In the former case certain traffic classes may receive preferential treatment in network links, thereby offering service differentiation. In the latter case the bandwidth sharing is strongly affected by the protocol that governs the transfer of traffic along the end-to-end route. Part I of this thesis is devoted to case i), whereas Part II considers case ii). For both cases, various *bandwidth-sharing disciplines* can be identified for either implementing or modeling bandwidth sharing. We apply *queueing theory* as a tool to analyze the performance of several such mechanisms.

1.1 Modeling of communication networks

In this section we show how a communication network may be modeled as a queueing system, i.e., a network of queues. To see the connection with a queueing system, it is important to distinguish between 1) the network itself, and 2) the traffic of the applications that it supports. Below we first describe the basic characteristics of 1) and 2), before making the connection with a queueing system.

1.1.1 Network characteristics

We focus on a wired communication network that supports several heterogeneous applications. These networks consist of nodes that communicate over links. Nodes

are computers, switches, routers, servers and other devices. Some nodes are classified as end-users, which provide the interface between the users and the network. Other nodes, e.g. switches and routers, are not identified with any user, but forward traffic as it is sent between users. Links are physical channels over which traffic is transmitted. We assume that all traffic in the network is digitized, i.e., traffic consists of small *packets*. Such networks are commonly referred to as *packet-switched* networks.

1.1.2 Traffic characteristics

Below we mention four important properties that network traffic usually obeys.

Stationarity

Traffic on network links, averaged over suitable time periods, typically exhibits systematic variations. These variations usually follow a daily pattern, with a clearly identifiable busy period, which can last several hours. During this busy period, traffic arrival processes approximately show *stationary* behavior in the sense that the statistical properties are nearly time-invariant. Therefore, a cumulative traffic input process is usually modeled as a stochastic process with stationary increments, see also Section 1.5. This implies that the statistical properties of traffic are assumed to be constant for an indefinite period.

High level of aggregation

The input stream of each node in communication networks usually consists of a superposition of a large number of individual streams. To give an indication, at the core network, resources are commonly shared by thousands of users, whereas at the access of a network, the number of aggregated streams is typically at least in the order of tens.

Streaming and elastic traffic

The majority of traffic can broadly be categorized into *streaming* and *elastic* traffic, each having its own *Quality-of-Service* (QoS) requirements, see [157].

Streaming traffic is produced by audio and video applications for both real-time communication and reproduction of stored sequences. Usually, the transmission rate has some intrinsic time profile, which may either be nearly constant or highly bursty, depending on the specific application. In both these cases, the application-level QoS is mainly determined by integrity of the time profile, making small packet delay and low loss crucial requirements.

Elastic traffic, on the other hand, results from the transfer of digital documents such as Web pages, files and e-mails. In contrast to streaming traffic, the transmission

rate can be adapted over time, based on the level of congestion in the network, because these applications are typically more tolerant of packet delays.

Time scale separation

In packet-switched networks it is common to distinguish between different time scales. At the lowest level, the *packet-level*, the main interest concerns the individual packets that are transmitted through the network. At the highest level, the *flow-level*, we leave out all the packet-level details, and consider the sequence of all packets from the beginning of a transfer until the end as a single *flow*. During a transfer, periods in which bursts of packets are sent usually alternate with intervals in which no packets are transmitted, which gives rise to an intermediate time scale, the *burst-level*.

At the packet-level we deal with the fluctuations of the packet arrivals within a burst. At the burst-level we consider the fluctuations of the level of activity of users, as opposed to the flow-level, where we focus on the fluctuations of the number of users.

If one, in case of elastic traffic, is interested in the performance as perceived by end-users, then it is appropriate to study models at the flow-level. In this case chief interests concern sojourn times (time between arrival and departure) of flows, but also include issues of fairness (concerning bandwidth sharing among various types of flows) and bandwidth utilization. In contrast, if one aims to study the performance as perceived by end-users in case of streaming traffic, then it is more appropriate to apply analysis at the packet-level or the burst-level. Now one typically studies the performance (of a system) in terms of packet losses (due to buffer overflow) and packet delays. Despite the fact that burst-level models leave out the packet-level details, they lend themself very well for this goal, the underlying idea being that the packet loss probability can be approximated by the buffer overflow probability, whereas the packet delay can be approximated by the virtual delay, i.e., the delay experienced by a packet having arrived at an arbitrary point in time.

Below we discuss the main characteristics of each of these three levels. We refer to [82, 159] for more details concerning the various levels.

Packet-level characteristics
Measurements of network traffic as performed over the past decade showed [110, 152] that traffic at the packet-level, but also at the burst-level and the flow-level, is typically highly variable, or *bursty*, over a wide range of times scales. Such traffic is known to be *long-range dependent* (LRD), meaning that the autocorrelations decay relatively slowly, see Chapter 2 for formal definitions. Traffic exhibiting long-range dependence can claim a substantial part of the available capacity for a relatively long period. In the absence of a protection mechanism, this can prevent other traffic from receiving proper service.

Several papers offered explanations for traffic to be LRD, see e.g. [46, 47, 150, 174].

These papers argue that LRD may be caused by the fact that certain traffic characteristics are *heavy-tailed*. In particular, it is indicated that file sizes and transmission times of files in the Internet, which are typical examples of flows, may have infinite variance. However, LRD may also be due to other traffic-related or user-related characteristics, see e.g. [47, 174] for an overview.

As opposed to traffic exhibiting long-range dependence, traffic can also be *short-range dependent* (SRD). In this case the autocorrelations decay relatively quickly, implying that SRD traffic behaves smoothly over long time scales. Not surprisingly, SRD is intimately related to traffic with *light-tailed* properties.

Flow-level characteristics
As mentioned before, measurements at various locations of flow sizes showed that their distribution has a heavy tail in general. The precise distribution clearly depends on the type of flow under consideration. A reasonable fit to the shape of the heavy tail is provided by the Pareto distribution:

$$\mathbb{P}(\text{size} > x) \approx \frac{\alpha}{x^{\beta}}, \quad \text{for large values of } x,$$

where $1 < \beta \leq 2$ to ensure that this distribution has a finite mean and infinite variance, and α is some positive constant.

The above distribution has the property that the majority of the flows is small, whereas most of the traffic volume is contained in the large flows. This property has also been verified by extensive measurements, see e.g. [46, 73].

A typical and important example of a light-tailed flow size distribution is the exponential distribution:

$$\mathbb{P}(\text{size} > x) = e^{-\lambda x}, \quad x \geq 0,$$

where $\lambda > 0$ is the rate parameter. It is easy to verify that this distribution has finite mean $1/\lambda$ and variance $1/\lambda^2$. Generalizations of the exponential distribution, such as phase-type distributions, are other examples of distributions with light tails.

Burst-level characteristics
In packet-switched networks traffic is divided into small packets that are sent over the network. When considering somewhat larger time scales, traffic may approximately be viewed as a continuous flow of *fluid*, thereby neglecting the discrete nature of the relatively small packets. In particular, there is usually an alternation of inactive periods, in which no packets are sent, and active periods, in which bursts of packets are transmitted, such that the generated traffic can be modeled as an *On-Off* source.

The notions of heavy-tailedness (light-tailedness) and LRD (SRD) are closely related in case of an On-Off source as well. If the On-period or Off-period are drawn from a distribution having infinite variance, then the On-Off process can shown to be

LRD, see [38]. Other results can be found in e.g. [79]. In case both the On-period and Off-period have a light-tailed distribution, the On-Off process is SRD.

1.1.3 Queueing system

We now relate the elements described above to a queueing system. In general, a queueing system describes a system where limited resources are used to perform certain tasks. These resources are often referred to as servers, whereas the tasks to be performed can be viewed as customers that arrive to the servers, each of them bringing along a certain amount of work to be executed by the servers.

Sections 1.1.1 and 1.1.2 indicate how a communication network can be represented by a queueing system. Clearly, the nodes of the network correspond to the available resources in a queueing system. Depending on whether we analyze the network at packet-, flow-, or burst-level, we see that a customer can be identified with a packet, flow, or burst of packets, respectively.

In this monograph we study communication systems at either the burst-level or the flow-level. In other words, we leave out all packet-level details, and focus on somewhat larger time scales. In particular, in Part I we analyze models at the burst-level, implying that we assume the number of flows to be fixed in the system, whereas in Part II we study models at the flow-level, thus assuming the number of flows to be variable.

Important references on queueing theory are e.g. Asmussen [12], Cohen [44], and Tijms [163]. In e.g. [100, 101, 171] it is shown how queueing theory can be applied to communication networks.

1.2 Basic queueing models

Before presenting various bandwidth-sharing disciplines in the next two sections, we first need to introduce some queueing theory terminology, which we do in the context of the classical G/G/1 queue and the fluid queue.

1.2.1 Classical G/G/1 queue

The most basic queueing model is the single-server queue. In this system customers arrive one at a time. The time between two consecutive arrivals is called the *interarrival time*. In practice one assumes that the sequence of interarrival times consists of independent and identically distributed (i.i.d.) random variables. As reflected by the name of this model, there is one server, which works at some predefined speed c. The service requirements of customers are assumed to be i.i.d. random variables. Moreover, the sequences of interarrival times and service requirements are assumed to be independent. As soon as a customer is completely served, it leaves the system.

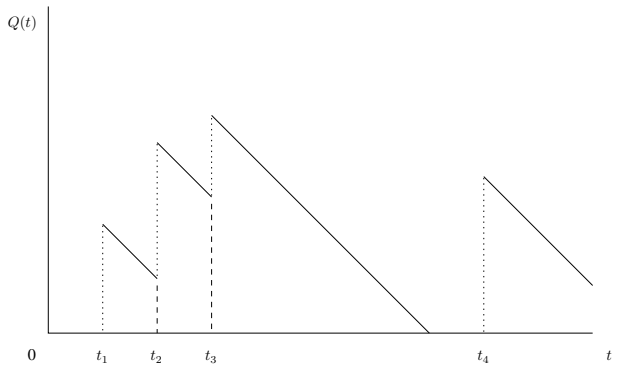

Figure 1.1: An example of the workload process $\{Q(t), t \geq 0\}$ in a classical G/G/1 queue.

The above model is usually referred to as the G/G/1 queue, a notation that is due to Kendall [92]. Here the first G reflects that the interarrival time distribution is of a *general* form, whereas the second G indicates that the same holds for the service requirement distribution. In the special case of *Poisson* arrivals, i.e., exponentially distributed interarrival times, we denote the model as the M/G/1 queue, where the M stands for *memoryless* or *Markovian*. If the service requirements are exponentially distributed as well , we obtain the well-known M/M/1 queue.

Clearly, as long as there are customers in the system, the server works at the predefined speed c, which is enough to describe the evolution of the queue length.

Figure 1.1 shows a typical sample path of the workload process $\{Q(t), t \geq 0\}$ (i.e., the sum of the service requirements of all the customers in the system) in a classical G/G/1 queue with unit capacity. Here t_i denotes the time at which the ith customer arrives. The heights of the small dashed lines represent the corresponding service requirements of these customers. These jumps in the workload process illustrate that traffic, in a classical G/G/1 queue, arrives *instantaneously*.

We remark that the G/G/1 queue is a widely used model in various fields of research. For example, many problems in inventory, risk theory, communication networks, etc., can often be reformulated in terms of these classical queueing systems, see e.g. [12].

1.2.2 Fluid queue

In Section 1.1 we already mentioned that network traffic is inherently bursty. When studying traffic at the burst-level, traffic is usually modeled as a continuous fluid flow, thereby neglecting the discrete nature of relatively small packets. More generally, fluid models may be valuable when a separation of time scales applies. That is, fluctuations

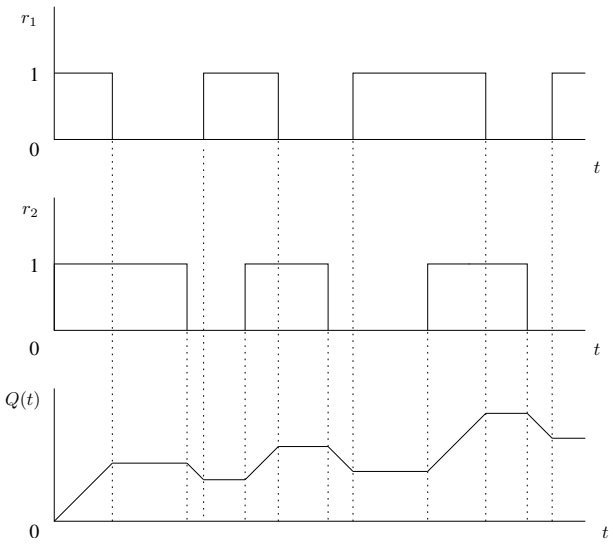

Figure 1.2: An example of the workload process $\{Q(t), t \geq 0\}$ in a fluid queue fed by a superposition of two On-Off sources.

around a certain drift on a shorter time scale may sometimes be neglected on a longer time scale. The main difference with classical queues is that traffic does not arrive instantaneously, but *gradually* over time.

On-Off source
A popular way of modeling bursty traffic is by means of On-Off sources, as there is usually an alternation between periods in which packets are sent, and in which no packets are transmitted, see e.g. [11, 43, 102, 54, 160, 180].

Figure 1.2 illustrates an example in which a fluid queue with unit capacity is fed by a superposition of two On-Off sources. It is assumed that both sources generate traffic at constant rate $r_1 = r_2 = 1$ in On-periods, and that the buffer is empty at time zero. Therefore, we find that the total workload $Q(t)$ at time $t \geq 0$, with $Q(0) = 0$, builds up with rate one if both sources are active, remains constant if only one of the sources is transmitting, or drains with rate one if both sources are silent.

Reflection of a process
One way to model the workload process of a queue, either with instantaneous or gradual input, is to define it as the reflection at zero of some process $\{A(t) - ct, t \geq 0\}$, where $\{A(t), t \geq 0\}$ is a continuous-time stochastic process, denoting the amount of traffic entering the system in the interval $[0, t]$, and $c > 0$ is the service capacity of

the node. By reflection at zero we mean that the workload at time $t \geq 0$ can be represented as

$$\begin{aligned} Q(t) &= A(t) - ct - \inf_{0 \leq s \leq t} \{A(s) - cs\} \\ &= \sup_{0 \leq s \leq t} \{A(t) - A(s) - c(t-s)\} \geq 0, \end{aligned}$$

given that we start with an empty system, i.e., $Q(0) = 0$. The reflection at zero ensures that the workload process is non-negative.

The above-mentioned approach will be applied in this monograph to analyze queues at the burst-level. In particular, in Part I we assume $A(t)$ to be a so-called *Gaussian* process. Gaussian processes cover both SRD and LRD traffic, see Chapter 2 for more details.

Figure 1.3 depicts an example of the reflection of the process $\{B(t) - ct, t \geq 0\}$ at zero, where $B(t)$ is a so-called *Brownian motion*, which is a special case of a Gaussian process. Brownian motions, and Gaussian processes in general, play an important role in this monograph.

1.3 Scheduling in network nodes

An instrument that can be used to accomplish service differentiation, is the so-called *scheduling mechanism*. Such a mechanism has to be implemented in the switches or routers of a network, and it determines for each arriving packet at what time it is forwarded to the next router or switch on its route. The goal of these mechanisms is to implement differentiated sharing or ensure fairness, such that the proper QoS can be provided to each application.

The most important packet-based scheduling mechanisms in current communication networks are variants of Weighted Fair Queueing (WFQ) and Weighted Round-Robin (WRR). Both WFQ and WRR are weighted versions of standard Round-Robin (RR) scheduling, where the various traffic classes may receive different service quotas, as specified by class-specific service weights. In this section, we focus on an ideal fluid-based variant of WFQ, the so-called Generalized Processor Sharing (GPS) discipline. In other words, GPS assumes that traffic of the various classes is infinitely divisible and that it can serve several classes simultaneously. In reality, however, traffic consists of small packets which have to be processed sequentially, implying that GPS is a convenient idealization, but not implementable. In [148, 149] it was shown that WFQ, which *is* implementable, closely approximates the behavior of GPS. Therefore, we expect that results for GPS carry over to WFQ, especially the ones that relate to burst-level performance metrics.

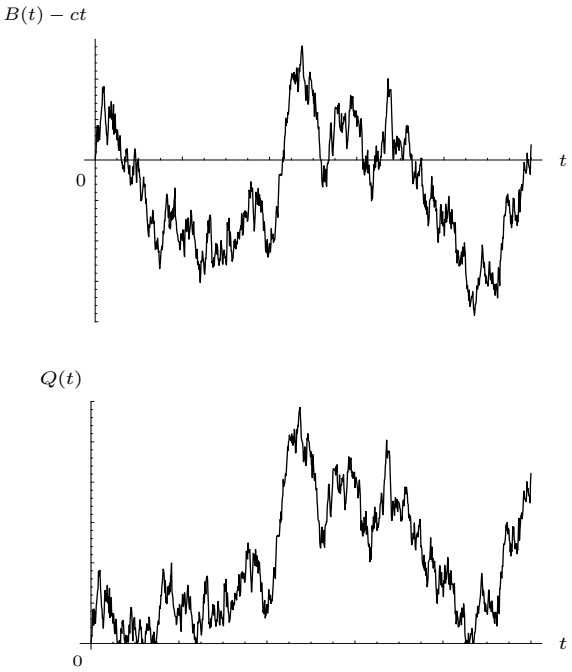

Figure 1.3: An example of the reflection of the process $\{B(t) - ct, t \geq 0\}$ at zero, where $B(t)$ is a Brownian motion.

1.3.1 Generalized Processor Sharing

Assume that there are M different traffic classes that require service at a particular node in the network with service rate c. Each class is assigned a weight ϕ_i, $i = 1, \ldots, M$. Without loss of generality, assume that the weights add up to one, i.e., $\sum_{i=1}^{M} \phi_i = 1$. The GPS weight ϕ_i determines the guaranteed service rate $\phi_i c$ for class i. If all classes are *backlogged*, i.e., if the queues of all classes are non-empty, then class i receives service at rate $\phi_i c$.

Let us first assume that each of the traffic classes consists of flows that generate instantaneous traffic bursts. In that case a class either fully uses its allocated share of the service capacity or does not use any service capacity. In the latter case, its service share becomes available to the other backlogged classes, and is also shared according to these weights. That is, denoting the set of backlogged classes by $\mathcal{B}$, the service rate allocated to class $i \in \mathcal{B}$ equals

$$\frac{\phi_i c}{\sum_{j \in \mathcal{B}} \phi_j} \geq \phi_i c,$$

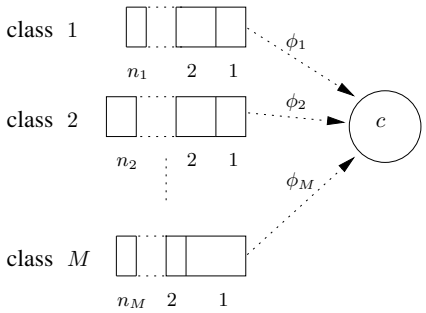

Figure 1.4: GPS mechanism

see Figure 1.4 for an illustration.

In case of fluid input for some class, some subtleties may arise, as a class may then receive service equal to its input rate, without being backlogged. A more formal description of GPS is therefore given in [148, 149]. Let $S_i(s,t)$ denote the amount of traffic of class i served in the time interval $[s,t]$, $i = 1, \ldots, M$. If the queue of class i is backlogged in the corresponding interval, then GPS satisfies the following property:

$$\frac{S_i(s,t)}{S_j(s,t)} \geq \frac{\phi_i}{\phi_j}, \quad i,j = 1,\ldots,M, \quad i \neq j. \tag{1.1}$$

Obviously, there is equality in (1.1) if class j is also continuously backlogged in the interval $[s,t]$.

From the above we conclude that GPS achieves statistical multiplexing gains by re-allocating capacity from non-backlogged classes. Note that GPS is a *work-conserving* scheduling discipline, i.e., the server always works at maximum speed if at least one of the queues is non-empty. Also, notice that GPS aims to describe the performance at the burst-level, where the population of flows may be assumed nearly static.

Assigning weight one to a single class, implies that the other classes can only be served if there is no traffic of this single class queued; i.e., *priority queueing* can be regarded as a special case of GPS. By assigning positive weights to all classes, GPS is capable of protecting a class against starvation when some other class 'misbehaves', as opposed to priority scheduling, where the low-priority classes may be excluded from service over substantial time intervals. Therefore, GPS can be regarded as a protection mechanism.

We already mentioned that measurements showed that network traffic is typically highly variable or bursty over a wide range of times scales, which underscores the importance of a protection mechanism like GPS.

We remark that the GPS model is in fact a special case of the coupled processors model [45]. Also, the GPS discipline shows resemblance with so-called *cycle stealing policies*, see [145] for more details.

1.4 Rate control by end-users

In the previous section we focused on 'explicit scheduling' (of packets) in a network node. We now assume that bandwidth sharing is a consequence of the end-to-end rate control by end-users. In that case the bandwidth shares are strongly affected by the protocol that governs the transfer of packets along the end-to-end route. Hence, there is 'implicit scheduling'. In the current Internet, the dominant transport protocols are variants of the Transmission Control Protocol (TCP).

TCP is capable of providing both error control and congestion control. In order to guarantee error control, the receiver sends an acknowledgment (ack) to the source after each (group of) correctly received packet(s). In case the source does not receive an ack before a time-out occurs or it receives a duplicate ack indicating that some packet is missing, the packet is assumed to be lost. When a packet is lost or when a negative ack (indicating that the packet contains errors) is received, the source retransmits the lost packet.

The congestion control of TCP is based on a so-called *window size*, which specifies the maximum number of packets that can be sent by the source without having received an ack. TCP infers the level of congestion in the network from the returned ack's. In case packets are lost, TCP concludes that the level of congestion is high and reduces the window size. In contrast, in case no packets are lost, TCP concludes that the network is lightly loaded and increases the window size up to some maximum.

A TCP-based data transfer starts with a *slow-start* phase, in which the window size increases at an exponential rate over time. Next follows a *congestion-avoidance* phase, in which the window size increases linearly at rate 1/RTT, where RTT stands for the *round-trip time* of each correctly received ack. Note that this is effectively done by increasing the window size W by 1/W for each acknowledged packet. We refer to [83, 106] for more details concerning TCP.

We remark that the congestion-avoidance phase of TCP can in fact be viewed as a special case of the family of Additive-Increase-Multiplicative-Decrease (AIMD) congestion control protocols, in which the window size increases linearly when no losses occur, whereas the window size is reduced by a multiplicative factor when a loss is detected, see [139].

Below we first discuss two single-node flow-level systems that may properly model the bandwidth sharing realized by TCP in a common link, namely: Processor Sharing (PS) and Discriminatory Processor Sharing (DPS). Recall that at the flow-level we leave out packet-level details, and focus on somewhat larger time scales.

1.4.1 Processor Sharing

The last decade, the PS discipline has emerged as a useful paradigm for evaluating the performance of a variety of resource allocation mechanisms. Under the PS dis-

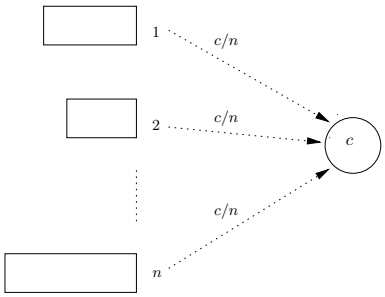

Figure 1.5: PS mechanism

cipline, the server simultaneously serves each of the n users present with rate c/n, see Figure 1.5 for an illustration. While the PS discipline originally emerged as an idealization of RR scheduling mechanisms in time-shared computer systems [99], in recent years the PS discipline has received renewed attention as a convenient abstraction for modeling the flow-level performance of bandwidth-sharing protocols in packet-switched networks, in particular TCP. During the congestion-avoidance phase, all active users receive approximately equal bandwidth, assuming they have identical access rates and RTTs. This motivates the use of PS to model the dynamic behavior of TCP flows sharing a common link, see e.g. [18, 81, 135, 136, 141, 158]. Here it is assumed that, at the moment a flow-level transition takes place, the allocation of the transmission rates to the individual flows adapts instantly. We remark that this is an idealization, as under TCP the adaptation does not occur instantly, but takes some time. More references on PS can be found in [36, 143].

Although the PS model provides valuable insights, it critically relies on the assumption that the service capacity is equally shared among competing flows, i.e., it assumes *fair* sharing. In [10] it was, however, argued that the actual service rates realized by TCP may show substantial variation among flows with heterogeneous RTTs and access rate limitations, implying that the PS model is not always appropriate. In the next subsection we therefore discuss DPS, which models differentiated bandwidth sharing.

1.4.2 Discriminatory Processor Sharing

The DPS discipline, which is closely related to PS, is useful for modeling the flow-level performance of bandwidth-sharing protocols such as TCP in packet-switched networks. In addition, DPS provides a natural framework for modeling the flow-level performance of differentiated bandwidth-sharing disciplines such as WFQ and WRR. Hence, DPS is appropriate for modeling differentiated bandwidth sharing. For applications of DPS in communication networks, see e.g. [1, 10, 40, 42].

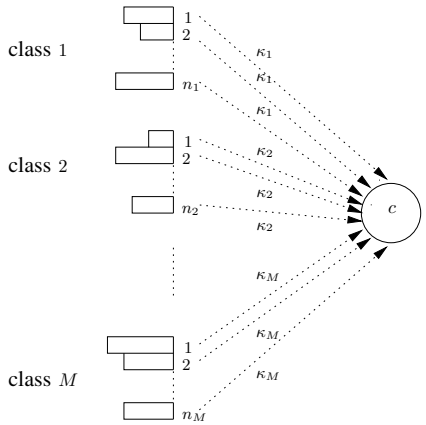

Figure 1.6: DPS mechanism

Assume that there are M classes of flows that require service at a particular node in the network. Let n_i denote the number of class-i flows, $i = 1, \ldots, M$. Each class of flows is assigned a non-negative weight κ_i, $i = 1, \ldots, M$. Without loss of generality, assume that the weights add up to one, i.e., $\sum_{i=1}^{M} \kappa_i = 1$. The M classes are simultaneously served, and each of the n_i flows of class i receives a service rate

$$\frac{\kappa_i c}{\sum_{j=1}^{M} \kappa_j n_j}, \quad i = 1, \ldots, M,$$

see Figure 1.6 for an illustration.

In case of DPS the service rate is, as opposed to GPS, in addition to the weights, determined by the number of flows in the system. Hence, the DPS discipline serves to evaluate the flow-level performance. In case $\kappa_i = \kappa$, $i = 1, \ldots, M$, i.e., if all weights are equal, then DPS reduces to PS. We remark that DPS models are much harder to analyze than PS models.

We argued before that heterogeneous RTTs have an impact on bandwidth sharing realized by TCP. In fact, in TCP the classes with lower RTTs obtain a higher share of the bandwidth [10]. Hence, by setting the DPS class weights inversely proportional to the respective RTTs, DPS is useful to examine the dynamic behavior of TCP flows sharing a common link.

Although PS and DPS can be used for modeling the flow-level performance of bandwidth-sharing protocols in packet-switched networks, they do not explicitly take into account that a flow may require service at several nodes *simultaneously*, meaning that it receives the same rate at each of these nodes. For example, consider the so-called *linear* network depicted in Figure 1.7. This network consists of L nodes and supports $L+1$ classes of users: class-i users require service at node i only, $i = 1, \ldots, L$,

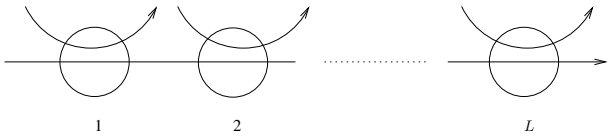

Figure 1.7: A linear bandwidth-sharing network.

whereas class-$(L+1)$ users require service at all L nodes simultaneously.

In the remainder of this section we represent the network as a set of nodes $\mathcal{L} = \{1, \ldots, L\}$, where node $l \in \mathcal{L}$ has finite capacity $c_l > 0$. We distinguish a total number M of classes in the network, where flows of class i use the same route r_i, consisting of a particular nonempty subset of nodes, $i = 1, \ldots, M$. As mentioned above, in case a class requires service at multiple nodes, we assume that users of this class need to be served simultaneously at these nodes. Let $S(l)$ denote the set of classes that require service at node l, $l = 1, \ldots, L$.

Below we discuss two sharing policies that are able to capture the above-mentioned effects, namely: Alpha-Fair Sharing (AFS) and Balanced Fairness (BFS). We note that AFS covers both PS and DPS as special cases. It is remarked that PS is also a special case of BFS.

1.4.3 Alpha-Fair Sharing

When the network is in state $n = (n_1, \ldots, n_M) \in \mathbb{N}_0^M \backslash \{\vec{0}\}$, with n_j denoting the number of class-j users in the network, the AFS service rate x_i^* allocated to each of the class-i users is obtained by solving the following optimization problem [140]:

$$\begin{array}{rl} \max & \sum_{i=1}^{M} n_i U_i(x_i) \\ \text{subject to} & \sum_{i \in S(l)} n_i x_i \leq c_l, \quad l = 1, \ldots, L \\ \text{over} & x_i \geq 0, \quad i = 1, \ldots, M, \end{array} \tag{1.2}$$

where the *utility function* $U_i(x_i)$ is defined by

$$U_i(x_i) = \begin{cases} \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if} \quad \alpha \in (0, \infty) \backslash \{1\}; \\ \kappa_i \log x_i & \text{if} \quad \alpha = 1. \end{cases} \tag{1.3}$$

The κ_is are non-negative weights, and $\alpha \in (0, \infty)$ can be interpreted as a fairness coefficient. The cases $\alpha \to 0$, $\alpha \to 1$ and $\alpha \to \infty$ correspond to allocations which achieve *maximum throughput*, *proportional fairness*, and *max-min fairness*, respectively. In [146] it has also been shown that the case $\alpha = 2$, with additional class weights set inversely proportional to the respective RTTs, provides a reasonable modeling abstraction for the bandwidth sharing realized by TCP. Bandwidth sharing in a

network with AFS is thus evaluated in terms of a utility function, an approach that was first introduced in [89], but we also refer to [88, 107].

Let $s_i(n) := n_i x_i^*$ denote the service rate allocated to class i under an AFS policy when the network is in state n, $i = 1, \ldots, M$. In general no closed-form solution for $s_i(n)$ is known, and one has to obtain it numerically. A few exceptions are known in which there are explicit expressions: linear, grid, and cyclic networks, see [23].

If we consider a single-node network with capacity c, then it can be verified that the solution of the optimization problem (1.2) is given by [14]

$$x_i^* = \frac{\kappa_i^{1/\alpha} c}{\sum_{j=1}^{M} \kappa_j^{1/\alpha} n_j}, \quad i = 1, \ldots, M.$$

In this case we observe that the capacity is shared according to a vector of weights $\kappa_i^{1/\alpha}$, $i = 1, \ldots, M$, in a DPS fashion, i.e., AFS covers DPS (and PS).

As we will argue in Section 1.6, the flow-level performance of AFS networks is hard to analyze in general. To gain insight, a theoretical sharing policy has been constructed [25], known as BFS, which provides performance results that are insensitive to the detailed traffic characteristics. In particular, BFS is constructed in such a way that the performance only depends on the various traffic characteristics through the average *load* of each class, which is defined as the product of the class-dependent arrival rate of flows and the class-dependent average service requirement of a flow, given that flows arrive according to a Poisson process.

1.4.4 Balanced Fairness

Let $\phi_i(n)$ denote the capacity allocated to class i when the network is in state $n \in \mathbb{N}_0^M$, $i = 1, \ldots, M$, under BFS. An allocation is said to be balanced if for all states n, with $n_i, n_j > 0$,

$$\frac{\phi_i(n)}{\phi_i(n - e_j)} = \frac{\phi_j(n)}{\phi_j(n - e_i)}, \quad i, j = 1, \ldots, M, \tag{1.4}$$

where e_i represents the M-dimensional unit vector whose components are equal to 0 except for component i which is equal to 1, $i = 1, \ldots, M$. Define the capacity set

$$\mathcal{C} = \left\{ y \geq 0 : \sum_{i \in S(l)} y_i \leq c_l, \quad l = 1, \ldots, L \right\}.$$

Note that $\phi(n) \in \mathcal{C}$ for all $n \in \mathbb{N}_0^M \backslash \{\vec{0}\}$. There are several allocations that satisfy the balance property (1.4), but there exists a unique allocation such that $\phi(n)$ belongs to the boundary of the capacity set $\mathcal{C}$ in any state $n \in \mathbb{N}_0^M \backslash \{\vec{0}\}$. This allocation is commonly referred to as the BFS allocation.

All balanced service rates can be expressed in terms of a unique balance function $\Phi(\cdot)$, so that $\Phi(0) = 1$ and

$$\phi_i(n) = \frac{\Phi(n - e_i)}{\Phi(n)}, \quad \forall n : n_i > 0, \quad i = 1, \ldots, M.$$

Hence, characterization of $\Phi(n)$ implies that $\phi(n)$ is characterized as well. Define $\Phi(n) = 0$ if $n \notin \mathbb{N}_0^M$. It can be shown [27] that

$$\Phi(n) = \max_{l=1,\ldots,L} \frac{\sum_{k \in S(l)} \Phi(n - e_k)}{c_l}, \quad n \in \mathbb{N}_0^M \backslash \{\vec{0}\},$$

so that $\Phi(\cdot)$ can be obtained recursively. There exist a few networks for which explicit formulae are known for $\Phi(\cdot)$, but in general no explicit expressions are available, which implies that we have to obtain them recursively. In this case it is clear that for large state spaces it is time-consuming to obtain $\Phi(\cdot)$.

As mentioned earlier, the advantage of BFS is that it makes the flow-level analysis tractable, see Section 1.6. Besides being an interesting concept in its own, BFS can also be viewed as an approximation tool. The performance of a network under max-min fairness and proportional fairness is often accurately approximated by that under BFS, see [24].

1.4.5 Extensions

Below we show that the above descriptions of AFS and BFS can be extended.

Access-link rate limitations

So far we assumed that flow rates are constrained by the network links only. In practice this is not true, as the rate of a flow may additionally be constrained by a fixed maximum that represents, for instance, the user's *access-link rate*. These additional constraints can easily be incorporated in utility maximization problems such as (1.2), so that AFS can cope with this. BFS can also be extended such that it covers access-link rate limitations [25].

In Chapter 6 we consider an alternative approach for modeling class-dependent rate limitations in AFS networks. That is, we first determine the AFS allocation (without access-link rate constraints), and then truncate the resulting rates at the access-link rates. It can be verified that these two methods result in different allocations. We prefer the latter approach, as in general this allows fairly explicit analysis, whereas this is considerably harder under the former method, see also [14].

Multi-path routing

We assumed above that each class of users corresponds to a unique route in the network. However, in practice it is likely that some classes of users have multiple

alternative paths through the network. The question that arises is how flows are routed in those cases. Several papers address this question by studying multi-path utility maximization problems, see e.g. [74, 90, 95, 96, 121, 164, 170, 172]. In [22, 84, 108, 109] it was shown how BFS can be applied in networks with multi-path routing. Chapter 8 considers multi-path routing in a simple AFS or BFS network.

1.5 Queues analyzed at the burst-level

In Part I we study communication networks at the burst-level. When analyzing systems at the burst-level, it is often assumed that traffic approximately behaves as a continuous stream of work, thereby neglecting the discrete nature of the small packets, i.e., the traffic behaves as fluid, see Section 1.1.

Although this monograph deals with the performance of bandwidth-sharing disciplines, the first two chapters of Part I, Chapters 2 and 3, do not consider such mechanisms. In particular, in Chapter 2 we study a single-node queue, whereas Chapter 3 analyzes a two-node parallel queue and a two-node tandem queue. We have incorporated these two chapters as we develop techniques there that are extensively used in Chapters 4 and 5, where we study a two-class GPS model. The aim of this section is to give a flavor of the methods used and results derived throughout Part I.

1.5.1 Single-node queue

To gain insight into fluid queues, we first consider a single-node network with capacity c. Let $A := \{A(t), t \in \mathbb{R}\}$ be a continuous-time stochastic process, with $A(0) \equiv 0$. Also, let $A(s,t) := A(t) - A(s)$ denote the amount of traffic entering the system in the interval $[s,t]$, $t > s$. Note that $A(t)$ $(-A(t))$ denotes the amount of traffic generated in the interval $[0,t]$ $([t,0])$ if $t \geq 0$ $(t \leq 0)$. Furthermore, assume that A has *stationary increments*, i.e., $A(t) - A(s)$ has the same distribution as $A(t-s)$ for $t > s \geq 0$. Let $Q(t)$ denote the buffer content at time t of this fluid queue. As a first step to understand this fluid queue, let us consider the discrete-time setting. In particular, consider the buffer content of the queue at time $t = 0$. Let A_{-i} denote the amount of work arriving at the $-i$th epoch, $i = 1, 2, \ldots$. Also, let Q_{-i} denote the buffer content at time $-i$, $i = 0, 1, \ldots$. Then using *Lindley's recursion*, we find

$$\begin{aligned} Q_0 &= \max\{Q_{-1} + A_{-1} - c, 0\} \\ &= \max\{\max\{Q_{-2} + A_{-2} - c, 0\} + A_{-1} - c, 0\} \\ &= \max\{Q_{-2} + A_{-2} + A_{-1} - 2c, A_{-1} - c, 0\} \\ &= \ldots \\ &= \max\left\{Q_{-i} + \sum_{j=1}^{i} A_{-j} - ic, \sum_{j=1}^{i-1} A_{-j} - (i-1)c, \ldots, A_{-1} - c, 0\right\}. \end{aligned}$$

In queueing theory, one is often interested in the behavior of a system after initial effects have vanished, i.e., when the system is in *steady state*. This corresponds to letting $-i \to -\infty$. Therefore, we need to impose assumptions to ensure that the buffer content does not blow up, i.e., we need assumptions for *stability* of the system. First note that, due to the assumption of stationary increments, the mean arrival rate at the queue equals $\mu := \mathbb{E}A_{-1}$. It is then intuitively clear that a sufficient condition for stability of the system is that the mean arrival rate μ is smaller than the capacity c of the system. In case this stability condition is satisfied, there exists a random $-i < 0$ such that $Q_{-i} = 0$, and we thus find that Q_0 converges in distribution to the random variable

$$\sup_{i \geq 0} \left\{ \sum_{j=1}^{i} A_{-j} - ci \right\}. \tag{1.5}$$

Equation (1.5) suggests that $Q(0)$ converges in distribution to

$$Q := \sup_{t \geq 0} \{A(-t, 0) - ct\}, \tag{1.6}$$

given that $\mu < c$. This can be shown to be correct, but it is somewhat harder to prove. Equation (1.6) is often referred to as *Reich's formula* [155]. The steady-state distribution is thus equivalent to the distribution of the supremum of the so-called free process $\{A(-t, 0) - ct,\ t \geq 0\}$. Note that this free process can be negative, but that the supremum of it is always non-negative, as the free process equals 0 at time $t = 0$. This implies that the steady-state distribution lives on $[0, \infty)$.

We remark that if the arrival process A is *time-reversible*, then $\sup_{t \geq 0} \{A(t) - ct\}$ has the same distribution as Q. Also, note that, due to the assumption of stationary increments, $Q(t)$ converges in distribution to Q for all $t \in \mathbb{R}$.

1.5.2 Tandem queue

We proceed with more complex systems: tandem queues. We first analyze the two-node tandem queue, before studying tandem queues with an arbitrary number of nodes. In this fluid queue, traffic is first served with rate c_1 at the first queue, and then immediately sent to the second queue where it is served with rate c_2. To exclude the trivial case where the second buffer is continuously empty, we assume that $c_1 > c_2$.

Let us focus on the steady-state distribution of the content of the two buffers. Clearly, the steady-state first buffer content can be represented as

$$Q_1 = \sup_{t \geq 0} \{A(-t, 0) - c_1 t\},$$

given that $\mu < c_1$. It requires more work to obtain the steady-state buffer content distribution of the second queue. A crucial observation is that the aggregate buffer

content, i.e., the content of the first and second buffer combined, drains with rate c_2, see e.g. [87]. Hence, we find that the steady-state total buffer content can be expressed as

$$Q_{1,2} = \sup_{t\geq 0} \{A(-t,0) - c_2 t\},$$

given that $\mu < c_2$.

It is noted that both Q_1 and $Q_{1,2}$ relate to the same time point ($t = 0$), meaning that $(Q_1, Q_{1,2})$ has the 'correct' joint distribution. We thus find that $Q_2 = Q_{1,2} - Q_1$, i.e., the second buffer content is equal to the total buffer content minus the first buffer content. From the above we conclude that the steady-state second buffer content can be written as

$$Q_2 = \sup_{t\geq 0} \{A(-t,0) - c_2 t\} - \sup_{t\geq 0} \{A(-t,0) - c_1 t\},$$

given that $\mu < c_2$.

The above reasoning also applies in case of an arbitrary number of nodes. Let us assume that we have an L-node tandem queue, with $L > 1$. Furthermore, assume that $c_1 > \ldots > c_L > \mu$ to guarantee that none of the queues is always empty and that the system is stable. Then we find [87] that the steady-state buffer contents can be represented as

$$Q_1 = \sup_{t\geq 0} \{A(-t,0) - c_1 t\};$$

$$Q_l = Q_{1,\ldots,l} - Q_{1,\ldots,l-1} = \sup_{t\geq 0} \{A(-t,0) - c_l t\} - \sup_{t\geq 0} \{A(-t,0) - c_{l-1} t\},$$

for $l = 2, \ldots, L$.

1.5.3 Priority queue

The priority queue is closely related to the tandem queue, as can be seen as follows. Consider a single-node queue with capacity c, where M different classes of users compete for service, each one of them having its own buffer. Let $\{A_i(t), t \in \mathbb{R}\}$ denote the input process with stationary increments of class i, with $A_i(0) \equiv 0$, and let $\mu_i := \mathbb{E}A_i(1)$, $i = 1, \ldots M$. Also, let $A_i(s,t) := A_i(t) - A_i(s)$ for $t > s$.

Assume that lower indexed classes have priority over higher indexed classes, i.e., traffic of class i is only served if no traffic of class j is in the system, $j = 1, \ldots, i-1$. This means that the class-i traffic does not 'see' traffic of class j, $j = i+1, \ldots, M$, at all. The steady-state total buffer content of classes 1 up to m is therefore given by

$$Q_{1,\ldots,m} = \sup_{t\geq 0} \left\{ \sum_{i=1}^{m} A_i(-t,0) - ct \right\}, \quad m = 1, \ldots, M,$$

given that the stability condition $\sum_{i=1}^{m} \mu_i < c$ is satisfied. As $Q_{1,\ldots,m}$, $m = 1, \ldots, M$, all relate to the same time point ($t = 0$), we thus find that $Q_p = Q_{1,\ldots,p} - Q_{1,\ldots,p-1}$, $p = 2, \ldots, M$. It is now straightforward to derive that the steady-state buffer contents can be expressed as

$$Q_1 = \sup_{t \geq 0} \{A_1(-t, 0) - ct\};$$

$$Q_m = \sup_{t \geq 0} \left\{ \sum_{i=1}^{m} A_i(-t, 0) - ct \right\} - \sup_{t \geq 0} \left\{ \sum_{i=1}^{m-1} A_i(-t, 0) - ct \right\}, \quad m = 2, \ldots, M.$$

1.5.4 Generalized Processor Sharing queue

In Chapters 4 and 5 we consider a two-class GPS queue with capacity c, assuming that each class has its own buffer. As before, let $\{A_i(t), t \in \mathbb{R}\}$ denote the external input process of class i and ϕ_i the GPS weight of class i, $i = 1, 2$. Since the GPS discipline is work-conserving, a first observation is that the steady-state total buffer content can be represented as

$$Q_{1,2} = \sup_{t \geq 0} \{A_1(-t, 0) + A_2(-t, 0) - ct\},$$

given that $\mu_1 + \mu_2 < c$. If $S_i(s, t)$ denotes the amount of service received by class-i users in the interval $[s, t]$, $t > s$, then we find that the steady-state ith buffer content can alternatively be written as

$$Q_i = \sup_{s_i \geq 0} \{A_i(-s_i, 0) - S_i(-s_i, 0)\}, \quad i = 1, 2.$$

Clearly, $S_i(-t, 0)$ is strongly affected by $A_j(-t, 0)$ and ϕ_j, $i, j = 1, 2$, and it is therefore hard to give a closed-form expression for $S_i(-t, 0)$, implying that Q_i remains elusive in general. This illustrates why a two-class GPS system is, compared to a priority queue, very hard to analyze. GPS systems with more than two classes will therefore be even harder to analyze. We remark that there exist cases in which Q_i is tractable, as a GPS system is a special case of a coupled processors model, whose solution gives rise to so-called boundary value problems in case of two classes, see e.g. [71, 72].

1.6 Queues analyzed at the flow-level

In Part II we analyze queueing models at the flow-level, meaning that, as opposed to the previous section, we also have to take the random nature of the flows into account. We assume that both the interarrival times of flows and the corresponding service requirements are drawn from some distribution. In this section we present some well-known results for queueing models that are analyzed at the flow-level. We mainly focus on queues where classes share capacity according to DPS, AFS and BFS.

In the remainder of this section we assume that there are M classes of flows that compete for bandwidth in a network. We assume that class-i flows arrive according to a Poisson process of rate λ_i, and have exponentially distributed service requirements with mean ν_i^{-1}, $i = 1, \ldots, M$. Let the average load of class i be denoted by $\rho_i := \lambda_i/\nu_i$. Furthermore, let $N(t) = (N_1(t), \ldots, N_M(t))$ denote the state of the network at time t, with $N_i(t)$ denoting the number of class-i flows.

1.6.1 M/M/1-Discriminatory Processor Sharing queue

We first consider an M-class DPS queue with unit capacity. This implies that $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \nu_i \frac{n_i \kappa_i}{\sum_{j=1}^{M} n_j \kappa_j}, \quad i = 1, \ldots, M,$$

where $n \in \mathbb{N}_0^M$. We assume that the total load of the system is smaller than the available capacity, i.e., $\sum_{i=1}^{M} \rho_i < 1$, so that the process $N(t)$ is stable.

Depending on whether we have equal class weights, i.e., $\kappa_i = \kappa$, $i = 1, \ldots, M$, or unequal class weights, one needs to use different techniques to determine the flow-level performance of the M/M/1-DPS queue, which is illustrated below.

Equal class weights

In case of equal class weights DPS reduces to PS. It can be verified that the transition rates are independent of κ in that case, and that the steady-state distribution of $N(t)$ is given by [16]

$$\pi(n) = \pi(n_1, \ldots, n_M) = \left(1 - \sum_{i=1}^{M} \rho_i\right) \left(\begin{array}{c} \sum_{i=1}^{M} n_i \\ n_1, n_2, \ldots, n_M \end{array} \right) \prod_{i=1}^{M} \rho_i^{n_i}, \quad n \in \mathbb{N}_0^M, \tag{1.7}$$

where we write

$$\left(\begin{array}{c} \sum_{i=1}^{M} n_i \\ n_1, n_2, \ldots, n_M \end{array} \right) = \frac{\left(\sum_{i=1}^{M} n_i\right)!}{\prod_{i=1}^{M} (n_i!)}.$$

Here $\pi(n)$ denotes the fraction of the time that the network is in state n in the long-run. We remark that (1.7) in fact also holds in case of generally distributed service requirements. Using (1.7), we can also determine the mean number of flows of each class in the system:

$$\mathbb{E}N_i = \sum_{n_1=1}^{\infty} \cdots \sum_{n_M=1}^{\infty} n_i \pi(n) = \frac{\rho_i}{1 - \sum_{i=1}^{M} \rho_i}, \quad i = 1, \ldots, M.$$

Exploiting *Little's formula*, we can derive the mean sojourn time of each class:

$$\mathbb{E}S_i = \mathbb{E}N_i/\lambda_i = \frac{1/\nu_i}{1 - \sum_{i=1}^{M} \rho_i}, \quad i = 1, \ldots, M.$$

Unequal class weights

In case of unequal class weights no explicit expression is available for the steady-state distribution of $N(t)$, implying that it is considerably harder to obtain expressions for $\mathbb{E}N_i$ and $\mathbb{E}S_i$, $i = 1, \ldots, M$.

In [154] it was shown that one can obtain the mean number of users of each class by solving the following set of linear equations for $\mathbb{E}N_i$:

$$\mathbb{E}N_i - \lambda \sum_{j=1}^{M} \kappa_j \frac{\frac{\lambda_j}{\lambda}\mathbb{E}N_i + \frac{\lambda_i}{\lambda}\mathbb{E}N_j}{\kappa_j\nu_j + \kappa_i\nu_i} = \rho_i, \quad i = 1, \ldots, M,$$

where $\lambda := \sum_{i=1}^{M} \lambda_i$. In general the expression for $\mathbb{E}N_i$, $i = 1, \ldots, M$, is very cumbersome. Only in case $M = 2$ one can find clean expressions:

$$\mathbb{E}N_1 = \frac{\rho_1}{1-\rho_1-\rho_2}\left(1 + \frac{\nu_1\rho_2\,(\kappa_2 - \kappa_1)}{\kappa_1\nu_1(1-\rho_1) + \kappa_2\nu_2(1-\rho_2)}\right);$$

$$\mathbb{E}N_2 = \frac{\rho_2}{1-\rho_1-\rho_2}\left(1 + \frac{\nu_2\rho_1\,(\kappa_1 - \kappa_2)}{\kappa_1\nu_1(1-\rho_1) + \kappa_2\nu_2(1-\rho_2)}\right).$$

Clearly, using Little's formula, we can obtain an expression for $\mathbb{E}S_i$, $i = 1, \ldots, M$.

1.6.2 Alpha-Fair Sharing networks

We proceed by considering a general network topology where capacity is shared according to AFS, see Section 1.4.3. Now $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \nu_i s_i(n), \quad i = 1, \ldots, M,$$

where $n \in \mathbb{N}_0^M$. In Theorem 1 of [23] it was shown that an AFS network is stable if $\sum_{i \in S(l)} \rho_i < c_l$, $l = 1, \ldots, L$, see also [169, 176] for instance. That is, the network is stable if no individual link is overloaded.

In Section 1.4 we already mentioned that only for linear, grid, and cyclic networks explicit expressions are known for $s_i(n)$, $i = 1, \ldots, M$. Hence, it seems reasonable to expect that only for some of these special networks closed-form expressions can be derived for the corresponding steady-state distribution of $N(t)$, the mean number of users and mean sojourn time of each class. So far only for linear networks and grid networks these have been found [23, 135], given equal node capacities and unweighted proportional fairness ($\alpha = 1$). In fact, in [25] it was shown that the performance of AFS networks is tractable only in case of unweighted proportional fairness in homogeneous hypercubes. That is, even under the simplest Markovian assumptions the performance of most AFS networks has remained elusive so far. Fortunately, the performance of networks becomes much more tractable in case of BFS.

1.6.3 Balanced Fairness Sharing networks

We now focus on networks where capacity is shared according to BFS, see Section 1.4.4. In this case $N(t)$ is a Markov process with transition rates:

$$q(n, n+e_i) = \lambda_i; \quad q(n, n-e_i) = \nu_i \phi_i(n), \quad i = 1, \ldots, M,$$

where $n \in \mathbb{N}_0^M$. In [24] the authors showed that the network is stable under BFS if $\sum_{i \in S(l)} \rho_i < c_l$, $l = 1, \ldots, L$, i.e., the same stability condition as in the previous subsection.

From the balance property (1.4) it may be readily verified that the steady-state distribution of $N(t)$ equals

$$\pi(n) = \pi(n_1, \ldots, n_M) = \frac{1}{G(\rho)} \Phi(n) \prod_{i=1}^{M} \rho_i^{n_i}, \quad n \in \mathbb{N}_0^M, \tag{1.8}$$

where the normalization constant $G(\rho)$ equals

$$G(\rho) = G(\rho_1, \ldots, \rho_M) = \sum_{n_1=0}^{\infty} \ldots \sum_{n_M=0}^{\infty} \Phi(n) \prod_{i=1}^{M} \rho_i^{n_i}.$$

We mention that (1.8) is in fact valid for much more general traffic characteristics. Equation (1.8) is insensitive to all traffic characteristics beyond the traffic loads ρ_is provided that flows are generated within sessions. A session consists of a finite, random number of flows separated by intervals of inactivity referred to as think times. The number of flows per session, flow sizes and think times may have arbitrary distributions, and need not be independent. The only requirement is that sessions arrive as a Poisson process, see [25] for more details. We note that this result is valid for any network topology.

From Little's formula it follows that

$$\mathbb{E}N_i = \rho_i \frac{\frac{\partial G(\rho)}{\partial \rho_i}}{G(\rho)} = \rho_i \frac{\partial \log G(\rho)}{\partial \rho_i}, \quad i = 1, \ldots, M,$$

i.e., characterization of $G(\rho)$ implies that $\mathbb{E}N_i$, $i = 1, \ldots, M$, is known as well. The problem is, however, that it is often extremely hard to derive an explicit expression for $G(\rho)$. In fact, there exist only a couple of network topologies for which $G(\rho)$ is known, viz. linear and tree networks, see [27, 29].

1.7 Literature overview

Below we review the most important results that have appeared on GPS, DPS, AFS, and BFS, respectively. Recall that some elementary results were already presented in Sections 1.5 and 1.6.

1.7.1 Generalized Processor Sharing

We first review the available literature on GPS. Although the selection of GPS weights is, at least from an operational point of view, a key problem, most of the work on GPS describes the queueing performance of a GPS system for fixed weights. Parekh & Gallager [148, 149] performed the first mathematical analysis of GPS. They derived deterministic worst-case delay guarantees for leaky-bucket controlled traffic. This approach was revisited in Pereira *et al.* [153], where a so-called fractal leaky-bucket is used to shape traffic. The authors argued that such a leaky bucket is more effective in case of traffic exhibiting long-range dependence. Again deterministic worst-case delay guarantees are derived. Subsequent papers focused on statistical performance guarantees, such as loss probabilities, delay characteristics and workload distributions. The exact analysis of GPS is in general intractable. Hence, most of the other work on GPS is based on asymptotic approximations. We mention the most important results below.

Yaron & Sidi [175] and Zhang *et al.* [179] derived bounds for GPS queues fed by so-called exponentially-bounded burstiness traffic. Bertsimas *et al.* [20], Massoulié [133], and Zhang [177, 178] established large deviations results for light-tailed traffic, i.e., SRD, sources. So-called large-buffer asymptotics for heavy-tailed traffic, i.e., LRD, processes were obtained in Borst *et al.* [32, 33] and Kotopoulos *et al.* [103]. Van Uitert & Borst [166, 167] extended these results to networks of GPS queues. Borst *et al.* [34, 35] analyzed the buffer asymptotics in a two-class GPS system with a mixture of heavy-tailed and light-tailed traffic. Dębicki & Mandjes [49] and Dębicki & Van Uitert [51] derived the logarithmic large-buffer asymptotics for a two-class GPS system with Gaussian inputs.

Mannersalo & Norros [132] developed accurate approximations for the overflow probabilities in a so-called many-sources asymptotic regime, see Chapter 2. They considered a GPS system shared by two heterogeneous classes of Gaussian sources, with a relatively large number of sources in both classes. The obtained approximations were validated by extensive simulations. Mandjes & Van Uitert [129] further justified and refined these approximations, and established an interesting connection with tandem queues fed by Gaussian traffic, see also Mandjes & Van Uitert [130]. For the special case of Brownian inputs, Mandjes [126] showed the exactness of the resulting decay rates.

Giordano *et al.* [66] proposed a fast simulation approach for the evaluation of the loss probability in a GPS scheduler. The proposed algorithms are based on the large deviations results of Zhang [177].

In the literature hardly any results are available on the *delay* asymptotics in a GPS system. Paschalidis [151] focused on a two-class GPS system, in a discrete-time setting, in which the input traffic was assumed to be SRD, and derived the logarithmic delay asymptotics using known logarithmic large-buffer asymptotics.

As mentioned earlier, the inverse problem of mapping the QoS requirements on suitable GPS weights has received considerably less attention in the literature. Dukkipati *et al.* [56] and Panagakis *et al.* [147] developed algorithms to allocate optimal weights to leaky-bucket constrained traffic with deterministic service guarantees, in the presence of best-effort traffic, i.e., weights are chosen such that the throughput of the best-effort class is maximized. Again for leaky-bucket regulated traffic, Elwalid & Mitra [60] first derived the admissible region for a two-class GPS system for fixed weights (i.e., all combinations of flows that satisfy the QoS for both classes), and then showed that nearly the entire realizable region (i.e., the union of the admissible regions over all possible weight values) is obtained by selecting either one or two specific weights. Further results along these lines may be found in Kumaran *et al.* [105]. Guillemin & Dupuis [71] considered a class with smooth traffic and a class with bursty traffic (in the context of ATM networks), and conducted simulation experiments in which it was shown that the throughput of the bursty traffic class is hardly affected in case the smooth traffic class is prioritized.

We finally mention the thesis of Van Uitert [165], where one can find an excellent overview of performance analysis aspects of GPS.

The above references illustrate that the GPS queue is well studied. However, two important problems have not been been addressed in the literature so far. The first problem is to quantify the performance gain that can be achieved by using GPS instead of priority queueing. The second problem deals with the derivation of the delay asymptotics in case of a continuous-time setting and/or LRD traffic, thus generalizing the results of [151]. The main goal of Part I of this thesis is to solve these two problems.

1.7.2 Discriminatory Processor Sharing

DPS was introduced in Kleinrock [99] as a multi-class extension of PS. The analysis of the DPS discipline is much harder compared to that of the PS discipline, which is reflected by the fact that results for DPS are scarce in literature. O'Donovan [144] was the first to derive expressions for the expected conditional sojourn times, assuming Poisson arrivals and generally distributed service requirements (with finite second moments). The author showed that the expected conditional sojourn times can be found as a solution of a system of integrodifferential equations. Unfortunately, [144] contained an error, and the corrected form of the equations appeared in Fayolle *et al.* [63]. In case of exponentially distributed service requirements, the authors also derived explicit expressions for the expected conditional sojourn times and obtained the expected unconditional sojourn times from a system of linear equations.

In case of exponentially distributed service requirements, Rege & Sengupta [154] derived the moments of the steady-state queue length distribution as the solutions to linear equations. These results were further extended to phase-type distributions in Van Kessel *et al.* [93]. Kim & Kim [98] derived the moments of the sojourn times

as the solutions to linear equations, again in case of exponentially distributed service requirements. Haviv & Van der Wal [78] extended these results (for the mean sojourn times) to phase-type distributions, where the DPS weights may depend both on class and phase.

Fayolle *et al.* [63] also showed that the so-called slowdown in a DPS system approaches the slowdown of the PS system, as the service requirement tends to infinity. This result was strengthened in Avrachenkov *et al.* [13], where the authors derived the asymptotic slope of the expected conditional sojourn time of a class and the asymptotic bias. In [13] it was also shown that the expected queue lengths of all classes are finite under the usual stability condition that the total load is less than the capacity of the node, regardless of the higher moments of the service requirements. We also refer to Borst *et al.* [37] where the authors considered a different asymptotic regime in which DPS exhibits some sort of insensitivity.

Van Kessel *et al.* [94] considered a DPS queue with general service requirements and assumed a so-called time-scale decomposition, implying that the flow dynamics of the various classes occur on separate time scales. The analysis of this limiting regime results in explicit expressions for the queue length distributions.

Grishechkin [68] studied the DPS queue in heavy traffic, assuming finite second moments of the service requirement distributions. Rege & Sengupta [154] derived a heavy-traffic limit theorem, assuming exponentially distributed service requirements. This theorem was extended to phase-type distributions in [93]. Altman *et al.* [10] studied the DPS queue in overload. For results on weight setting and more references on DPS we refer to the survey paper of Altman *et al.* [9].

1.7.3 Alpha-Fair Sharing

AFS networks were first considered by Massoulié & Roberts [135, 136] and are useful for modeling the dynamic interaction among competing elastic flows that traverse several links. Most of the available results deal with the stability of the system. De Veciana *et al.* [169] and Ye [176] showed that, assuming Poisson arrivals and exponentially distributed service requirements, weighted max-min fairness and proportional fairness achieve stability, given that no individual link is overloaded. Bonald & Massoulié [23] extended this result to the class of AFS policies as introduced by Mo & Walrand [140]. Massoulié [134] further generalized it to phase-type service requirement distributions. Bramson [39] showed that max-min fairness guarantees stability of the network in case of general distributed service requirements and renewal arrival processes, given that the above-mentioned stability condition holds. Gromoll & Williams [69, 70] extended this stability result to AFS policies for some special topologies.

While valuable stability results have been established, it is very hard to derive other explicit results, as illustrated in Bonald & Proutière [25]. In fact, even under the

simplest Markovian assumptions the distribution of the number of users in the network has remained intractable in all but a few special networks [23, 135]. The available results therefore are limited to a mean-field approximation of max-min fairness in a homogeneous star network in Fayolle *et al.* [62], a fluid limit analysis of AFS policies in Kang *et al.* [85] and Kelly & Williams [91], and the behavior of AFS networks in overload in Egorova *et al.* [58].

The above illustrates that the flow-level performance of general AFS network topology is not well understood. Although it is known that the performance of AFS networks is often accurately estimated by that of BFS networks, see Bonald *et al.* [24], this is not always useful, as only for a few BFS networks explicit expressions are available for the mean number of users. Therefore, interesting research directions include 1) the development of novel approximations. Furthermore, 2) the case of AFS when a flow may choose between alternative routes or can use several routes simultaneously, so-called *traffic splitting*, remains largely to be evaluated. We finally remark that 3) so far hardly any techniques have been developed to estimate rare event probabilities in AFS networks. In Part II of this thesis we will focus on the three above-mentioned problems.

1.7.4 Balanced Fairness

The notion of BFS was introduced by Bonald & Proutière [25], and was initially applied to wired networks with single-path routing. In [26, 27, 29, 30, 31] the authors derived some approximations and bounds on performance measures for BFS, and discussed various computational aspects.

BFS can also be applied to wired networks with multi-path routing. Similar to the previous subsection, one needs to distinguish between the cases that a flow may choose between alternative routes or can split its traffic over several routes simultaneously. The first case we refer to as load balancing at the flow-level, whereas the second case we refer to as load balancing at the packet-level. Optimal BFS load balancing at the flow-level utilizing local state information was addressed in Bonald *et al.* [22]. Jonckheere & Virtamo [84] showed that one can achieve even better performance if capacity allocation and load balancing are optimized jointly. BFS load balancing at the packet-level was introduced in Leino & Virtamo [108]. A comparison between packet-level and flow-level BFS load balancing was conducted in Leino & Virtamo [109].

1.8 Outline

This thesis consists of two parts. In Part I, consisting of Chapters 2-5, we study bandwidth sharing as a result of explicit packet scheduling in network nodes. In particular, our goal is to analyze a mechanism that can implement differentiated sharing. In contrast, Part II, consisting of Chapters 6-8, considers bandwidth sharing

as a consequence of the rate control by end-users.

As stated in Section 1.2, when analyzing the burst-level performance it is common to model the input processes as fluid. In Part I of this monograph we consider Gaussian input processes, which are a general and versatile class of fluid input processes, covering a broad range of correlation structures, including both SRD and LRD traffic.

In Chapter 2 we first present the machinery that we need to analyze queues with Gaussian input. We also illustrate the use of this machinery for a single queue with Brownian input (a special case of Gaussian input). We determine the joint distribution function of the workloads at two different times, which also allows us to calculate their covariance and exact large-buffer asymptotics. The nature of these asymptotics depends on the model parameters, i.e., there are different regimes. By using sample-path large deviations, these regimes can be interpreted: we explicitly characterize the most likely way for the buffers to fill, given that some large buffer level is reached. This chapter is partly based on [115].

In the next chapter we analyze simple networks of Brownian queues, namely: a two-node parallel queue and two-node tandem queues. Using the methodology of Chapter 2, we derive for both systems the joint distribution function of the workloads of the first and second queue, and obtain their exact large-buffer asymptotics. We show that different regimes of asymptotics exist, and interpret these. It is also shown that one can use similar techniques to derive results for a two-class priority queue. The results on the parallel queue and the tandem queue are published in [114].

Although Chapters 2 and 3 do not have a direct relation to bandwidth sharing at first sight, they are very useful, as we develop techniques there that are extensively used in Chapters 4 and 5, where we consider a two-class GPS queue. In particular, we deal with the two open problems mentioned in Section 1.7.1.

Chapter 4 studies the probability that the virtual delay of a particular class exceeds some threshold. We first derive bounds on this probability for general input processes, and use these to obtain the delay asymptotics in the important case of Gaussian inputs. We show that, depending on the GPS weights, three kinds of asymptotics appear. The results of this chapter are published in [116].

In Chapter 5 we study the choice of optimal GPS weights. In order to do so, we first characterize the admissible region of the system for fixed GPS weights. Then we obtain the realizable region by taking the union of the admissible regions over all possible GPS weights. The results indicate that nearly the entire realizable region can be obtained by strict priority scheduling disciplines, i.e., the results suggest that the weight-setting is not so crucial, and that simple priority strategies may suffice for practical purposes. This chapter is based on [118], while a short version appeared in [120].

In Part II, which consists of the remaining three chapters of this thesis, we analyze models at the flow-level. In Chapters 6-8 it is assumed that bandwidth is shared according to an AFS policy, and we study the three open problems mentioned in

Section 1.7.3. We assume that flows arrive according to a Poisson process, and have exponentially distributed service requirements.

In Chapter 6 we consider a general AFS network topology and focus on the probability that, conditional on the network population being in a given state at time zero, the network is in some set of states after some predefined time. In particular, we assume that the underlying event is rare, so that the corresponding probability is small. As in general no explicit expressions are known for this probability, an attractive approach may be to resort to Monte Carlo (MC) simulation. However, due to the rarity of the event under consideration, 'naive' MC simulation is inefficient. A natural approach to speed up the simulation is to use Importance Sampling (IS). The idea underlying IS is to simulate the system with a new set of input probability distributions, i.e., new interarrival and service time distributions, such that the rare event becomes more likely. We devise an IS scheme to obtain an unbiased estimate of the corresponding probability in a fast way. The results of this chapter appeared in [119].

In the next chapter we analyze a special network: a linear network, again as in Figure 1.7. We first explicitly derive the Laplace-Stieltjes Transform (LST) of the joint workload process at the various nodes in case of unweighted proportional fairness. Next we study the performance of this network by focusing on the mean number of users of each class. In case of unweighted proportional fairness, these can explicitly be derived, whereas in all other cases these are intractable. In this chapter we therefore derive approximations for the mean number of users of each class, by assuming that one or two of the nodes are heavily loaded. In case of a single heavily loaded node we exploit the fact that this node approximately behaves as a two-class DPS model. In case that there are two nodes critically loaded, we rely on the observation that the joint workload process at these nodes is asymptotically independent of the fairness coefficient α, provided all classes have equal weights. This chapter is based on [113].

Chapter 8 considers a stylized network in which, besides classes of users that use specific routes, one class of users can split its traffic over several routes. We consider load balancing at the packet-level, implying that traffic of this class of users can be divided among several routes at the same time. Assuming that load balancing is based on an AFS policy, we show that the network has different types of dynamics. In particular, we show that some classes of users, depending on the state of the network, share capacity according to DPS, whereas each of the remaining classes of users behaves as in a single-class single-node model. We compare the flow-level performance of this network with that of a similar network, where load balancing is based on BFS. We derive explicit expressions for the number of users under BFS, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under AFS. Furthermore, we examine the performance gain that can be achieved by using packet-level load balancing instead of flow-level load balancing. The results of this chapter are partly published in [112].

Part I

Gaussian queues with differentiated bandwidth sharing

Chapter 2

Gaussian methodology

Part I of this thesis is mainly devoted to differentiated bandwidth sharing in network links. In particular, our goal is to study a single-node system that operates under the GPS discipline. As explained in Chapter 1, such systems are typically analyzed at the burst-level. At this time scale traffic that enters the system approximately behaves as a continuous stream of work, i.e., as fluid. We consider a general and versatile class of fluid input processes, viz. the class of *Gaussian inputs*. The present chapter serves as an introductory chapter, in which we present the basic concepts and machinery that are needed in this part. Furthermore, we illustrate the use of this machinery on a simple system. We refer to [127] for an overview on queues with Gaussian input, so-called *Gaussian queues*.

2.1 Preliminaries on Gaussian random variables

Before introducing Gaussian inputs in Section 2.2, we first explain the concept of Normal (or Gaussian) random variables. In addition, we discuss some well-known properties of these variables in this section. Below we present the results that are of importance in Part I.

A Normal random variable X with mean μ and variance σ^2 has density

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

which is denoted by $X \sim N(\mu, \sigma^2)$. The Moment Generating Function (MGF) of X is given by

$$\mathbb{E}e^{sX} = e^{\mu s + \sigma^2 s^2/2}, \quad s \in \mathbb{R}.$$

In the special case of $\mu = 0$ and $\sigma = 1$, we call X *standard Normal*. Throughout Part I, we denote the density function of a standard Normally distributed variable X by

$$\phi(x) := \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}},$$

its distribution function by

$$\Phi(x) := \int_{-\infty}^{x} \phi(y)\mathrm{d}y,$$

and the corresponding tail distribution by $\Psi(x) := 1 - \Phi(x)$. A well-known (double) inequality is (see page 5 of [137]):

$$\frac{1}{x + 1/x}\phi(x) \leq \Psi(x) \leq \frac{1}{x}\phi(x), \quad x > 0. \tag{2.1}$$

It follows that, for $x \to \infty$,

$$\Psi(x) \sim \frac{1}{\sqrt{2\pi}x} e^{-x^2/2} =: \zeta(x), \tag{2.2}$$

where we write $f(x) \sim g(x)$ when $f(x)/g(x) \to 1$ if $x \to \infty$.

A random variable X is d-variate Normally distributed, $d \in \mathbb{N}$, with d-dimensional mean vector μ and (non-singular) $d \times d$ covariance-matrix

$$\Sigma = \begin{pmatrix} \mathbb{V}\mathrm{ar}\, X_1 & \mathbb{C}\mathrm{ov}(X_1, X_2) & \dots & \mathbb{C}\mathrm{ov}(X_1, X_d) \\ \mathbb{C}\mathrm{ov}(X_1, X_2) & \mathbb{V}\mathrm{ar}\, X_2 & \dots & \mathbb{C}\mathrm{ov}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(X_1, X_d) & \mathbb{C}\mathrm{ov}(X_2, X_d) & \dots & \mathbb{V}\mathrm{ar}\, X_d \end{pmatrix}, \tag{2.3}$$

i.e., $X \sim N_d(\mu, \Sigma)$, if X has density

$$\frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)}{2}\right),$$

where $\det(\Sigma)$ is the determinant of the matrix Σ, and Σ^{-1} denotes the inverse of Σ.

Suppose now that (Y, X) is $(q + d)$-variate Normally distributed, where Y is q-dimensional and X is d-dimensional. The mean vector μ and covariance matrix Σ are partitioned as follows:

$$\mu = \begin{pmatrix} \mu_Y \\ \mu_X \end{pmatrix}, \quad \text{with sizes } \begin{pmatrix} q \times 1 \\ d \times 1 \end{pmatrix};$$

$$\Sigma = \begin{pmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{pmatrix}, \quad \text{with sizes } \begin{pmatrix} q \times q & q \times d \\ d \times q & d \times d \end{pmatrix}.$$

Then the random variable $(Y|X = x)$, for some $x \in \mathbb{R}^d$, is Normally distributed with mean vector $\overline{\mu}$ and covariance matrix $\overline{\Sigma}$, where

$$\overline{\mu} = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x - \mu_X), \tag{2.4}$$

and

$$\overline{\Sigma} = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY},$$

i.e., $(Y|X = x) \sim N_q\left(\overline{\mu}, \overline{\Sigma}\right)$. The above indicates that knowing the value of X to be x alters the mean vector and the covariance matrix of Y, as $Y \sim N_q(\mu_Y, \Sigma_{YY})$. It is noted that, as opposed to the conditional mean $\overline{\mu}$, the conditional variance $\overline{\Sigma}$ does not depend on the value x.

2.2 Gaussian input

In the previous section we introduced Normal random variables. Below we explain the connection between these and Gaussian input.

Let $\{A(t), t \in \mathbb{R}\}$ be an input process, with $A(0) \equiv 0$. Also, let $A(s,t) := A(t) - A(s)$ denote the amount of traffic arriving in $[s,t]$, $s < t$. Note that $A(t)$ $(-A(t))$ denotes the amount of traffic generated in the interval $[0,t]$ $([t,0])$ if $t \geq 0$ $(t \leq 0)$. The input process $A(t)$ is called a Gaussian process with stationary increments, if for all $s < t$, $A(s,t)$ is Normally distributed with mean $\mu \cdot (t-s)$ and variance $v(t-s)$, where $\mu := \mathbb{E}A(1)$ and $v(t-s) := \mathbb{V}\mathrm{ar}\,A(s,t)$. Hence, the entire probabilistic behavior of the Gaussian input process can be expressed in terms of a mean traffic rate μ and a variance function $v(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$. The assumption of stationary increments entails that the law of $A(s,t)$ only depends on the *length* of the interval, and not on its position. We also define the *centered* process $\overline{A}(t) := A(t) - \mu t$.

The class of Gaussian inputs is extremely rich, and this richness is best illustrated by the variety of possible choices for the variance function $v(\cdot)$, see Chapter 2 of [127] for more details. The variance function fully determines the correlation structure. To see this, first note that, assuming $0 < s < t$, we have

$$\mathbb{C}\mathrm{ov}(A(s), A(t)) = \mathbb{C}\mathrm{ov}(A(s), A(s) + A(s,t)) = \mathbb{V}\mathrm{ar}\,A(s) + \mathbb{C}\mathrm{ov}(A(s), A(s,t)).$$

Then, using that

$$\mathbb{V}\mathrm{ar}\,A(t) = \mathbb{V}\mathrm{ar}(A(s) + A(s,t)) = \mathbb{V}\mathrm{ar}\,A(s) + \mathbb{V}\mathrm{ar}\,A(s,t) + 2\mathbb{C}\mathrm{ov}(A(s), A(s,t)),$$

we find that

$$\Gamma(s,t) := \mathbb{C}\mathrm{ov}(A(s), A(t)) = \mathbb{C}\mathrm{ov}(\overline{A}(s), \overline{A}(t)) = \frac{1}{2}(v(t) + v(s) - v(t-s)).$$

Throughout Part I we impose the following (weak) assumptions on $v(\cdot)$.

Assumption 2.2.1 *The variance function $v(\cdot)$ satisfies:*
A1 $v(\cdot) \in C_1([0,\infty))$;
A2 For some $\alpha < 2$ it holds that $v(t)t^{-\alpha} \to 0$, as $t \to \infty$;
A3 $v(\cdot)$ is strictly increasing.

We need the first two assumptions to apply certain techniques, which will be defined in Section 2.3. Assumption A3 is needed in the proofs of some lemmas and theorems. It is noted that various measurement studies have confirmed that A1-A3 are natural assumptions.

The class of Gaussian inputs covers a broad range of correlation structures. Importantly, Gaussian models include both SRD and LRD traffic. We say that $A(\cdot)$ exhibits long-range dependence if

$$\sum_{k=1}^{\infty} \mathbb{C}\mathrm{ov}(A(0,1), A(k,k+1)) = \infty,$$

and that $A(\cdot)$ is SRD otherwise ($< \infty$).

Closely related to the notion of LRD, is the notion of *self-similarity*. A process $A(\cdot)$ is self-similar with Hurst parameter H, $H \in (0,1)$, if

$$a^{-H} A(at) \stackrel{d}{=} A(t), \quad \forall a > 0,$$

where $\stackrel{d}{=}$ denotes equality in distribution.

We now mention two examples of Gaussian inputs that are of importance in this monograph, both satisfying A1-A3. We start with a fractional Brownian motion (fBm), which has variance function $v(t) = t^{2H}$, with $H \in (0,1)$, implying that fBm exhibits self-similarity with Hurst parameter H. For $H \in (0,1/2)$ it is easy to show that fBm is LRD, whereas for $H \in (1/2,1)$ fBm is SRD. In the special case of $H = 1/2$, fBm reduces to an ordinary Brownian motion, which has independent increments. This illustrates that the notions of self-similarity and LRD are related in some cases, but not equivalent. Another example of Gaussian inputs is the integrated Ornstein Uhlenbeck (iOU) process, which has variance function $v(t) = t - 1 + e^{-t}$. It is an easy exercise to show that iOU exhibits short-range dependence.

We remark that Gaussian inputs are often useful as approximations of well-known non-Gaussian input processes. We say that an input process $\{\tilde{A}(t), t \in \mathbb{R}\}$ with stationary increments has the *Gaussian counterpart* $\{A(t), t \in \mathbb{R}\}$ if $A(\cdot)$ is Gaussian and furthermore $\mathbb{E}\tilde{A}(t) = \mathbb{E}A(t)$ and $\mathbb{V}\mathrm{ar}\tilde{A}(t) = \mathbb{V}\mathrm{ar}A(t)$ for all t. In other words, $\tilde{A}(\cdot)$ inherits the correlation structure of $A(\cdot)$. Typical examples are the Gaussian counterpart of the Poisson stream, the M/G/∞ input model, and the purely periodic stream. For more details we refer to Section 2.5 of [127].

A natural question that arises is whether Gaussian traffic describes real traffic accurately. Before addressing this question, let us first recall four main characteristics of real traffic. Typically, 1) it can be assumed that the real traffic input process is stationary, at least over suitable time periods. Furthermore, 2) the aggregation level at the core network is usually quite high, as the total input stream to each node in the network consists of a superposition of a large number of individual streams. We already argued in the previous chapter that extensive measurements showed that

3) network traffic exhibits significant positive correlation over a wide range of time scales, and 4) the traffic rate is bursty over a wide range of time scales, i.e., it exhibits extreme irregularity.

Let us now verify whether Gaussian inputs can capture these four properties. Clearly, Gaussian sources have stationary increments, so property 1) is fulfilled. Gaussian traffic arises as limiting process of the superposition of a large number of independent traffic sources, and is thus appropriate if the aggregation level is sufficiently large. In [64] it was empirically shown that a relatively low aggregation level is already sufficient for Gaussianity. A complicating issue is the fact that elastic traffic is controlled through feedback loops like TCP. In [97] it was, however, argued that (non-feedback) Gaussian traffic models are still justified as long as the level of aggregation is sufficiently large (both in time and number of flows), implying that property 2) is satisfied. Properties 3) and 4) are respectively satisfied if the arrival process is LRD and if the traffic rate process behaves irregularly at small time scales (i.e., it could have non-differentiable trajectories). Clearly, not all Gaussian inputs satisfy these last two properties, but for example fBm with $H > 1/2$ is a suitable candidate.

An issue associated with Gaussian traffic is that the cumulative input process will be (locally) *decreasing*, whereas the amount of real traffic generated in some interval cannot be negative. This fact may seem troublesome at first sight, however, similar problems appear in different settings. For example, consider the situation where the number of successes in n Bernoulli trials is approximated by a Gaussian random variable for large n. In this case the real distribution of the number of successes is also closely approximated by a Normal distribution, which has $\mathbb{R}$, i.e., also negative values, as support. Although the cumulative input process will be (locally) decreasing, this occurs with a small probability, as the cumulative input process typically has a positive drift. In addition, as we will see in Section 2.4, the steady-state buffer content of a Gaussian queue can always be evaluated and lives on $[0, \infty)$.

Above we provided qualitative arguments suggesting that Gaussian inputs can describe real network traffic accurately. For more validation and justification of this claim we refer to [138] and Chapter 3 of [127].

2.3 Large deviations for Gaussian processes

In this section we consider *large deviation* results for Gaussian processes. As we will see below, large deviations are closely related to *rare events*. To explain the concept of large deviations in general, we start with a number of results relating to a finite-dimensional setting. Next we consider the infinite-dimensional framework, which is the one corresponding to Gaussian processes, and present the theorems that are of interest in this monograph. This section is based on Chapter 4 in [127].

2.3.1 Finite-dimensional framework: Cramér's theorem

Consider a sequence of i.i.d. random variables $X_1, \ldots, X_n$ that are distributed like a random variable X, which has mean $\mu := \mathbb{E}X$, with $-\infty < \mu < \infty$. The law of large numbers states that the sample mean $n^{-1}\sum_{i=1}^n X_i$ converges to μ almost surely as $n \to \infty$. Let us now focus on the probability that, although $n \to \infty$, this sample mean *does* deviate severely from μ. Below we wish to analyze:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i > x\right),$$

for $x > \mu$, where n and x are fixed.

Define the MGF of X by $M(\theta) := \mathbb{E}e^{\theta X}$, and assume that the MGF is finite in a neighborhood of 0, so that all moments of X are finite. It is now straightforward to show that

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i > x\right) &= \mathbb{P}\left(e^{\theta\sum_{i=1}^n X_i} > e^{n\theta x}\right) \\ &\leq e^{-n\theta x}\mathbb{E}e^{\theta\sum_{i=1}^n X_i} = e^{-n\theta x}(M(\theta))^n,\end{aligned} \tag{2.5}$$

for any $\theta \geq 0$, where we use the *Markov inequality*: $\mathbb{P}(Y \geq y) \leq \mathbb{E}Y/y$ for any non-negative random variable Y, where $\mathbb{E}Y < \infty$. As (2.5) holds for any $\theta \geq 0$, it also holds for the θ that gives the tightest upper bound:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n X_i > x\right) \leq \inf_{\theta\geq 0} e^{-n\theta x}(M(\theta))^n = \exp\left(-n\sup_{\theta\geq 0}\left(\theta x - \log M(\theta)\right)\right). \tag{2.6}$$

Equation (2.6) is known as the *Chernoff bound* and shows that the probability that the sample mean exceeds μ decays exponentially as n increases, i.e., the *decay rate*, or equivalently, the *rate function* equals:

$$J(x) := \sup_{\theta\geq 0}\left(\theta x - \log M(\theta)\right),$$

where $J(x)$ is referred to as the *Fenchel-Legendre transform* of $\log M(\theta)$. Here $J(x)$ can be interpreted as a cost function: the larger the distance to the mean μ is, the higher the cost are. It is an easy exercise to show that $J(x) > 0$ if $x \neq \mu$, $J(\mu) = 0$, and $J(\cdot)$ is convex, see Exercise 4.1.1 in [127].

In turns out that the Chernoff bound is tight on a logarithmic scale. Before we state this result, known as Cramér's theorem [52], we first need the following definition.

Definition 2.3.1 *A sequence $Y_1, Y_2, \ldots$ obeys the large deviations principle (LDP) with rate function $K(\cdot)$ if:*

(a) For any closed set F,

$$\limsup_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i \in F\right) \leq -\inf_{x\in F} K(x);$$

(b) For any open set G,

$$\liminf_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i \in G\right) \geq -\inf_{x\in G} K(x).$$

Theorem 2.3.2 [Cramér] *Let $X_i \in \mathbb{R}$ be i.i.d. random variables, distributed as a random variable X with mean μ and MGF $M(\theta) = \mathbb{E}e^{\theta X}$ that is finite in a neighborhood of 0. Then $X_1, X_2, \ldots$ obeys the LDP with rate function $J(\cdot)$.*

Using Theorem 2.3.2 and the fact that $J(\cdot)$ is convex, it can be proved that

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > x\right) = -J(x). \tag{2.7}$$

Hence, Cramér's theorem gives information on the *logarithm* of the probability, rather than the probability itself. From (2.7) we conclude that

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > x\right) = f(x,n)e^{-nJ(x)},$$

where $f(x,n)$ is not given explicitly, but known to be *subexponential*, i.e.,

$$\lim_{n\to\infty} \frac{\log f(x,n)}{n} = 0.$$

In the absence of an explicit expression for $f(x,n)$, one may use the approximation

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i > x\right) \approx e^{-nJ(x)}. \tag{2.8}$$

We remark that in some cases this approximation may be inaccurate, as some polynomial function n^α, where α can be both positive and negative, or a function of the type $\exp\left(n^{1-\epsilon}\right)$, where ϵ is a small positive number, can be part of $f(x,n)$. However, often this approximation is useful to gain insight.

Let us now consider the probability that $n^{-1}\sum_{i=1}^{n} X_i$ is contained in some set B. Then we find the approximation

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \in B\right) \approx e^{-n\inf_{x\in B} J(x)}.$$

That is, the rate function is determined by the most likely point in the set B, i.e., the point $x^* := \arg\inf_{x\in B} J(x)$. Clearly, if $\mu \in B$ then $x^* = \mu$ and $\inf_{x\in B} J(x) = 0$.

As may be expected, Theorem 2.3.2 can also be extended to a multivariate, say d-dimensional with $d \in \mathbb{N}$, version. Let $\langle a, b\rangle$ denote the inner product $\sum_{i=1}^{d} a_i b_i$.

Theorem 2.3.3 [Multivariate Cramér] *Let $X_i \in \mathbb{R}^d$ be i.i.d. d-dimensional random variables, distributed as a random variable X with mean μ and MGF $M(\theta) = \mathbb{E}e^{\langle\theta,X\rangle}$ that is finite in a neighborhood of 0. Then the sequence $X_1, X_2, \ldots$ obeys the LDP with rate function $J_d(\cdot)$, where*

$$J_d(x) := \sup_{\theta\in\mathbb{R}^d} (\langle\theta, x\rangle - \log M(\theta)). \tag{2.9}$$

Considering the specific case that X is d-dimensional Normally distributed with mean vector μ and non-singular covariance matrix Σ, see Section 2.1, we find that $\log M(\theta) = \log \mathbb{E}e^{\langle\theta,X\rangle} = \langle\theta, \mu\rangle + \frac{1}{2}\theta^{\mathrm{T}}\Sigma\theta$. Consequently, with $(x-\mu)^{\mathrm{T}} := (x_1 - \mu_1, \ldots, x_d - \mu_d)$, we deduce that

$$\theta^* = \Sigma^{-1}(x-\mu) \quad \text{and} \quad J_d(x) = \frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu),$$

where θ^* denotes the optimizer in (2.9). The following theorem follows directly from the above, and will be used in Chapter 3. We refer to Exercise 4.1.9 in [127] for more details.

Theorem 2.3.4 *Let $(X, Y) \sim N_2(0,\Sigma)$, for a non-degenerate 2-dimensional covariance-matrix Σ. Then,*

$$\begin{array}{ll} (i) & -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \geq x\right) = \frac{1}{2}x^2/(\Sigma_{XX})^2; \\ (ii) & -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i \geq x, \frac{1}{n}\sum_{i=1}^{n} Y_i \geq y\right) = \inf_{a\geq x}\inf_{b\geq y} \Lambda(a,b), \end{array}$$

where $\Lambda(a,b) := \frac{1}{2}\begin{pmatrix} a & b \end{pmatrix}\Sigma^{-1}\begin{pmatrix} a \\ b \end{pmatrix}$ *and* $x, y > 0$.

2.3.2 Infinite-dimensional framework: Schilder's theorem

Below we present an extension of Cramér's theorem that relates to an infinitely-dimensional setting: the generalized version of *Schilder's theorem* [15]. Whereas 'Cramer' can be applied to describe the likelihood of a sample mean of Normal random variables or vectors attaining a rare value, 'Schilder' describes the large deviations of the sample mean of Gaussian processes.

Let $A_1(\cdot), A_2(\cdot), \ldots$ be a sequence of i.i.d. Gaussian processes, distributed as a Gaussian process with variance function $v(\cdot)$. For large values of n it is clear that the sample mean path $n^{-1}\sum_{i=1}^{n} A_i(t)$ approaches μt almost surely, where $\mu := \mathbb{E}A_1(1)$.

'Schilder' can be applied to determine the probability that the sample mean path deviates from some mean path. In particular, it characterizes the exponential decay rate of the sample mean path being contained in some specific set.

We continue with a description of the framework of Schilder's sample-path large deviations principle (LDP) (see [15], and also Theorem 1.3.27 of [53] for a more detailed treatment). Below we assume that the processes $A_1(\cdot), A_2(\cdot), \ldots$ are centered, but it is clear that the results for centered processes can be translated immediately into results for non-centered processes. Define the path space Ω as

$$\Omega := \left\{ \omega : \mathbb{R} \to \mathbb{R}, \ \text{ continuous}, \ \ \omega(0) = 0, \ \lim_{t\to\infty} \frac{\omega(t)}{1+|t|} = \lim_{t\to-\infty} \frac{\omega(t)}{1+|t|} = 0 \right\}, \quad (2.10)$$

which is a separable Banach space by imposing the norm

$$||\omega||_\Omega := \sup_{t\in\mathbb{R}} \frac{|\omega(t)|}{1+|t|}.$$

We note that in [7] it was pointed out that $A_i(\cdot)$ can be realized on Ω under Assumption A2. Then one can construct a *reproducing kernel Hilbert space* $R \subseteq \Omega$, consisting of elements that are roughly as smooth as the covariance function $\Gamma(s,\cdot)$; for details, see [8]. We start from a 'smaller' space R^*, defined by

$$R^* := \left\{ \omega : \mathbb{R} \to \mathbb{R}, \ \ \omega(\cdot) = \sum_{i=1}^{n} a_i \Gamma(s_i, \cdot), \ \ a_i, s_i \in \mathbb{R}, n \in \mathbb{N} \right\}.$$

The inner product on this space R^* is, for ω_a, $\omega_b \in R^*$, defined as

$$\langle \omega_a, \omega_b \rangle_R := \left\langle \sum_{i=1}^{n} a_i \Gamma(s_i,\cdot), \sum_{j=1}^{n} b_j \Gamma(s_j,\cdot) \right\rangle_R = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i b_j \Gamma(s_i, s_j); \quad (2.11)$$

notice that this implies $\langle \Gamma(s,\cdot), \Gamma(\cdot,t)\rangle_R = \Gamma(s,t)$. This inner product has the following useful property, which is known as the *reproducing kernel* property,

$$\omega(t) = \sum_{i=1}^{n} a_i \Gamma(s_i,t) = \left\langle \sum_{i=1}^{n} a_i \Gamma(s_i,\cdot), \Gamma(t,\cdot) \right\rangle_R = \langle \omega(\cdot), \Gamma(t,\cdot)\rangle_R.$$

From this we introduce the norm $||\omega||_R := \sqrt{\langle \omega, \omega \rangle_R}$. The closure of R^* under this norm is defined as space R. Now we can define the rate function:

$$I(\omega) := \begin{cases} \frac{1}{2}||\omega||_R^2 & \text{if} \quad \omega \in R; \\ \infty & \text{otherwise.} \end{cases} \quad (2.12)$$

Theorem 2.3.5 [Generalized Schilder] *Let $A_i(\cdot) \in \Omega$ be i.i.d. centered Gaussian processes, with variance function $v(\cdot)$ satisfying Assumptions A1 and A2. Then the sequence $A_1(\cdot), A_2(\cdot), \ldots$ obeys the LDP with rate function $I(\cdot)$.*

Recall that an LDP consists of an upper and lower bound, which apply to closed and open sets, respectively. We will use Theorem 2.3.5 for certain open sets (to be defined in the next chapters). It can be verified that these sets U are such that

$$\inf_{\omega \in U} I(\omega) = \inf_{\omega \in \overline{U}} I(\omega),$$

where $\overline{U}$ is the closure of U. The way to prove this is to show that an arbitrarily chosen path in $\overline{U}$ can be approximated by a path in U, see [142] and Appendix A of [130].

Now consider the probability that the sample mean path of n i.d.d. Gaussian processes is contained in some set of paths E. Then 'Schilder' yields the approximation

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} A_i(\cdot) \in E\right) \approx \exp\left(-n \inf_{f \in E} I(f)\right). \tag{2.13}$$

Hence, the decay rate is dominated by the path in the set E that minimizes $I(f)$, i.e., the path $f^* = \arg\inf_{f \in E} I(f)$. We refer to f^* as the *most probable path* (MPP), as the decay rate of (2.13) is fully determined by the likelihood of this most likely path in E. That is, given that the sample mean path is contained in the set E, with overwhelming probability this happens by a path that is close to f^*.

A problem that arises is that, as we saw above, there is only an explicit expression for $I(f)$ available if f corresponds to a linear combination of covariance functions. Another difficulty is that the optimization should be performed over all paths $f \in E$, which are infinitely dimensional objects. Nevertheless, if we find such a minimizing path f^*, then this is useful in order to gain insight into the dynamics of a problem. In Section 2.5 we explicitly derive the MPPs in a simple system.

There exists also a version of Schilder's theorem relating to multi-dimensional Gaussian processes. In particular, we will use the framework that corresponds to two-dimensional Gaussian processes in Chapters 3 and 4. The formulation of this framework is nearly identical to the above, but more involved, and is therefore left out.

2.4 Gaussian queues

Consider the process $\{A(t) - ct, t \geq 0\}$, where $A(t)$ is a Gaussian process and $c > 0$ is a scalar. The reflection of this process at zero is referred to as a Gaussian queue. Due to the stationary increments, it is clear that a sufficient condition for stability of this system is that $\mu < c$. In Chapter 1 we already argued that the steady-state

buffer content of such a queue can be represented as

$$Q := \sup_{t\geq 0}\{A(-t,0) - ct\},$$

given that this stability condition is satisfied.

As mentioned before, an inherent conceptual problem of Gaussian queues is that the input process can be negative. However, irrespective of whether $A(t)$ corresponds to negative traffic or not, Q can always be evaluated and lives on $[0,\infty)$.

We remark that Gaussian queues are in general hard to analyze. In particular, only the cases of the Brownian motion and the Brownian bridge (that is, a standard Brownian motion conditioned on $B(1) = 0$) result in explicit expressions for the steady-state buffer content distribution, see Section 2.5. To gain insight, one often resorts to either approximations or asymptotics.

2.4.1 Approximation

As the steady-state buffer content distribution of Gaussian queues is intractable in general, this has motivated the derivation of approximations for the situation of a general correlation structure. Let us focus on the overflow probability $\mathbb{P}(Q > b)$, with $b \geq 0$. In e.g. [65, 127] the following approximation was suggested:

$$\mathbb{P}(Q > b) \approx \exp\left(-\inf_{t\geq 0}\frac{(b+(c-\mu)t)^2}{2v(t)}\right).$$

The above approximation is obtained by using that

$$\mathbb{P}(Q > b) = \mathbb{P}\left(\sup_{t\geq 0}\{A(-t,0) - ct\} > b\right) \approx \sup_{t\geq 0}\mathbb{P}(A(-t,0) > b + ct),$$

realizing that $A(-t,0) \sim N(\mu t, v(t))$, and subsequently applying the Chernoff bound. It is noted that the analysis of Chapter 5 is based on this approximation. Interestingly, it turns out that the approximation is exact for the Brownian motion and the Brownian bridge, see Example 5.4.2 in [127].

2.4.2 Asymptotics

The relevance of asymptotics can best be illustrated by considering two examples of interest. We already mentioned in Chapter 1 that both packet losses (due to buffer overflow) and packet delay strongly determine the QoS as perceived by users. Particularly for data applications, the loss is only allowed to exceed some specific value with extremely small probability. Hence, the (exponential) decay rate of the loss probability is an important performance measure, as it can be used to approximate the loss probability. Similarly, for most real-time applications the delay can only

exceed some specific threshold with extremely small probability, implying that the decay rate of the delay probability is also a useful measure.

Asymptotics may clearly serve as approximations of the probabilities of interest, and they have the important additional advantage that they often provide useful qualitative insights, while they remain computationally tractable. Recall that asymptotics are closely connected to the probabilities of rare events. Typically, the most likely way for a rare event to occur is fairly simple, and can be directly deduced from the results, as will be illustrated in the next section.

Two types of asymptotics are widely used, namely: *large-buffer asymptotics* and *many-sources asymptotics*. Within each of these two regimes, we also distinguish between *logarithmic* and *exact* asymptotics. Below we briefly discuss each of the four cases of asymptotics. As a side remark we mention that in practice most (real-time) applications do not tolerate large delays, hence the large-buffer asymptotics are not always appropriate. It can be argued that in those situations the many-sources asymptotic regime is more justified.

Logarithmic large-buffer asymptotics

In order to find the logarithmic large-buffer asymptotics, we need to derive a function $f_1(b) \in \mathbb{R}_+$, such that

$$\log \mathbb{P}(Q > b) \sim -f_1(b), \quad b \to \infty, \tag{2.14}$$

i.e., we need to find the decay rate.

Exact large-buffer asymptotics

In case the logarithmic large-buffer asymptotics are characterized, i.e., $f_1(b)$ is known, it follows from (2.14) that

$$\mathbb{P}(Q > b) \sim g_1(b) e^{-f_1(b)}, \quad b \to \infty,$$

where the function $g_1(b)$ is such that

$$\lim_{b\to\infty} \frac{\log g_1(b)}{f_1(b)} = 0.$$

If the functions $f_1(b)$ and $g_1(b)$ are both explicitly found, then we say that one has determined the exact large-buffer asymptotics. It is clear that the exact asymptotics are more refined than the logarithmic asymptotics, i.e., if the exact asymptotics are known, then they effectively also yield the logarithmic asymptotics. Exact asymptotics are often considerably harder to obtain though.

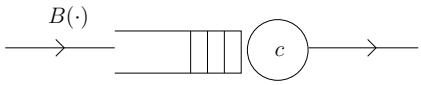

Figure 2.1: Single-node queue

Logarithmic many-sources asymptotics

In order to find the logarithmic many-sources asymptotics, we need to derive a function $f_2(b) \in \mathbb{R}_+$, such that

$$\frac{1}{n} \log \mathbb{P}(Q_n > nb) \sim -f_2(b), \quad n \to \infty, \tag{2.15}$$

where Q_n denotes the steady-state buffer content of the queue under consideration in a system with n i.i.d. inputs, and where the link capacity is also scaled by n, i.e., $c \leftarrow nc$. We note that the decay rates $f_1(b)$ and $f_2(b)$ are not necessarily equal.

Exact many-sources asymptotics

From (2.15) it follows that

$$\mathbb{P}(Q_n > nb) \sim g_2(b,n) e^{-nf_2(b)}, \quad n \to \infty,$$

where $g_2(b,n)$ is typically not calculated, but known to be a subexponential function. In case both $f_2(b)$ and $g_2(b,n)$ are explicitly known, one has found the exact many-sources asymptotics. Again, the exact asymptotics are more refined than the logarithmic asymptotics.

2.5 Brownian queues

In the remainder of this chapter we consider the reflection of the process $\{B(t)-ct, t \geq 0\}$, where $B(t)$ is a standard Brownian motion (with $B(0) \equiv 0$), denoting the amount of traffic entering the system in the interval $[0,t]$, and $c > 0$ is the service capacity of the node. The reflection of $\{B(t) - ct, t \geq 0\}$ at zero is referred to as a *Brownian queue*, which is a special kind of Gaussian queue, see Figure 2.1 for an illustration. Ordinary Brownian input plays an important role in this monograph, as the use of Brownian input often results in explicit expressions for performance measures, thereby providing valuable insight. Brownian motions can be used to approximate weakly-dependent traffic streams, cf. also the celebrated 'Central Limit Theorem in functional form'. Its mean and variance function are characterized through $\mu = 0$ and $v(t) = t$, respectively. It can be verified that $\Gamma(s,t) = \mathbb{C}\mathrm{ov}(B(s), B(t)) = \min\{|s|, |t|\}$ if $s, t \geq 0$ or $s, t < 0$, and $\Gamma(s,t) = 0$ otherwise. Let $B(s,t) = B(t) - B(s)$ denote the amount of traffic generated in the interval $[s,t]$, $s < t$. The goal of this section is to show how

the machinery that was presented earlier in this chapter can be used. We remark that some of the results that are derived below are already known, as the Brownian queue has been well-studied in the past, see e.g. [2, 3, 4, 75]. However, those results were obtained in a completely different and perhaps less transparent manner.

2.5.1 Useful properties

We already mentioned that the steady-state buffer content distribution of a Brownian queue is tractable. In fact, in [156] it was shown that it is exponentially distributed with mean $1/(2c)$. That is,

$$\begin{aligned}\mathbb{P}(Q \leq b) &= \mathbb{P}\left(\sup_{t\geq 0}\{B(-t,0) - ct\} \leq b\right) = \mathbb{P}\left(\sup_{t\geq 0}\{B(0,t) - ct\} \leq b\right) \\ &= \mathbb{P}\left(\forall t \geq 0 : B(t) \leq b + ct\right) = 1 - e^{-2bc}, \end{aligned} \tag{2.16}$$

with $b, c \geq 0$, i.e., the probability that a standard Brownian motion stays below the function $b + ct$ for all $t \geq 0$, equals $1 - \exp(-2bc)$.

Another useful property is that [125]

$$\mathbb{P}(\forall t \in [0,1] : B(t) \leq b + ct | B(1) = 0) = 1 - e^{-2b(b+c)}, \tag{2.17}$$

with $b, c \geq 0$, i.e., the probability that a Brownian bridge stays below the function $b + ct$, for all $t \in [0, 1]$, equals $1 - \exp(-2b(b + c))$. We can exploit (2.17) to derive that [125]

$$\begin{aligned}&\mathbb{P}\left(\forall t \in [0,u] : B(t) \leq b + ct | B(u) = x\right) \\ &\quad= \mathbb{P}\left(\forall s \in [0,1] : B(su) \leq b + csu | B(u) = x\right) \\ &\quad= \mathbb{P}\left(\forall s \in [0,1] : \sqrt{u}B(s) \leq b + csu | \sqrt{u}B(1) = x\right) \\ &\quad= \mathbb{P}\left(\forall s \in [0,1] : B(s) \leq \frac{b}{\sqrt{u}} + cs\sqrt{u} | B(1) = \frac{x}{\sqrt{u}}\right) \\ &\quad= \mathbb{P}\left(\forall s \in [0,1] : B(s) \leq \frac{b}{\sqrt{u}} + cs\sqrt{u} - \frac{sx}{\sqrt{u}} \middle| B(1) = 0\right), \\ &\quad= \mathbb{P}\left(\forall s \in [0,1] : B(s) \leq \frac{b}{\sqrt{u}} + \left(c\sqrt{u} - \frac{x}{\sqrt{u}}\right) s \middle| B(1) = 0\right) \\ &\quad= 1 - e^{-2\frac{b}{\sqrt{u}}\left(\frac{b}{\sqrt{u}} + c\sqrt{u} - \frac{x}{\sqrt{u}}\right)}, \end{aligned} \tag{2.18}$$

with $b, c, u \geq 0$ and $x \in [0, b + cu]$, where we use that a standard Brownian motion is self-similar with $H = 1/2$. The above results can also easily be extended to general Brownian input, with drift $\mu > 0$ and variance $v(t) = \lambda t$, $\lambda > 0$, as we will see below and in the next chapters.

The rate function given in Equation (2.12) simplifies considerably in case of standard Brownian input. Using (2.11) and the definition of $\Gamma(s,t)$ for standard Brownian input, it can be shown (see Theorem 5.2.3 of [52]) that (2.12) is equivalent to

$$I(\omega) = \begin{cases} \frac{1}{2}\int_{-\infty}^{\infty}(\omega'(t))^2 \mathrm{d}t & \text{if} \quad \omega \in R; \\ \infty & \text{otherwise.} \end{cases} \tag{2.19}$$

In the remainder of this section we analyze the transient behavior of a Brownian queue. In particular, we explicitly derive the joint distribution function

$$p(\overline{b},T) := \mathbb{P}(Q_0 > b_0, Q_T > b_T),$$

where $b_0, b_T \geq 0$, $\overline{b} = (b_0, b_T)$, $T > 0$, and Q_t denotes the workload at time $t \geq 0$, assuming that the workload process is in stationarity at $t = 0$. This also allows us to explicitly calculate the covariance of Q_0 and Q_T. By setting $b_0 = b$, $b_T = \alpha b$, and $T = \gamma b$, with $\alpha, \gamma \geq 0$, and letting $b \to \infty$, we also obtain the exact large-buffer asymptotics, i.e., we find a function $f(\cdot)$ such that $\mathbb{P}(Q_0 > b, Q_{\gamma b} > \gamma b)/f(b) \to 1$ as $b \to \infty$. It turns out that the nature of the asymptotics depends on the value of α, γ, and the service rate c of the queue, i.e., there are various regimes. These regimes can be further interpreted relying on Schilder's sample-path large deviations theorem. In particular, we obtain the MPP, i.e., the most likely way for the buffer to fill.

2.5.2 Joint distribution function

In this subsection we derive a closed-form expression for $p(\overline{b},T)$. It turns out that it is easier to first calculate $\overline{p}(\overline{b},T) := \mathbb{P}(Q_0 \leq b_0, Q_T \leq b_T)$. Recall that $\Phi(\cdot)$ denotes the distribution function of a standard Normal random variable. According to Reich's formula [155],

$$Q_0 = \sup_{t\geq 0}\{B(-t,0) - ct\} \quad \text{and} \quad Q_T = \sup_{s\geq 0}\{B(T-s,T) - cs\}. \tag{2.20}$$

Hence, we find that

$$\begin{aligned} \overline{p}(\overline{b},T) &= \mathbb{P}\left(\sup_{t\geq 0}\{B(-t,0) - ct\} \leq b_0, \sup_{s\geq 0}\{B(T-s,T) - cs\} \leq b_T\right) \\ &= \mathbb{P}(\forall s,t \geq 0 : B(-t,0) \leq b_0 + ct, B(T-s,T) \leq b_T + cs) \\ &= \mathbb{P}(\forall s,t \geq 0 : B(T,t+T) \leq b_0 + ct, B(0,s) \leq b_T + cs), \end{aligned}$$

where the last line is obtained by using time reversibility of Brownian motion. Now, conditioning on the value of $B(0,T)$, we obtain that

$$\begin{aligned} \overline{p}(\overline{b},T) = \int_{-\infty}^{b_T+cT} & \mathbb{P}(\forall s \in [0,T) : B(0,s) \leq b_T + cs | B(0,T) = x) \times \\ & \mathbb{P}(\forall t \geq 0 : \forall s \geq T : B(T,t+T) \leq b_0 + ct, B(0,s) \leq b_T + cs | B(0,T) = x) \\ & \mathrm{d}\mathbb{P}(N(0,T) \leq x). \end{aligned}$$

Let us first focus on the first term in the above integral. Using (2.18), we obtain that

$$\begin{aligned}\mathbb{P}(\forall s \in [0,T) : B(0,s) \leq b_T + cs | B(0,T) = x) \\ = 1 - \exp\left(-2b_T c - 2b_T(b_T - x)/T\right).\end{aligned} \tag{2.21}$$

Proceeding with the second term in the integral, we find that

$$\begin{aligned}&\mathbb{P}(\forall t \geq 0 : \forall s \geq T : B(T,T+t) \leq b_0 + ct, B(0,s) \leq b_T + cs | B(0,T) = x) \\ &= \mathbb{P}(\forall t \geq 0 : \forall s \geq T : B(T,T+t) \leq b_0 + ct, B(T,s) \leq b_T + cs - x) \\ &= \mathbb{P}(\forall s,t \geq 0 : B(T,T+t) \leq b_0 + ct, B(T,T+s) \leq b_T + (s+T)c - x) \\ &= \mathbb{P}(\forall s,t \geq 0 : B(0,t) \leq b_0 + ct, B(0,s) \leq b_T + (s+T)c - x) \\ &= \mathbb{P}(\forall t \geq 0 : B(0,t) \leq \min\{b_0, b_T + cT - x\} + ct).\end{aligned}$$

Exploiting (2.16), we deduce that

$$\begin{aligned}&\mathbb{P}(\forall t \geq 0 : \forall s \geq T : B(T,T+t) \leq b_0 + ct, B(0,s) \leq b_T + cs | B(0,T) = x) \\ &= \mathbb{P}(\forall t \geq 0 : B(0,t) \leq \min\{b_0, b_T + cT - x\} + ct) \\ &= \begin{cases} 1 - \exp(-2b_0 c) & \text{if } x \leq b_T + cT - b_0; \\ 1 - \exp(-2(b_T + cT - x)c) & \text{if } x > b_T + cT - b_0. \end{cases}\end{aligned} \tag{2.22}$$

Theorem 2.5.1 *For each $b_0, b_T, T \geq 0$,*

$$\begin{aligned}p(\overline{b},T) \quad = \quad & -\Phi\left(k_1(\overline{b},T)\right) + e^{-2b_T c}\Phi\left(k_2(\overline{b},T)\right) + \\ & e^{-2b_0 c}\Phi\left(k_3(\overline{b},T)\right) + e^{-2(b_0+b_T)c}\Phi\left(k_4(\overline{b},T)\right),\end{aligned}$$

where

$$k_1(\overline{b},T) = \frac{-b_T - cT - b_0}{\sqrt{T}}; \quad k_2(\overline{b},T) = \frac{b_T - cT - b_0}{\sqrt{T}};$$

$$k_3(\overline{b},T) = \frac{-b_T - cT + b_0}{\sqrt{T}}; \quad k_4(\overline{b},T) = \frac{-b_T + cT - b_0}{\sqrt{T}}.$$

Proof: Using (2.21) and (2.22), we obtain that $\overline{p}(\overline{b},T)$ equals

$$\begin{aligned}&\int_{-\infty}^{b_T+cT-b_0} \left(1 - \exp\left(-2b_T c - 2\frac{b_T(b_T-x)}{T}\right)\right) \times \\ &\qquad (1 - \exp(-2b_0 c))\, \mathrm{d}\mathbb{P}(N(0,T) \leq x) \\ &\qquad + \\ &\int_{b_T+cT-b_0}^{b_T+cT} \left(1 - \exp\left(-2b_T c - 2\frac{b_T(b_T-x)}{T}\right)\right) \times \\ &\qquad (1 - \exp(-2(b_T + cT - x)c))\, \mathrm{d}\mathbb{P}(N(0,T) \leq x).\end{aligned}$$

It is a straightforward exercise to show that the first integral is equal to

$$(1-\exp(-2b_0c))\left(\Phi\left(\frac{b_T+cT-b_0}{\sqrt{T}}\right)-\exp(-2b_Tc)\Phi\left(\frac{-b_T+cT-b_0}{\sqrt{T}}\right)\right),$$

whereas the second integral equals

$$1-\Phi\left(\frac{-b_T-cT-b_0}{\sqrt{T}}\right)-\Phi\left(\frac{b_T+cT-b_0}{\sqrt{T}}\right)+$$
$$\exp(-2b_Tc)\left(\Phi\left(\frac{-b_T+cT-b_0}{\sqrt{T}}\right)+\Phi\left(\frac{b_T-cT-b_0}{\sqrt{T}}\right)-1\right).$$

Using that $\mathbb{P}(Q_i \leq b_i) = 1-\exp(-2b_ic)$, $i=0,T$, see (2.16), and that $1-\Phi(x) = \Phi(-x)$, the stated follows from

$$p(\overline{b},T) = 1-\mathbb{P}(Q_0\leq b_0)-\mathbb{P}(Q_T\leq b_T)+\overline{p}(\overline{b},T). \qquad \square$$

2.5.3 Covariance function

In the previous subsection we derived a closed-form expression for $p(\overline{b},T)$. This result also allows us to calculate the covariance of Q_0 and Q_T, i.e., $\mathbb{C}\mathrm{ov}(Q_0,Q_T)$, which we present in the next theorem.

Theorem 2.5.2 *For each $T\geq 0$,*

$$\begin{aligned}\theta(T) &:= \mathbb{C}\mathrm{ov}(\mathrm{Q}_0,\mathrm{Q}_\mathrm{T}) \qquad (2.23)\\ &= \left(-\frac{c^2T^2}{2}-T+\frac{1}{2c^2}\right)\left(1-\Phi(c\sqrt{T})\right)+\phi(c\sqrt{T})\left(\frac{cT\sqrt{T}}{2}+\frac{\sqrt{T}}{2c}\right).\end{aligned}$$

Proof: First recall that $\mathbb{C}\mathrm{ov}(Q_0,Q_T) = \mathbb{E}Q_0Q_T-\mathbb{E}Q_0\mathbb{E}Q_T$. Then use the well-known fact that Q_0 and Q_T are both exponentially distributed with mean $1/(2c)$, i.e., $\mathbb{E}Q_0\mathbb{E}Q_T = 1/(4c^2)$. Hence, we are left with the computation of $\mathbb{E}Q_0Q_T$. Using Theorem 2.5.1, we find that

$$\begin{aligned}\mathbb{E}Q_0Q_T &= \int_0^\infty\int_0^\infty p(\overline{b},T)\mathrm{d}b_0\mathrm{d}b_T\\ &= -\int_0^\infty\int_0^\infty \Phi\left(k_1(\overline{b},T)\right)\mathrm{d}b_0\mathrm{d}b_T+\int_0^\infty\int_0^\infty e^{-2b_Tc}\Phi\left(k_2(\overline{b},T)\right)\mathrm{d}b_0\mathrm{d}b_T\\ &\quad+\int_0^\infty\int_0^\infty e^{-2b_0c}\Phi\left(k_3(\overline{b},T)\right)\mathrm{d}b_0\mathrm{d}b_T\\ &\quad+\int_0^\infty\int_0^\infty e^{-2(b_0+b_T)c}\Phi\left(k_4(\overline{b},T)\right)\mathrm{d}b_0\mathrm{d}b_T.\end{aligned}$$

By interchanging the order of integration, and applying integration by parts, straightforward (though tedious) calculus yields that

$$-\int_0^\infty\int_0^\infty \Phi\left(k_1(\overline{b},T)\right)\mathrm{d}b_0\mathrm{d}b_T \qquad (2.24)$$

$$= -\left(\frac{T}{2} + \frac{c^2T^2}{2}\right)\left(1 - \Phi(c\sqrt{T})\right) + \frac{cT\sqrt{T}}{2}\phi(c\sqrt{T});$$

$$\int_0^\infty \int_0^\infty e^{-2b_T c}\Phi\left(k_2(\overline{b}, T)\right) \mathrm{d}b_0 \mathrm{d}b_T \tag{2.25}$$

$$= \left(\frac{1}{2c^2} - \frac{T}{2}\right)\left(1 - \Phi(c\sqrt{T})\right) + \frac{\sqrt{T}}{2c}\phi(c\sqrt{T});$$

$$\int_0^\infty \int_0^\infty e^{-2b_0 c}\Phi\left(k_3(\overline{b}, T)\right) \mathrm{d}b_0 \mathrm{d}b_T \tag{2.26}$$

$$= \left(\frac{1}{2c^2} - \frac{T}{2}\right)\left(1 - \Phi(c\sqrt{T})\right) + \frac{\sqrt{T}}{2c}\phi(c\sqrt{T});$$

$$\int_0^\infty \int_0^\infty e^{-2(b_0+b_T)c}\Phi\left(k_4(\overline{b}, T)\right) \mathrm{d}b_0 \mathrm{d}b_T$$

$$= \left(\frac{T}{2} - \frac{1}{4c^2}\right)\left(1 - \Phi(c\sqrt{T})\right) + \frac{1}{4c^2}\Phi(c\sqrt{T}) - \frac{\sqrt{T}}{2c}\phi(c\sqrt{T}). \tag{2.27}$$

Adding up (2.24)-(2.27), and subtracting $1/(4c^2)$ yields the stated. □

It is noted that $\theta(0) = \mathbb{V}\mathrm{ar}Q_0 = 1/(4c^2)$, which is equivalent to the variance of an exponentially distributed variable with mean $1/(2c)$, as required. Also, note that $\lim_{T\to\infty}\theta(T) \to 0$, as Q_0 and Q_T become less and less correlated as $T \to \infty$. The following proposition summarizes three properties of $\theta(\cdot)$. This proposition implies that $(1 - \theta(\cdot))/\mathbb{V}\mathrm{ar}Q_0$ is a distribution function on $[0, \infty)$.

Proposition 2.5.3 *$\theta(\cdot)$ is non-increasing, convex and non-negative on $[0, \infty)$.*

Proof: $\theta(T)$ is non-increasing on $[0, \infty)$ if $\theta'(T) \leq 0$, i.e.,

$$-\left(1 + c^2T\right)\left(1 - \Phi(c\sqrt{T}\right) + c\sqrt{T}\phi\left(c\sqrt{T}\right) \leq 0,$$

which is equivalent to

$$\frac{\phi\left(c\sqrt{T}\right)}{1 - \Phi\left(c\sqrt{T}\right)} \leq c\sqrt{T} + \frac{1}{c\sqrt{T}}. \tag{2.28}$$

Likewise, $\theta(T)$ is convex on $[0, \infty)$ if $\theta''(T) \geq 0$, i.e.,

$$-c^2\left(1 - \Phi(c\sqrt{T}\right) + \frac{c}{\sqrt{T}}\phi\left(c\sqrt{T}\right) \geq 0,$$

or equivalently,

$$\frac{\phi\left(c\sqrt{T}\right)}{1 - \Phi\left(c\sqrt{T}\right)} \geq c\sqrt{T}. \tag{2.29}$$

Recalling the standard equality (2.1), it is easily seen that both (2.28) and (2.29) hold. Since $\theta(T)$ is non-increasing and $\lim_{T\to\infty}\theta(T) \to 0$, we also must have that $\theta(T)$ is non-negative. □

The next proposition presents the exact asymptotics of $\theta(T)$.

Proposition 2.5.4 *If* $T \to \infty$,

$$\theta(T) \sim \frac{4}{c^5 T\sqrt{T}}\phi\left(c\sqrt{T}\right). \tag{2.30}$$

Proof: First use that [5]

$$(1-\Phi(g(x))) \sim \left(\frac{1}{g(x)} - \frac{1}{(g(x))^3} + \frac{3}{(g(x))^5} - \frac{15}{(g(x))^7}\right)\phi(g(x)) \tag{2.31}$$

if $g(x)$ is increasing and $x \to \infty$. Using (2.31) and Theorem 2.5.2, it can then be verified that

$$\theta(T) \sim \left(\frac{4}{c^5 T\sqrt{T}} + \frac{16\frac{1}{2}}{c^7 T^2\sqrt{T}} - \frac{7\frac{1}{2}}{c^9 T^3\sqrt{T}}\right)\phi\left(c\sqrt{T}\right) \sim \frac{4}{c^5 T\sqrt{T}}\phi\left(c\sqrt{T}\right).$$

We note that the correct exact asymptotics of $\theta(T)$ can only be obtained, if all four terms of the right-hand side of (2.31) are used. □

Remark: The correlation coefficient of Q_0 and Q_T is given by

$$\rho(T) := \mathbb{C}\mathrm{or}(Q_0, Q_T) = \frac{\mathbb{C}\mathrm{ov}(Q_0, Q_T)}{\sqrt{\mathbb{V}\mathrm{ar}Q_0}\sqrt{\mathbb{V}\mathrm{ar}Q_T}} = 4c^2\theta(T), \tag{2.32}$$

as both Q_0 and Q_T are exponentially distributed with mean $1/(2c)$. Note that $\rho(0) = 1$ and $\lim_{T\to\infty}\rho(T) \to 0$. Due to (2.32), we also have that $\rho(T)$ is non-increasing, convex and non-negative on $[0,\infty)$, and that

$$\rho(T) \sim \frac{16}{c^3 T\sqrt{T}}\phi\left(c\sqrt{T}\right).$$

Hence, the exponential decay rate of both $\theta(T)$ and $\rho(T)$ equals $\left(c^2T\right)/2$.

It is noted that Theorem 2.5.2 and Propositions 2.5.3 and 2.5.4 have already (partly) appeared (for $\rho(T)$, instead of $\theta(T)$) in [4]. However, it is noted that our derivations are completely different compared to the ones given in [4]. We rely on Reich's formula to obtain the results, whereas [4] does not use this formula implicitly. It turns out that Proposition 2.5.3 also extends to the class of Lévy inputs, i.e., arrival processes with stationary, independent increments, see Theorem 3.6 in [61]. This class comprises, besides Brownian input, also compound Poisson input as special case.

2.5.4 Exact large-buffer asymptotics

In this subsection we derive the exact asymptotics of $p(\overline{b},T)$. We first present the following useful lemma.

Lemma 2.5.5 *Let $b_0 = b$, $b_T = \alpha b$ and $T = \gamma b$, with $\alpha, \gamma \geq 0$. If $b \to \infty$, then*

$$\Phi(k_1(\overline{b},T)) \sim -\zeta(k_1(\overline{b},T));$$

$$\Phi(k_2(\overline{b},T)) \sim \begin{cases} -\zeta(k_2(\overline{b},T)) & \text{if } \alpha < 1+c\gamma; \\ 1/2 & \text{if } \alpha = 1+c\gamma; \\ 1 & \text{otherwise}; \end{cases}$$

$$\Phi(k_3(\overline{b},T)) \sim \begin{cases} -\zeta(k_3(\overline{b},T)) & \text{if } \alpha > 1-c\gamma; \\ 1/2 & \text{if } \alpha = 1-c\gamma; \\ 1 & \text{otherwise}; \end{cases}$$

$$\Phi(k_4(\overline{b},T)) \sim \begin{cases} -\zeta(k_4(\overline{b},T)) & \text{if } \alpha > c\gamma-1; \\ 1/2 & \text{if } \alpha = c\gamma-1; \\ 1 & \text{otherwise}, \end{cases}$$

where $\zeta(\cdot)$ is as defined in (2.2).

Proof: First determine for which values of $b_T/b_0 = \alpha$, $k_i(\overline{b},T)$, $i \in \{1,2,3,4\}$, is positive or negative. Note that $k_1(\overline{b})$ is always negative. Hence, we obtain $1+c\gamma$, $1-c\gamma$ and $c\gamma-1$ as critical values from $k_i(\overline{b})$, $i = 2,3,4$, respectively. Next use the fact that $\Phi(-u) \sim \zeta(u)$ and $\Phi(u) \sim 1$ as $u \to \infty$. Observe that $\Phi(0) = 1/2$. □

We remark that $-\zeta(k_i(\overline{b},T))$ is positive in Lemma 2.5.5, as $\zeta(k_i(\overline{b},T))$ is negative in the listed cases, $i = 1,\ldots,4$. Define

$$\gamma(\overline{b},T) := 2b_0c + \frac{(-b_T - cT + b_0)^2}{2T}.$$

Theorem 2.5.6 *Let $b_0 = b$, $b_T = \alpha b$, $T = \gamma b$, with $\alpha, \gamma \geq 0$. Suppose $c\gamma > 1$. For $b \to \infty$,*

$$p(\overline{b},T) \sim \begin{cases} e^{-2(b_0+b_T)c} & \text{if } 0 \leq \alpha < \left(\sqrt{c\gamma}-1\right)^2; \\ \left(1 - \frac{1}{\sqrt{2\pi}k_2(\overline{b},T)} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-2(b_0+b_T)c} & \text{if } \alpha = \left(\sqrt{c\gamma}-1\right)^2; \\ \left(-\frac{1}{\sqrt{2\pi}k_2(\overline{b},T)} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-\gamma(\overline{b},T)} & \text{if } \left(\sqrt{c\gamma}-1\right)^2 < \alpha < 1+c\gamma; \\ \left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-2b_Tc} & \text{if } \alpha = 1+c\gamma; \\ e^{-2b_Tc} & \text{if } \alpha > 1+c\gamma. \end{cases}$$

Proof: We only prove the last statement, as the other four statements follow in a similar way. We have to prove that

$$p(\overline{b},T)e^{2b_Tc} \to 1 \text{ as } b \to \infty, \text{ for } \alpha > 1 + c\gamma.$$

From Lemma 2.5.5 we obtain that for $\alpha > 1 + c\gamma$,

$$\Phi(k_1(\overline{b},T)) \sim -\zeta(k_1(\overline{b},T)); \quad \Phi(k_2(\overline{b},T)) \sim 1;$$

$$\Phi(k_3(\overline{b},T)) \sim -\zeta(k_3(\overline{b},T)); \quad \Phi(k_4(\overline{b},T)) \sim -\zeta(k_4(\overline{b},T)).$$

Now straightforward calculus shows that, as $b \to \infty$,

$$\Phi(k_1(\overline{b},T)) = o\left(e^{-2b_Tc}\right),$$

and the same applies for $\Phi(k_3(\overline{b},T))e^{-2b_0c}$ and $\Phi(k_4(\overline{b},T))e^{-2(b_0+b_T)c}$. Using that $\Phi(k_2(\overline{b},T)) \sim 1$, Theorem 2.5.1 implies the stated. □

The following two theorems can be proven in a similar fashion as Theorem 2.5.6.

Theorem 2.5.7 *Let $b_0 = b$, $b_T = \alpha b$, $T = \gamma b$, with $\alpha, \gamma \geq 0$. Suppose $c\gamma = 1$. For $b \to \infty$,*

$$p(\overline{b},T) \sim \begin{cases} e^{-2b_0c} & \text{if } \alpha = 0; \\ \left(-\frac{1}{\sqrt{2\pi}k_2(\overline{b},T)} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-\gamma(\overline{b},T)} & \text{if } 0 < \alpha < 1 + c\gamma; \\ \left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-2b_Tc} & \text{if } \alpha = 1 + c\gamma; \\ e^{-2b_Tc} & \text{if } \alpha > 1 + c\gamma. \end{cases}$$

Theorem 2.5.8 *Let $b_0 = b$, $b_T = \alpha b$, $T = \gamma b$, with $\alpha, \gamma \geq 0$. Suppose $c\gamma < 1$. For $b \to \infty$,*

$$p(\overline{b},T) \sim \begin{cases} e^{-2b_0c} & \text{if } 0 \leq \alpha < 1 - c\gamma; \\ \left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}k_2(\overline{b},T)}\right) e^{-2b_0c} & \text{if } \alpha = 1 - c\gamma; \\ \left(-\frac{1}{\sqrt{2\pi}k_2(\overline{b},T)} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-\gamma(\overline{b},T)} & \text{if } 1 - c\gamma < \alpha < 1 + c\gamma; \\ \left(\frac{1}{2} - \frac{1}{\sqrt{2\pi}k_3(\overline{b},T)}\right) e^{-2b_Tc} & \text{if } \alpha = 1 + c\gamma; \\ e^{-2b_Tc} & \text{if } \alpha > 1 + c\gamma. \end{cases}$$

2.5.5 Most probable path

In the previous subsection it was shown that the nature of the large-buffer asymptotics strongly depends on the model parameters α, γ, and c, i.e., there are multiple regimes. In this subsection we will interpret these regimes by exploiting sample-path large

deviations results. Schilder's theorem, as introduced in Section 2.3, implies that the exponential decay rate of the joint overflow probability is characterized by the path that minimizes the decay rate. Among all paths such that the buffer exceeds b_0 and b_T at time 0 and T respectively, this is the MPP: informally speaking, given that this rare event occurs, with overwhelming probability (b_0, b_T) is reached by a path 'close to' the MPP.

In order to apply 'Schilder', we feed the single-node network by n i.i.d. standard Brownian sources. The link rate and buffer thresholds are also scaled by n: nc, nb_0 and nb_T, respectively. Using (2.20), $p_n(\overline{b}, T)$ can be expressed as

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} B_i(\cdot) \in S\right),$$

where

$$S := \{f \in \Omega | \exists s, t \geq 0 : -f(-t) > b_0 + ct, f(T) - f(T-s) > b_T + cs\},$$

and Ω is as defined in Equation (2.10).

We already argued in Section 2.3 that we can replace '$>$' by '$\geq$' in S, which is denoted as the set $\overline{S}$, without any impact on the decay rate of $p_n(\overline{b}, T)$. From 'Schilder' it then follows that

$$J(\overline{b}, T) := -\lim_{n\to\infty} \frac{1}{n} \log p_n(\overline{b}, T) = \inf_{f\in S} I(f) = \inf_{f\in\overline{S}} I(f).$$

As we will see below, depending on the value of b_0, b_T, c, and T, various regimes of asymptotics exist. Recall from Section 2.3 that knowledge of the MPP automatically implies that the decay rate is characterized, as the MPP translates in the decay rate through Equation (2.19). In the remainder of this subsection we explicitly derive $J(\overline{b}, T)$ by determining the MPPs corresponding to the various regimes.

Let us first define

$$\overline{U} := \{f \in \Omega | \exists t \geq 0 : -f(-t) \geq b_0 + ct\};$$

$$\overline{V} := \{f \in \Omega | \exists s \geq 0 : f(T) - f(T-s) \geq b_T + cs\},$$

i.e., $\overline{U}$ ($\overline{V}$) is the collection of all paths that yield a buffer content of at least b_0 (b_T) at time 0 (T). It follows that $\overline{S}$ is a subset of both $\overline{U}$ and $\overline{V}$, i.e., $\overline{S} \subseteq \overline{U}$ and $\overline{S} \subseteq \overline{V}$, implying that

$$J(\overline{b}, T) \geq \inf_{f\in\overline{U}} I(f); \quad J(\overline{b}, T) \geq \inf_{f\in\overline{V}} I(f). \tag{2.33}$$

From the above it follows that there is equality in one of the inequalities of (2.33), if either the MPP in $\overline{U}$ or $\overline{V}$ (or both) is also contained in the set $\overline{S}$.

Fortunately, the MPPs in $\overline{U}$ and $\overline{V}$ are already available, see e.g. [7]. The MPP in $\overline{U}$ is given by, for $r \in [-b_0/c, 0]$,

$$f^*(r) = \mathbb{E}(B(r)| - B(-b_0/c) = 2b_0).$$

The MPP is only specified on the interval $[-b_0/c, 0]$, because outside this interval the MPP generates traffic with mean rate $\mu = 0$. Using (2.4), it can then be verified that, for $r \in [-b_0/c, 0]$, $(f^*)'(r) = 2c$, whereas the derivative of this path is equal to zero outside this interval. In other words, the buffer starts to build up with constant rate $2c - c = c$ at time $-b_0/c$, which leads to $Q_0 = (b_0/c)c = b_0$, as required. Let us now determine the cost of this MPP. Using (2.19), we find that

$$I(f^*) = \frac{1}{2}\frac{b_0}{c}(2c)^2 = 2b_0c.$$

The MPP in $\overline{V}$ has a similar structure as the one above, but now the buffer grows with constant rate c in the interval $[T - b_T/c, T]$, which eventually gives $Q_T = b_T$, as required. The cost of this path can be derived in a similar manner and equal $2b_Tc$.

We are now ready to provide some explanation for each of the regimes of Theorems 2.5.6-2.5.8. Let us start with the regime $\alpha \geq 1 + c\gamma$ in Theorems 2.5.6-2.5.8. Using that $\alpha = b_T/b_0$ and $\gamma = T/b_0$, it is easily seen that we can rewrite $\alpha \geq 1 + c\gamma$ as $b_T - cT \geq b_0$. Subsequently, it is straightforward to show that the MPP in $\overline{V}$ is also contained in $\overline{S}$ under this regime, i.e., overflow of the buffer at time T implies overflow at time 0 without any additional effort. As the MPP in the set $\overline{V}$ is contained in $\overline{S} \subseteq \overline{V}$, it is also the MPP in the set $\overline{S}$. In other words, $J(\overline{b}, T)$ is equal to $2b_Tc$, given that $b_T - cT \geq b_0$. The MPP is depicted in Figure 2.2 (top, left).

Next consider the regime $0 \leq \alpha \leq 1 - c\gamma$ in Theorems 2.5.7-2.5.8, or equivalently $b_T \leq b_0 - cT$. In this case one can verify that the MPP in the set $\overline{U}$ is also contained in the set $\overline{S}$, and therefore it is the MPP in $\overline{S}$. Thus, overflow at time 0 implies overflow at time T without any extra effort. We conclude that $J(\overline{b}, T)$ equals $2b_0c$, given that $b_T \leq b_0 - cT$. The MPP is depicted in Figure 2.2 (top, right).

We proceed with the regime $0 \leq \alpha \leq (\sqrt{c\gamma} - 1)^2$ in Theorem 2.5.6, or equivalently $T \geq (\sqrt{b_0} + \sqrt{b_T})^2/c$. Consider the path that is such that the buffer builds up with rate c in the interval $[-b_0/c, 0]$, empties with rate c in the interval $(0, b_0/c)$, is empty in the interval $[b_0/c, T - b_T/c)$, and is growing again with rate c in the interval $[T - b_T/c, T]$, i.e., the MPP of $\overline{U}$ and $\overline{V}$ combined. It can be verified that this path is contained in the set $\overline{S}$ if $T \geq (\sqrt{b_0} + \sqrt{b_T})^2/c$. In Section 2.5.6 we show that this path is in fact the MPP in $\overline{S}$. In that case, $J(\overline{b}, T)$ can be obtained by using (2.19), and equals $2b_0c + 2b_Tc$. Clearly, this is no surprise, as the path consists of the MPP of $\overline{U}$ and $\overline{V}$. Note that this suggests that Q_0 and Q_T behave (almost) independently if, compared to b_0 and b_T, T is large enough, as may be expected. The MPP is depicted in Figure 2.2 (bottom, left).

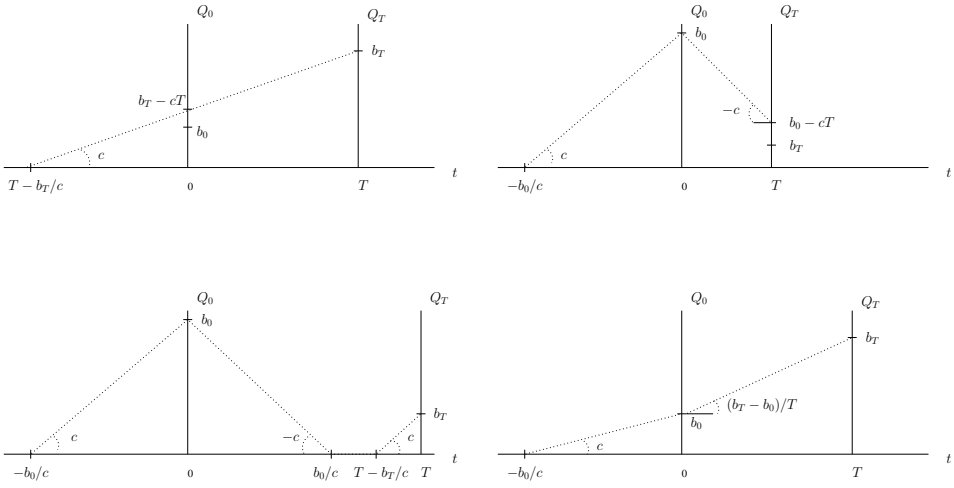

Figure 2.2: The most probable storage paths in $\{Q_0 \geq b_0, Q_T \geq b_T\}$.

We now focus on the remaining regimes of Theorems 2.5.6-2.5.8. Consider the path that is such that the buffer builds up with rate c in the interval $[-b_0/c, 0]$, and grows with rate $(b_T - b_0)/T$ in the interval $(0, T]$. Clearly, this path yields $Q_0 = b_0$ and $Q_T = b_T$, and is thus contained in $\overline{S}$. In Section 2.5.6 we show that this path is in fact the MPP for the remaining regimes. Assuming that this is indeed the case, $J(\overline{b}, T)$ is obtained by using (2.19), and equals $\gamma(\overline{b}, T)$. The MPP is depicted in Figure 2.2 (bottom, right).

The following two theorems are presented without proof, as they summarize the above-mentioned statements.

Theorem 2.5.9 *Suppose $cT > b_0$. Then it holds that*

$$J(\overline{b}, T) \sim \begin{cases} 2(b_0 + b_T)c & \text{if } 0 \leq b_T \leq \left(\sqrt{cT} - \sqrt{b_0}\right)^2; \\ \gamma(\overline{b}, T) & \text{if } \left(\sqrt{cT} - \sqrt{b_0}\right)^2 < b_T < b_0 + cT; \\ 2b_T c & \text{if } b_T \geq b_0 + cT. \end{cases}$$

Theorem 2.5.10 *Suppose $cT \leq b_0$. Then it holds that*

$$J(\overline{b}, T) \sim \begin{cases} 2b_0 c & \text{if } 0 \leq b_T \leq b_0 - cT; \\ \gamma(\overline{b}, T) & \text{if } b_0 - cT < b_T < b_0 + cT; \\ 2b_T c & \text{if } b_T \geq b_0 + cT. \end{cases}$$

2.5.6 Discussion

Using Theorems 2.5.6-2.5.8, the logarithmic large-buffer asymptotics can easily be derived as well. That is, we need to find a function $J^*(\overline{b}_\alpha, T_\gamma)$, with $\overline{b}_\alpha \equiv (b, \alpha b)$ and $T_\gamma \equiv \gamma b$, such that

$$\lim_{b\to\infty} \frac{-\log \mathbb{P}\left(Q_0 > b, Q_{\gamma b} > \alpha b\right)}{J^*(\overline{b}_\alpha, T_\gamma)} = 1,$$

where $\alpha, \gamma \geq 0$. With $b = b_0$, $\alpha b = b_T$, i.e., $\overline{b}_\alpha = \overline{b}$, and $\gamma b = T_\gamma = T$, it is not hard to see that $J^*(\overline{b}_\alpha, T_\gamma)$ equals $J(\overline{b}, T)$; compare Theorems 2.5.6-2.5.8 with Theorems 2.5.9-2.5.10, respectively. Indeed, since we assumed that in the many-sources framework the standard Brownian sources are i.i.d., and because a standard Brownian motion is characterized by independent increments, $J^*(\overline{b}_\alpha, T_\gamma)$ and $J(\overline{b}, T)$ should match, see for instance Example 7.4 in [65]. Recall that in the previous subsection we argued that the paths depicted in Figure 2.2 are MPPs in the set $\overline{S}$. From the above we conclude that this is indeed correct.

In the analysis we assumed that the input process was a standard Brownian motion, i.e., no drift and $v(t) = t$. We now show how the results can be extended to general Brownian input, with drift $\mu > 0$ and variance $v(t) = \lambda t$, $\lambda > 0$. Clearly, we should have that $c > \mu > 0$ to ensure stability. We denote the input process of a general Brownian motion by $\{B^*(t), t \in \mathbb{R}\}$. Then

$$\begin{aligned}
\overline{p}(\overline{b}, T) &= \mathbb{P}\left(\sup_{t\geq 0}\{B^*(-t,0) - ct\} > b_0, \sup_{s\geq 0}\{B^*(T-s,T) - cs\} > b_T\right) \\
&= \mathbb{P}\left(\exists s, t \geq 0 : B^*(-t,0) > b_0 + ct, B^*(T-s,T) > b_T + cs\right) \\
&= \mathbb{P}\left(\exists s, t \geq 0 : B(-t,0) > \frac{b_0}{\sqrt{\lambda}} + \frac{(c-\mu)t}{\sqrt{\lambda}}, B(T-s,T) > \frac{b_T}{\sqrt{\lambda}} + \frac{(c-\mu)s}{\sqrt{\lambda}}\right).
\end{aligned}$$

Hence, in order to generalize the results of this section, it follows that we have to set $c \leftarrow (c-\mu)/\sqrt{\lambda}$ and $b_i \leftarrow b_i/\sqrt{\lambda}$, $i = 0, T$ there. In order to generalize the results of Section 2.5.3 on the covariance, in addition we need to multiply the right-hand side of (2.23) and (2.30) by $\sqrt{\lambda}\sqrt{\lambda} = \lambda$. The results on the correlation coefficient can be generalized in a similar way.

In this section we studied the joint distribution function of the workloads at time 0 and time T, the covariance of these workloads, large-buffer asymptotics, and the MPP leading to overflow. It is noted that one may also derive an explicit expression for

$$q(\overline{b}, T) := \mathbb{P}(Q_T > b_T | Q_0 = b_0),$$

by using $p(\overline{b}, T)$, see [115] for more details.

Chapter 3

Simple networks of Brownian queues

In the previous chapter we considered a Brownian queue, i.e., a single-node network with Brownian input. Before analyzing GPS systems in Chapters 4 and 5, we first need to gain more insight by extending the results of Chapter 2 to more complicated systems. In particular, in this chapter we study simple networks of Brownian queues, namely: a two-node parallel Brownian queue and a two-node tandem Brownian queue. In addition, we consider priority queueing in a two-class Brownian queue, which is in fact a special case of GPS scheduling.

The case of networks of Brownian queues has, compared to single Brownian queues, been studied considerably less. In [125] and [50] a two-node tandem queue is analyzed: [125] derives the joint distribution function of the first and total queue length, whereas [50] focuses on the distribution function of the second queue. Also, several papers consider the more general case of tandem systems with Lévy input. We remark that the solution presented in [87] and [48] is in terms of a joint Laplace transform; no explicit expression for the joint distribution function is given. In [86] it is shown that a tandem system with dependent or independent Lévy inputs to the nodes can be seen as a special case of a parallel queue with dependent Lévy inputs to the nodes.

The case where different types of Brownian inputs compete for service on a single link is not well-understood either. In [130, 131] a two-class priority queue is considered, and the decay rate of the overflow probability of the low-priority class is derived. We also refer to [128] for related results on priority queueing. In [126, 129, 132] a two-class GPS system is analyzed, and the decay rate of the overflow probability of a particular class is obtained.

Besides the use of Brownian motions as input processes, they also appear in the analysis of queueing models where the input process is no Brownian motion. Multi-dimensional reflected Brownian motions are often used to approximate the joint queue length or joint workload processes of open networks under heavy-traffic conditions, see e.g. [77, 122].

The remainder of the chapter is organized as follows. In Section 3.1 we present

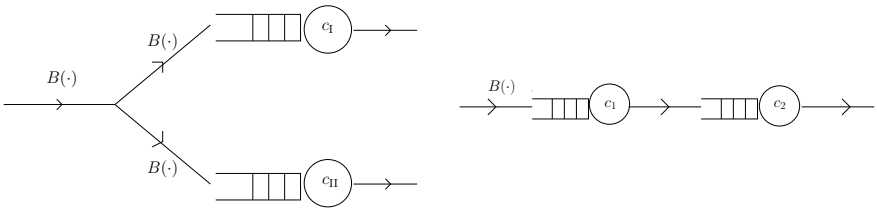

Figure 3.1: Left: Two-node parallel queue. Right: Two-node tandem queue

a detailed description of the two-node tandem queue, as well as a closely related two-node parallel queue. We also give formal implicit expressions for the overflow probabilities. In Section 3.2 the two-node parallel queue is analyzed: we derive an exact expression for the joint distribution function, large-buffer asymptotics, and the most probable path (MPP). Then we argue that the two-node parallel queue is closely related to the two-node tandem queue. Exploiting this property, we obtain in Section 3.3 similar results for the tandem system. In Section 3.4 we consider another related system, viz. the two-class priority queue.

3.1 Preliminaries

In this section we first describe our main queueing models: the two-node parallel queue and the two-node tandem queue. We proceed by presenting an implicit expression for the joint overflow probability in each of the two models.

3.1.1 Queueing models

Section 3.2 considers a two-node parallel queue with service rate $c_{\rm I}$ at queue I, and $c_{\rm II}$ at queue II. Traffic that enters the system has to be served at both queues I and II, which is done in parallel; see Figure 3.1 for an illustration. The case $c_{\rm I} = c_{\rm II}$ being trivial, we assume without loss of generality that $c_{\rm I} > c_{\rm II} > 0$.

We assume that the input process is a standard Brownian motion $\{B(t), t \geq 0\}$, with $B(0) \equiv 0$. Recall that this implies that $B(s,t) = B(t) - B(s) \sim N(0, t-s)$, i.e., the amount of traffic that enters in the interval $[s,t]$ is standard Normally distributed with mean 0 and variance $t-s$.

In Section 3.3 we consider a two-node tandem queue, again with standard Brownian input. Thus, the output of the first queue is fed into the second queue; see Figure 3.1. Assume constant service rates c_1 and c_2, respectively. To avoid the trivial situation of the second queue remaining empty, it is assumed that $c_1 > c_2 > 0$. We note that this model corresponds to the heavy-traffic limit of the two-node tandem queue with Poisson arrivals, see [125].

3.1.2 Joint overflow probabilities

In this subsection we present an implicit expression for the joint overflow probability in each of the two queueing models.

Let Q_{I} and Q_{II} denote the steady-state workload of queues I and II, respectively, in the two-node parallel queue. We study the joint distribution of the steady-state workloads of queues I and II:

$$\mathbb{P}(Q_{\mathrm{I}} > b_{\mathrm{I}}, Q_{\mathrm{II}} > b_{\mathrm{II}}). \tag{3.1}$$

Note that if $b_{\mathrm{II}} < b_{\mathrm{I}}$, then (due to $c_{\mathrm{I}} > c_{\mathrm{II}}$) the event $\{Q_{\mathrm{I}} > b_{\mathrm{I}}\}$ automatically implies $\{Q_{\mathrm{II}} > b_{\mathrm{II}}\}$. Hence, we concentrate on $b_{\mathrm{II}} \geq b_{\mathrm{I}}$. Reich's formula [155] states that

$$Q_{\mathrm{I}} = \sup_{s\geq 0}\{-B(-s) - c_{\mathrm{I}}s\} \quad \text{and} \quad Q_{\mathrm{II}} = \sup_{t\geq 0}\{-B(-t) - c_{\mathrm{II}}t\}. \tag{3.2}$$

Let s^* and t^* denote an optimizing s and t in (3.2). Now, $-s^*$ ($-t^*$) can be interpreted as the beginning of the busy period of queue I (queue II) containing time 0. Hence, $c_{\mathrm{I}} > c_{\mathrm{II}}$ implies that $s^* \leq t^*$, and therefore (3.1) can be rewritten as $\mathbb{P}\left(B(\cdot) \in S\right)$, where

$$S := \{f \in \Omega | \exists t \geq 0 : \exists s \in [0,t] : -f(-s) > b_{\mathrm{I}} + c_{\mathrm{I}}s, -f(-t) > b_{\mathrm{II}} + c_{\mathrm{II}}t\}, \tag{3.3}$$

and Ω is as defined in Equation (2.10).

In the two-node tandem queue we focus on the joint probability that the stationary workloads of the first and second queue, Q_1 and Q_2, respectively, exceed thresholds b_1 and b_2, with $b_1, b_2 \geq 0$. For any queue in which traffic leaves the first queue as fluid, the steady-state *total* workload Q_T in the two-node tandem queue behaves as a single queue emptied at rate c_2, see e.g. [130] and references therein. As a consequence,

$$Q_1 = \sup_{s\geq 0}\{-B(-s) - c_1 s\} \quad \text{and} \quad Q_T = \sup_{t\geq 0}\{-B(-t) - c_2 t\}. \tag{3.4}$$

Like for the parallel system, we have that the optimizing s is not larger than the optimizing t in (3.4). Hence, for $b_T \geq b_1 \geq 0$, $\mathbb{P}(Q_1 > b_1, Q_T > b_T)$ equals $\mathbb{P}\left(B(\cdot) \in T\right)$, with

$$T := \{f \in \Omega | \exists t \geq 0 : \exists s \in [0,t] : -f(-s) > b_1 + c_1 s, -f(-t) > b_T + c_2 t\}. \tag{3.5}$$

Note that (3.3) and (3.5) coincide if $c_1 = c_{\mathrm{I}}$, $c_2 = c_{\mathrm{II}}$, $b_1 = b_{\mathrm{I}}$, and $b_T = b_{\mathrm{II}}$. We will exploit this property in Section 3.3. Evidently, the distribution of (Q_1, Q_T) uniquely determines the distribution of (Q_1, Q_2). Using that $Q_2 = Q_T - Q_1$, we obtain that $\mathbb{P}(Q_1 > b_1, Q_2 > b_2)$, with $b_1, b_2 \geq 0$, equals $\mathbb{P}\left(B(\cdot) \in U\right)$, where

$$U := \left\{ f \in \Omega \;\middle|\; \begin{array}{l} \exists t \geq 0 : \exists s \in [0,t] : \forall u \in [0,t] : \\ -f(-s) > b_1 + c_1 s, \\ f(-u) - f(-t) > b_2 + c_2 t - c_1 u \end{array} \right\}. \tag{3.6}$$

3.2 Two-node parallel queue

In this section we focus on the two-node parallel queue. We derive the joint distribution function of queues I and II, large-buffer asymptotics, and the MPP leading to overflow.

3.2.1 Joint distribution function

In this subsection we derive an exact expression for $p(\overline{b}) := \mathbb{P}(Q_{\rm I} > b_{\rm I}, Q_{\rm II} > b_{\rm II})$, with $\overline{b} \equiv (b_{\rm I}, b_{\rm II})$. For the sake of brevity, write $\chi \equiv \chi(\overline{b}) := (b_{\rm II} - b_{\rm I})/(c_{\rm I} - c_{\rm II})$. Recall that $\Phi(\cdot)$ denotes the distribution function of a standard Normal random variable, $\phi(\cdot) := \Phi'(\cdot)$, and $\Psi(\cdot) := 1 - \Phi(\cdot)$. We first present the main theorem of this subsection.

Theorem 3.2.1 *For each* $b_{\rm II} \geq b_{\rm I} \geq 0$,

$$\begin{aligned} p(\overline{b}) &= -\Psi(k_1(\overline{b})) + \Psi(k_2(\overline{b}))e^{-2b_{\rm I}c_{\rm I}} + \\ &\quad \Psi(k_3(\overline{b}))e^{-2b_{\rm II}c_{\rm II}} + (1 - \Psi(k_4(\overline{b})))e^{-2(b_{\rm I}(c_{\rm I}-2c_{\rm II})+b_{\rm II}c_{\rm II})}, \end{aligned}$$

where

$$k_1(\overline{b}) := \frac{b_{\rm I} + c_{\rm I}\chi}{\sqrt{\chi}}; \quad k_2(\overline{b}) := \frac{-b_{\rm I} + c_{\rm I}\chi}{\sqrt{\chi}};$$

$$k_3(\overline{b}) := \frac{b_{\rm I} + (c_{\rm I} - 2c_{\rm II})\chi}{\sqrt{\chi}}; \quad k_4(\overline{b}) := \frac{-b_{\rm I} + (c_{\rm I} - 2c_{\rm II})\chi}{\sqrt{\chi}}.$$

Proof: In [125] an expression was derived for $\overline{p}(\overline{b}) := \mathbb{P}(Q_{\rm I} \leq b_{\rm I}, Q_{\rm II} \leq b_{\rm II})$ in case of standard Brownian input. We give a short sketch of the proof. First note that, due to time-reversibility arguments,

$$\overline{p}(\overline{b}) = \mathbb{P}(\forall t \geq 0 : B(t) \leq \min\{b_{\rm I} + c_{\rm I}t, b_{\rm II} + c_{\rm II}t\}).$$

Let $y \equiv y(\overline{b}) := b_{\rm I} + c_{\rm I}\chi$. Hence, (χ, y) is the point where $b_{\rm I} + c_{\rm I}t$ and $b_{\rm II} + c_{\rm II}t$ intersect, see Figure 3.2 for an illustration. For $t \in [0, \chi]$ the minimum is given by $b_{\rm I} + c_{\rm I}t$, whereas for $t \in [\chi, \infty)$ the minimum is $b_{\rm II} + c_{\rm II}t$. Now, conditioning on the value of $B(\chi)$, being Normally distributed with mean 0 and variance χ, one obtains that $\overline{p}(\overline{b})$ equals

$$\int_{-\infty}^{y} \frac{1}{\sqrt{\chi}}\phi\left(\frac{x}{\sqrt{\chi}}\right)\mathbb{P}(\forall t \in [0,\chi] : B(t) \leq b_{\rm I} + c_{\rm I}t | B(\chi) = x)\mathbb{P}(\forall t \geq 0 : B(t) \leq y - x + c_{\rm II}t)\mathrm{d}x.$$

The first probability in the above integral can be obtained by using (2.18), whereas the second probability is obtained by using (2.16). After substantial calculus we obtain that $\overline{p}(\overline{b})$ equals

$$\Phi(k_1(\overline{b}) - \Phi(k_2(\overline{b}))e^{-2b_{\rm I}c_{\rm I}} - \Phi(k_3(\overline{b}))e^{-2b_{\rm II}c_{\rm II}} + \Phi(k_4(\overline{b}))e^{-2(b_{\rm I}(c_{\rm I}-2c_{\rm II})+b_{\rm II}c_{\rm II})}.$$

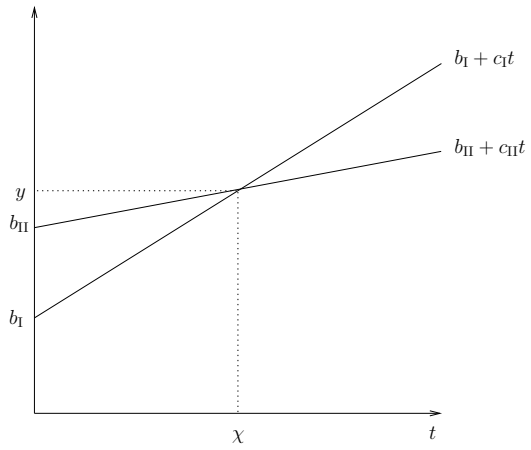

Figure 3.2: Illustration for the proof of Theorem 3.2.1.

Again using (2.18), we find that $\mathbb{P}(Q_i > b_i) = e^{-2b_i c_i}$, $i = \mathrm{I}, \mathrm{II}$. The stated now follows from

$$p(\overline{b}) = 1 - \mathbb{P}(Q_{\rm I} \leq b_{\rm I}) - \mathbb{P}(Q_{\rm II} \leq b_{\rm II}) + \overline{p}(\overline{b}). \qquad \square$$

3.2.2 Exact large-buffer asymptotics

In this subsection we derive the exact asymptotics of the joint buffer content distribution. We write as before $f(u) \sim g(u)$ when $f(u)/g(u) \to 1$ if $u \to \infty$. Also, let

$$\alpha_+ := \frac{c_{\rm I}}{2c_{\rm I} - c_{\rm II}}; \quad \alpha_0 := \frac{2c_{\rm II} - c_{\rm I}}{c_{\rm II}}; \quad \alpha_- := \frac{c_{\rm I} - 2c_{\rm II}}{2c_{\rm I} - 3c_{\rm II}}.$$

It can be verified that $\alpha_0 < 0 < \alpha_- < \alpha_+ < 1$ if $c_{\rm I} > 2c_{\rm II}$, whereas $0 \leq \alpha_0 < \alpha_+ < 1$ if $c_{\rm I} \leq 2c_{\rm II}$. Let us first present the following useful lemma.

Lemma 3.2.2 *Let $b_{\rm I} = \alpha b$ and $b_{\rm II} = b$, with $\alpha \in [0,1]$. If $b \to \infty$, then*

$$\Psi(k_1(\overline{b})) \sim \zeta(k_1(\overline{b}));$$

$$\Psi(k_2(\overline{b})) \sim \begin{cases} \zeta(k_2(\overline{b})) & \text{if } \alpha < \alpha_+; \\ 1/2 & \text{if } \alpha = \alpha_+; \\ 1 & \text{otherwise}; \end{cases}$$

$$\Psi(k_3(\overline{b})) \sim \begin{cases} \zeta(k_3(\overline{b})) & \text{if } \alpha > \alpha_0; \\ 1/2 & \text{if } \alpha = \alpha_0; \\ 1 & \text{otherwise}; \end{cases}$$

$$1-\Psi(k_4(\overline{b})) \sim \begin{cases} 1 & \text{if } \alpha < \alpha_- \text{ and } c_{\rm I} > 2c_{\rm II}; \\ 1/2 & \text{if } \alpha = \alpha_- \text{ and } c_{\rm I} \geq 2c_{\rm II}; \\ -\zeta(k_4(\overline{b})) & \text{otherwise.} \end{cases}$$

Proof: First determine for which values of $b_{\rm I}/b_{\rm II} = \alpha$, $k_i(\overline{b})$, $i \in \{1,2,3,4\}$, is positive or negative. Note that $k_1(\overline{b})$ is always positive, given that $b_{\rm II} \geq b_{\rm I} \geq 0$. Also, $k_4(\overline{b})$ is always negative if $c_{\rm I} \leq 2c_{\rm II}$ and $b_{\rm I} > 0$. Hence, we obtain α_+, α_0 and α_- as critical values from $k_i(\overline{b})$, $i = 2,3,4$, respectively. Next use the fact that $\Psi(u) \sim \zeta(u)$, where $\zeta(\cdot)$ is as defined (2.2), and $\Psi(-u) \sim 1$ as $u \to \infty$. Observe that $\Psi(0) = 1/2$. □

Define

$$\beta(\overline{b}) := \frac{1}{\sqrt{2\pi}} \left(-\frac{1}{k_1(\overline{b})} + \frac{1}{k_2(\overline{b})} + \frac{1}{k_3(\overline{b})} - \frac{1}{k_4(\overline{b})} \right); \quad \gamma(\overline{b}) := \frac{(b_{\rm II}c_{\rm I} - b_{\rm I}c_{\rm II})^2}{2(b_{\rm II}-b_{\rm I})(c_{\rm I}-c_{\rm II})}.$$

Straightforward calculus also shows the following equalities:

$$\begin{aligned} \exp\left(-\frac{k_1(\overline{b})^2}{2}\right) &= \exp\left(-\frac{k_2(\overline{b})^2}{2}\right) \exp\left(-2b_{\rm I}c_{\rm I}\right) \\ &= \exp\left(-\frac{k_3(\overline{b})^2}{2}\right) \exp\left(-2b_{\rm II}c_{\rm II}\right) \\ &= \exp\left(-\frac{k_4(\overline{b})^2}{2}\right) \exp\left(-2(b_{\rm I}(c_{\rm I}-2c_{\rm II}) + b_{\rm II}c_{\rm II})\right) \\ &= \exp\left(-\gamma(\overline{b})\right). \end{aligned} \tag{3.7}$$

Theorem 3.2.3 *Let $b_{\rm I} = \alpha b$ and $b_{\rm II} = b$, with $\alpha \in [0,1]$. Suppose $c_{\rm I} > 2c_{\rm II}$. For $b \to \infty$,*

$$p(\overline{b}) \sim \begin{cases} e^{-2(b_{\rm I}(c_{\rm I}-2c_{\rm II})+b_{\rm II}c_{\rm II})} & \text{if } \alpha \in [0,\alpha_-); \\ \frac{1}{2}e^{-2(b_{\rm I}(c_{\rm I}-2c_{\rm II})+b_{\rm II}c_{\rm II})} & \text{if } \alpha = \alpha_-; \\ \beta(\overline{b})e^{-\gamma(\overline{b})} & \text{if } \alpha \in (\alpha_-,\alpha_+); \\ \frac{1}{2}e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha = \alpha_+; \\ e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha \in (\alpha_+,1]. \end{cases}$$

Proof: We only prove the first statement, as the other four statements follow in a similar way. We have to prove that

$$p(\overline{b}) \exp(2(b_{\rm I}(c_{\rm I}-2c_{\rm II}) + b_{\rm II}c_{\rm II})) \to 1 \text{ as } b \to \infty, \text{ for } \alpha \in [0,\alpha_-).$$

From Lemma 3.2.2 we obtain that for $\alpha \in [0,\alpha_-)$,

$$\begin{aligned} &\Psi(k_1(\overline{b})) \sim \zeta(k_1(\overline{b})); \qquad \Psi(k_2(\overline{b})) \sim \zeta(k_2(\overline{b})); \\ &\Psi(k_3(\overline{b})) \sim \zeta(k_3(\overline{b})); \quad 1-\Psi(k_4(\overline{b})) \sim 1-\zeta(k_4(\overline{b})). \end{aligned}$$

Now it can be checked from (3.7) that, as $b \to \infty$,

$$\Psi(k_1(\overline{b})) = o\left(e^{-2(b_{\rm I}(c_{\rm I}-2c_{\rm II})+b_{\rm II}c_{\rm II})}\right),$$

and the same applies for $\Psi(k_2(\overline{b}))e^{-2b_{\rm I}c_{\rm I}}$ and $\Psi(k_3(\overline{b}))e^{-2b_{\rm II}c_{\rm II}}$. With $1-\Psi(k_4(\overline{b})) \sim 1$, Theorem 3.2.1 implies the stated. □

Theorem 3.2.4 *Let $b_{\rm I} = \alpha b$ and $b_{\rm II} = b$, with $\alpha \in [0,1]$. Suppose $c_{\rm I} < 2c_{\rm II}$. For $b \to \infty$,*

$$p(\overline{b}) \sim \begin{cases} e^{-2b_{\rm II}c_{\rm II}} & \text{if } \alpha \in [0,\alpha_0); \\ \frac{1}{2}e^{-2b_{\rm II}c_{\rm II}} & \text{if } \alpha = \alpha_0; \\ \beta(\overline{b})e^{-\gamma(\overline{b})} & \text{if } \alpha \in (\alpha_0,\alpha_+); \\ \frac{1}{2}e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha = \alpha_+; \\ e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha \in (\alpha_+,1]. \end{cases}$$

Proof: The proof is similar to that of Theorem 3.2.3. □

Remark: Note that for $c_{\rm I} = 2c_{\rm II}$, one obtains $\alpha_0 = 0$. It can be verified that in this special case Theorem 3.2.4 reduces to

$$p(\overline{b}) \sim \begin{cases} e^{-2b_{\rm II}c_{\rm II}} & \text{if } \alpha = 0; \\ \beta(\overline{b})e^{-\gamma(\overline{b})} & \text{if } \alpha \in (0,\alpha_+); \\ \frac{1}{2}e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha = \alpha_+; \\ e^{-2b_{\rm I}c_{\rm I}} & \text{if } \alpha \in (\alpha_+,1]. \end{cases}$$

3.2.3 Most probable path

In the previous subsection it was shown that the nature of the large-buffer asymptotics strongly depends on the model parameters α, $c_{\rm I}$ and $c_{\rm II}$, i.e., there are different regimes. In this subsection we will interpret and explain these regimes by using sample-path large deviations. In particular, by using Schilder's theorem (Theorem 2.3.5), we show that in each of these regimes the system has a typical (most likely) behavior, and we characterize this behavior for each regime.

Schilder's theorem implies that the exponential decay rate of the joint overflow probability in the parallel system is characterized by the path in S that minimizes the decay rate. Among all paths such that queue I exceeds $b_{\rm I}$ and queue II exceeds $b_{\rm II}$, this is the MPP: informally speaking, given that this rare event occurs, with overwhelming probability $(b_{\rm I}, b_{\rm II})$ is reached by a path 'close to' the MPP. The goal of this subsection is to find the MPP in S, and to relate its form to the regimes identified in Section 3.2.2.

Consider the two-node parallel queue as described before. Now, in order to apply 'Schilder', we feed this network by n i.i.d. standard Brownian sources. The link rates

and buffer thresholds are also scaled by n: $nc_{\rm I}$, $nc_{\rm II}$, $nb_{\rm I}$ and $nb_{\rm II}$ respectively. Now, $p_n(\overline{b}) := \mathbb{P}(Q_{\rm I,n} > nb_{\rm I}, Q_{\rm II,n} > nb_{\rm II})$ can be expressed as

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} B_i(\cdot) \in S\right).$$

As mentioned in Chapter 2, we can replace '$>$' by '$\geq$' in S, the resulting set being denoted by $\overline{S}$, without any impact on the decay rate. From 'Schilder' it then follows that

$$J(\overline{b}) := -\lim_{n\to\infty} \frac{1}{n} \log p_n(\overline{b}) = \inf_{f\in S} I(f) = \inf_{f\in \overline{S}} I(f) = \inf_{t\geq 0} \inf_{s\in[0,t]} \Upsilon(s,t), \tag{3.8}$$

with

$$\Upsilon(s,t) := \inf_{f\in\overline{S}^{s,t}} I(f) \text{ and } \overline{S}^{s,t} := \{f \in \Omega | -f(-s) \geq b_{\rm I} + c_{\rm I}s, -f(-t) \geq b_{\rm II} + c_{\rm II}t\},$$

and $I(\cdot)$ as defined in (2.19), using the fact that the decay rate of a union of events is the minimum of the decay rates of the individual events.

We first show how, for fixed s, t, with $0 \leq s \leq t$, the infimum of $\Upsilon(s,t)$ over $\overline{S}^{s,t}$ can be computed. Define

$$g_1(s) := \frac{b_{\rm II}s}{b_{\rm I} + (c_{\rm I} - c_{\rm II})s} \quad \text{and} \quad g_2(s) := s\frac{c_{\rm I}}{c_{\rm II}} + \frac{b_{\rm I} - b_{\rm II}}{c_{\rm II}}, \quad s \geq 0.$$

Note that $g_1(\cdot)$ is a concave function, whereas $g_2(\cdot)$ is a linear function. Furthermore, $g_1(s) > g_2(s)$ if $s < \chi$, $g_1(s) = g_2(s)$ if $s = \chi$, and otherwise $g_1(s) < g_2(s)$. Also, define

$$\begin{aligned} A_1 &:= \{(s,t) | 0 \leq s \leq t \leq g_1(s)\};\\ A_2 &:= \{(s,t) | 0 \leq s \leq t \leq g_2(s)\};\\ A_3 &:= \{(s,t) | t > \max\{g_1(s), g_2(s)\}, s \geq 0\}. \end{aligned}$$

Note that $A := \{(s,t) | 0 \leq s \leq t\} = A_1 \cup A_2 \cup A_3$, for disjoint A_1, A_2 and A_3, as illustrated in Figure 3.3.

Lemma 3.2.5 *For $t \geq 0$, and $s \in [0,t]$,*

$$\Upsilon(s,t) = \begin{cases} h_1(t) := \frac{(b_{\rm II}+c_{\rm II}t)^2}{2t} & \text{if } (s,t) \in A_1;\\ h_2(s) := \frac{(b_{\rm I}+c_{\rm I}s)^2}{2s} & \text{if } (s,t) \in A_2;\\ h_3(s,t) := \frac{(b_{\rm I}+c_{\rm I}s)^2}{2s} + \frac{(b_{\rm II}+c_{\rm II}t-b_{\rm I}-c_{\rm I}s)^2}{2(t-s)} & \text{if } (s,t) \in A_3. \end{cases}$$

Proof: The proof is analogous to Lemma 3.4 of [130]. First note that the values of the Brownian input at times $-s$ and $-t$ are bivariate Normally distributed, i.e., as

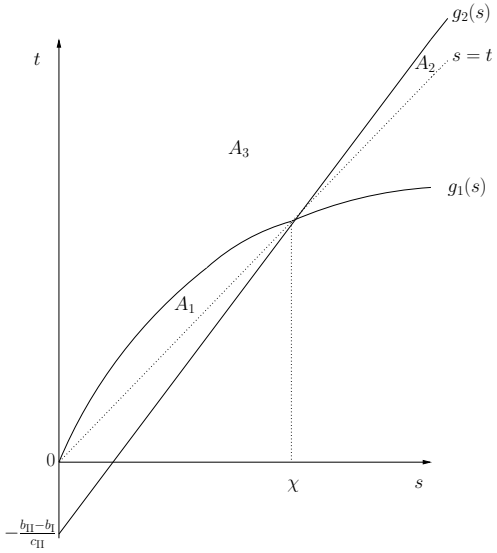

Figure 3.3: The partitioning of A.

$(-B(-s), -B(-t))$. Now, by using Theorem 2.3.4, we find for $y, z \in \mathbb{R}$ and $t \geq 0$, $s \in [0, t]$,

$$\Upsilon(s,t) = \inf_{y \geq b_{\rm I} + c_{\rm I} s} \inf_{z \geq b_{\rm II} + c_{\rm II} t} \Lambda(y,z), \tag{3.9}$$

with

$$\Lambda(y,z) = \frac{1}{2} \begin{pmatrix} y & z \end{pmatrix} \begin{pmatrix} s & s \\ s & t \end{pmatrix}^{-1} \begin{pmatrix} y \\ z \end{pmatrix}.$$

One can show that if

$$y_0 := \mathbb{E}(-B(-s) | -B(-t) = b_{\rm II} + c_{\rm II} t) \geq b_{\rm I} + c_{\rm I} s,$$

or, equivalently, $t \leq g_1(s)$, then the optimum in (3.9) is attained at $(y^*, z^*) = (y_0, b_{\rm II} + c_{\rm II} t)$. Hence, the rate function is independent of s, and given by $\Lambda(y_0, b_{\rm II} + c_{\rm II} t) = h_1(t)$.

In a similar way, if

$$z_0 := \mathbb{E}(-B(-t) | -B(-s) = b_{\rm I} + c_{\rm I} s) \geq b_{\rm II} + c_{\rm II} t,$$

or, after rewriting, $t \leq g_2(s)$, then the optimum in (3.9) is attained at $(y^*, z^*) = (b_{\rm I} + c_{\rm I} s, z_0)$. The rate function is then given by $\Lambda(b_{\rm I} + c_{\rm I} s, z_0) = h_2(s)$ (independently of t).

If $y_0 < b_{\rm I} + c_{\rm I}s$ and $z_0 < b_{\rm II} + c_{\rm II}t$, then the optimum in (3.9) is attained at $(y^*, z^*) = (b_{\rm I} + c_{\rm I}s, b_{\rm II} + c_{\rm II}t)$. It is readily verified that this yields $h_3(s,t)$ for $t > \max\{g_1(s), g_2(s)\}$. □

In order to obtain $J(\overline{b})$, it follows from (3.8) that we have to compute

$$\inf_{(s,t)\in A} \Upsilon(s,t). \tag{3.10}$$

We will obtain (3.10) by first deriving

$$\inf_{(s,t)\in A_1} \Upsilon(s,t) = \inf_{(s,t)\in A_1} h_1(t); \tag{3.11}$$

$$\inf_{(s,t)\in A_2} \Upsilon(s,t) = \inf_{(s,t)\in A_2} h_2(s); \tag{3.12}$$

$$\inf_{(s,t)\in A_3} \Upsilon(s,t) = \inf_{(s,t)\in A_3} h_3(s,t), \tag{3.13}$$

and subsequently taking the minimum of (3.11)-(3.13) (recall that $A = A_1 \cup A_2 \cup A_3$). We start by computing (3.11).

Area A_1

The optimization over A_1 reduces to

$$\inf_{(s,t)\in A_1} \Upsilon(s,t) = \inf_{(s,t)\in A_1} h_1(t) = \inf_{t\in[0,\chi]} h_1(t). \tag{3.14}$$

It can be verified that $h_1(t)$ is strictly decreasing on the interval $[0, b_{\rm II}/c_{\rm II})$, and strictly increasing on the interval $(b_{\rm II}/c_{\rm II}, \infty)$. Therefore, if $b_{\rm II}/c_{\rm II} \le \chi$ then $t^* = b_{\rm II}/c_{\rm II}$ and $s^* \in [g_1^{-1}(t^*), t^*]$, whereas otherwise $t^* = s^* = \chi$.

Lemma 3.2.6 *Expression (3.14) equals*

$$\begin{cases} 2b_{\rm II}c_{\rm II} & \text{if } c_{\rm I} \le 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in [0, \alpha_0]; \\ \gamma(\overline{b}) & \text{if } c_{\rm I} \le 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in (\alpha_0, 1]; \\ \gamma(\overline{b}) & \text{if } c_{\rm I} > 2c_{\rm II}. \end{cases}$$

Proof: The condition $b_{\rm II}/c_{\rm II} \le \chi$ is equivalent to $b_{\rm I}/b_{\rm II} \le (2c_{\rm II} - c_{\rm I})/c_{\rm II} = \alpha_0$. Note that α_0 is only non-negative if $c_{\rm I} \le 2c_{\rm II}$. Hence, evaluation of (3.14) for $t^* = b_{\rm II}/c_{\rm II}$ proves the first statement. Similarly, evaluation of (3.14) for $t^* = \chi$ proves the second statement. □

Area A_2

The approach is very similar to above. We are to solve the following optimization problem:

$$\inf_{(s,t)\in A_2} \Upsilon(s,t) = \inf_{(s,t)\in A_2} h_2(s) = \inf_{s\in[\chi,\infty)} h_2(s). \tag{3.15}$$

The function $h_2(s)$ has a global minimum that is attained at $s = b_{\mathrm{I}}/c_{\mathrm{I}}$. Thus, if $b_{\mathrm{I}}/c_{\mathrm{I}} \geq \chi$, then $s^* = b_{\mathrm{I}}/c_{\mathrm{I}}$ and $t^* \in [s^*, g_2(s^*)]$, whereas otherwise $s^* = t^* = \chi$. The following lemma is proven analogously to Lemma 3.2.6.

Lemma 3.2.7 *Expression (3.15) equals*

$$\begin{cases} \gamma(\overline{b}) & \text{if } b_{\mathrm{I}}/b_{\mathrm{II}} \in [0,\alpha_+); \\ 2b_{\mathrm{I}}c_{\mathrm{I}} & \text{if } b_{\mathrm{I}}/b_{\mathrm{II}} \in [\alpha_+,1]. \end{cases}$$

Area A_3

Now we are to solve the following optimization problem:

$$\inf_{(s,t)\in A_3} \Upsilon(s,t) = \inf_{(s,t)\in A_3} h_3(s,t) = \inf_{s\geq 0} \inf_{t>\max\{g_1(s),g_2(s)\}} h_3(s,t).$$

We can divide area A_3 in two parts, namely: $s \in [0,\chi]$ and $t \in (g_1(s),\infty)$, and $s \in (\chi,\infty)$ and $t \in (g_2(s),\infty)$ (see Figure 3.3). Let us start with the second part:

$$\inf_{s\in(\chi,\infty)} \inf_{t\in(g_2(s),\infty)} h_3(s,t). \tag{3.16}$$

Clearly, (3.16) is bounded from below by

$$\inf_{s\in(\chi,\infty)} \inf_{t\in(g_2(s),\infty)} h_2(s).$$

One can show that $h_3(s,t)$ reduces to $h_2(s)$ if $t = g_2(s)$ ($s \in [\chi,\infty)$). Therefore, analogously to area A_2, if $b_{\mathrm{I}}/c_{\mathrm{I}} \geq \chi$, then $s^* = b_{\mathrm{I}}/c_{\mathrm{I}}$ and $t^* = g_2(s^*) = (2b_{\mathrm{I}} - b_{\mathrm{II}})/c_{\mathrm{II}}$, whereas otherwise $s^* = t^* = \chi$. We thus obtain the following result.

Lemma 3.2.8 *Expression (3.16) equals*

$$\begin{cases} \gamma(\overline{b}) & \text{if } b_{\mathrm{I}}/b_{\mathrm{II}} \in [0,\alpha_+); \\ 2b_{\mathrm{I}}c_{\mathrm{I}} & \text{if } b_{\mathrm{I}}/b_{\mathrm{II}} \in [\alpha_+,1]. \end{cases}$$

We now turn to the first part:

$$\inf_{s\in[0,\chi]} \inf_{t\in(g_1(s),\infty)} h_3(s,t). \tag{3.17}$$

First concentrate on the minimum of $h_3(s,t)$ over $t \geq 0$, which is attained at

$$t = \frac{b_{\rm II} - b_{\rm I}}{c_{\rm II}} + s\frac{2c_{\rm II} - c_{\rm I}}{c_{\rm II}} =: g_3(s)$$

if $s \in [0,\chi]$ (for $s > \chi$ it is attained at $t = g_2(s)$, but this case is irrelevant here). Note that $g_3(s)$ is linearly decreasing (increasing) if $c_{\rm I} > 2c_{\rm II}$ ($c_{\rm I} < 2c_{\rm II}$). Also, $g_3(\chi) = \chi$. Hence, we have to distinguish between two cases:

- First concentrate on $c_{\rm I} > 2c_{\rm II}$. Then $g_3(s) > g_1(s)$ for all $s \in [0,\chi)$ (as $g_3(s)$ is non-increasing and $g_3(\chi) = \chi$). Substituting $t = g_3(s)$ in (3.17) gives

$$\inf_{s\in[0,\chi]} \frac{b_{\rm I}^2 + 2b_{\rm I}(c_{\rm I} - 2c_{\rm II})s + 4b_{\rm II}c_{\rm II}s + (c_{\rm I} - 2c_{\rm II})^2 s^2}{2s}. \tag{3.18}$$

  This is minimized for $s^* = b_{\rm I}/(c_{\rm I} - 2c_{\rm II})$ and $t^* = g_3(s^*) = (b_{\rm II} - 2b_{\rm I})/c_{\rm II}$ if $b_{\rm I}/(c_{\rm I} - 2c_{\rm II}) \leq \chi$, whereas otherwise $s^* = \chi = t^*$.

- Next consider $c_{\rm I} \leq 2c_{\rm II}$. In this case it is not clear a priori whether $g_3(s) \geq g_1(s)$ for all $s \in [0,\chi)$. For the moment assume that this is true. Then (3.18) is again appropriate, and this is minimized for $s^* = b_{\rm I}/(2c_{\rm II} - c_{\rm I})$ and $t^* = g_3(s) = b_{\rm II}/c_{\rm II}$ if $b_{\rm I}/(2c_{\rm II} - c_{\rm I}) \leq \chi$, whereas otherwise $s^* = \chi = t^*$. Now, in the former case it can be checked that $g_3(s^*) = g_1(s^*) = b_{\rm II}/c_{\rm II}$, and in the latter case we find $g_3(s^*) = g_1(s^*) = \chi$, i.e., the minimizers satisfy $g_3(s^*) \geq g_1(s^*)$, and hence we are done.

This reasoning leads to the following result.

Lemma 3.2.9 *Expression (3.17) equals*

$$\begin{cases} 2(b_{\rm I}(c_{\rm I} - 2c_{\rm II}) + b_{\rm II}c_{\rm II}) & \text{if } c_{\rm I} > 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in [0,\alpha_-]; \\ \gamma(\overline{b}) & \text{if } c_{\rm I} > 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in (\alpha_-,1]; \\ 2b_{\rm II}c_{\rm II} & \text{if } c_{\rm I} \leq 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in [0,\alpha_0]; \\ \gamma(\overline{b}) & \text{if } c_{\rm I} \leq 2c_{\rm II} \text{ and } b_{\rm I}/b_{\rm II} \in (\alpha_0,1]. \end{cases}$$

Exponential decay rate

In order to find $J(\overline{b})$, we have to determine the minimum of (3.11)-(3.13). This minimum can be obtained by combining Lemmas 3.2.6-3.2.9. From this, we already see that the minimum depends on the value of $b_{\rm I}/b_{\rm II} \in [0,1]$ and the sign of $c_{\rm I} - 2c_{\rm II}$. We now present an exact expression for the rate function $J(\overline{b})$. We start with the case $c_{\rm I} > 2c_{\rm II}$.

Theorem 3.2.10 *Suppose $c_{\rm I} > 2c_{\rm II}$. Then it holds that*

$$J(\overline{b}) = \begin{cases} 2(b_{\rm I}(c_{\rm I} - 2c_{\rm II}) + b_{\rm II}c_{\rm II}) & \text{if } b_{\rm I}/b_{\rm II} \in [0, \alpha_-]; \\ \gamma(\overline{b}) & \text{if } b_{\rm I}/b_{\rm II} \in (\alpha_-, \alpha_+); \\ 2b_{\rm I}c_{\rm I} & \text{if } b_{\rm I}/b_{\rm II} \in [\alpha_+, 1]. \end{cases}$$

Proof: By combining Lemmas 3.2.6-3.2.9, we find that there exist two critical values of $b_{\rm I}/b_{\rm II} \in [0,1]$, given that $c_{\rm I} > 2c_{\rm II}$: α_- and α_+. Recall from Section 3.2.2 that $0 < \alpha_- < \alpha_+ < 1$ if $c_{\rm I} > 2c_{\rm II}$. Now, if $b_{\rm I}/b_{\rm II} \in [0, \alpha_-]$, then it follows from Lemmas 3.2.6-3.2.9 that $J(\overline{b}) = \min\big\{2(b_{\rm I}(c_{\rm I} - 2c_{\rm II}) + b_{\rm II}c_{\rm II}), \gamma(\overline{b})\big\}$. Straightforward calculus shows that the first argument is smaller for these values of $b_{\rm I}/b_{\rm II}$. Similarly, if $b_{\rm I}/b_{\rm II} \in (\alpha_-, \alpha_+)$, then $J(\overline{b}) = \gamma(\overline{b})$. Finally, if $b_{\rm I}/b_{\rm II} \in [\alpha_+, 1]$, then $J(\overline{b}) = \min\big\{2b_{\rm I}c_{\rm I}, \gamma(\overline{b})\big\}$. Applying straightforward calculus yields that the first argument is smaller if $b_{\rm I}/b_{\rm II} \in (\alpha_+, 1]$. □

Schilder's theorem says that knowledge of the MPP f^* for the buffers to fill, also implies that the exponential decay rate is known: $J(\overline{b}) = I(f^*)$. Luckily, we do not have to derive the MPPs corresponding to the three decay rates of Theorem 3.2.10, because we have already implicitly obtained them. The values of $-s^*$ and $-t^*$, where s^* and t^* are the optimizers in Sections 3.2.3-3.2.3 associated with the three decay rates of Theorem 3.2.10, can be interpreted as the time where the first and second queue, respectively, start to build up in the corresponding MPP.

The s^* and t^* associated with the decay rate of the first regime in Theorem 3.2.10 are $s^* = b_{\rm I}/(c_{\rm I} - 2c_{\rm II})$ and $t^* = (b_{\rm II} - 2b_{\rm I})/c_{\rm II}$, see Section 3.2.3. Hence, in the MPP of the first regime, queue I starts to build up at $-s^*$, whereas queue II starts to build up at $-t^*$. The MPP is given by, for $r \in [-t^*, 0]$,

$$f^*(r) = \mathbb{E}\left(B(r)| - B(-s^*) = b_{\rm I} + c_{\rm I}s^*, -B(-t^*) = b_{\rm II} + c_{\rm II}t^*\right).$$

Using (2.4) it can be verified that

$$\begin{aligned} (f^*)'(r) &= 2c_{\rm II} && \text{if } r \in [-t^*, -s^*); \\ (f^*)'(r) &= 2(c_{\rm I} - c_{\rm II}) && \text{if } r \in [-s^*, 0]. \end{aligned}$$

Applying 'Schilder', i.e., using (2.19), one can verify that, as expected,

$$I(f^*) = \frac{1}{2}\left((2c_{\rm II})^2(t^* - s^*) + (2(c_{\rm I} - c_{\rm II}))^2 s^*\right) = 2(b_{\rm I}(c_{\rm I} - 2c_{\rm II}) + b_{\rm II}c_{\rm II}),$$

so f^* is indeed the MPP. Note that given service rates $c_{\rm I}$ and $c_{\rm II}$ at queues I and II, respectively, with $c_{\rm I} > 2c_{\rm II}$, the MPP yields $Q_{\rm I}(0) = b_{\rm I}$ and $Q_{\rm II}(0) = b_{\rm II}$. Also note that we have not specified the MPP outside $[-t^*, 0]$, because outside this interval the MPP produces traffic according to the average rate $\mathbb{E}B(1)$, which equals 0 (we are dealing with standard Brownian input), and therefore this does not affect $I(f^*)$.

Below we will therefore not specify the MPPs outside $[-t^*, 0]$ either (for different values of t^*).

The s^* and t^* associated with the decay rate of the second regime in Theorem 3.2.10 are $s^* = t^* = (b_{\rm II} - b_{\rm I})/(c_{\rm I} - c_{\rm II}) = \chi$, see Sections 3.2.3-3.2.3, i.e., in the second regime, both queues I and II start to build up at $-t^*$. The MPP is given by, for $r \in [-t^*, 0]$,

$$f^*(r) = \mathbb{E}\left(B(r)| - B(-t^*) = b_{\rm I} + c_{\rm I} t^*\right). \tag{3.19}$$

Using (2.4), it can be verified that this MPP is such that traffic enters the network with constant rate $(b_{\rm I}/(b_{\rm II} - b_{\rm I}))(c_{\rm I} - c_{\rm II}) + c_{\rm I}$ in the interval $[-t^*, 0]$, and this yields $Q_{\rm I}(0) = b_{\rm I}$ and $Q_{\rm II}(0) = b_{\rm II}$. Using (2.19), we find

$$I(f^*) = \frac{1}{2}\left(\frac{b_{\rm I}}{b_{\rm II} - b_{\rm I}}(c_{\rm I} - c_{\rm II}) + c_{\rm I}\right)^2 t^* = \gamma(\overline{b}),$$

so f^* is indeed the MPP.

The s^* and t^* associated with the decay rate of the third regime in Theorem 3.2.10 are $s^* = t^* = b_{\rm I}/c_{\rm I}$, see Section 3.2.3, i.e., in the third regime, both queues start to build up at $-t^*$. The MPP is given by, for $r \in [-t^*, 0]$,

$$f^*(r) = \mathbb{E}\left(B(r)| - B(-t^*) = b_{\rm I} + c_{\rm I} t^*\right).$$

Again, using (2.4), we find that this MPP is such that traffic is produced at constant rate $2c_{\rm I}$ in the interval $[-t^*, 0]$, and this gives $Q_{\rm I}(0) = b_{\rm I}$ and $Q_{\rm II}(0) = (b_{\rm I}/c_{\rm I})(2c_{\rm I} - c_{\rm II})$. Note that $Q_{\rm II}(0)$ is larger than $b_{\rm II}$ if $b_{\rm I}/b_{\rm II} \in (\alpha_+, 1]$, so there is indeed exceedance of $b_{\rm II}$. From (2.19), it follows that

$$I(f^*) = \frac{1}{2}\left(2c_{\rm I}\right)^2 t^* = 2b_{\rm I}c_{\rm I},$$

as required.

Theorem 3.2.11 *Suppose $c_{\rm I} \leq 2c_{\rm II}$. Then it holds that*

$$J(\overline{b}) = \begin{cases} 2b_{\rm II}c_{\rm II} & \text{if } b_{\rm I}/b_{\rm II} \in [0, \alpha_0]; \\ \gamma(\overline{b}) & \text{if } b_{\rm I}/b_{\rm II} \in (\alpha_0, \alpha_+); \\ 2b_{\rm I}c_{\rm I} & \text{if } b_{\rm I}/b_{\rm II} \in [\alpha_+, 1]. \end{cases}$$

Proof: The proof is similar to that of Theorem 3.2.10. □

The s^* and t^* associated with the decay rate of the first regime in Theorem 3.2.11 are $s^* = t^* = b_{\rm II}/c_{\rm II}$, see Section 3.2.3. Hence, in the MPP corresponding to the first regime of Theorem 3.2.11, both queues start to build up at $-t^*$. The MPP is given by, for $r \in [-t^*, 0]$,

$$f^*(r) = \mathbb{E}\left(B(r)| - B(-t^*) = b_{\rm II} + c_{\rm II} t^*\right).$$

Using (2.4), we find that traffic is generated at a constant rate $2c_{\rm II}$ in the interval $[-t^*, 0]$, and this results in $Q_{\rm I}(0) = (b_{\rm II}/c_{\rm II})(2c_{\rm II} - c_{\rm I}) > b_{\rm I}$ and $Q_{\rm II}(0) = b_{\rm II}$. Applying 'Schilder', yields

$$I(f^*) = \frac{1}{2}\left(2c_{\rm II}\right)^2 t^* = 2b_{\rm II}c_{\rm II}.$$

The MPPs corresponding to the second and third regime are similar to the MPPs corresponding to the second and third regime of Theorem 3.2.10.

3.2.4 Discussion

Using Theorems 3.2.3 and 3.2.4, the logarithmic large-buffer asymptotics follow directly as well. That is, we need to find a function $J^*(\overline{b}_\alpha)$, with $\overline{b}_\alpha \equiv (\alpha b, b)$, such that

$$\lim_{b\to\infty} \frac{-\log \mathbb{P}\left(Q_{\rm I} > \alpha b, Q_{\rm II} > b\right)}{J^*(\overline{b}_\alpha)} = 1,$$

where $\alpha \in [0,1]$. With $\alpha b = b_{\rm I}$ and $b = b_{\rm II}$, i.e., $\overline{b}_\alpha = \overline{b}$, it is not hard to see that $J^*(\overline{b}_\alpha)$ equals $J(\overline{b})$; compare Theorems 3.2.10 and 3.2.11 with Theorems 3.2.3 and 3.2.4, respectively. As we already argued in the previous chapter, since we assumed that in the many-sources framework the standard Brownian sources are i.i.d., and because a standard Brownian motion is characterized by independent increments, $J^*(\overline{b}_\alpha)$ and $J(\overline{b})$ should match.

In the analysis of the two-node parallel queue we assumed that the input process was a standard Brownian motion, i.e., no drift and $v(t) = t$. We now show how the results can be extended to general Brownian input, with drift $\mu > 0$ and variance $v(t) = \lambda t$, $\lambda > 0$. Clearly, we should have that $c_{\rm I} > c_{\rm II} > \mu > 0$ to ensure stability. We denote the input process of a general Brownian motion by $\{B^*(t), t \in \mathbb{R}\}$. Then, analogously to (3.3), $p(\overline{b}) = \mathbb{P}(B^*(\cdot) \in S) = \mathbb{P}(B(\cdot) \in S^*)$, with

$$S^* := \left\{ f \in \Omega \,\middle|\, \exists t \geq 0 : \exists s \in [0,t] : \begin{array}{l} -f(-s) > \frac{b_{\rm I} + (c_{\rm I} - \mu)s}{\sqrt{\lambda}}; \\ -f(-t) > \frac{b_{\rm II} + (c_{\rm II} - \mu)t}{\sqrt{\lambda}} \end{array} \right\}.$$

Hence in order to generalize the results of Section 3.2 to general Brownian input, we have to set $c_i \leftarrow (c_i - \mu)/\sqrt{\lambda}$ and $b_i \leftarrow b_i/\sqrt{\lambda}$, $i = {\rm I}, {\rm II}$.

3.3 Two-node tandem queue

In this section we focus on the two-node tandem queue. Exploiting the results of the two-node parallel queue in Section 3.2, we derive similar results for the two-node tandem queue.

3.3.1 Joint distribution function

In this subsection we derive an exact expression for $q(\overline{b}) := \mathbb{P}(Q_1 > b_1, Q_2 > b_2)$, with $\overline{b} \equiv (b_1, b_2)$. In Section 3.1.2 we argued that $p(b_{\rm I}, b_{\rm II})$ equals $q(\overline{b}_T) := \mathbb{P}(Q_1 > b_1, Q_T > b_T)$, with $\overline{b}_T \equiv (b_1, b_T)$, given that $b_{\rm I} = b_1$, $b_{\rm II} = b_T$, $c_{\rm I} = c_1$ and $c_{\rm II} = c_2$. In a first step to obtain $q(\overline{b})$, we derive $q_f(\overline{b}_T) := -\partial q(\overline{b}_T)/\partial b_1$. With mild abuse of notation, we also write $q_f(\overline{b}_T) = \mathbb{P}(Q_1 = b_1, Q_T > b_T)$. Define $\tau_T \equiv \tau(\overline{b}_T) := (b_T - b_1)/(c_1 - c_2)$ and $\tau \equiv \tau(b_2) := b_2/(c_1 - c_2)$.

Lemma 3.3.1 *For each $b_T \geq b_1 \geq 0$,*

$$\begin{aligned} q_f(\overline{b}_T) &= -\frac{\partial \ell_1(\overline{b}_T)}{\partial b_1}\phi(\ell_1(\overline{b}_T)) + 2c_1\Psi(\ell_2(\overline{b}_T))e^{-2b_1c_1} + \\ &\quad \frac{\partial \ell_2(\overline{b}_T)}{\partial b_1}\phi(\ell_2(\overline{b}_T))e^{-2b_1c_1} + \frac{\partial \ell_3(\overline{b}_T)}{\partial b_1}\phi(\ell_3(\overline{b}_T))e^{-2b_Tc_2} + \\ &\quad 2(c_1 - 2c_2)(1 - \Psi(\ell_4(\overline{b}_T)))e^{-2(b_1(c_1-2c_2)+b_Tc_2)} - \\ &\quad \frac{\partial \ell_4(\overline{b}_T)}{\partial b_1}\phi(l_4(\overline{b}_T))e^{-2(b_1(c_1-2c_2)+b_Tc_2)}, \end{aligned}$$

where

$$\ell_1(\overline{b}_T) := \frac{b_1 + c_1\tau_T}{\sqrt{\tau_T}}; \quad \ell_2(\overline{b}_T) := \frac{-b_1 + c_1\tau_T}{\sqrt{\tau_T}};$$

$$\ell_3(\overline{b}_T) := \frac{b_1 + (c_1 - 2c_2)\tau_T}{\sqrt{\tau_T}}; \quad \ell_4(\overline{b}_T) := \frac{-b_1 + (c_1 - 2c_2)\tau_T}{\sqrt{\tau_T}}.$$

Proof: Use Theorem 3.2.1, with $b_{\rm I} = b_1$, $b_{\rm II} = b_T$, $c_{\rm I} = c_1$ and $c_{\rm II} = c_2$, to obtain $q(\overline{b}_T)$. Then recall that $q_f(\overline{b}_T) = -\partial q(\overline{b}_T)/\partial b_1$. We extensively use the chain rule:

$$\frac{\partial \Psi(f(u))}{\partial u} = -\frac{\partial f(u)}{\partial u}\phi(f(u)).$$

Applying straightforward calculus now gives the desired result. □

Note that

$$\begin{aligned} q(\overline{b}) &= \mathbb{P}(Q_1 > b_1, Q_T > b_2 + Q_1) \\ &= \int_{b_1}^{\infty} q_f(\overline{x})\mathrm{d}x, \end{aligned} \tag{3.20}$$

where $\overline{x} \equiv (x, b_2 + x)$. Define

$$m_1(\overline{b}) := \frac{b_1 + c_1\tau}{\sqrt{\tau}}; \quad m_2(\overline{b}) := \frac{-b_1 + c_1\tau}{\sqrt{\tau}}; \quad m_4(\overline{b}) := \frac{-b_1 + (c_1 - 2c_2)\tau}{\sqrt{\tau}}.$$

We directly present the main theorem on tandem queues.

Theorem 3.3.2 *For each $b_1, b_2 \geq 0$,*

$$\begin{aligned} q(\overline{b}) &= \frac{c_2}{c_1 - c_2}\Psi(m_1(\overline{b})) + \Psi(m_2(\overline{b}))e^{-2b_1c_1} + \\ &\quad \frac{c_1 - 2c_2}{c_1 - c_2}(1 - \Psi(m_4(\overline{b})))e^{-2(b_1(c_1-c_2)+b_2c_2)}. \end{aligned}$$

Proof: Use (3.20) in combination with Lemma 3.3.1. Note that $q_f(\overline{x})$ consists of six terms. Let us start with the first term:

$$\int_{b_1}^{\infty} -\frac{\partial \ell_1(\overline{x})}{\partial x}\phi(\ell_1(\overline{x}))\mathrm{d}x = \Psi(\ell_1(\overline{x}))\Big|_{b_1}^{\infty} = -\Psi(m_1(\overline{b})). \tag{3.21}$$

Similarly, for the second and third term:

$$\begin{aligned} \int_{b_1}^{\infty} \left(2c_1\Psi(\ell_2(\overline{x}))e^{-2c_1x} + \frac{\partial \ell_2(\overline{x})}{\partial x}\phi(\ell_2(\overline{x}))e^{-2c_1x}\right)\mathrm{d}x &= -\Psi(\ell_2(\overline{x}))e^{-2c_1x}\Big|_{b_1}^{\infty} \\ &= \Psi(m_2(\overline{b}))e^{-2b_1c_1}. \end{aligned} \tag{3.22}$$

Proceeding with the fourth term:

$$\begin{aligned} \int_{b_1}^{\infty} \frac{\partial \ell_3(\overline{x})}{\partial x}\phi(l_3(\overline{x}))e^{-2c_2(b_2+x)}\mathrm{d}x &= \int_{b_1}^{\infty} \frac{\partial \ell_3(\overline{x})}{\partial x}\frac{1}{\sqrt{2\pi}}e^{-\frac{\ell_1(\overline{x})^2}{2}}\mathrm{d}x \\ &= \int_{b_1}^{\infty} \frac{\partial \ell_1(\overline{x})}{\partial x}\frac{1}{\sqrt{2\pi}}e^{-\frac{\ell_1(\overline{x})^2}{2}}\mathrm{d}x \\ &= -\Psi(\ell_1(\overline{x}))\Big|_{b_1}^{\infty} = \Psi(m_1(\overline{b})); \end{aligned} \tag{3.23}$$

here the first equality in (3.23) follows from the fact that

$$e^{-\ell_3(\overline{x})^2/2}e^{-2c_2(b_2+x)} = e^{-\ell_1(\overline{x})^2/2},$$

whereas the second equality holds due to $\partial\ell_3(\overline{x})/\partial x = \partial\ell_1(\overline{x})/\partial x$. We decompose the fifth term into two parts:

$$\begin{aligned} &2(c_1 - 2c_2)(1 - \Psi(\ell_4(\overline{x})))e^{-2(x(c_1-c_2)+b_2c_2)} \\ &\quad = 2(c_1-c_2)(1-\Psi(\ell_4(\overline{x})))e^{-2(x(c_1-c_2)+b_2c_2)} + 2c_2(\Psi(\ell_4(\overline{x}))-1)e^{-2(x(c_1-c_2)+b_2c_2)}. \end{aligned}$$

Now, taking the first decomposed fifth term and the sixth term:

$$\begin{aligned} &\int_{b_1}^{\infty} \left(2(1-\Psi(\ell_4(\overline{x})))(c_1-c_2)e^{-2(x(c_1-c_2)+b_2c_2)} - \frac{\partial \ell_4(\overline{x})}{\partial x}\phi(\ell_4(\overline{x}))e^{-2(x(c_1-c_2)+b_2c_2)}\right)\mathrm{d}x \\ &\qquad = -(1 - \Psi(\ell_4(\overline{x})))e^{-2(x(c_1-c_2)+b_2c_2)}\Big|_{b_1}^{\infty} \\ &\qquad = (1 - \Psi(m_4(\overline{b})))e^{-2(b_1(c_1-c_2)+b_2c_2)}. \end{aligned} \tag{3.24}$$

We are left with the second decomposed fifth term:

$$\begin{aligned}
&\int_{b_1}^{\infty} 2c_2(\Psi(\ell_4(\overline{x}))-1)e^{-2(x(c_1-c_2)+b_2c_2)}\mathrm{d}x \\
&\qquad = \frac{c_2}{c_1-c_2}\int_{b_1}^{\infty} 2(c_1-c_2)(\Psi(\ell_4(\overline{x}))-1)e^{-2(x(c_1-c_2)+b_2c_2)}\mathrm{d}x \\
&\qquad = \frac{c_2}{c_1-c_2}\Psi(m_1(\overline{b})) - \frac{c_2}{c_1-c_2}(1-\Psi(m_4(\overline{b})))e^{-2(b_1(c_1-c_2)+b_2c_2)}; \quad (3.25)
\end{aligned}$$

here the second equality in (3.25) is obtained by applying integration by parts, but requires tedious calculus. Adding up (3.21)-(3.25) yields the stated. □

Remark: For $b_1 > 0$ and $b_2 = 0$, we find $q(b_1, 0) = \mathbb{P}(Q_1 > b_1) = \exp(-2b_1c_1)$ in Theorem 3.3.2, i.e., the well-known exponential distribution with mean $1/(2c_1)$. For $b_1 = 0$ and $b_2 > 0$, Theorem 3.3.2 yields

$$\begin{aligned}
q(0,b_2) &= \mathbb{P}(Q_2 > b_2) \\
&= \frac{c_1}{c_1-c_2}\Psi\left(\frac{c_1}{\sqrt{c_1-c_2}}\sqrt{b_2}\right) + \frac{c_1-2c_2}{c_1-c_2}e^{-2b_2c_2}\left(1-\Psi\left(\frac{c_1-2c_2}{\sqrt{c_1-c_2}}\sqrt{b_2}\right)\right),
\end{aligned}$$

which is in line with Theorem 4.3 in [50].

3.3.2 Exact large-buffer asymptotics

In this subsection we derive the exact asymptotics of the joint buffer content distribution. Define

$$\alpha_+ := \frac{c_1}{2c_1-c_2};\ \alpha_- := \frac{c_1-2c_2}{2c_1-3c_2}.$$

It can be verified that $0 < \alpha_- < \alpha_+ < 1$ if $c_1 > 2c_2$, and $0 < \alpha_+ < 1$ if $c_1 \le 2c_2$. Recall that $\zeta(x) = \left(\sqrt{2\pi}x\right)^{-1}\exp(-x^2/2)$. First we present the counterpart of Lemma 3.2.2.

Lemma 3.3.3 *Let $b_1 = \alpha b$ and $b_2 = (1-\alpha)b$, with $\alpha \in [0,1]$. If $b \to \infty$, then*

$$\Psi(m_1(\overline{b})) \sim \zeta(m_1(\overline{b}));$$

$$\Psi(m_2(\overline{b})) \sim \begin{cases} \zeta(m_2(\overline{b})) & \text{if } \alpha < \alpha_+; \\ 1/2 & \text{if } \alpha = \alpha_+; \\ 1 & \text{otherwise;} \end{cases}$$

$$1-\Psi(m_4(\overline{b})) \sim \begin{cases} 1 & \text{if } \alpha < \alpha_- \text{ and } c_1 > 2c_2; \\ 1/2 & \text{if } \alpha = \alpha_- \text{ and } c_1 \ge 2c_2; \\ -\zeta(m_4(\overline{b})) & \text{otherwise.} \end{cases}$$

Proof: The proof is as in Lemma 3.2.2. □

Define

$$\theta(\overline{b}) := \frac{1}{\sqrt{2\pi}} \left(\frac{c_2}{c_1 - c_2} \frac{1}{m_1(\overline{b})} + \frac{1}{m_2(\overline{b})} - \frac{c_1 - 2c_2}{c_1 - c_2} \frac{1}{m_4(\overline{b})} \right); \tag{3.26}$$

$$\delta(\overline{b}) := \frac{(b_1(c_1 - c_2) + b_2 c_1)^2}{2b_2(c_1 - c_2)}. \tag{3.27}$$

The following equalities can shown to hold true:

$$\begin{aligned} \exp\left(-\frac{m_1(\overline{b})^2}{2} \right) &= \exp\left(-\frac{m_2(\overline{b})^2}{2} \right) \exp\left(-2b_1 c_1 \right) \\ &= \exp\left(-\frac{m_4(\overline{b})^2}{2} \right) \exp\left(-2(b_1(c_1 - c_2) + b_2 c_2) \right) \\ &= \exp\left(-\delta(\overline{b}) \right). \end{aligned} \tag{3.28}$$

The proof of the following two theorems is similar to the proof of Theorem 3.2.3, but now requires Lemma 3.3.3 and Equations (3.26)-(3.28). We omit the proofs.

Theorem 3.3.4 *Let $b_1 = \alpha b$ and $b_2 = (1-\alpha)b$, with $\alpha \in [0,1]$. Suppose $c_1 > 2c_2$. For $b \to \infty$,*

$$q(\overline{b}) \sim \begin{cases} \frac{c_1 - 2c_2}{c_1 - c_2} e^{-2(b_1(c_1 - c_2) + b_2 c_2)} & \text{if } \alpha \in [0, \alpha_-); \\ \frac{1}{2} \frac{c_1 - 2c_2}{c_1 - c_2} e^{-2(b_1(c_1 - c_2) + b_2 c_2)} & \text{if } \alpha = \alpha_-; \\ \theta(\overline{b}) e^{-\delta(\overline{b})} & \text{if } \alpha \in (\alpha_-, \alpha_+); \\ \frac{1}{2} e^{-2b_1 c_1} & \text{if } \alpha = \alpha_+; \\ e^{-2b_1 c_1} & \text{if } \alpha \in (\alpha_+, 1]. \end{cases}$$

Theorem 3.3.5 *Let $b_1 = \alpha b$ and $b_2 = (1-\alpha)b$, with $\alpha \in [0,1]$. Suppose $c_1 \leq 2c_2$. For $b \to \infty$,*

$$q(\overline{b}) \sim \begin{cases} \theta(\overline{b}) e^{-\delta(\overline{b})} & \text{if } \alpha \in [0, \alpha_+); \\ \frac{1}{2} e^{-2b_1 c_1} & \text{if } \alpha = \alpha_+; \\ e^{-2b_1 c_1} & \text{if } \alpha \in (\alpha_+, 1]. \end{cases}$$

Remark: We note that for $c_1 < 2c_2$ and $b_1 = 0$ ($\alpha = 0$) the asymptotics are not given by $\theta(\overline{b}) \exp(-\delta(\overline{b}))$, as it can be verified that $\theta(\overline{b})$ equals 0 in this special case. Therefore we have to rely here on the sharper asymptotic $\left(\sqrt{2\pi}u\right)^{-1} \exp(-u^2/2) - \Psi(u) \sim \left(\sqrt{2\pi}u^3\right)^{-1} \exp(-u^2/2)$. Using this, it can be shown [50] that

$$q(0, b_2) \sim \frac{1}{\sqrt{2\pi}} \left(\frac{c_1 - c_2}{b_2} \right)^{3/2} \frac{4c_2}{c_1^2 (c_1 - 2c_2)^2} e^{-\frac{c_1^2}{2(c_1 - c_2)} b_2}.$$

3.3.3 Most probable path

Similar to the parallel system, the large-buffer asymptotics now depend on the model parameters α, c_1 and c_2. Again, we will interpret the corresponding regimes by determining the structure of the MPPs.

We feed n i.i.d. standard Brownian sources into the tandem system, and also scale the link rates and buffer thresholds by n: nc_1, nc_2, nb_1 and nb_2 respectively. By using (3.6), we can write

$$q_n(\overline{b}) := \mathbb{P}(Q_{1,n} > nb_1, Q_{2,n} > nb_2) = \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} B_i(\cdot) \in U\right).$$

Define the set $\overline{U}$ as U, but with $>$' replaced by '$\geq$'. Clearly, $\overline{U} \subseteq \overline{U}^* \subseteq \overline{V}$, with

$$\overline{U}^* := \{f \in \Omega | \exists t \geq 0 : \exists s \in [0,t] : -f(-s) \geq b_1 + c_1 s, f(-s) - f(-t) \geq b_2 + c_2 t - c_1 s\};$$

$$\overline{V} := \{f \in \Omega | \exists t \geq 0 : \exists s \in [0,t] : -f(-s) \geq b_1 + c_1 s, -f(-t) \geq b_1 + b_2 + c_2 t\}.$$

Hence, 'Schilder' gives

$$K(\overline{b}) := -\lim_{n\to\infty} \frac{1}{n} \log q_n(\overline{b}) = \inf_{f\in U} I(f) = \inf_{f\in\overline{U}} I(f) \geq \inf_{f\in\overline{V}} I(f). \tag{3.29}$$

Let the MPP in the set $\overline{V}$ be denoted by f^*. If $f^* \in \overline{U}$, then there is clearly equality in (3.29).

Theorem 3.3.6 *Suppose $c_1 > 2c_2$. Then it holds that*

$$K(\overline{b}) = \begin{cases} 2(b_1(c_1 - c_2) + b_2 c_2) & \text{if } b_1/(b_1+b_2) \in [0, \alpha_-]; \\ \delta(\overline{b}) & \text{if } b_1/(b_1+b_2) \in (\alpha_-, \alpha_+); \\ 2b_1 c_1 & \text{if } b_1/(b_1+b_2) \in [\alpha_+, 1]. \end{cases}$$

Proof: Consider Theorem 3.2.10 with $c_{\rm I} = c_1$, $c_{\rm II} = c_2$, $b_{\rm I} = b_1$ and $b_{\rm II} = b_1 + b_2$, i.e., we have $\overline{U} \subseteq \overline{V} = \overline{S}$. The MPPs (in $\overline{S} = \overline{V}$) corresponding to each of the regimes of Theorem 3.2.10 were derived in Section 3.2.3. It can easily be checked that these MPPs are also contained in $\overline{U}$, and consequently they are the MPPs in $\overline{U}$. This implies that $K(\overline{b})$ is given by Theorem 3.2.10. □

Theorem 3.3.7 *Suppose $c_1 \leq 2c_2$. Then it holds that*

$$K(\overline{b}) = \begin{cases} \delta(\overline{b}) & \text{if } b_1/(b_1+b_2) \in [0, \alpha_+); \\ 2b_1 c_1 & \text{if } b_1/(b_1+b_2) \in [\alpha_+, 1]. \end{cases}$$

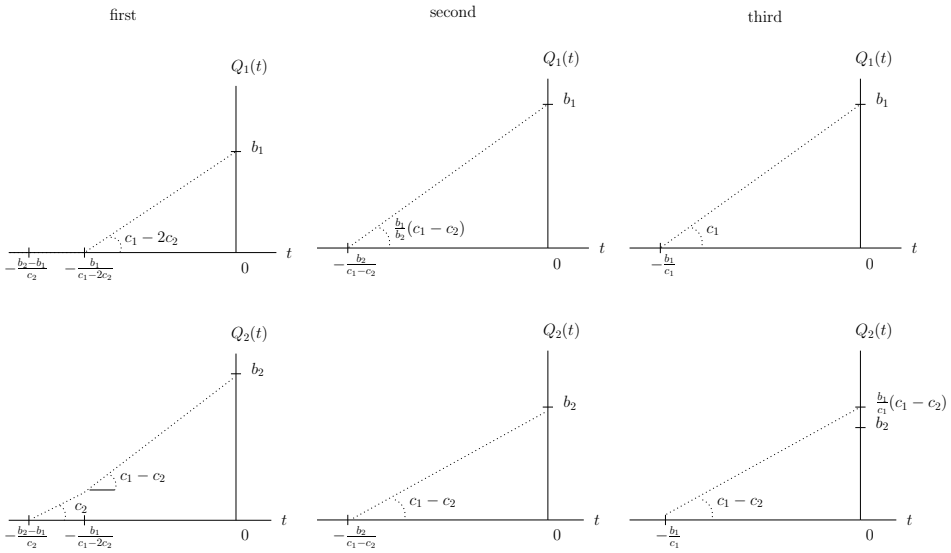

Figure 3.4: The most probable storage path in $\{Q_1 \geq b_1, Q_2 \geq b_2\}$ corresponding to each of the regimes of Theorem 3.3.6. The most probable storage path corresponding to each of the two regimes of Theorem 3.3.7, is also given by the most probable storage paths of the last two regimes of Theorem 3.3.6.

Proof: Consider Theorem 3.2.11 with $c_{\mathrm{I}} = c_1$, $c_{\mathrm{II}} = c_2$, $b_{\mathrm{I}} = b_1$ and $b_{\mathrm{II}} = b_1 + b_2$. Again, the MPPs corresponding to the second and third regime of Theorem 3.2.11, are also contained in the set $\overline{U}$, so $K(\overline{b})$ is given by Theorem 3.2.11 for $b_1/(b_1+b_2) \in (\alpha_0, 1]$. However, the MPP corresponding to the first regime, i.e., $b_1/(b_1+b_2) \in [0, \alpha_0]$, is not contained in $\overline{U}$, so we need a different approach here. In order to obtain a workload in queue 2 at least as large as b_2 at time 0, queue 2 needs to start building up at $-\tau = -b_2/(c_1 - c_2)$ at the latest. The set $\overline{U}$ can now be rewritten as

$$\left\{ f \in \Omega \,\middle|\, \exists t \geq \tau : \exists s \in [0,t] : \forall u \in [0,t] : \begin{array}{l} -f(-s) \geq b_1 + c_1 s, \\ f(-u) - f(-t) \geq b_2 + c_2 t - c_1 u \end{array} \right\},$$

which is contained in

$$\{ f \in \Omega | \exists t \geq \tau : \exists s \in [0,t] : -f(-s) \geq b_1 + c_1 s, -f(-t) \geq b_1 + b_2 + c_2 t \} =: \overline{W}.$$

Using the results of Section 3.2.3, with $b_{\mathrm{I}} = b_1$, $b_{\mathrm{II}} = b_1 + b_2$, $c_{\mathrm{I}} = c_1$ and $c_{\mathrm{II}} = c_2$, one can show that if $b_1/(b_1 + b_2) \in [0, \alpha_+)$ and $c_1 \leq 2c_2$, then the MPP in $\overline{W}$ is given by (3.19). As (3.19) is contained in $\overline{U}$, it is also the MPP in $\overline{U}$, implying that $K(\overline{b}) = \delta(\overline{b})$. □

Figure 3.4 depicts for each of the regimes of Theorem 3.3.6 the most likely way for the buffers to fill. Clearly, the most likely way for the buffers to fill for each of the two regimes of Theorem 3.3.7, coincides with the most probable storage paths of the last two regimes of Theorem 3.3.6. Interestingly, three types of MPPs are possible. In the first type queue 2 starts to build up earlier than queue 1, but they reach b_1 and b_2 at the same time. In the second type both queues start to grow at the same time, and reach b_1 and b_2 at the same time, whereas in the third type both queues start to build up at the same time, but at the time queue 1 reaches b_1, queue 2 is strictly larger than b_2.

Remark: If we set $b_1 > 0$ and $b_2 = 0$, then Theorems 3.3.6 and 3.3.7 give $K(\overline{b}) = 2b_1c_1$, which indeed is the exponential decay rate of the overflow probability in a single queue with standard Brownian input, emptied at rate c_1. For $b_1 = 0$ and $b_2 > 0$, Theorems 3.3.6 and 3.3.7 yield

$$K(\overline{b}) = \begin{cases} 2b_2c_2 & \text{if } c_1 > 2c_2; \\ \frac{c_1^2}{2(c_1-c_2)}b_2 & \text{otherwise,} \end{cases}$$

which is in line with Section 4.1 in [130].

3.3.4 Discussion

As in the two-node parallel queue, we can derive the logarithmic large-buffer asymptotics by using Theorems 3.3.4 and 3.3.5. That is, we need to obtain a function $K^*(\overline{b}_\alpha)$, with $\overline{b}_\alpha \equiv (\alpha b, (1-\alpha)b)$, such that

$$\lim_{b\to\infty} \frac{-\log \mathbb{P}\left(Q_1 > \alpha b, Q_2 > (1-\alpha)b\right)}{K^*(\overline{b}_\alpha)} = 1,$$

where $a \in [0,1]$. With $b_1 = \alpha b$ and $b_2 = (1-\alpha)b$, i.e., $\overline{b}_\alpha = \overline{b}$, it is not hard to see that $K^*(\overline{b}_\alpha)$ and $K(\overline{b})$ coincide; compare Theorems 3.3.6 and 3.3.7 with Theorems 3.3.4 and 3.3.5, respectively.

Again the results can also be generalized immediately to general Brownian input. Assuming that $c_1 > c_2 > \mu > 0$, this is done by setting $c_i \leftarrow (c_i - \mu)/\sqrt{\lambda}$ and $b_i \leftarrow b_i/\sqrt{\lambda}$, $i = 1, 2$.

The main approach used in this section relies on the fact that Brownian motions are characterized by stationary independent increments. Therefore, it can be expected that our approach is also valid for other input processes that have stationary independent increments (and an LDP). In subsequent research we showed that the asymptotics results can indeed be extended to the class of spectrally-positive Lévy inputs, i.e., input processes with stationary independent increments which do not have negative jumps [19]. This class covers Brownian motion and compound Poisson input as important special cases. As these results are somewhat out of the scope of this

monograph, we decided to only include the main findings, and to leave out details and proofs.

Below we associate with $A(\cdot) = \{A(t), t \geq 0\}$ the spectrally-positive Lévy input process of the tandem, where $A(0) \equiv 0$. As before, assume that both service rates (i.e., c_1 and c_2) are larger than $\mu := \mathbb{E}A(1) > 0$ to ensure stability, and that $c_1 > c_2$ to avoid the trivial situation that the second queue is always empty. Spectrally-positive Lévy processes are uniquely given through their Laplace exponent $\kappa(\cdot)$:

$$\mathbb{E}e^{-sA(t)} = e^{t\kappa(s)}, \quad s \geq 0.$$

If $\kappa(s)$ also exists for some negative s, then the Lévy process is called light-tailed, as the tail of the distribution of $A(1)$ decays exponentially or faster. If $\kappa(s)$ is only defined for non-negative s, then the process is called heavy-tailed, as the tail of the distribution of $A(1)$ decays more slowly than any exponential.

For the case of light-tailed Lévy input, we prove that the probability

$$\pi_\alpha(b) := \mathbb{P}(Q_1 > \alpha b, Q_2 > (1-\alpha)b),$$

with $\alpha \in (0,1)$, decays essentially exponentially, and we identify the corresponding decay rate

$$-\lim_{b\to\infty} \frac{1}{b} \log \pi_\alpha(b).$$

Recall that we already obtained the decay rate for the special case of Brownian input, see Theorems 3.3.6 and 3.3.7. In order to prove that $\pi_\alpha(b)$ decays exponentially for the general case of light-tailed Lévy input, we can exploit similar ideas as before. However, we also need to rely on an alternative approach, as explicit calculations seem no longer possible in this case.

Before presenting the next two theorems, we first need to define some notation. Let $\vartheta(s) := \kappa(s) + c_1 s$. Also, let $\bar{\vartheta}$ be $\inf_s \vartheta(s)$, and $\bar{s}$ the minimizing argument. Define $\bar{t}$ as the (non-zero) root of $\vartheta(t) = (c_1 - c_2)t$. Furthermore, let $s(\alpha)$ be the solution to $\vartheta'(s) = -(c_1 - c_2)\alpha/(1-\alpha)$; $t(\alpha)$ is defined as $\vartheta(s(\alpha))/(c_1 - c_2)$. Let $s_- < 0$ solve $\vartheta(s) = 0$, and $s_+ < 0$ is defined as the smaller solution to $\vartheta(s)/(c_1 - c_2) = \bar{t}$. Finally, define

$$\gamma_+ := -\frac{\vartheta'(s_-)}{c_1 - c_2 - \vartheta'(s_-)}; \quad \gamma_- := -\frac{\vartheta'(s_+)}{c_1 - c_2 - \vartheta'(s_+)}.$$

Theorem 3.3.8 *Suppose $\bar{s} < \bar{t}$. Then it holds that*

$$-\lim_{b\to\infty} \frac{1}{b} \log \pi_\alpha(b) = \begin{cases} -\alpha s_+ - (1-\alpha)\bar{t} & \text{if } \alpha \leq \gamma_-; \\ -\alpha s(\alpha) - (1-\alpha)t(\alpha) & \text{if } \gamma_- < \alpha < \gamma_+; \\ -\alpha s_- & \text{if } \alpha \geq \gamma_+. \end{cases}$$

Theorem 3.3.9 *Suppose $\bar{s} \geq \bar{t}$. Then it holds that*

$$-\lim_{b\to\infty} \frac{1}{b} \log \pi_\alpha(b) = \begin{cases} -\alpha s(\alpha) - (1-\alpha)t(\alpha) & \text{if } \alpha < \gamma_+; \\ -\alpha s_- & \text{if } \alpha \geq \gamma_+. \end{cases}$$

The proof of these two theorems is along the following lines. Relying on the classical Chernoff bound, we find a lower bound to this decay rate in the form of the solution of a convex programming problem. The analysis of this lower bound is based on the joint Laplace transform $\mathbb{E}e^{-sQ_1-tQ_2}$, for $(s,t) \in \mathbb{R}^2_+$, see [48]. In the light-tailed case, however, this expression is valid for some $(s,t) \notin \mathbb{R}^2_+$ as well. These (s,t) provide us with the crucial information to identify a lower bound on the decay rate. Relying on sample-path large deviations for Lévy processes [6], it is shown that this lower bound is actually tight. To this end, we construct a trajectory whose rate function coincides with the solution of the above-mentioned convex programming problem; as this trajectory is 'feasible' (in that it is such that indeed queue 1 exceeds αb and queue 2 exceeds $(1-\alpha)b$), this yields the desired result. Recall that, in the Brownian case, a pictorial illustration of the paths to overflow is given in Figure 3.4; the paths in the non-Brownian case look similar. It is a straightforward exercise to show that Theorems 3.3.8 and 3.3.9 coincide with Theorems 3.3.6 and 3.3.7, respectively, in case of standard Brownian input. The interested reader can find more details in [117].

In the case of heavy-tailed Lévy input the above line of reasoning does not apply. The rare event of joint overflow is then typically the result of just a single big jump, rather than a sequence of somewhat unlikely outcomes. In other words, the event of interest is essentially due to one large service requirement. In order for the workload of queue 1 to exceed αb, and for the workload of queue 2 to exceed $(1-\alpha)b$, with overwhelming probability this is due to a single job, whose size is roughly of the order b. This idea leads to a procedure that provides the exact asymptotics of $\pi_\alpha(b)$ in the heavy-tailed case.

The theorem below identifies the exact asymptotics of $\pi_\alpha(b)$ in the case of compound Poisson input with regularly varying jobs. That is, jobs arrive according to a Poisson process of rate ν, and the jobs $D_1, D_2, \ldots$ are i.i.d. samples from a distribution with $\mathbb{P}(D > b) = b^{-\delta}L(b)$, for some $\delta > 1$ and $L(\cdot)$ being a slowly varying function, i.e., $L(b)/L(tb) \to 1$ for $b \to \infty$, for any t.

Theorem 3.3.10 *As $b \to \infty$,*

$$\pi_\alpha(b) \sim \frac{\nu}{c_1-\mu} \frac{1}{\delta-1} \left(\frac{c_1-\mu}{c_1-c_2} - \alpha \left(\frac{c_2-\mu}{c_1-c_2} \right) \right)^{1-\delta} \cdot b^{1-\delta} L(b).$$

The proof consists of a lower bound that focuses on the probability of the single most likely event, in conjunction with an upper bound that shows that all other scenarios (for instance those with multiple big jumps) yield negligible contributions.

The lower bound is relatively straightforward, and provides interesting insights into the way the rare event under consideration occurs. The upper bound requires more work; the line of reasoning resembles that of earlier papers, e.g. [17, 181]. For more details we refer to [117].

3.4 Two-class priority queue

In the previous section we analyzed joint overflow in the first and second queue of a tandem system. This analysis was possible due to the fact that we had explicit knowledge of both the first buffer and total buffer contents. In this section we use the same arguments to solve the two-class priority system. We remark that the similarity between tandem and priority systems was already observed, see e.g. [59].

We consider a priority queue with service rate c, fed by traffic of two classes, each with its own queue. Traffic of class h does not 'see' class l, and therefore class h is referred to as the *high-priority* class, whereas the other class l is referred to as the *low-priority* class. The input process of class i is a Brownian motion $\{B_i(t), t \in \mathbb{R}\}$, $i = h, l$. Throughout this section it is assumed that $B_h(\cdot)$ is independent of $B_l(\cdot)$. The mean traffic rate of class i is denoted by μ_i, and the variance function of class i is given by $v_i(t) = \lambda_i t$, with $\lambda_i > 0$, $i = h, l$. It turns out that in this priority setting we cannot restrict ourself, without loss of generality, to centered processes, as was the case in the parallel and tandem settings, see also [130]. That is, we cannot assume, without loss of generality, that $\mu_l = \mu_h = 0$. To ensure stability, we assume that $\mu_h + \mu_l < c$. Also $\Gamma_h(s,t)$ and $\Gamma_l(s,t)$ are as defined before.

Remark: Notice that the above setting also covers the case where the number of sources of both classes are *not* equal. Assume for instance that there are n i.i.d. Brownian high-priority sources and $n\kappa$, with $\kappa > 0$, i.i.d. Brownian low-priority sources. Multiplying μ_l and $v_l(\cdot)$ by κ, and applying the fact that the Normal distribution is infinitely divisible, we arrive at n i.i.d. Brownian sources for both classes.

It is well-known that the steady-state workload of a single queue with Brownian input having mean rate μ and $v(t) = \lambda t$, is exponentially distributed with mean $\lambda/(2(c-\mu))$. Since class h is not influenced by class l, it is also clear how the high-priority steady-state buffer content Q_h behaves: it is exponentially distributed with mean $\lambda_h/(2(c-\mu_h))$. Also, due to the work-conserving property of the priority queue, the steady-state total workload $Q_T := Q_l + Q_h$ is characterized: it is exponentially distributed with mean $(\lambda_l+\lambda_h)/(2(c-\mu_l-\mu_h))$. However, no closed-form expressions for the joint overflow probabilities

$$\mathbb{P}(Q_h > b_h, Q_T > b_T), \tag{3.30}$$

and

$$\mathbb{P}(Q_h > b_h, Q_l > b_l), \tag{3.31}$$

where $b_T \geq b_h, b_l \geq 0$ are scalars, are known.

In contrast to the two-class parallel and two-class tandem queue, where we were able to obtain closed-form expression for the joint overflow probabilities, it is now hard to derive explicit expressions for (3.30) and (3.31), if possible at all. Fortunately, we are able to calculate the corresponding decay rates by using a version of 'Schilder' that relates to two-dimensional Gaussian processes. The description of this framework is nearly similar to the one presented in Chapter 2, but slightly more involved. We refer to [130] for a complete description of this setting. Since one requires similar techniques as were used before to derive the decay rates, we present the theorems below without proof.

Reich's formula [155] states that

$$Q_h = \sup_{s\geq 0}\{-B_h(-s) - cs\} \quad \text{and} \quad Q_T = \sup_{t\geq 0}\{-B_h(-t) - B_l(-t) - ct\}. \tag{3.32}$$

Let s^* and t^* denote an optimizing s and t in (3.32). Now, $-s^*$ ($-t^*$) can be interpreted as the beginning of the busy period of Q_h (Q_T) containing time 0. Clearly $s^* \leq t^*$, and therefore (3.30) can be rewritten as $\mathbb{P}(B_h(\cdot), B_l(\cdot)) \in Z)$, with

$$Z := \left\{ \overline{f} \in \Omega \times \Omega \;\middle|\; \begin{array}{l} \exists t \geq 0 : \exists s \in [0,t] : \\ -f_h(-s) > b_h + cs; \\ -f_h(-t) - f_l(-t) > b_T + ct \end{array} \right\}, \tag{3.33}$$

where

$$\overline{f}(t) = (\overline{f}_h(t), \overline{f}_l(t)) := \left(\frac{f_h(t) - \mu_h t}{\sqrt{\lambda_h}}, \frac{f_l(t) - \mu_l t}{\sqrt{\lambda_l}} \right).$$

Using that $Q_l = Q_T - Q_h$, (3.31) can be expressed as $\mathbb{P}(B_h(\cdot), B_l(\cdot)) \in Z^*)$, with

$$Z^* := \left\{ \overline{f} \in \Omega \times \Omega \;\middle|\; \begin{array}{l} \exists t \geq 0 : \exists s \in [0,t] : \forall u \in [0,t] : \\ -f_h(-s) > b_h + cs; \\ f_h(-u) - f_h(-t) - f_l(-t) > b_l + ct - cu \end{array} \right\}. \tag{3.34}$$

In order to apply 'Schilder', we feed this network by n i.i.d. high-priority Brownian sources and n i.i.d. low-priority Brownian sources. The link rates and buffer thresholds are also scaled by n: nc, nb_h, nb_l, and nb_T, respectively.

Using (3.33), $\mathbb{P}(Q_{h,n} > nb_h, Q_{T,n} > nb_T)$ can be expressed as

$$\mathbb{P}\left(\left(\frac{1}{n}\sum_{i=1}^{n} B_{h,i}(\cdot), \frac{1}{n}\sum_{i=1}^{n} B_{l,i}(\cdot) \right) \in Z \right).$$

From 'Schilder' it follows that

$$M(\overline{b}) := -\lim_{n\to\infty}\frac{1}{n}\log\mathbb{P}(Q_{h,n} > nb_h, Q_{T,n} > nb_T) = \inf_{f\in Z} I(\overline{f}),$$

where $\overline{b} = (b_h, b_T)$.

Similarly, using (3.34), we can write

$$\mathbb{P}(Q_{h,n} > nb_h, Q_{l,n} > nb_l) = \mathbb{P}\left(\left(\frac{1}{n}\sum_{i=1}^{n} B_{h,i}(\cdot), \frac{1}{n}\sum_{i=1}^{n} B_{l,i}(\cdot)\right) \in Z^*\right).$$

Hence, 'Schilder' gives

$$N(\tilde{b}) := -\lim_{n\to\infty}\frac{1}{n}\log\mathbb{P}(Q_{h,n} > nb_h, Q_{l,n} > nb_l) = \inf_{f\in Z^*} I(\overline{f}),$$

where $\tilde{b} = (b_h, b_l)$.

Define

$$\beta_- = \frac{\lambda_h(c-\mu_h-2\mu_l)-\lambda_l(c-\mu_h)}{\lambda_h(c-\mu_h-3\mu_l)-\lambda_l(3c-3\mu_h-\mu_l)};$$

$$\beta_0 = \frac{\lambda_h(c-\mu_h-2\mu_l)-\lambda_l(c-\mu_h)}{(\lambda_h+\lambda_l)(c-\mu_l-\mu_h)};$$

$$\beta_+ = \frac{c-\mu_h}{c-\mu_h+\mu_l}.$$

It can be verified that $\beta_- = \beta_0 = 0 < \beta_+ < 1$ if $(\lambda_h-\lambda_l)c = \lambda_h(\mu_h+2\mu_l)-\lambda_l\mu_h$, $\beta_- < 0 < \beta_0 < \beta_+ < 1$ if $(\lambda_h-\lambda_l)c > \lambda_h(\mu_h+2\mu_l)-\lambda_l\mu_h$, and $\beta_0 < 0 < \beta_- < \beta_+ < 1$ if $(\lambda_h-\lambda_l)c < \lambda_h(\mu_h+2\mu_l)-\lambda_l\mu_h$. Also, let

$$p(t) := \frac{(b_h+(c-\mu_h)t)^2}{2\lambda_h t} + \frac{(b_T-b_h-\mu_l t)^2}{2\lambda_l t},$$

and

$$\gamma := \sqrt{\frac{\lambda_h(b_T-b_h)^2+\lambda_l b_h^2}{\lambda_h\mu_l^2+\lambda_l(c-\mu_h)^2}}.$$

Theorem 3.4.1 *Suppose* $(\lambda_h-\lambda_l)c < \lambda_h(\mu_h+2\mu_l)-\lambda_l\mu_h$. *Then it holds that*

$$M(\overline{b}) = \begin{cases} \frac{2(b_T\lambda_h(c-\mu_h-\mu_l)+b_h(\lambda_l(c-\mu_h)-\lambda_h(c-\mu_h-2\mu_l)))}{\lambda_h(\lambda_h+\lambda_l)} & \text{if } b_h/b_T \in [0,\beta_-]; \\ p(\gamma) & \text{if } b_h/b_T \in (\beta_-,\beta_+); \\ \frac{2(c-\mu_h)b_h}{\lambda_h} & \text{if } b_h/b_T \in [\beta_+,1]. \end{cases}$$

Theorem 3.4.2 *Suppose* $(\lambda_h - \lambda_l)c \geq \lambda_h(\mu_h + 2\mu_l) - \lambda_l\mu_h$. *Then it holds that*

$$M(\overline{b}) = \begin{cases} \frac{2(c-\mu_h-\mu_l)b_T}{\lambda_h+\lambda_l} & \text{if } b_h/b_T \in [0, \beta_0]; \\ p(\gamma) & \text{if } b_h/b_T \in (\beta_0, \beta_+); \\ \frac{2(c-\mu_h)b_h}{\lambda_h} & \text{if } b_h/b_T \in [\beta_+, 1]. \end{cases}$$

Theorem 3.4.3 *Suppose* $(\lambda_h - \lambda_l)c < \lambda_h(\mu_h + 2\mu_l) - \lambda_l\mu_h$. *Then it holds that*

$$N(\tilde{b}) = \begin{cases} \frac{2(b_l\lambda_h(c-\mu_h-\mu_l)+b_h(\lambda_l(c-\mu_h)+\lambda_h\mu_l))}{\lambda_h(\lambda_h+\lambda_l)} & \text{if } b_h/(b_h+b_l) \in [0, \beta_-]; \\ p(\gamma) & \text{if } b_h/(b_h+b_l) \in (\beta_-, \beta_+); \\ \frac{2(c-\mu_h)b_h}{\lambda_h} & \text{if } b_h/(b_h+b_l) \in [\beta_+, 1]. \end{cases}$$

Theorem 3.4.4 *Suppose* $(\lambda_h - \lambda_l)c \geq \lambda_h(\mu_h + 2\mu_l) - \lambda_l\mu_h$. *Then it holds that*

$$N(\tilde{b}) = \begin{cases} p(\gamma) & \text{if } b_h/(b_h+b_l) \in [0, \beta_+); \\ \frac{2(c-\mu_h)b_h}{\lambda_h} & \text{if } b_h/(b_h+b_l) \in [\beta_+, 1]. \end{cases}$$

Chapter 4

Delay in Generalized Processor Sharing

In the previous two chapters we considered (simple networks of) Brownian queues. As mentioned before, the goal of Chapters 2 and 3 was to develop techniques that can be used to analyze GPS systems. In this chapter we will exploit these techniques to derive the delay asymptotics in a GPS system, which was one of the problems stated in Section 1.7.

In the literature, hardly any results are available on the delay asymptotics in a GPS system. A two-class GPS system, in a discrete-time setting, in which the input traffic is assumed to be SRD, was studied in [151], and the logarithmic asymptotics of the probability that the delay exceeds some large value were derived. In this chapter we generalize the results of [151] by deriving the delay asymptotics in case of a continuous-time setting and/or LRD traffic.

We first derive bounds on the delay probability in a two-class GPS system with general input processes, assuming that the inputs have stationary increments. We focus on a two-class system, as the majority of the traffic can broadly be categorized into streaming and elastic traffic, see e.g. [157], each one having its own QoS requirements. We next consider the situation of n input processes of both classes, scale the link capacity with n as well, and let n grow large. This many-sources regime is motivated by the fact that, particularly in the core of the network, resources are commonly shared by a large number of flows at the same time. In this many-sources framework, we apply Schilder's sample-path large deviations theorem to calculate the decay rates of these bounds in the important case of Gaussian inputs, which cover both SRD and LRD traffic. We note that this work is related to [129], where the authors derive (lower bounds on) the decay rate of the overflow probability in a two-class GPS system; for other related work, see [7, 126, 132]. We show that there exist two closed intervals of GPS weight values in which the bounds are tight: one containing the special case that class 1 has priority, and the other containing the case that class 2 has priority. For the remaining middle interval, we derive bounds on the decay rate. In the special case of Brownian inputs we obtain transparent closed-form expressions.

The remainder of this chapter is organized as follows. In Section 4.1 we describe the two-class GPS model. In Section 4.2 we derive bounds on the delay probability. In Section 4.3 we specialize to Gaussian traffic in a many-sources setting: using Schilder's theorem and the bounds mentioned above, we derive (bounds on) the corresponding decay rate.

4.1 Queueing model

In this chapter we consider a two-class GPS system, served with rate c. Each class has its own queue, and is assigned a weight $\phi_i \geq 0$, $i = 1, 2$. Without loss of generality it is assumed that $\phi_1 + \phi_2 = 1$. The weight ϕ_i determines the guaranteed minimum rate $\phi_i c$ for class i. If a class does not fully use this minimum rate, then the excess capacity becomes available for the other class. Note that GPS is a work-conserving scheduling discipline, i.e., the server works at full speed as long as at least one of the queues is non-empty.

We focus on the delay experienced by a packet ('fluid molecule') of a particular class, say class 1, having arrived at an arbitrary point in time, the so-called *virtual delay*. We assume that the system is stable, so that the delay is bounded almost surely. Also, without loss of generality we assume that the packet arrives at time 0. We denote the delay experienced by this packet by $D_1 \equiv D_1(0)$. Clearly,

$$p(d) := \mathbb{P}(D_1 > d) = \mathbb{P}(Q_1 > S_1(0, d)), \tag{4.1}$$

where $Q_i \equiv Q_i(0)$ is the steady-state workload of class i, and $S_i(s, t)$ is the amount of service received by class i in the interval $(s, t]$. Let $X_i(s, t)$ be the amount of traffic generated by class i in the interval $[s, t]$. Throughout this chapter we assume that $X_1(s, t)$ is independent of $X_2(s, t)$, so $\mathbb{C}\mathrm{ov}(X_1(s, t), X_2(s, t)) = 0$, $s \leq t$.

4.2 Bounds on the virtual delay probability

In this section we derive bounds on $p(d)$, which apply to all input processes that have stationary increments; stationarity of the increments means that the distribution of $X_i(s, s + t)$ does not depend on s, but just on the interval length t. We will use these bounds in Section 4.3 to derive (bounds on) the exponential decay rate of $p(d)$ in the many-sources setting.

To derive a lower bound on $p(d)$, we need to find an upper bound on $S_1(0, d)$, as follows from (4.1). As $S_1(0, d) \leq cd - S_2(0, d)$, this is equivalent to finding a lower bound on $S_2(0, d)$. Now, we have to distinguish between two scenarios: (i) queue 2 is continuously backlogged in the interval $(0, d]$ and (ii) queue 2 is empty at some time in $(0, d]$. In case (i) we have that $S_2(0, d) = \phi_2 cd$, because the second class receives at least its guaranteed service rate in the interval $(0, d]$, and class 1 is continuously

backlogged by definition (otherwise it cannot experience a delay of d), thus claiming at least its guaranteed rate in the interval $(0, d]$. In case (ii) we need a different approach to derive a lower bound on $S_2(0, d)$. Let z denote the last time in $(0, d]$ that the second queue was empty, that is $z := \max\{v \in (0, d] : Q_2(v) = 0\}$. This yields

$$\begin{aligned} S_2(0,d) &= S_2(0,z) + S_2(z,d) = Q_2 + X_2(0,z) + \phi_2 c(d-z) \\ &\geq \inf_{u\in(0,d]}\{X_2(0,u) + \phi_2 c(d-u)\}. \end{aligned} \tag{4.2}$$

Note that (4.2) will not exceed $\phi_2 cd$. That is, it is also a lower bound on $S_2(0, d)$ in case (i). Therefore, we find the following upper bound:

$$S_1(0,d) \leq cd - S_2(0,d) \leq cd - \inf_{u\in(0,d]}\{X_2(0,u) + \phi_2 c(d-u)\}.$$

Hence, we obtain

$$p(d) \geq \mathbb{P}\left(Q_1 > cd - \inf_{u\in(0,d]}\{X_2(0,u) + \phi_2 c(d-u)\}\right). \tag{4.3}$$

So far no explicit expressions have been found for the steady-state buffer content distribution of a particular class in a GPS system. In other words: we do not know the distribution of Q_1, which makes the lower bound (4.3) not very useful; we would rather like to have a bound that is in terms of the input processes X_1 and X_2 only. Using that, for $b \geq 0$,

$$\mathbb{P}(Q_1 > b) = \mathbb{P}\left(\bigcup_{x\geq 0}\{Q_1 + Q_2 > x + b, Q_2 \leq x\}\right),$$

we find that (4.3) can be rewritten as

$$p(d) \geq \mathbb{P}\left(\bigcup_{x\geq 0}\left\{Q_1 + Q_2 > x + cd - \inf_{u\in(0,d]}\{X_2(0,u) + \phi_2 c(d-u)\}, Q_2 \leq x\right\}\right). \tag{4.4}$$

But now observe that $Q_1 + Q_2$ is the steady-state workload of the total queue, and hence, due to the work-conserving nature of GPS, Reich's identity [155] implies that

$$Q_1 + Q_2 = \sup_{t\geq 0}\{X_1(-t,0) + X_2(-t,0) - ct\}. \tag{4.5}$$

Also, again by Reich's identity,

$$Q_2 = \sup_{s\geq 0}\{X_2(-s,0) - S_2(-s,0)\}. \tag{4.6}$$

The negative of the optimizing t (s), denoted by t^* (s^*), can be interpreted as the beginning of the busy period of the total (second) queue containing time 0. Clearly,

this entails that $s^* \leq t^*$. Now, using (4.5) and (4.6), we have that (4.4) can be expressed as

$$p(d) \geq \mathbb{P}\left(\begin{array}{l} \exists x \geq 0, t \geq 0 : \forall s \in [0,t] : \forall u \in (0,d] : \\ X_1(-t,0) + X_2(-t,u) > x + ct + \phi_1 cd + \phi_2 cu; \\ X_2(-s,0) \leq x + S_2(-s,0) \end{array} \right). \tag{4.7}$$

From (4.7) we conclude that, in order to find a lower bound on $p(d)$ that only depends on the input processes X_1 and X_2, we have to find a lower bound on $S_2(-s,0)$.

Lemma 4.2.1 *$p(d)$ is lower bounded by*

$$\mathbb{P}\left(\begin{array}{l} \exists x \geq 0, t \geq 0 : \forall s \in [0,t] : \forall u \in (0,d] : \\ X_1(-t,0) + X_2(-t,u) > x + ct + \phi_1 cd + \phi_2 cu; \\ X_2(-s,0) \leq x + \phi_2 cs \end{array} \right).$$

Proof: Since $-s^*$ denotes the beginning of the the busy period, queue 2 is continuously backlogged in the interval $(-s^*, 0]$, and therefore $S_2(-s^*,0) \geq \phi_2 cs^*$. This implies that the right-hand side of (4.7) is lower bounded by the stated, and therefore also $p(d)$. □

Likewise, to derive an upper bound on $p(d)$ we need to find a lower bound on $S_1(0,d)$. A first lower bound on $S_1(0,d)$ is clearly given by $S_1(0,d) \geq cd - Q_2 - X_2(0,d)$. This is a direct implication of the fact that, in an interval $(0,d]$, a queue never claims more than the workload at time 0, increased by the amount of traffic arriving at this queue in $(0,d]$.

Lemma 4.2.2 *$p(d)$ is upper bounded by*

$$\mathbb{P}\left(\exists t \geq 0 : X_1(-t,0) + X_2(-t,d) > ct + cd\right).$$

Proof: Since $S_1(0,d) \geq cd - Q_2 - X_2(0,d)$, we have

$$p(d) \leq \mathbb{P}(Q_1 > cd - Q_2 - X_2(0,d)) = \mathbb{P}(Q_1 + Q_2 > cd - X_2(0,d)).$$

Using (4.5), it is easily seen that the right-hand side is equivalent to the stated. □

Class 1 can only experience a delay of d if class 1 is continuously backlogged in the interval $(0,d]$. This implies that $S_1(0,d) \geq \phi_1 cd$, from which we deduce the following second upper bound.

Lemma 4.2.3 *$p(d)$ is upper bounded by*

$$\mathbb{P}\left(\begin{array}{l} \exists x \geq 0, t \geq 0 : \forall s \in [0,t] : \exists v \in [0,s] : \\ X_1(-t,0) + X_2(-t,0) > x + ct + \phi_1 cd; \\ X_1(-s,-v) + X_2(-s,0) \leq x + cs - \phi_1 cv \end{array} \right).$$

Proof: Since $S_1(0,d) \geq \phi_1 cd$, we have that $p(d) \leq \mathbb{P}(Q_1 > \phi_1 cd)$. In Section 3 of [129] it is shown that $\mathbb{P}(Q_1 > \phi_1 cd)$ is upper bounded by the stated. □

Notice the similarity between the lower bound of Lemma 4.2.1 and the upper bound of Lemma 4.2.3.

4.3 Decay rate of the virtual delay probability

In this section we derive (bounds on) the decay rate of the virtual delay probability in case of Gaussian inputs. We consider a many-sources setting, where the link capacity is scaled proportionally to the number of sources. In the special case of Brownian inputs we obtain closed-form expressions.

4.3.1 Gaussian input traffic

Let class i consist of a superposition of n, $n \in \mathbb{N}$, i.i.d. flows (or: sources), $i = 1, 2$; the analysis can easily be extended to the case of an unequal number of sources, see the remark in Section 3.4. Let the service capacity be nc. A class-i flow behaves as a Gaussian process with stationary increments $\{A_i(t), t \in \mathbb{R}\}$, with $A_i(0) \equiv 0$. Also, let the mean traffic rate and variance function of a single class-i flow be denoted by $\mu_i > 0$ and $v_i(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$, respectively, $i = 1, 2$. This mean rate and variance curve fully characterize the probabilistic behavior of the traffic process $A_i(\cdot)$. To guarantee stability we assume that $\mu_1 + \mu_2 < c$. With $A_i(s,t) := A_i(t) - A_i(s)$ denoting the amount of traffic generated by a single flow of type i in the interval $[s,t]$, $A_i(s,t)$ has a Normal distribution with $\mathbb{E}A_i(s,t) = \mu_i \cdot (t-s)$ and $\mathbb{V}\mathrm{ar}A_i(s,t) = v_i(t-s)$. As before, $\overline{A}_i(t) := A_i(t) - \mu_i t$ denotes the centered process. Recall that the covariance function $\Gamma_i(s,t)$ can be written as

$$\begin{aligned} \Gamma_i(s,t) &:= \mathbb{C}\mathrm{ov}\left(A_i(s), A_i(t)\right) = \mathbb{C}\mathrm{ov}\left(\overline{A}_i(s), \overline{A}_i(t)\right) \\ &= \frac{1}{2}\left(v_i(s) + v_i(t) - v_i(t-s)\right), \end{aligned} \tag{4.8}$$

for all $0 < s < t$. We impose Assumptions A1-A3 on $v_i(\cdot)$, $i = 1, 2$, see Chapter 2.

4.3.2 Decay rate

In this subsection we derive (bounds on) the decay rate corresponding to the virtual delay probability

$$p_n(d) := \mathbb{P}(Q_{1,n} > S_{1,n}(0,d)), \quad n \to \infty,$$

where $Q_{1,n} \equiv Q_{1,n}(0)$ is the steady-state class-1 buffer content and $S_{1,n}(0,d)$ is the amount of service received in the interval $(0,d]$ by class 1, in a system with n class-i inputs, $i = 1, 2$, that has service capacity nc.

Using 'Schilder' in a two-dimensional framework, it follows that

$$J(d) := -\lim_{n\to\infty} \frac{1}{n} \log p_n(d) = \inf_{f\in L} I(f) = \inf_{f\in \overline{L}} I(f), \tag{4.9}$$

where the open (closed) set L ($\overline{L}$) consists of all paths (f_1, f_2) that give a delay larger (larger or equal) than d. Recall that we refer to the path in $\overline{L}$ (and likewise in L) that minimizes the decay rate, i.e., $f^* = (f_1^*, f_2^*)$, as the most probable path (MPP). Informally speaking, given that the rare event occurs, with overwhelming probability a delay of d is achieved by a path 'close to' the MPP, cf. [7].

Recall that in Section 4.2 we derived bounds on $p(d)$. In this subsection we will exploit these bounds, to derive (lower bounds on) $J(d)$. Note that the decay rates of the upper (lower) bounds on $p(d)$ are lower (upper) bounds on $J(d)$ for all $\phi_2 \in [0, 1]$.

Class 2 in overload

We first focus on the regime $\phi_2 \in [0, \mu_2/c]$, i.e., class 2 in overload, and we derive an exact expression for the decay rate of $p_n(d)$. Recall that $S_1(d) \geq \phi_1 cd$ is required in order to have a delay of d. This yields

$$p_n(d) \leq \mathbb{P}(Q_{1,n} \geq n\phi_1 cd) \leq \mathbb{P}(Q_{1,n}^{n\phi_1 c} \geq n\phi_1 cd),$$

where $Q_{1,n}^a \equiv Q_{1,n}^a(0)$ denotes the stationary workload of queue 1 if it is served in isolation at constant rate a.

Lemma 4.3.1 *If $\phi_2 \in [0, \mu_2/c]$, then*

$$J(d) = \inf_{t\geq 0} \frac{(\phi_1 cd + (\phi_1 c - \mu_1)t)^2}{2v_1(t)}. \tag{4.10}$$

Let t^* be the optimizer in the above equation. Then, the MPP is given by

$$\begin{aligned} f_1^*(r) &= \begin{cases} -\mathbb{E}\left(A_1(r,0) | A_1(-t^*,0) = \phi_1 c(t^*+d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_1(0,r) | A_1(-t^*,0) = \phi_1 c(t^*+d)\right) & \text{for } r > 0; \end{cases} \\ f_2^*(r) &= \begin{cases} -\mathbb{E}\left(A_2(r,0) | A_1(-t^*,0) = \phi_1 c(t^*+d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_2(0,r) | A_1(-t^*,0) = \phi_1 c(t^*+d)\right) & \text{for } r > 0. \end{cases} \end{aligned} \tag{4.11}$$

Proof: The decay rate $J^L(d)$ of $\mathbb{P}(Q_{1,n}^{n\phi_1 c} \geq n\phi_1 cd)$ in case $\phi_2 \in [0, \mu_2/c]$ is given in Theorem 6.1 of [129]. Note that because $\mathbb{P}(Q_{1,n}^{n\phi_1 c} \geq n\phi_1 cd)$ is an upper bound on the delay probability, its decay rate $J^L(d)$ is a lower bound on $J(d)$. In addition, in Section 6 of [129] the authors derived the MPP $\tilde{f} = (\tilde{f}_1, \tilde{f}_2)$ corresponding to $J^L(d)$ using (2.12). The decay rate $J^L(d)$ is given by (4.10) and $\tilde{f}$ is given by (4.11). What is left to show is that $J(d) = J^L(d)$ and $f^* = \tilde{f}$.

Finding an upper bound $J^U(d)$ on $J(d)$ is a matter of computing the rate function ('cost') of a path in $\overline{L}$, i.e., a path that produces a delay of at least d. Using (2.4), it can be verified that

$$\tilde{f}_1(r) = \mu_1 r - \frac{(\phi_1 cd + (\phi_1 c - \mu_1)t^*)}{v_1(t^*)}\Gamma_1(-r, t^*) \quad \text{for } r \in (-t^*, 0];$$

$$\tilde{f}_2(r) = \mu_2 r \quad \text{for } r \in (-t^*, d].$$

This path is such that, at time 0, queue 1 has buffer content $\phi_1 cd$ (as $Q_1(-t^*) = 0$), and such that queue 2 continuously claims service rate $\phi_2 c$ in the interval $(0, d]$ (as $\mu_2 \geq \phi_2 c$), i.e., a service rate $\phi_1 c$ is available for class 1 in the interval $(0, d]$. Hence, we conclude that the path $\tilde{f}$ results in a delay of exactly d, i.e., $\tilde{f} \in \overline{L}$, implying that $f^* = \tilde{f}$ and, using (2.12), $J^U(d) = J^L(d) = J(d)$. □

Class 2 in underload

We now consider the regime $\phi_2 \in (\mu_2/c, 1]$ and derive the decay rate $J(d)$. In the analysis below the following critical class 2 weight is of importance:

$$\phi_2^F := \frac{\mu_2}{c} + \frac{v_2'(d - r^*) + v_2'(t^* + r^*)}{2(v_1(t^*) + v_2(t^* + d))}\left(\left(1 - \frac{\mu_1 + \mu_2}{c}\right)t^* + \left(1 - \frac{\mu_2}{c}\right)d\right), \quad (4.12)$$

where t^* is minimizer of

$$\inf_{t \geq 0} \frac{(c(t + d) - \mu_1 t - \mu_2(t + d))^2}{2v_1(t) + 2v_2(t + d)}, \quad (4.13)$$

and where r^* is maximizer of

$$\sup_{r \in (-t^*, d]} v_2'(d - r) + v_2'(t^* + r). \quad (4.14)$$

Note that $\phi_2^F > \mu_2/c$, as $v'(\cdot) > 0$ by Assumption A3, and possibly larger than 1. The next theorem presents the exact decay rate in case $\phi_2 \in [\phi_2^F, 1]$ (if this interval is non-empty).

Lemma 4.3.2 *If $\phi_2 \in [\phi_2^F, 1]$, then*

$$J(d) = \inf_{t \geq 0} \frac{(c(t + d) - \mu_1 t - \mu_2(t + d))^2}{2v_1(t) + 2v_2(t + d)}. \quad (4.15)$$

Let t^* be the optimizer in the above equation. Then, the MPP is given by

$$\begin{aligned} f_1^*(r) &= \begin{cases} -\mathbb{E}\left(A_1(r, 0) | A_1(-t^*, 0) + A_2(-t^*, d) = c(t^* + d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_1(0, r) | A_1(-t^*, 0) + A_2(-t^*, d) = c(t^* + d)\right) & \text{for } r > 0; \end{cases} \\ f_2^*(r) &= \begin{cases} -\mathbb{E}\left(A_2(r, 0) | A_1(-t^*, 0) + A_2(-t^*, d) = c(t^* + d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_2(0, r) | A_1(-t^*, 0) + A_2(-t^*, d) = c(t^* + d)\right) & \text{for } r > 0. \end{cases} \end{aligned} \quad (4.16)$$

Proof: Lemma 4.2.2 gives an upper bound on the delay probability. The decay rate $J^L(d)$ of this upper bound and the corresponding MPP $\tilde{f}$, the latter obtained by using (2.12), are well known (see for instance [7]), and given by (4.15) and (4.16), respectively. Below we show that $J^L(d) = J(d)$, or equivalently, that $\tilde{f} \in \overline{L}$ if $[\phi_2^F, 1]$ (similarly to the proof of Lemma 4.3.1).

Using (2.4), it can be verified that

$$\tilde{f}_1(r) = \mu_1 r - \frac{\Gamma_1(-r, t^*)}{v_1(t^*) + v_2(t^* + d)} \left((c - \mu_2)(t^* + d) - \mu_1 t^*\right) \quad \text{for } r \in (-t^*, 0];$$

$$\tilde{f}_2(r) = \mu_2 r - \frac{\Gamma_2(t^*, t^* + d) - \Gamma_2(t^* + r, t^* + d)}{v_1(t^*) + v_2(t^* + d)} \left((c - \mu_2)(t^* + d) - \mu_1 t^*\right) \text{ for } r \in (-t^*, d];$$

$$g_2(r) := \frac{\mathrm{d}\tilde{f}_2(r)}{\mathrm{d}r} = \mu_2 + \frac{v_2'(d - r) + v_2'(t^* + r)}{2v_1(t^*) + 2v_2(t^* + d)} \left((c - \mu_2)(t^* + d) - \mu_1 t^*\right),$$

i.e., $g_2(\cdot)$ represents the input rate of the path $\tilde{f}_2$, which is derived using (4.8). Note that $-t^*$ denotes the beginning of the busy period of the total queue, i.e., $Q_1(-t^*) = Q_2(-t^*) = 0$. Hence, if the input rate of class 2 is smaller than the guaranteed minimum rate $\phi_2 c$ for all $r \in (-t^*, d]$, then clearly queue 2 is empty in the interval $(-t^*, d]$. Let

$$r^* := \arg \max_{r \in (-t^*, d]} g_2(r),$$

i.e., r^* is the maximizer of (4.14). Then queue 2 is empty in the interval $(-t^*, d]$ if $\phi_2 \geq g_2(r^*)/c = \phi_2^F$. Now, note that the path $\tilde{f}$ is such that

$$Q_1 + Q_2 = Q_1 = cd - \tilde{f}_2(d) = \int_0^d (c - g_2(r))\mathrm{d}r,$$

in case $\phi_2 \in [\phi_2^F, 1]$. As class 2 only uses rate $g_2(r) \leq \phi_2 c$ in this case, this implies that rate $c - g_2(r)$ is available for the first class, $r \in (-t^*, d]$. It is not hard to see that, given $Q_1(0) = \int_0^d (c - g_2(r))\mathrm{d}r$ and service rate $c - g_2(r)$ for the first class, the experienced delay in steady state is exactly d. This proves that $\tilde{f} \in \overline{L}$, i.e., $f^* = \tilde{f}$ and $J(d) = J^L(d)$. □

We now focus on the remaining interval of weights: $\phi_2 \in \left(\mu_2/c, \phi_2^F\right)$. We have not succeeded in finding the exact decay rate in this middle regime, but we present two lower bounds; it is noted that lower bounds on the decay rate, which correspond to upper bounds on the probability $p_n(d)$, are usually of practical interest, as typically the network has to be designed such that $p_n(d)$ is small.

Clearly, the decay rate of Lemma 4.3.2 is also a lower bound on $J(d)$ in case $\phi_2 \in (\mu_2/c, \phi_2^F)$, as it is independent of ϕ_2 (see proof of Lemma 4.3.2). However, the corresponding path f^* is not necessarily contained in $\overline{L}$, and therefore it is not known whether the bound is tight.

We proceed by presenting a second lower bound on $J(d)$.

Lemma 4.3.3 *$J(d)$ is lower bounded by*

$$\inf_{x\geq 0}\inf_{t\geq 0}\sup_{s\in[0,t]}\inf_{v\in[0,s]}\inf_{\substack{z_1\geq x+ct+\phi_1 cd\\ z_2\leq x+cs-\phi_1 cv}}\Lambda(z_1,z_2),$$

where

$$\Lambda(z_1,z_2):=\tfrac{1}{2}\begin{pmatrix} z_1-(\mu_1+\mu_2)t\\ z_2-(\mu_1+\mu_2)s+\mu_2 v\end{pmatrix}^{\mathrm{T}}\Sigma^{-1}\begin{pmatrix} z_1-(\mu_1+\mu_2)t\\ z_2-(\mu_1+\mu_2)s+\mu_2 v\end{pmatrix},$$

and

$$\Sigma=\begin{pmatrix} v_1(t)+v_2(t) & \Gamma_1(s,t)-\Gamma_1(v,t)+\Gamma_2(s,t)\\ \Gamma_1(s,t)-\Gamma_1(v,t)+\Gamma_2(s,t) & v_1(s-v)+v_2(s)\end{pmatrix}.$$

Proof: Let the exact decay rate of the upper bound in Lemma 4.2.3 be denoted by $J^L(d)$. Define the set of paths

$$S^{s,t,v,x}:=\left\{\overline{f}\in\Omega\times\Omega\,\middle|\,\begin{array}{l}-f_1(-t)-f_2(-t)\geq x+ct+\phi_1 cd;\\ f_1(-v)-f_1(-s)-f_2(-s)\leq x+cs-\phi_1 cv\end{array}\right\},$$

where

$$S:=\bigcup_{x\geq 0}\bigcup_{t\geq 0}\bigcap_{s\in[0,t]}\bigcup_{v\in[0,s]}S^{s,t,v,x},$$

$\overline{f}(t)=(\overline{f}_1(t),\overline{f}_2(t)):=(f_1(t)-\mu_1 t, f_2(t)-\mu_2 t)$ is the centered path, and Ω is as defined in Equation (2.10). Then using Lemma 4.2.3 and 'Schilder' (recall that Schilder's theorem relates to *centered* Gaussian inputs), we obtain that

$$J(d)\geq\inf_{f\in S}I(\overline{f})=J^L(d)\geq\inf_{x\geq 0}\inf_{t\geq 0}\sup_{s\in[0,t]}\inf_{v\in[0,s]}\inf_{f\in S^{s,t,v,x}}I(\overline{f}).$$

The last inequality above was given in Theorem 4.1 of [129]. We now focus on the calculation of $\inf_{f\in S^{s,t,v,x}}I(\overline{f})$ for fixed s,t,v and x. Recognize that $\Lambda(z_1,z_2)$ is the large deviations rate function of the bivariate random variable

$$(A_1(-t,0)+A_2(-t,0), A_1(-s,-v)+A_2(-s,0)).$$

Finally, using Theorem 2.3.4,

$$\inf_{f\in S^{s,t,v,x}}I(\overline{f})=\inf_{\substack{z_1\geq x+ct+\phi_1 cd\\ z_2\leq x+cs-\phi_1 cv}}\Lambda(z_1,z_2).$$

This proves the stated. □

The following theorem summarizes Lemmas 4.3.1-4.3.3.

Theorem 4.3.4 *Suppose that class-1 and class-2 sources are Gaussian inputs. Then, under Assumptions A1-A3,*

$$J(d) = \begin{cases} (i) & \inf_{t\geq 0} \frac{(\phi_1 cd + (\phi_1 c - \mu_1)t)^2}{2v_1(t)} \quad \text{for } \phi_2 \in [0, \mu_2/c]; \\ (iii) & \inf_{t\geq 0} \frac{(c(t+d) - \mu_1 t - \mu_2(t+d))^2}{2v_1(t)+2v_2(t+d)} \quad \text{for } \phi_2 \in [\phi_2^F, 1], \end{cases}$$

and (ii) $J(d) \geq$

$$\max\left\{ \inf_{t\geq 0} \frac{(c(t+d) - \mu_1 t - \mu_2(t+d))^2}{2v_1(t) + 2v_2(t+d)}, \inf_{x\geq 0} \inf_{t\geq 0} \sup_{s\in[0,t]} \inf_{v\in[0,s]} \inf_{\substack{z_1 \geq x+ct+\phi_1 cd \\ z_2 \leq x+cs-\phi_1 cv}} \Lambda(z_1, z_2) \right\}$$

for $\phi_2 \in (\mu_2/c, \phi_2^F)$*, where* $\Lambda(z_1, z_2)$ *is as in Lemma 4.3.3.*

4.3.3 Brownian inputs

For most Gaussian inputs that satisfy A1-A3 there does not exist a closed-form expression for (bounds on) the decay rates as presented in Theorem 4.3.4. In case of Brownian inputs, however, we can derive explicit expressions for (bounds on) the decay rate $J(d)$. Brownian motions can be used to approximate weakly-dependent traffic streams as suggested by the celebrated Central Limit Theorem in functional form, see also [126]. We let the variance functions be characterized through $v_i(t) = \lambda_i t$, with $\lambda_i > 0$, $i = 1, 2$.

Straightforward calculus shows that (4.13) is minimized for

$$t^* = \begin{cases} d\left(\frac{c-\mu_2}{c-\mu_1-\mu_2} - 2\frac{\lambda_2}{\lambda_1+\lambda_2}\right) & \text{if } \frac{\mu_2}{c} + \frac{2\lambda_2}{\lambda_1+\lambda_2}\left(1 - \frac{\mu_1+\mu_2}{c}\right) \leq 1; \\ 0 & \text{otherwise.} \end{cases}$$

Since $v_i'(t) = \lambda_i$, we obtain from (4.12) that

$$\phi_2^F = \min\left\{ \frac{\mu_2}{c} + \frac{2\lambda_2}{\lambda_1+\lambda_2}\left(1 - \frac{\mu_1+\mu_2}{c}\right), 1 \right\}.$$

The following theorem characterizes the decay rate $J(d)$.

Proposition 4.3.5 *Suppose that class-1 and class-2 sources are Brownian inputs. Then,*

$$J(d) = \begin{cases} (i) & 2\phi_1 cd \frac{\phi_1 c - \mu_1}{\lambda_1} \quad \text{for } \phi_2 \in [0, \mu_2/c]; \\ (iii) & 2d\left(\frac{c-\mu_1-\mu_2}{\lambda_1+\lambda_2}\right)\left(\frac{(c-\mu_2)\lambda_1 + \mu_1\lambda_2}{\lambda_1+\lambda_2}\right) \quad \text{for } \phi_2 \in [\phi_2^F, 1], \end{cases}$$

and (ii) $J(d) \in$

$$\left[\tfrac{1}{2}d\left(\tfrac{(\phi_1 c + (\phi_1 c - \mu_1)u^*)^2}{\lambda_1 u^*} + \tfrac{(\phi_2 c - \mu_2)^2}{\lambda_2} u^*\right), \tfrac{1}{2}d\left(\tfrac{(\phi_1 c + (\phi_1 c - \mu_1)u^*)^2}{\lambda_1 u^*} + \tfrac{(\phi_2 c - \mu_2)^2}{\lambda_2}(u^* + 1)\right)\right]$$

for $\phi_2 \in (\mu_2/c, \phi_2^F)$, with the 'critical time scale' u^* given by

$$u^* := \frac{\phi_1 c}{\sqrt{(\phi_1 c - \mu_1)^2 + (\phi_2 c - \mu_2)^2 \frac{\lambda_1}{\lambda_2}}}.$$

Proof: Straightforward calculus shows that the optimizer of Lemma 4.3.1 is

$$t^* = \frac{\phi_1 cd}{\phi_1 c - \mu_1},$$

from which we obtain (i). In Lemma 4.3.2 the optimizer is

$$t^* = d\left(\frac{c - \mu_2}{c - \mu_1 - \mu_2} - 2\frac{\lambda_2}{\lambda_1 + \lambda_2}\right),$$

which yields (iii). The lower bound in (ii) follows from Lemma 4.3.3, and was proved in Theorem 5.6 of [126]. The upper bound in (ii) is a matter of calculating the cost of a path in $\overline{L}$. Consider the following path:

$$f_1(r) = \begin{cases} -\mathbb{E}\left(A_1(r,0) | A_1(-t^*,0) = \phi_1 c(t^* + d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_1(0,r) | A_1(-t^*,0) = \phi_1 c(t^* + d)\right) & \text{for } r > 0; \end{cases}$$

$$f_2(r) = \begin{cases} -\mathbb{E}\left(A_2(r,0) | A_2(-t^*,d) = \phi_2 (t^* + d)\right) & \text{for } r \leq 0; \\ \mathbb{E}\left(A_2(0,r) | A_2(-t^*,d) = \phi_2 c(t^* + d)\right) & \text{for } r > 0, \end{cases}$$

where $t^* = u^* d$. Using (2.4), it can be verified that this path is such that class 1 produces traffic at constant rate $\phi_1 c(t^* + d)/t^* > \phi_1 c$ in the interval $(-t^*, 0]$ and at constant rate μ_1 elsewhere, whereas class 2 produces traffic at constant rate $\phi_2 c$ in the interval $(-t^*, d]$ and at constant rate μ_2 elsewhere. This obviously leads to $Q_1(0) = \phi_1 cd$ (as $Q_1(-t^*) = Q_2(-t^*) = 0$), and thus a delay of d, as class 2 continuously claims its guaranteed rate in the interval $(-t^*, d]$. Using (2.19), the decay rate associated with f is therefore given by $I(\tilde{f}_1, \tilde{f}_2)$, with $\tilde{f}_i(t) := (f_i(t) - \mu_i t)/\sqrt{\lambda_i}$, $i = 1, 2$ (recall that (2.19) relates to *standard* Brownian inputs), which is equivalent to the desired expression. □

Chapter 5

Selection of optimal weights in Generalized Processor Sharing

In the previous chapter we derived the delay asymptotics in a GPS queue, which was one of the problems mentioned in Section 1.7. In this chapter we turn to the other problem mentioned. That is, in contrast to the previous chapter, where the GPS weights were assumed to be fixed beforehand, we now analyze the selection of optimal GPS weights.

The problem of mapping the QoS requirements on suitable GPS weights has received little attention in the literature, see the overview in Chapter 1. The results of [60] on the weight setting problem rely on the restrictive assumption of leaky-bucket controlled traffic. The contribution of this chapter is that we extend their results on the weight setting to a general and versatile class of input processes, covering a broad range of correlation structures, viz. the class of Gaussian inputs.

We consider a two-class GPS system with Gaussian traffic sources. The QoS criterion is that the loss probability should be kept below some class-specific value. The large deviations approximations of [132] on GPS for Gaussian inputs are the key tool in our analysis. As a first step, we use these approximations to find the admissible region for class 1 for fixed weights, i.e., all numbers of sources n_1, n_2 of class 1 and class 2 such that the QoS requirement of class 1 is met. By taking the intersection of the admissible region of both classes, we then obtain the admissible region (of the system), i.e., all combinations of flows that satisfy the QoS for both classes. In the special case of Brownian inputs, we explicitly determine the boundary of the admissible region.

We then explicitly derive the realizable region as the union of the admissible regions over all possible weight values, in case of Brownian inputs. A remarkable finding is that nearly the entire realizable region is achieved by one of the strict priority scheduling disciplines. A further key observation is that the QoS requirements and the buffer thresholds fully determine which class should have high priority, if

such a strict priority policy would be imposed. Importantly, the above two remarkable conclusions also hold for general Gaussian inputs. In the absence of an explicit description of the boundary of the realizable region, we have relied on extensive numerical experimentation.

The above results indicate that from an efficiency point of view GPS does *not* outperform a simple priority discipline. In other words, it suggests that there is hardly any efficiency improvement to be achieved by implementing GPS (compared to priority scheduling), in that the admissible region corresponding to some GPS weight vector, is contained in the admissible region corresponding to one of the priority cases. It is worth pointing out one important caveat. By assigning positive weights to all classes, GPS is capable of protecting a class against starvation when some other class misbehaves, as opposed to priority scheduling, where the low-priority class may be excluded from service over substantial time intervals.

The remainder of this paper is organized as follows. In Section 5.1 we describe our two-class GPS model with Gaussian inputs, and review the Mannersalo-Norros approximations [132] for loss probabilities, which consist of three regimes. In Section 5.2 the stable region is partitioned into three subsets, each subset corresponding to one of the three regimes. Using the partitioning of the stable region and the Mannersalo-Norros approximations, we derive the admissible region in Section 5.3. In Section 5.4 we consider Brownian inputs, and explicitly derive the boundary of the admissible region and the boundary of the realizable region. In Section 5.5 we perform numerical analysis. In particular, we consider systems shared by two types of applications with heterogeneous QoS requirements, and numerically derive the realizable regions.

5.1 Preliminaries

In this section we introduce the notation of the two-class GPS model and discuss Gaussian sources. Then we present approximations for the overflow probabilities.

5.1.1 Queueing model

We consider a model with two queues that share a server of rate c. Traffic of class i is buffered in queue i, $i = 1, 2$. The scheduling discipline is GPS, with weight $\phi_i \geq 0$ assigned to class i, $i = 1, 2$. Without loss of generality we assume that $\phi_1 + \phi_2 = 1$. The weight ϕ_i determines the guaranteed minimum rate for class i. If a class does not fully use the minimum rate, then the excess capacity becomes available to the other class.

5.1.2 Gaussian input traffic, overflow probabilities

As our first goal in Sections 5.2 and 5.3 is to characterize the admissible region (for a given weight vector), we first present the Mannersalo-Norros approximations [132]

for the overflow probabilities for given numbers of sources of both classes.

Let class 1 (class 2) consist of a superposition of n_1 (n_2) i.i.d. flows (or: sources), modeled as Gaussian processes with stationary increments. Clearly $n_1, n_2 \in \mathbb{N}_0$, but for convenience we let $n_1, n_2 \in \mathbb{R}_+$. We denote the mean traffic rate and variance function of a single class-i flow by $\mu_i > 0$ and $v_i(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$, respectively, for $i = 1, 2$; this mean rate and variance curve fully characterize the probabilistic behavior of the flow. Hence, if $A_i(s,t)$ denotes the amount of traffic generated by a single flow of type i in the interval $[s,t]$, then $\mathbb{E}A_i(s,t) = \mu_i \cdot (t-s)$ and $\mathbb{V}\mathrm{ar}A_i(s,t) = v_i(t-s)$. To guarantee stability we assume that $n_1\mu_1 + n_2\mu_2 \leq c$ (which we refer to as the 'capacity constraint'). We impose Assumptions A1-A3 on $v_i(\cdot)$, see Chapter 2.

The derivation of the admissible regions relies on the Mannersalo-Norros approximations [132] for the overflow probabilities; these require Assumptions A1 and A2. On the basis of extensive simulation experiments, Mannersalo & Norros [132] showed the accuracy of their approximations. Assumption A3 is needed in the proofs of some lemmas.

Let Q_i denote the stationary buffer content in the GPS model of class i, and $\triangle_i(n_1, n_2)$ the Mannersalo-Norros approximation of $-\log \mathbb{P}(Q_i > B_i)$. Define

$$\psi(t|n_1, n_2) := \frac{1}{2} \inf_{t \geq 0} \frac{(b_1 + (c - n_1\mu_1 - n_2\mu_2)t)^2}{n_1 v_1(t) + n_2 v_2(t)}. \tag{5.1}$$

We impose the following assumption on $\psi(t|n_1, n_2)$.

Assumption 5.1.1 *For any $(n_1, n_2) \in \mathbb{R}_+^2$ such that $n_1\mu_1 + n_2\mu_2 \leq c$, $\psi(t|n_1, n_2)$ has a unique minimizer $t^F(n_1, n_2)$.*

Clearly, $t^F(n_1, n_2)$ depends on (n_1, n_2), but for ease of notation, we will denote it by t^F in the remainder of this chapter. Due to Assumption A2, for any $(n_1, n_2) \in \mathbb{R}_+^2$ such that $n_1\mu_1 + n_2\mu_2 \leq c$, $\lim_{t\to 0} \psi(t|n_1, n_2) = \lim_{t\to\infty} \psi(t|n_1, n_2) = \infty$, and thus a minimizer t^F of $\psi(t|n_1, n_2)$ clearly exists, but it is not necessarily unique. We performed extensive numerical experiments with the often used variance functions $v_i(\cdot)$, e.g., fractional Brownian motions, and the Gaussian counterpart of the Anick-Mitra-Sondhi (AMS) [11] model (see also Section 5.5), and observed that t^F was unique in all considered cases, making this uniqueness assumption a weak assumption. In fact, it turned out to be a non-trivial exercise to find a situation with multiple minimizers, see Figure 5.1 for a rare example with two minimizers. By slightly increasing (n_1, n_2), we see that the minimizing t jumps from 0.2775 to 32.3631. For a related example, see Section 5 of [124].

Also, define

$$\phi_2^F := \frac{n_2\mu_2}{c} + \left(\frac{n_2 v_2(t^F) \left(b_1 + (c - n_1\mu_1 - n_2\mu_2)t^F\right)}{ct^F \left(n_1 v_1(t^F) + n_2 v_2(t^F)\right)} \right). \tag{5.2}$$

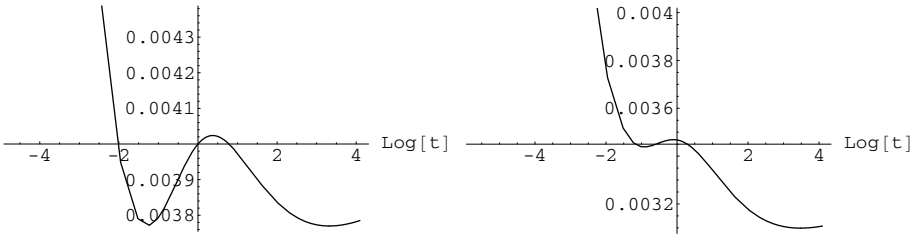

Figure 5.1: Left: The function $\psi(t|n_1, n_2)$ evaluated with parameters $n_1 = n_2 = 1$, $c = 1$, $b_1 = 0.04$, $\mu_1 = 0.4$, $\mu_2 = 0.5$, $v_1(t) = t^{0.55}$ and $v_2(t) = 1.45t^{1.98}$. The minimizers are $t_1^F = 0.2775$ and $t_2^F = 27.6741$, with $\psi(t_1^F|1,1) = \psi(t_2^F|1,1) = 0.00377$. Right: The same setting, but now with $n_1 = n_2 = 1.01$. The minimizer is $t^F = 32.3631$, with $\psi(t^F|1.01, 1.01) = 0.00310$.

Due to the uniqueness of t^F, ϕ_2^F is unique as well, and is larger than $n_2\mu_2/c$. Then

$$\triangle_1(n_1,n_2) = \begin{cases} (i) & \frac{1}{2}\inf_{t\geq 0}\frac{(b_1+(\phi_1 c-n_1\mu_1)t)^2}{n_1 v_1(t)} & \text{for } \phi_2 \in [0, \frac{n_2\mu_2}{c}]; \\ (ii) & \frac{1}{2}\inf_{t\geq 0}\left(\frac{(b_1+(\phi_1 c-n_1\mu_1)t)^2}{n_1 v_1(t)} + \frac{(\phi_2 c-n_2\mu_2)^2 t^2}{n_2 v_2(t)}\right) & \text{for } \phi_2 \in (\frac{n_2\mu_2}{c}, \phi_2^F); \\ (iii) & \frac{1}{2}\inf_{t\geq 0}\frac{(b_1+(c-n_1\mu_1-n_2\mu_2)t)^2}{n_1 v_1(t)+n_2 v_2(t)} & \text{for } \phi_2 \in [\phi_2^F, 1]. \end{cases}$$

The approximations $\triangle_2(n_1, n_2)$ are analogous; evidently, we can now approximate $\mathbb{P}(Q_i > B_i)$ by $\exp(-\triangle_i(n_1, n_2))$. We now heuristically explain the three regimes (i), (ii), (iii). As the first and the third have the easiest explanation we start there, before turning to the second regime.

In regime (i) we have that $\phi_2 c \leq n_2\mu_2$. That is, the mean traffic rate generated by class 2 exceeds the guaranteed rate of service to class 2 (we call this: class 2 in overload). Therefore, it is very likely that type-2 sources claim their guaranteed service rate $\phi_2 c$ essentially all the time. Hence, overflow in queue 1 resembles overflow in a FIFO queue with service rate $\phi_1 c$. The approximation $\triangle_1(n_1, n_2)$ of regime (i) is based on this principle, cf. [7]. The minimizing t represents the (most likely) length of the interval between the epoch queue 1 starts to build up, until it reaches buffer content b_1.

Regime (iii) requires ϕ_2 to be at least as large as ϕ_2^F. It can be verified (by using the explicit formulae for conditional means of Normal random variables) that ϕ_2^F is equal to the value of ϕ_2 for which

$$\mathbb{E}\left(A_2(-t^F, 0)|A_1(-t^F, 0) + A_2(-t^F, 0) = b_1 + ct^F\right) = \phi_2 c t^F. \tag{5.3}$$

Hence, if $\phi_2 \geq \phi_2^F$, conditioned on the total queue building up b_1 in t^F time units, then all this traffic is in queue 1, and queue 2 is essentially empty.

Regime (ii) applies if class 2 is underloaded, but $\phi_2 \leq \phi_2^F$. When the total queue reaches level b_1, it is now very likely that the queue of class 2 is non-empty. Hence, an additional constraint must be imposed to keep the buffer content of queue 2 small. The approximation is such that the flows of class 1 generate $b_1+\phi_1 ct$, while the class-2 sources generate $\phi_2 ct$ (i.e., the class-2 sources claim their guaranteed rate). Note that in the approximation it is used that the interval in which the class-2 sources claim rate $\phi_2 c$ coincides with the interval in which queue 1 builds up. For a refinement of this approximation we refer to [129], which allows scenarios in which the first queue starts to build up *before* the second queue reaches traffic rate $\phi_2 c$.

5.2 Partitioning of the stable region

In order to derive the admissible region (for given weights) of the two-class GPS system, we have to determine the admissible region of each class separately and then take the intersection of these two regions. In Sections 5.2 and 5.3, without loss of generality, we focus on the admissible region of the first class (i.e., the set of sources (n_1, n_2) for which the class-1 sources receive the desired QoS), as the second one can be treated in the same fashion. Before the admissible region of the first class can be obtained, which we will do in Section 5.3, we first determine all (n_1, n_2) for which (i) $\phi_2 \in [0, n_2\mu_2/c]$, (ii) $\phi_2 \in (n_2\mu_2/c, \phi_2^F)$ and (iii) $\phi_2 \in [\phi_2^F, 1]$, thus partitioning the stable region $T := \{(n_1, n_2) : n_1\mu_1 + n_2\mu_2 \leq c\}$ into three sets. In these three sets we can use the approximation of $\triangle_1(n_1, n_2)$ presented in Section 5.1.2.

Lemma 5.2.1 *Let $\phi_1 \in (0, 1)$. Then $T = T_1^i(\phi_1) \cup T_1^{ii}(\phi_1) \cup T_1^{iii}(\phi_1)$ for disjoint non-empty $T_1^i(\phi_1)$, $T_1^{ii}(\phi_1)$ and $T_1^{iii}(\phi_1)$, where*

$$\begin{aligned}
T_1^i(\phi_1) &:= \left\{(n_1, n_2) \in T : n_2 \geq \frac{\phi_2 c}{\mu_2}\right\};\\
T_1^{ii}(\phi_1) &:= \left\{(n_1, n_2) \in T : n_2 < \frac{\phi_2 c}{\mu_2}, \quad \frac{b_1 + (\phi_1 c - n_1\mu_1)t^F}{n_1 v_1(t^F)} > \frac{(\phi_2 c - n_2\mu_2)t^F}{n_2 v_2(t^F)}\right\};\\
T_1^{iii}(\phi_1) &:= \left\{(n_1, n_2) \in T : n_2 < \frac{\phi_2 c}{\mu_2}, \quad \frac{b_1 + (\phi_1 c - n_1\mu_1)t^F}{n_1 v_1(t^F)} \leq \frac{(\phi_2 c - n_2\mu_2)t^F}{n_2 v_2(t^F)}\right\},
\end{aligned}$$

such that regime (j) applies in $T_1^j(\phi_1)$, for $j \in \{i, ii, iii\}$.

Proof: $T_1^i(\phi_1)$ follows from the fact that we must have $\phi_2 \in [0, n_2\mu_2/c]$. In order to be in $T_1^{ii}(\phi_1)$ we must have that $\phi_2 \in (n_2\mu_2/c, \phi_2^F)$, or equivalently $n_2 < \phi_2 c/\mu_2$ and $\phi_2 < \phi_2^F$. The latter inequality can be rewritten as

$$\phi_2 < \frac{n_2\mu_2}{c} + \left(\frac{n_2 v_2(t^F)\left(b_1 + (c - n_1\mu_1 - n_2\mu_2)t^F\right)}{ct^F\left(n_1 v_1(t^F) + n_2 v_2(t^F)\right)}\right).$$

Multiply both sides with ct^F, and rearrange the right-hand side to obtain

$$\phi_2 ct^F < \left(\frac{n_2 v_2(t^F) \left(b_1 + ct^F - n_1\mu_1 t^F\right)}{n_1 v_1(t^F) + n_2 v_2(t^F)} \right) + \frac{n_1 v_1(t^F) n_2 \mu_2 t^F}{n_1 v_1(t^F) + n_2 v_2(t^F)}.$$

Multiplying both sides with $n_1 v_1(t^F) + n_2 v_2(t^F)$ and collecting 'equivalent terms' leads to

$$n_1 v_1(t^F) \left(\phi_2 ct^F - n_2\mu_2 t^F\right) < n_2 v_2(t^F) \left(b_1 + \phi_1 ct^F - n_1\mu_1 t^F\right).$$

Dividing both sides by $n_1 v_1(t^F)$ and $n_2 v_2(t^F)$ respectively gives

$$\frac{b_1 + (\phi_1 c - n_1\mu_1)t^F}{n_1 v_1(t^F)} > \frac{(\phi_2 c - n_2\mu_2)t^F}{n_2 v_2(t^F)}. \tag{5.4}$$

The characterization of $T_1^{iii}(\phi_1)$ follows similarly.

In case $\phi_1 \in (0,1)$, all three sets are non-empty, and this proves the stated. Note that $T = T_1^{iii}(0)$ for $\phi_1 = 0$ and $T = T_1^{i}(1)$ for $\phi_1 = 1$. □

Now consider the boundary between $T_1^{ii}(\phi_1)$ and $T_1^{iii}(\phi_1)$, i.e., combinations of (n_1, n_2) such that (5.4) holds with equality. For most of the $v_i(\cdot)$ curves we considered, this boundary could not be explicitly expressed in terms of a function $f_1(n_2) = n_1$; to compute the boundary, one needs to resort to numerical methods. However, some characteristics of $f_1(\cdot)$ can be derived and are presented in the following lemma.

Lemma 5.2.2 *The following statements can be made about $f(\cdot)$:*
(I) $f_1(0) = 0$*;*
(II) $f_1(\phi_2 c/\mu_2) = \phi_1 c/\mu_1$*;*
(III) $f_1(\cdot)$ *only intersects the capacity constraint at* $(n_1, n_2) = (\phi_1 c/\mu_1, \phi_2 c/\mu_2)$*;*
(IV) $f_1(\cdot)$ *only intersects the line* $n_2 = \phi_2 c/\mu_2$ *at* $(n_1, n_2) = (\phi_1 c/\mu_1, \phi_2 c/\mu_2)$*;*
(V) $f_1(\cdot)$ *only intersects the* n_1*-axis and* n_2*-axis at* $(n_1, n_2) = (0, 0)$.

Proof: If $(n_1, n_2) = (0, 0)$, then we have clearly equality in (5.4) (as both sides have value ∞), so this gives (I). We continue with (II). Take the point $(n_1, n_2) = (\phi_1 c/\mu_1, \phi_2 c/\mu_2)$. Then it follows that $t^F = \infty$, as $v_i(t)$ is increasing in t by A3, $i = 1, 2$. Plugging $t^F = \infty$ in (5.4), we find that there is equality there, no matter the value of b_1. We proceed with (III). Note that $f_1(\cdot)$ is the line where $\triangle_1(n_1, n_2)$ of regimes (ii) and (iii) have equal values. Next define $S := \{(n_1, n_2) | n_1\mu_1 + n_2\mu_2 = c\}$. We find that for all $(n_1, n_2) \in S$ we have that $\triangle_1(n_1, n_2)$ of regime (iii) equals 0 (as $v_i(t)$ is increasing in t by A3, $i = 1, 2$). Now note that the only $(n_1, n_2) \in S$ for which $\triangle_1(n_1, n_2)$ of regime (ii) equals zero is $(\phi_1 c/\mu_1, \phi_2 c/\mu_2)$. Thus, line $f_1(\cdot)$ only intersects the capacity constraint at $(n_1, n_2) = (\phi_1 c/\mu_1, \phi_2 c/\mu_2)$. We prove (IV) in a similar fashion. If $n_2 = \phi_2 c/\mu_2$, then for regime (ii):

$$\triangle_1(n_1, \phi_2 c/\mu_2) = \frac{1}{2} \inf_{t \geq 0} \frac{(b_1 + (\phi_1 c - n_1\mu_1)t)^2}{n_1 v_1(t)},$$

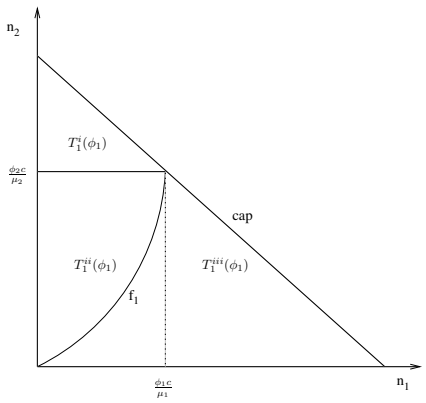

Figure 5.2: The typical partitioning of the stable region T

whereas for regime (iii):

$$\triangle_1(n_1, \phi_2 c/\mu_2) = \frac{1}{2} \inf_{t \geq 0} \frac{(b_1 + (\phi_1 c - n_1 \mu_1)t)^2}{n_1 v_1(t) + \frac{\phi_2 c}{\mu_2} v_2(t)}.$$

These two can only be equal if $n_1 = \phi_1 c/\mu_1$ as then the optimizer is $t = \infty$, and we obtain $\triangle_1(\phi_1 c/\mu_1, \phi_2 c/\mu_2) = 0$ for regimes (ii) and (iii). We conclude with (V). If $n_1 = 0$ or $n_2 = 0$ (but not both), then $\triangle_1(n_1, n_2)$ of regime (ii) equals ∞, whereas $\triangle_1(n_1, n_2)$ of regime (iii) is bounded. Hence, except for $(n_1, n_2) = (0, 0)$, $f_1(\cdot)$ cannot intersect the n_1-axis and n_2-axis. □

In our numerical experiments with the often used variance functions $v_i(\cdot)$, e.g., fractional Brownian motions, the Gaussian counterpart of the AMS model, and others as presented in [7], we observed that $f_1(\cdot)$ is strictly increasing, as depicted in Figure 5.2.

5.3 Analysis of the admissible region

In this section we analyze the admissible region of the first class (for given weights), i.e., all combinations of (n_1, n_2) that satisfy $\triangle_1(n_1, n_2) \geq \delta_1$, for some $\delta_1 > 0$. We show that this set consists of three disjoint subsets: $S_1(\phi_1) = S_1^i(\phi_1) \cup S_1^{ii}(\phi_1) \cup S_1^{iii}(\phi_1)$, with $S_1^j(\phi_1) \subset T_1^j$, $j \in \{i, ii, iii\}$, which we derive below. Finally, we present our main result that characterizes the boundary of $S_1(\phi_1)$. Again we concentrate on $S_1(\phi_1)$, but of course $S_2(\phi_1)$ can be treated analogously, thus determining the admissible region $S(\phi_1) := S_1(\phi_1) \cap S_2(\phi_1)$.

5.3.1 Region $S_1^i(\phi_1)$

We define $S_1^i(\phi_1)$ as the subset of $T_1^i(\phi_1)$ (see Section 5.2), for which $\triangle_1(n_1, n_2) \geq \delta_1$. That is,

$$\triangle_1(n_1, n_2) = \frac{1}{2} \inf_{t \geq 0} \frac{(b_1 + (\phi_1 c - n_1 \mu_1)t)^2}{n_1 v_1(t)} \geq \delta_1.$$

Rearranging and collecting terms yields

$$n_1 \leq \max \left\{ n_1 : \quad \forall t \geq 0 : X_t n_1^2 + Y_t n_1 + Z_t \geq 0 \right\},$$

where

$$\begin{aligned} X_t &:= \mu_1^2 t^2; \\ Y_t &:= -2b_1\mu_1 t - 2\phi_1 c \mu_1 t^2 - 2\delta_1 v_1(t); \\ Z_t &:= b_1^2 + \phi_1^2 c^2 t^2 + 2b_1 \phi_1 c t. \end{aligned}$$

This eventually leads to

$$\begin{aligned} n_1 \leq n_1^{Q_1} &:= \max \left\{ n_1 : \quad n_1 \leq \inf_{t \geq 0} \frac{-Y_t - \sqrt{Y_t^2 - 4X_t Z_t}}{2X_t} \right\} \\ &= \inf_{t \geq 0} \frac{-Y_t - \sqrt{Y_t^2 - 4X_t Z_t}}{2X_t}, \end{aligned} \tag{5.5}$$

and

$$n_1 \leq \max \left\{ n_1 : \quad n_1 \geq \inf_{t \geq 0} \frac{-Y_t + \sqrt{Y_t^2 - 4X_t Z_t}}{2X_t} \right\} = \infty.$$

Clearly, $n_1 \leq \infty$ always holds, so this constraint is redundant. It is noted that $\triangle_1(\phi_1 c/\mu_1, n_2)$ of regime (i) equals 0, as it is minimized for $t = \infty$ by A3. Since we require that $\triangle_1(n_1, n_2) \geq \delta_1 > 0$, this implies that $n_1^{Q_1} < \phi_1 c/\mu_1$. An example of a set $S_1^i(\phi_1)$ is depicted in Figure 5.3 (top, left).

5.3.2 Region $S_1^{ii}(\phi_1)$

In this regime $S_1^{ii}(\phi_1)$ consists of all combinations (n_1, n_2) in $T_1^{ii}(\phi_1)$ such that

$$\triangle_1(n_1, n_2) = \frac{1}{2} \inf_{t \geq 0} \left(\frac{(b_1 + (\phi_1 c - n_1 \mu_1)t)^2}{n_1 v_1(t)} + \frac{(\phi_2 c - n_2 \mu_2)^2 t^2}{n_2 v_2(t)} \right) \geq \delta_1.$$

Proceeding in the same manner as above, this reduces to

$$n_2 \leq g_1(n_1) := \inf_{t \geq 0} \frac{-Y_t - \sqrt{Y_t^2 - 4X_t Z_t}}{2X_t}, \tag{5.6}$$

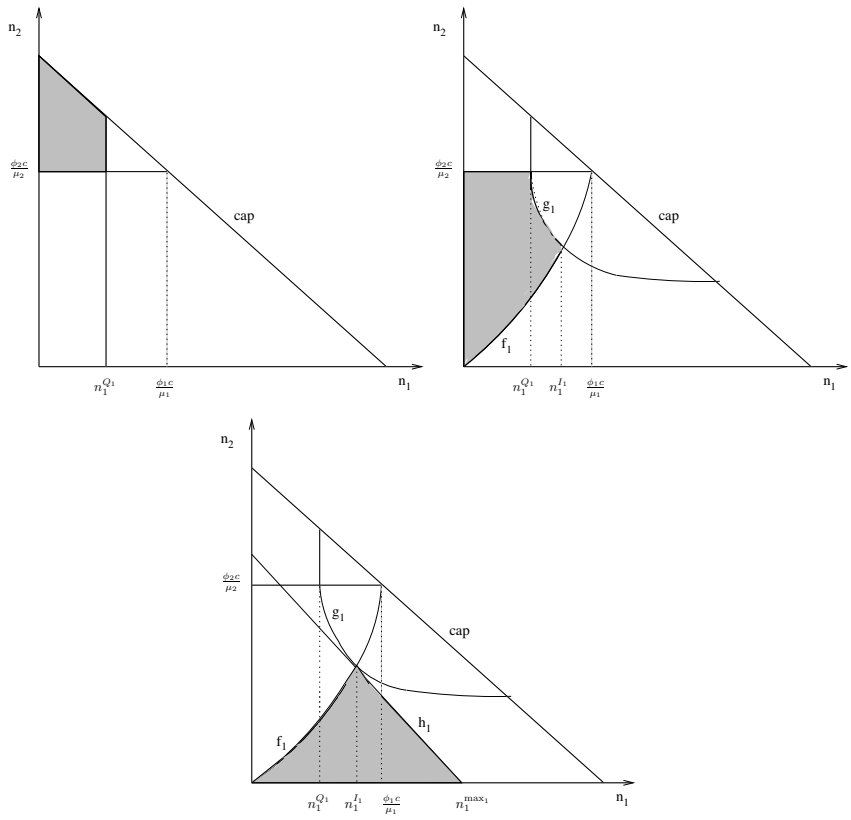

Figure 5.3: The typical partitioning of the admissible region of the first queue $S_1(\phi_1)$. Top, left: $S_1^i(\phi_1)$. Top, right: $S_1^{ii}(\phi_1)$. Bottom: $S_1^{iii}(\phi_1)$.

where

$$\begin{aligned} X_t &:= \mu_2^2 t^2 / v_2(t); \\ Y_t &:= \frac{(b_1 + (\phi_1 c - n_1\mu_1)t)^2}{n_1 v_1(t)} - \frac{2\phi_2 c \mu_2 t^2}{v_2(t)} - 2\delta_1; \\ Z_t &:= \phi_2^2 c^2 t^2 / v_2(t). \end{aligned}$$

As $g_1(\cdot)$ plays an important role in describing the boundary of $S_1(\phi_1)$, the remainder of this subsection is devoted to some structural properties of $g_1(\cdot)$. First notice that

$$\begin{aligned} &\frac{1}{2}\inf_{t\geq 0}\left(\frac{(b_1 + (\phi_1 c - n_1\mu_1)t)^2}{n_1 v_1(t)} + \frac{(\phi_2 c - n_2\mu_2)^2 t^2}{n_2 v_2(t)}\right) \\ &\qquad \geq \frac{1}{2}\inf_{t\geq 0}\frac{(b_1 + (\phi_1 c - n_1\mu_1)t)^2}{n_1 v_1(t)} + \frac{1}{2}\inf_{t\geq 0}\frac{(\phi_2 c - n_2\mu_2)^2 t^2}{n_2 v_2(t)}; \end{aligned}$$

the first part of the right-hand side of the last equation coincides with the loss probability of regime (i). By definition all $n_1 \leq n_1^{Q_1}$ satisfy the loss constraint of regime (i). Hence, all $n_1 \leq n_1^{Q_1}$ $(\forall n_2)$ satisfy the loss constraint of regime (ii) as well. One can easily see that if $n_2 = \phi_2 c/\mu_2$, then the loss probability of the middle regime reduces to that of the first regime. Thus, this implies that $g_1(n_1^{Q_1}) = \phi_2 c/\mu_2$ and that $g_1(\cdot)$ is only defined on the interval $[n_1^{Q_1}, \infty)$.

Lemma 5.3.1 *$g_1(\cdot)$ is strictly decreasing on the interval $[n_1^{Q_1}, \phi_1 c/\mu_1]$.*

Proof: First note that $g_1(\cdot)$ corresponds to all possible combinations (n_1, n_2) for which $\triangle_1(n_1, n_2)$ of regime (ii) equals δ_1, i.e., $\triangle_1(n_1, g_1(n_1)) = \delta_1$. Consider $(n_1, n_2) = (a, b)$, with $g_1(a) = b$, where $n_1^{Q_1} < a < \phi_1 c/\mu_1$ and $b < \phi_2 c/\mu_2$, or equivalently

$$\inf_{t \geq 0} \left(\frac{(b_1 + (\phi_1 c - a\mu_1)t)^2}{2av_1(t)} + \frac{(\phi_2 c - b\mu_2)^2 t^2}{2bv_2(t)} \right) = \delta_1.$$

Let an optimizer be denoted by t^o. Now, consider the point $(n_1, n_2) = (a + \epsilon_a, b + \epsilon_b)$, with $\epsilon_a \in (0, \phi_1 c/\mu_1 - a)$ and $\epsilon_b \in (0, \phi_2 c/\mu_2 - b)$. Clearly,

$$\inf_{t \geq 0} \left(\frac{(b_1 + (\phi_1 c - (a + \epsilon_a)\mu_1)t)^2}{2(a + \epsilon_a)v_1(t)} + \frac{(\phi_2 c - (b + \epsilon_b)\mu_2)^2 t^2}{2(b + \epsilon_b)v_2(t)} \right)$$

$$\leq \frac{(b_1 + (\phi_1 c - (a + \epsilon_a)\mu_1)t^o)^2}{2(a + \epsilon_a)v_1(t^o)} + \frac{(\phi_2 c - (b + \epsilon_b)\mu_2)^2 (t^o)^2}{2(b + \epsilon_b)v_2(t^o)} < \delta_1.$$

Thus, we have that $\triangle_1(a + \epsilon_a, b + \epsilon_b) < \delta_1$, implying that it is impossible that $g_1(a + \epsilon_a) = b + \epsilon_b$. In the same manner we can also prove that it is impossible that $g_1(a + \epsilon_a) = b$, $g_1(a) = b + \epsilon_b$, and $g_1(a - \epsilon_{\tilde{a}}) = b - \epsilon_{\tilde{b}}$, with $\epsilon_{\tilde{a}} \in [0, a - n_1^{Q_1})$ and $\epsilon_{\tilde{b}} \in [0, b)$, but not both 0. Hence, there must exist a value $x > b$ such that we have $\triangle_1(a - \epsilon_{\tilde{a}}, x) = \delta_1$, i.e., $g_1(a - \epsilon_{\tilde{a}}) = x$, and there must exist a value $y < b$ such that we have $\triangle_1(a + \epsilon_a, y) = \delta_1$, i.e., $g_1(a + \epsilon_a) = y$. This proves that $g_1(\cdot)$ must be a strictly decreasing function of n_1 on the interval $[n_1^{Q_1}, \phi_1 c/\mu_1]$. □

In Section 5.2 we remarked that for the often used variance functions, the function $f_1(\cdot)$, which separates regime (ii) from regime (iii), is increasing on the interval $[0, \phi_2 c/\mu_2]$, with $f(\phi_2 c/\mu_2) = \phi_1 c/\mu_1$. As $g_1(\cdot)$ is strictly decreasing on the interval $[n_1^{Q_1}, \phi_1 c/\mu_1]$, with $g_1(n_1^{Q_1}) = \phi_2 c/\mu_2$, we conjecture that $f_1(\cdot)$ and $g_1(\cdot)$ intersect at a unique point $(n_1, n_2) = (n_1^{I_1}, n_2^{I_1})$, with $n_1^{Q_1} < n_1^{I_1} < \phi_1 c/\mu_1$ and $n_2^{I_1} < \phi_2 c/\mu_2$; in Section 5.4 we will show that for Brownian motion inputs this claim is true. We also validated this conjecture by performing numerous numerical experiments with other Gaussian inputs. In none of these cases a counter example could be found. Then a typical shape of the region $S_1^{ii}(\phi_1)$ would be like Figure 5.3 (top, right).

5.3.3 Region $S_1^{iii}(\phi_1)$

$S_1^{iii}(\phi_1)$ consists of all combinations of (n_1, n_2) in $T_1^{iii}(\phi_1)$ such that

$$\triangle_1(n_1, n_2) = \frac{1}{2} \inf_{t \geq 0} \frac{(b_1 + (c - n_1\mu_1 - n_2\mu_2)t)^2}{n_1 v_1(t) + n_2 v_2(t)} \geq \delta_1.$$

Once again, standard rewriting yields

$$n_2 \leq h_1(n_1) = \inf_{t \geq 0} \frac{-Y_t - \sqrt{Y_t^2 - 4X_t Z_t}}{2X_t}, \tag{5.7}$$

where

$$\begin{aligned} X_t &:= \mu_2^2 t^2; \\ Y_t &:= 2n_1\mu_1\mu_2 t^2 - 2\delta_1 v_2(t) - 2b_1\mu_2 t - 2c\mu_2 t^2; \\ Z_t &:= b_1^2 + 2b_1 ct + c^2 t^2 + n_1^2\mu_1^2 t^2 - 2b_1 n_1\mu_1 t - 2cn_1\mu_1 t^2 - 2\delta_1 n_1 v_1(t). \end{aligned}$$

Let $n_1^{\max_1}$ denote the value of n_1 that solves $h_1(n_1) = 0$. The following lemma states some properties of $h_1(\cdot)$.

Lemma 5.3.2 *$h_1(\cdot)$ is strictly decreasing on the interval $[0, n_1^{\max_1}]$ and tighter than the capacity constraint. Furthermore, $g_1(n_1) \geq h_1(n_1)$ for all $n_1 \in [n_1^{Q_1}, n_1^{\max_1}]$.*

Proof: The proof of the first statement is similar to Lemma 5.3.1. We now show that $h_1(\cdot)$ is tighter than the capacity constraint. If $n_1\mu_1 + n_2\mu_2 = c$ (capacity constraint), then $\triangle_1(n_1, n_2)$ of regime (iii) equals 0, as the optimizer is $t^F = \infty$, due to A3. Note that the line $h_1(\cdot)$ are all (n_1, n_2) such that $\triangle_1(n_1, n_2)$ of regime (iii) equals δ_1, with $\delta_1 > 0$. Hence, the capacity constraint cannot be part of $h_1(\cdot)$, implying that $h_1(\cdot)$ is tighter than the capacity constraint, i.e., it lies below the capacity constraint.

We proceed with the proof of the last statement. We first prove that $\triangle_1(n_1, n_2)$ of regime (ii) is at least as large as the one of regime (iii), and then we use this to show that $g_1(n_1) \geq h_1(n_1)$ for all $n_1 \in [n_1^{Q_1}, n_1^{\max_1}]$. Let $a_1 := b_1 + (\phi_1 c - n_1\mu_1)t$, $a_2 := (\phi_2 c - n_2\mu_2)t$, $v_1 := n_1 v_1(t)$ and $v_2 := n_2 v_2(t)$. It can be seen that it suffices to prove that for all $t \geq 0$,

$$\frac{a_1^2}{v_1} + \frac{a_2^2}{v_2} \geq \frac{(a_1 + a_2)^2}{v_1 + v_2}. \tag{5.8}$$

Rearranging (5.8) yields $a_1^2 v_2^2 + a_2^2 v_1^2 - 2a_1 a_2 v_1 v_2 \geq 0$, which is equivalent to $(a_1 v_2 - a_2 v_1)^2 \geq 0$, thus proving $\triangle_1(n_1, n_2)$ of regime (ii) is at least as large as the one of regime (iii). Note that there is equality if $a_1 v_2 = a_2 v_1$, so in that case $\triangle_1(n_1, n_2)$ of regimes (ii) and (iii) coincide and they have the same optimizer t^F. Recall from Section 5.2 that $a_1 v_2 = a_2 v_1$, with $t = t^F$, corresponds to the line $f_1(\cdot)$.

By definition $g_1(n_1)$ ($h_1(n_1)$) is the value of n_2 where $\triangle_1(n_1, n_2)$ of regime (ii) ((iii)) equals δ_1. Let $\triangle_1(n_1, n_2)$ of regime (x), with $x \in \{i, ii, iii\}$, be denoted by $\triangle_1^x(n_1, n_2)$. Since we proved that $\triangle_1(n_1, n_2)$ of regime (ii) is at least as large as the one of regime (iii), we have that $\triangle_1^{iii}(n_1, g_1(n_1)) \leq \triangle_1^{ii}(n_1, g_1(n_1)) = \delta_1 = \triangle_1^{iii}(n_1, h_1(n_1))$. In the same manner as Lemma 5.3.1, we can prove that $\triangle_1^{iii}(n_1, n_2)$ decreases in n_2 for fixed n_1, given that $n_1\mu_1 + n_2\mu_2 \leq c$, implying that $g_1(n_1) \geq h_1(n_1)$ for all $n_1 \in [n_1^{Q_1}, n_1^{\max_1}]$. □

By definition, for $(n_1, n_2) = (f_1(n_2), n_2)$ the approximations of $\triangle_1(n_1, n_2)$ are equal for regimes (ii) and (iii) (see previous lemma). Hence, if $f_1(\cdot)$ and $g_1(\cdot)$ intersect at $(n_1^{I_1}, n_2^{I_1})$ (see Section 5.3.2), then this is also the point where $f_1(\cdot)$ and $h_1(\cdot)$ intersect. Figure 5.3 (bottom) illustrates the region $S_1^{iii}(\phi_1)$.

5.3.4 Region $S_1(\phi_1)$

$S_1(\phi_1)$ can be obtained by taking the union of the three described regions, i.e., $S_1(\phi_1) = S_1^i(\phi_1) \cup S_1^{ii}(\phi_1) \cup S_1^{iii}(\phi_1)$. We now state our main result, which follows from Sections 5.3.1-5.3.3.

Theorem 5.3.3 *The boundary of the admissible region of the first queue, $S_1(\phi_1)$, is defined as follows:*

$$\begin{array}{ll} 0 \leq n_1 \leq n_1^{Q_1}: & n_2 = (c - n_1\mu_1)/\mu_2; \\ n_1^{Q_1} < n_1 < n_1^{I_1}: & n_2 = g_1(n_1); \\ n_1^{I_1} \leq n_1 \leq n_1^{\max_1}: & n_2 = h_1(n_1). \end{array}$$

5.4 Brownian inputs

For most Gaussian inputs that satisfy A1-A3 the boundary of $S(\phi_1)$ cannot be explicitly computed; consequently, in those cases one has to rely on numerical techniques (as will be done in the numerical examples in Section 5.5). For the 'canonical model' with Brownian inputs though, we have succeeded in finding closed-form expressions for the boundary. As indicated in [126], Brownian motions can be used to approximate weakly-dependent traffic streams, cf. also the celebrated 'Central Limit Theorem in functional form'. We let the variance functions be characterized through $v_i(t) = \lambda_i t$, with $\lambda_i > 0$, $i = 1, 2$.

5.4.1 Region $S_1(\phi_1)$

It is a matter of straightforward calculus to show that $t^F = b_1/(c - n_1\mu_1 - n_2\mu_2)$. Now, the Mannersalo-Norros approximation reduces to the following. The critical

weight ϕ_2^F equals

$$1 - \frac{n_1\lambda_1 - n_2\lambda_2}{n_1\lambda_1 + n_2\lambda_2}\left(1 - \frac{n_1\mu_1 + n_2\mu_2}{c}\right) - \frac{n_1\mu_1}{c}. \tag{5.9}$$

Then we get the approximations

$$\triangle_1(n_1, n_2) = \begin{cases} (i) & 2b_1\frac{\phi_1 c - n_1\mu_1}{n_1\lambda_1} & \text{for } \phi_2 \in [0, \frac{n_2\mu_2}{c}]; \\ (ii) & \frac{1}{2}\Big(\frac{(b_1 + (\phi_1 c - n_1\mu_1)t^*)^2}{n_1\lambda_1 t^*} + \frac{(\phi_2 c - n_2\mu_2)^2}{n_2\lambda_2}t^*\Big) & \text{for } \phi_2 \in (\frac{n_2\mu_2}{c}, \phi_2^F); \\ (iii) & 2b_1\frac{c - n_1\mu_1 - n_2\mu_2}{n_1\lambda_1 + n_2\lambda_2} & \text{for } \phi_2 \in [\phi_2^F, 1], \end{cases}$$

with the 'critical time scale' t^* given by

$$\frac{b_1}{\sqrt{(\phi_1 c - n_1\mu_1)^2 + (\phi_2 c - n_2\mu_2)^2\frac{n_1\lambda_1}{n_2\lambda_2}}}. \tag{5.10}$$

In [126] it was shown that the resulting expressions are 'asymptotically exact' in the many-sources regime.

Let us first derive the function $f_1(\cdot)$. Recall from Section 5.2 that $f_1(\cdot)$ is equivalent to all pairs of (n_1, n_2) that satisfy (5.4) with equality. Plugging in the expression for t^F and some rearranging yields

$$n_1 = \frac{c\lambda_2(1 + \phi_1) - n_2\lambda_2\mu_2}{\frac{\phi_2 c\lambda_1}{n_2} + 2\lambda_2\mu_1 - \lambda_1\mu_2} =: f_1(n_2). \tag{5.11}$$

It can easily be verified that $f_1(0) = 0$ and $f_1(\phi_2 c/\mu_2) = \phi_1 c/\mu_1$. The following lemma states some properties of $f_1(\cdot)$; define

$$\xi := \frac{(1 + \phi_1)\mu_1}{\phi_1\mu_2}. \tag{5.12}$$

Lemma 5.4.1 *$f_1(\cdot)$ is continuous and has a continuous derivative on the interval $[0, \phi_2 c/\mu_2]$. Furthermore, $f_1(\cdot)$ is concave on $[0, \phi_2 c/\mu_2]$ if $\lambda_1 < \xi\lambda_2$; $f_1(\cdot)$ is convex on $[0, \phi_2 c/\mu_2]$ if $\lambda_1 > \xi\lambda_2$; $f_1(\cdot)$ has a constant positive derivative on $[0, \phi_2 c/\mu_2]$ if $\lambda_1 = \xi\lambda_2$ and this derivative has the value $\phi_1\mu_2/(\phi_2\mu_1)$.*

Proof: For any given α, β and γ such that $\beta \neq -\gamma n_2$, note that

$$\frac{\mathrm{d}^2}{\mathrm{d}n_2^2}\frac{1 - \alpha n_2}{\beta/n_2 + \gamma} = \frac{-2\beta(\alpha\beta + \gamma)}{(\beta + \gamma n_2)^3} =: p(n_2). \tag{5.13}$$

It is clear that $p(n_2)$ changes sign only at $n_2 = -\beta/\gamma$. Now, let

$$\alpha = \frac{\lambda_2\mu_2}{c\lambda_2(1 + \phi_1)}; \quad \beta = \frac{\phi_2 c\lambda_1}{c\lambda_2(1 + \phi_1)}; \quad \gamma = \frac{2\lambda_2\mu_1 - \lambda_1\mu_2}{c\lambda_2(1 + \phi_1)}.$$

Then, due to (5.11), $f''(n_2) = p(n_2)$, and therefore $f_1''(n_2)$ changes sign only at

$$n_2 = -\frac{\beta}{\gamma} = \frac{\phi_2 c \lambda_1}{\lambda_1\mu_2 - 2\lambda_2\mu_1}. \tag{5.14}$$

Note that expression (5.14) does not lie in $[0, \phi_2 c/\mu_2]$, so $f_1(\cdot)$ is either convex or concave on this interval. From (5.13) we conclude that there is concavity when $\lambda_1 < \xi\lambda_2$ (corresponding to $\alpha\beta > -\gamma$), and convexity otherwise. □

Subsequently, in order to fully characterize the areas $S_1^i(\phi_1)$, $S_1^{ii}(\phi_1)$, $S_1^{iii}(\phi_1)$, we now derive $n_1^{Q_1}$, $g_1(\cdot)$ and $h_1(\cdot)$. We do this by relying on (5.5), (5.6) and (5.7), respectively. This yields

$$\frac{2\phi_1 c b_1}{2b_1\mu_1 + \delta_1\lambda_1} =: n_1^{Q_1}; \tag{5.15}$$

$$\frac{(\phi_2 c - n_2\mu_2)^2 b_1^2}{n_2\lambda_2(\delta_1^2\lambda_1 + 2b_1\delta_1\mu_1)} + \frac{2\phi_1 c b_1 \delta_1}{\delta_1^2\lambda_1 + 2b_1\delta_1\mu_1} =: g_1^{-1}(n_2); \tag{5.16}$$

$$\frac{2cb_1}{2b_1\mu_2 + \delta_1\lambda_2} - n_1\frac{2b_1\mu_1 + \delta_1\lambda_1}{2b_1\mu_2 + \delta_1\lambda_2} =: h_1(n_1). \tag{5.17}$$

Note that $h_1(\cdot)$ is linear in n_1 and that

$$h_1(n_1^{\max_1}) = h_1\left(\frac{2cb_1}{2b_1\mu_1 + \delta_1\lambda_1}\right) = 0.$$

Due to Lemma 5.4.1, $f_1(\cdot)$, $g_1(\cdot)$ and $h_1(\cdot)$ have a unique intersection point (n_1, n_2) given by

$$(n_1^{I_1}, n_2^{I_1}) = \left(\frac{cb_1(\delta_1\lambda_2(1+\phi_1) + 2b_1\mu_2\phi_1)}{(\delta_1\lambda_1 + 2b_1\mu_1)(\delta_1\lambda_2 + b_1\mu_2)}, \frac{\phi_2 c b_1}{\delta_1\lambda_2 + b_1\mu_2}\right). \tag{5.18}$$

Now we have all the ingredients to describe the boundary of $S_1(\phi_1)$ explicitly. The admissible region of the second queue can be treated analogously. Both are depicted in Figure 5.4.

5.4.2 Region $S(\phi_1)$

A combination (n_1, n_2) is contained in $S(\phi_1)$ if it satisfies the QoS requirements for both classes. That is, if it is contained in $S_1(\phi_1) \cap S_2(\phi_1)$. In this subsection we characterize the boundary of $S(\phi_1)$. In the analysis below the ratios b_1/b_2 and δ_1/δ_2 turn out to be crucial. We therefore introduce $b := b_1/b_2$ and $d := \delta_1/\delta_2$. Let us first mention some useful facts.

Lemma 5.4.2 *If $b < d$ ($b > d$) then $h_2^{-1}(n_1) > h_1(n_1)$ ($h_2^{-1}(n_1) < h_1(n_1)$) for all n_1 that satisfy $h_2^{-1}(n_1) \geq 0$ and $h_1(n_1) \geq 0$.*

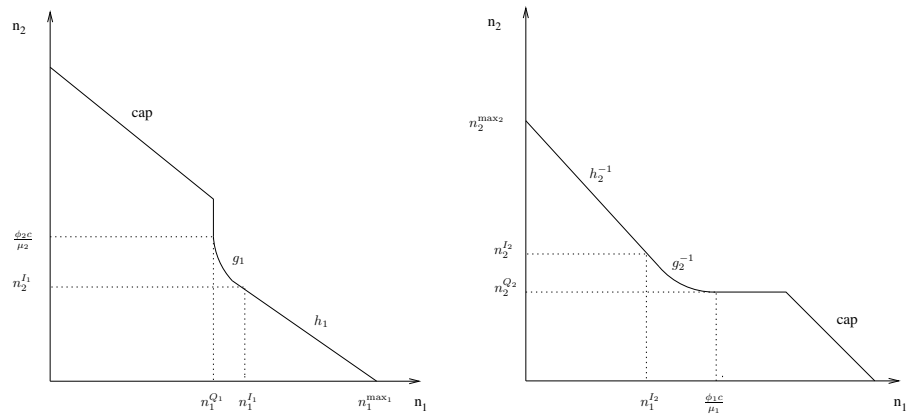

Figure 5.4: Left: $S_1(\phi_1)$. Right: $S_2(\phi_1)$.

Proof: We only prove the claim for $b < d$, as the claim for $b > d$ follows analogously. We know that

$$h_1(n_1) = \frac{2cb_1}{2b_1\mu_2 + \delta_1\lambda_2} - n_1\frac{2b_1\mu_1 + \delta_1\lambda_1}{2b_1\mu_2 + \delta_1\lambda_2};$$

$$h_2^{-1}(n_1) = \frac{2cb_2}{2b_2\mu_2 + \delta_2\lambda_2} - n_1\frac{2b_2\mu_1 + \delta_2\lambda_1}{2b_2\mu_2 + \delta_2\lambda_2}.$$

Now, $h_2^{-1}(0) > h_1(0)$ implies that

$$\frac{2cb_2}{2b_2\mu_2 + \delta_2\lambda_2} > \frac{2cb_1}{2b_1\mu_2 + \delta_1\lambda_2} \text{ or } b < d,$$

but we also have that $h_2(0) > h_1^{-1}(0)$ implies that

$$\frac{2cb_2}{2b_2\mu_1 + \delta_2\lambda_1} > \frac{2cb_1}{2b_1\mu_1 + \delta_1\lambda_1} \text{ or } b < d.$$

Since $h_1(\cdot)$ and $h_2^{-1}(\cdot)$ are linear, this proves the stated for $b < d$. Note that $h_1(\cdot)$ and $h_2^{-1}(\cdot)$ are identical if $b = d$. □

Lemma 5.4.3 *If $b < d/2$ ($b > 2d$) then $n_1^{Q_1} < n_1^{I_2}$ ($n_1^{Q_1} > n_1^{I_2}$) and $n_2^{Q_2} > n_2^{I_1}$ ($n_2^{Q_2} < n_2^{I_1}$). If $d/2 \le b \le 2d$ then $n_1^{Q_1} \ge n_1^{I_2}$ and $n_2^{Q_2} \ge n_2^{I_1}$.*

Proof: We only prove the claim for $b < d/2$, as the claims for $b > 2d$ and $d/2 \le b \le 2d$ follow in a similar fashion. Use the explicit expressions for $n_1^{Q_1}$ and $n_1^{I_2}$. Thus, $n_1^{Q_1} < n_1^{I_2}$ is equivalent to

$$\frac{2\phi_1 cb_1}{\delta_1\lambda_1 + 2b_1\mu_1} < \frac{\phi_1 cb_2}{\delta_2\lambda_1 + b_2\mu_1} \quad \text{or} \quad 2\delta_2 b_1\lambda_1 + 2b_1b_2\mu_1 < \delta_1 b_2\lambda_1 + 2b_1b_2\mu_1.$$

Omitting common terms and some rearranging directly yields $b < d/2$.

Likewise, it holds that $n_2^{Q_2} > n_2^{I_1}$ if $b < d/2$, since

$$\frac{2\phi_2 c b_2}{\delta_2\lambda_2 + 2b_2\mu_2} > \frac{\phi_2 c b_1}{\delta_1\lambda_2 + b_1\mu_2} \quad \text{or} \quad 2\delta_1 b_2\lambda_2 + 2b_1 b_2\mu_2 > \delta_2 b_1\lambda_2 + 2b_1 b_2\mu_2,$$

reduces to $b < 2d$. □

Combining the two previous lemmas leads to the conclusion that we have to distinguish between three cases: (a) $b < d/2$, (b) $d/2 \leq b \leq 2d$ and (c) $b > 2d$. Below we show that the shape of the boundary of $S(\phi_1)$ depends on (a), (b) or (c) if $\phi_1 \in (0,1)$. First we characterize the boundary of $S(\phi_1)$ for $\phi_1 = 0$ and $\phi_1 = 1$. The boundary of $S(0)$ is given by

$$0 \leq n_1 \leq n_1^O : \quad n_2 = n_2^{Q_2};$$

$$n_1^O < n_1 < n_1^{\max_1} : \quad n_2 = h_1(n_1),$$

where $n_2^{Q_2}$ is evaluated at $\phi_1 = 0$, and $n_1^O := h_1^{-1}(n_2^{Q_2})$. The boundary of $S(1)$ is

$$0 \leq n_1 \leq n_1^{Q_1} : \quad n_2 = h_2^{-1}(n_1),$$

where $n_1^{Q_1}$ is evaluated at $\phi_1 = 1$.

Remark: One can easily show that $S(0) \subset S(1)$ if $b < d$, $S(1) \subset S(0)$ if $b > d$ and $S(0) = S(1)$ if $b = d$.

In the following we show that there are different generic shapes of the boundary of $S(\phi_1)$, $\phi_1 \in (0,1)$, within each of the three cases.

Case $b < d/2$

It can easily be seen that the boundary of $S(\phi_1)$ has four possible shapes in this case (see Figure 5.5). The shape of the boundary ((a_1), (a_2), (a_3) or (a_4)) depends on the value of ϕ_1, but each shape occurs as will be shown in the following lemmas. Let n_1^V be the solution of $g_1(n_1) = h_2^{-1}(n_1)$. Furthermore, let n_1^W solve $g_1(n_1) = g_2^{-1}(n_1)$, and let $n_2^W = g_1(n_1^W)$. Finally, define $n_1^X := g_1^{-1}(n_2^{Q_2})$.

Lemma 5.4.4 *The boundary of $S(\phi_1)$ has shape (a_1) if $\phi_1 \in [X_3, 1)$, where*

$$X_3 := \frac{\delta_2\lambda_2(\delta_1\lambda_1 + 2b_1\mu_1)}{\delta_2\lambda_2(\delta_1\lambda_1 + 2b_1\mu_1) + 2\lambda_1\mu_2(\delta_1 b_2 - \delta_2 b_1)}. \tag{5.19}$$

Proof: In order to have shape (a_1) we must have that $h_2^{-1}(n_1^{Q_1}) \geq \phi_2 c/\mu_2$ for some value of $\phi_1 \in (0,1)$. That is,

$$\frac{2cb_2}{\delta_2\lambda_2 + 2b_2\mu_2} - \frac{2\phi_1 c b_1(\delta_2\lambda_1 + 2b_2\mu_1)}{(\delta_2\lambda_2 + 2b_2\mu_2)(\delta_1\lambda_1 + 2b_1\mu_1)} \geq \frac{(1-\phi_1)c}{\mu_2}. \tag{5.20}$$

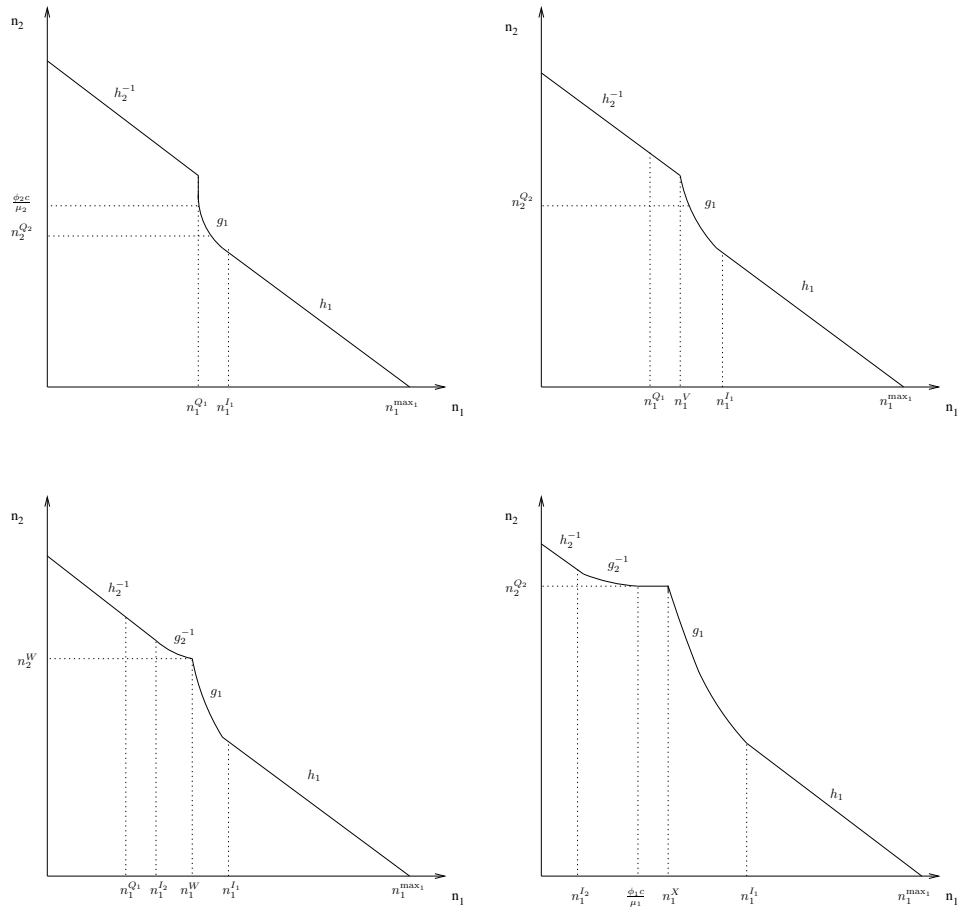

Figure 5.5: Top, from left to right: shape (a_1) and (a_2). Bottom, from left to right: shape (a_3) and (a_4).

One can easily show that this reduces to a constraint of the form $-A+B\phi_1 \geq 0$, with $A, B > 0$. For $\phi_1 = 0$ the left-hand side of the constraint (5.20) is equivalent to

$$\frac{2cb_2}{\delta_2\lambda_2 + 2b_2\mu_2} - \frac{c}{\mu_2},$$

which is smaller than 0 (assuming that $\delta_2, \lambda_2 > 0$). Hence, the constraint is not satisfied. For $\phi_1 = 1$ the left-hand side of (5.20) equals

$$\frac{2c\lambda_1(\delta_1 b_2 - \delta_2 b_1)}{(\delta_2\lambda_2 + 2b_2\mu_2)(\delta_1\lambda_1 + 2b_1\mu_1)},$$

which is larger than 0 if $b/d < 1$, which is true, as we required $b < d/2$. Thus, since the constraint is a linear function of ϕ_1, there must be value of $\phi_1 \in (0,1)$ for which $h_2^{-1}(n_1^{Q_1}) = \phi_2 c/\mu_2$. Straightforward calculus shows there is equality for $\phi_1 = X_3$. □

Lemma 5.4.5 *The boundary of $S(\phi_1)$ has shape (a_4) if $\phi_1 \in (0, X_1]$, where*

$$X_1 := \frac{\delta_2^2 b_1^2 \lambda_2 \mu_1}{\delta_2^2 b_1^2 \lambda_2 \mu_1 + 2\delta_1^2 b_2 \lambda_1 (\delta_2 \lambda_2 + 2 b_2 \mu_2)}. \tag{5.21}$$

Proof: The proof is analogous to that of Lemma 5.4.4. Shape (a_4) occurs if there exists a value of $\phi_1 \in (0,1)$ for which $g_1^{-1}(n_2^{Q_2}) \geq \phi_1 c/\mu_1$. This constraint can be rewritten as $A - B\phi_1 \geq 0$, with $A, B > 0$. Since it is satisfied for $\phi_1 = 0$, but not for $\phi_1 = 1$, there exists a unique value of $\phi_1 \in (0,1)$, X_1, such that there is equality, i.e., $g_1^{-1}(n_2^{Q_2}) = \phi_1 c/\mu_1$. □

Lemma 5.4.6 *The boundary of $S(\phi_1)$ has shape (a_3) if $\phi_1 \in (X_1, X_2)$, where X_2 is the value of ϕ_1 such that $n_2^{I_2} = h_2^{-1}(n_1^{I_2}) = g_1(n_1^{I_2})$.*

Proof: The shape of the boundary is like (a_3) if $h_2^{-1}(n_1^{I_2}) < g_1(n_1^{I_2})$ and if $g_1^{-1}(n_2^{Q_2}) < \phi_1 c/\mu_1$. The latter constraint is satisfied if $\phi_1 > X_1$ (Lemma 5.4.5). In contrast to the latter constraint, the former constraint does not reduce to a constraint that is a linear function of ϕ_1. It can be shown that there exists a unique value of ϕ_1, X_2, such that $h_2^{-1}(n_1^{I_2}) = g_1(n_1^{I_2})$. The expression of X_2 is not presented here (as it is quite complicated); it depends on the parameters $\delta_1, \delta_2, b_1, b_2, \lambda_1, \lambda_2, \mu_1$ and μ_2. Now, the constraint is satisfied for all $\phi_1 \in [0, X_2)$.

We now show that $X_2 \in (X_1, X_3)$. First recall that $g_1(n_1)$ is defined on the interval $(n_1^{Q_1}, n_1^{I_1})$ in $S_1(\phi_1)$, whereas $h_2^{-1}(n_1)$ is defined on the interval $[0, n_1^{I_2}]$ in $S_2(\phi_1)$. Therefore, if $g_1(\cdot)$ and $h_2^{-1}(\cdot)$ are part of the boundary of $S(\phi_1)$, then they are defined on (parts of) the mentioned intervals. If $\phi_1 \in (0, X_1]$, then $g_1(\cdot)$ is defined on the interval $(n_1^X, n_1^{I_1})$, with $n_1^X \geq \phi_1 c/\mu_1$ (see shape (a_4)). By definition $n_1^{I_2} < \phi_1 c/\mu_1$, so this implies that $g_1(\cdot)$ and $h_1^{-1}(\cdot)$ cannot intersect if $\phi_1 \in (0, X_1]$. Furthermore, if $\phi_1 \in [X_3, 1)$, then $h_2^{-1}(n_1) > g_1(n_1)$ for all $n_1 \in (n_1^{Q_1}, \min\{n_1^{I_1}, n_1^{I_2}\})$ (see shape (a_1)), so $X_2 \notin [X_3, 1)$. Hence, we conclude that $0 < X_1 < X_2 < X_3 < 1$. □

Lemma 5.4.7 *The boundary of $S(\phi_1)$ has shape (a_2) if $\phi_1 \in [X_2, X_3)$.*

Proof: One observes shape (a_2) if $h_2^{-1}(n_1^{Q_1}) < \phi_2 c/\mu_2$ and $h_2^{-1}(n_1^{I_2}) \geq g_1(n_1^{I_2})$. From Lemmas 5.4.4 and 5.4.6 we know that this coincides with $\phi_1 < X_3$ and $\phi_1 \geq X_2$ respectively. □

We now state our main result. The proof follows directly from Lemmas 5.4.4-5.4.7.

Proposition 5.4.8 *If $b < d/2$, then the boundary of $S(\phi_1)$ has*
shape (a_4) for $0 < \phi_1 \leq X_1$;
shape (a_3) for $X_1 < \phi_1 < X_2$;
shape (a_2) for $X_2 \leq \phi_1 < X_3$;
and shape (a_1) for $X_3 \leq \phi_1 < 1$.
Here X_1 is the value of ϕ_1 such that $g_1^{-1}(n_2^{Q_2}) = \phi_1 c/\mu_1$, X_2 is the value of ϕ_1 such that $n_2^{I_2} = h_2^{-1}(n_1^{I_2}) = g_1(n_1^{I_2})$, and X_3 is the value of ϕ_1 that solves $h_2^{-1}(n_1^{Q_1}) = \phi_2 c/\mu_2$.

Case $d/2 \leq b \leq 2d$

As proved in Lemma 5.4.3, this criterion leads to $n_1^{Q_1} \geq n_1^{I_2}$ and $n_2^{Q_2} \geq n_2^{I_1}$. Now, the boundary of $S(\phi_1)$ can have three shapes ((b_1), (b_2) and (b_3)). Shape (b_1) is depicted in Figure 5.6 (top, left). Shape (b_2) corresponds to (a_3), and (b_3) to (a_4).

As in the case of $b < d/2$, one can easily prove that each shape is observed. The proofs are omitted as they are similar to the proofs of Lemmas 5.4.4 and 5.4.5. We directly state the following proposition.

Proposition 5.4.9 *If $d/2 \leq b \leq 2d$, then the boundary of $S(\phi_1)$ has*
shape (b_3) for $0 < \phi_1 \leq Y_1$;
shape (b_2) for $Y_1 < \phi_1 < Y_2$;
and shape (b_3) for $Y_2 \leq \phi_1 < 1$.
Here Y_1 is the value of ϕ_1 such that $g_1^{-1}(n_2^{Q_2}) = \phi_1 c/\mu_1$ and Y_2 coincides with the value of ϕ_1 such that $g_2^{-1}(n_1^{Q_1}) = \phi_2 c/\mu_2$.

Case $b > 2d$

The last case is the counterpart of the first case. Therefore, the proofs are also omitted in the following. Now, $n_1^{Q_1} > n_1^{I_2}$ and $n_2^{Q_2} < n_2^{I_1}$. Define $n_1^Y := h_1^{-1}(n_2^{Q_2})$, and let n_1^Z be the solution of $g_2^{-1}(n_1) = h_1(n)$. There are four possible shapes of $S(\phi_1)$, $\phi_1 \in (0, 1)$. Shapes (c_1) and (c_2) are depicted in Figure 5.6. Shape (c_3) corresponds to (a_3), and (c_4) to (b_1).

Proposition 5.4.10 *If $b > 2d$, then the boundary of $S(\phi_1)$ has*
shape (c_1) for $0 < \phi_1 \leq Z_1$;
shape (c_2) for $Z_1 < \phi_1 \leq Z_2$;
shape (c_3) for $Z_2 < \phi_1 < Z_3$;
and shape (c_4) for $Z_3 \leq \phi_1 < 1$.
Here Z_1 corresponds to the value of ϕ_1 such that $h_1^{-1}(n_2^{Q_2}) = \phi_1 c/\mu_1$, Z_2 is the value of ϕ_1 that solves $n_1^{I_1} = h_1^{-1}(n_2^{I_1}) = g_2(n_2^{I_1})$ and Z_3 is the value of ϕ_1 such that $g_2^{-1}(n_1^{Q_1}) = \phi_2 c/\mu_2$.

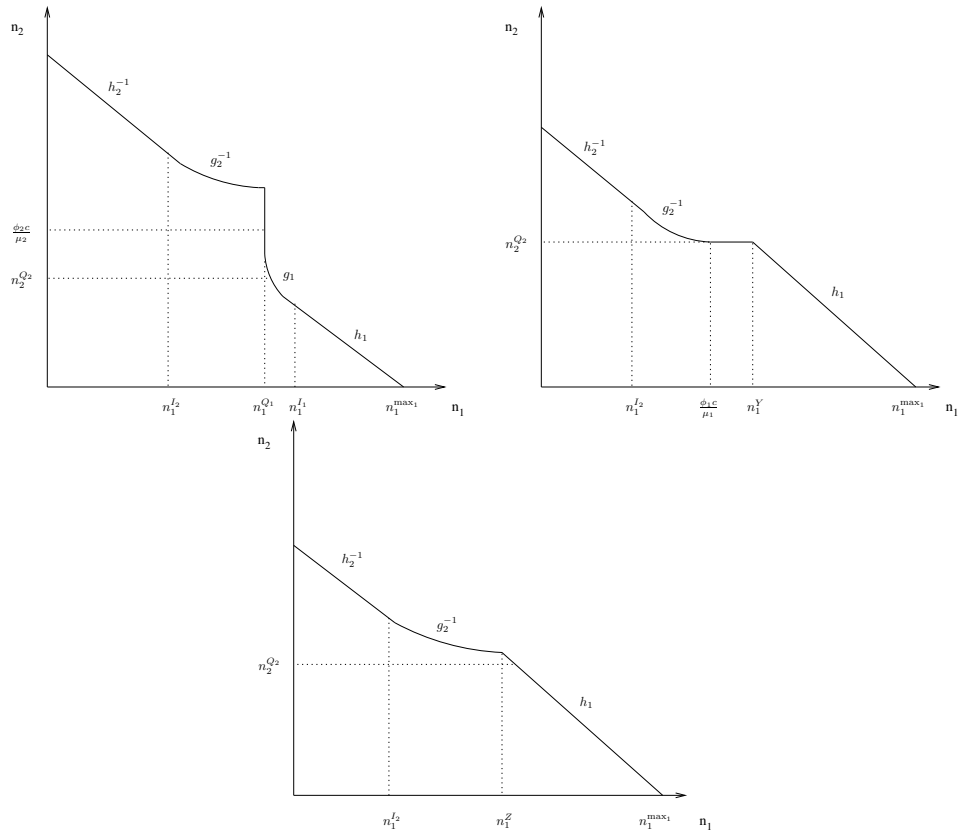

Figure 5.6: Top, from left to right: shape (b_1) and (c_1). Bottom: shape (c_2).

5.4.3 The realizable region

Let R denote the realizable region, i.e.,

$$R := \bigcup_{\phi_1 \in [0,1]} S(\phi_1). \tag{5.22}$$

In the following we show that we do not always need all values of $\phi_1 \in [0,1]$ to compose R. We now state our main result.

Theorem 5.4.11 *The realizable region R can be obtained as follows:*

$$b < d/2: \quad R = \bigcup_{\phi_1 \in (0, X_2) \cup \{1\}} S(\phi_1);$$

$$d/2 \leq b \leq d: \quad R = \bigcup_{\phi_1 \in (0,1]} S(\phi_1);$$

$$d < b \leq 2d: \quad R = \bigcup_{\phi_1 \in [0,1)} S(\phi_1);$$

$$b > 2d: \quad R = \bigcup_{\phi_1 \in \{0\} \cup (Z_2, 1)} S(\phi_1).$$

Proof: We only prove the first statement, as the other three statements can be proved in a similar fashion. We already remarked above that $S(0) \subset S(1)$ in case $b < d/2$. Furthermore, in this case we also have that $S(\phi_1) \subset S(1)$ for all values of $\phi_1 \in [X_2, 1)$. To see this, compare the boundaries (a_1) and (a_2) with the boundary of $S(1)$, and recall that $h_2^{-1}(\cdot) > h_1(\cdot)$ if $b < d/2$. What is left to prove is that we need all values of $\phi_1 \in (0, X_2)$ to compose R if $b < d/2$. Note that the boundary of $S(\phi_1)$ has shape (a_3) if $\phi_1 \in (X_1, X_2)$, implying that $S(\phi_1)$ contains (n_1^W, n_2^W), with $h_2^{-1}(n_1^W) < n_2^W$, which cannot be part of $S(1)$. From Lemma 5.4.6 it follows that for all values of $\phi_1 \in (X_1, X_2)$, n_1^W (n_2^W) increases (decreases) as ϕ_1 increases (but not linearly), implying that we need all values of $\phi_1 \in (X_1, X_2)$ to compose R if $b < d/2$. Likewise, shape (a_4) is seen if $\phi_1 \in (0, X_1]$. The point $(n_1^X, n_2^{Q_2})$ will then be contained in $S(\phi_1)$, which cannot be contained in $S(1)$ either. From Lemma 5.4.5 it follows that as ϕ_1 increases in the corresponding interval, n_1^X $(n_2^{Q_2})$ linearly increases (decreases), implying that we also need all values of $\phi_1 \in (0, X_1]$ to compose R if $b < d/2$, thus proving the first statement. □

Using Theorem 5.4.11, the boundary of R can now also be determined. Since there are four possible cases in Theorem 5.4.11, it follows that the boundary of R can have four different generic shapes. Below we discuss each one of these. Let us first introduce some notation. From now on, we write $z(\phi_1)$ if z depends on ϕ_1. Note that $\phi_2 = 1 - \phi_1$, thus if an expression contains ϕ_2, we can easily rewrite it as function of ϕ_1.

Case $b < d/2$

Theorem 5.4.11 shows that we need all values of $\phi_1 \in (0, X_2)$ and $\phi_1 = 1$ to compose R in this case. Straightforward calculus shows that all values of $\phi_1 \in (0, X_2)$ contribute to the boundary of R in the following way:

$$\phi_1 \in (0, X_1]: \quad (n_1, n_2) = (g_1^{-1}(n_2^{Q_2}(\phi_1)), n_2^{Q_2}(\phi_1)); \tag{5.23}$$

$$\phi_1 \in (X_1, X_2): \quad (n_1, n_2) = (n_1^W(\phi_1), n_2^W(\phi_1)), \tag{5.24}$$

with

$$n_1^H := g_1^{-1}(n_2^{Q_2}(0)) > 0; \quad n_2^{Q_2}(0) = h_2^{-1}(0);$$

$$g_1^{-1}(n_2^{Q_2}(X_1)) = n_1^W(X_1); \quad n_2^{Q_2}(X_1) = n_2^W(X_1);$$

$$n_2^W(X_2) = h_2^{-1}(n_1^W(X_2)).$$

It can be shown that (5.23) corresponds to a function $n_2 = k_1(n_1)$ which linearly decreases as n_1 increases. Furthermore, (5.24) corresponds to a function $n_2 = k_2(n_1)$ which non-linearly decreases as n_1 increases. Recall that the boundary of $S(1)$ is given by the line $h_2^{-1}(\cdot)$ on some predefined interval, so also the contribution of $\phi_1 = 1$ to the boundary of R can easily be derived. Moreover, $k_1(\cdot)$, $k_2(\cdot)$ and $h_2^{-1}(\cdot)$ perfectly connect, as one can show that

$$\left.\frac{\frac{\partial n_2^{Q_2}(\phi_1)}{\partial \phi_1}}{\frac{\partial g_1^{-1}(n_2^{Q_2}(\phi_1))}{\partial \phi_1}}\right|_{\phi_1=X_1} = \left.\frac{\frac{\partial n_2^{W}(\phi_1)}{\partial \phi_1}}{\frac{\partial n_1^{W}(\phi_1)}{\partial \phi_1}}\right|_{\phi_1=X_1}; \quad \left.\frac{\partial \frac{n_2^{W}(\phi_1)}{\partial \phi_1}}{\frac{\partial n_1^{W}(\phi_1)}{\partial \phi_1}}\right|_{\phi_1=X_2} = \frac{\partial h_2^{-1}(n_1)}{\partial n_1},$$

see Figure 5.7 (top, left) for an illustration. We are now able to describe to boundary of R, which follows from the above.

Proposition 5.4.12 *If $b < d/2$, then the boundary of R, denoted by r_1, is continuous.*

The approach that is required to derive the boundary of R in the other cases is very similar to that one in the current case. Therefore, we leave out the details in the remaining three cases.

Case $d/2 \leq b \leq d$

Let $n_2 = k_3(n_1)$ and $n_2 = k_4(n_1)$ be functions that correspond to the following equations, respectively:

$$\phi_1 \in (Y_1, Y_2): \quad (n_1, n_2) = (n_1^W(\phi_1), n_2^W(\phi_1));$$

$$\phi_1 \in [Y_2, 1): \quad (n_1, n_2) = (n_1^{Q_1}(\phi_1), g_2^{-1}(n_1^{Q_1}(\phi_1))).$$

It can be shown that $k_3(\cdot)$ is a non-linearly decreasing function, whereas $k_4(\cdot)$ is a linearly decreasing function. Furthermore, it can be shown that $k_1(\cdot)$, $k_3(\cdot)$ and $k_4(\cdot)$ connect perfectly, see Figure 5.7 (top, right). Recalling that $Y_1 = X_1$, we now have all the ingredients to describe the boundary.

Proposition 5.4.13 *If $d/2 \leq b \leq d$, then the boundary of R, denoted by r_2, is continuous.*

Case $d < b \leq 2d$

We directly state the result on the boundary of R, since it is very similar to the previous case.

Proposition 5.4.14 *If $d < b \leq 2d$, then the boundary of R, denoted by r_3, is continuous.*

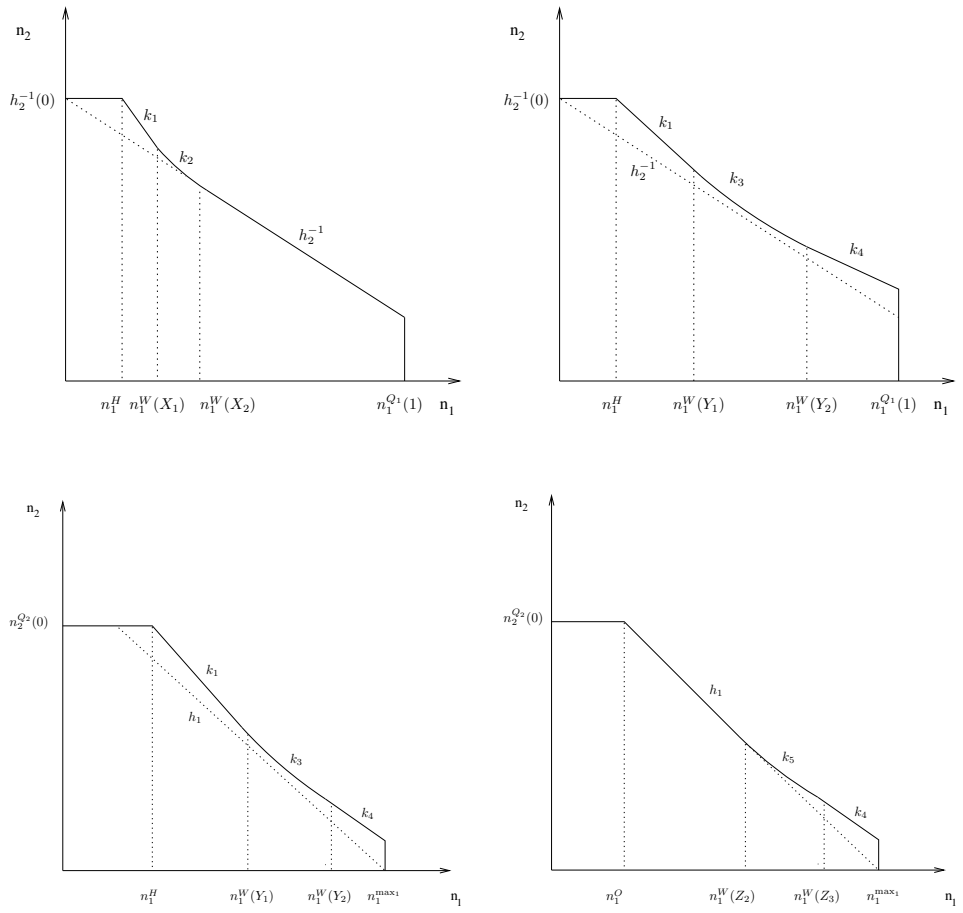

Figure 5.7: Top, from left to right: shape (r_1) and (r_2) (area below dotted line represents $S(1)$). Bottom, from left to right: shape (r_3) and (r_4) (area below dotted line represents $S(0)$).

Case $b > 2d$

Let $n_2 = k_5(n_1)$ be a function that corresponds to the following:

$$\phi_1 \in (Z_2, Z_3): \quad (n_1, n_2) = (n_1^W(\phi_1), n_2^W(\phi_1)).$$

Recalling that $Z_3 = Y_2$, we now directly state the following proposition.

Proposition 5.4.15 *If $b > 2d$, then the boundary of R, denoted by r_4, is continuous.*

Although Theorem 5.4.11 shows that we need a range of weights to obtain R, the above results suggest that almost all of R is obtained by the priority scheduling discipline, e.g., $\phi_1 = 0$ or $\phi_1 = 1$. This observation is established by comparing the boundary of R with the boundaries of $S(0)$ and $S(1)$, and showing that at least one of these two boundaries almost matches with the boundary of R. In particular, in case $b \leq d$, the admissible region $S(1)$ covers most of R, whereas in case $b > d$ the region $S(0)$ approximates R. We further explore this issue in the next section.

5.5 Numerical analysis

In this section we numerically compute the boundary of the realizable region for two realistic examples of Gaussian inputs, with very diverse parameter settings. As the inputs are non-Brownian, the boundary of the admissible region (and thus the realizable region) has to be obtained numerically. The goal is to compare the realizable region with the admissible region corresponding to the priority cases. Denoting by $|R|$, $|S(0)|$ and $|S(1)|$ the number of different pairs (n_1, n_2), $n_1, n_2 \in \mathbb{N}_0$, that are contained in sets R, $S(0)$ and $S(1)$, respectively, we define

$$O_1 \equiv \frac{|S(1)|}{|R|} \quad \text{and} \quad O_2 \equiv \frac{|S(0)|}{|R|}, \tag{5.25}$$

i.e., O_i is a measure that indicates what fraction of the realizable region can be obtained by prioritizing class i, $i = 1, 2$. Recall that $S(0) \subseteq R$ and $S(1) \subseteq R$, hence $O_1, O_2 \in [0, 1]$. The following examples illustrate that either $S(0)$ or $S(1)$ (or both) covers most of the realizable region (as was the case for Brownian inputs, see Section 5.4), i.e., either O_1 or O_2 (or both) is almost equal to 1.

5.5.1 Example 1

Consider two traffic classes sharing a total capacity (c) of 10 Mbps. The first class consists of data traffic, whereas the second class corresponds to voice traffic. Traffic of the first class is modeled as fractional Brownian motion, i.e., $v_1(t) = \alpha t^{2H_1}$, with $H_1 \in (0, 1)$. The mean traffic rate μ_1 is 0.2 Mbps and its variance function is given by $v_1(t) = 0.0025t^{2H_1}$. In measurement studies it was frequently found that H_1 lies between, say, 0.7 and 0.85. Below we take $H_1 \in \{0.5, 0.65, 0.8, 0.95\}$.

Traffic of the second class corresponds to the Gaussian counterpart of the AMS model. In the AMS model work arrives from sources in bursts which have peak rate h and exponentially distributed lengths with mean β^{-1}. After each burst, the source switches off for a period that is exponentially distributed with mean λ^{-1}. The variance curve of a single source is

$$v_2(t) = \frac{2\lambda\beta h^2}{(\lambda+\beta)^3}\left(t - \frac{1}{\lambda+\beta}(1 - \exp(-(\lambda+\beta)t))\right). \tag{5.26}$$

b_1	b_2	δ_1	δ_2	H_1	O_1	O_2
0.1	0.5	6.9	13.8	0.5	1.000	0.863
0.1	0.5	6.9	13.8	0.65	1.000	0.945
0.1	0.5	6.9	13.8	0.8	0.995	0.984
0.1	0.5	6.9	13.8	0.95	0.969	0.993
0.1	0.5	13.8	6.9	0.5	1.000	0.686
0.1	0.5	13.8	6.9	0.65	1.000	0.778
0.1	0.5	13.8	6.9	0.8	1.000	0.836
0.1	0.5	13.8	6.9	0.95	1.000	0.880
0.5	0.1	6.9	13.8	0.5	0.562	1.000
0.5	0.1	6.9	13.8	0.65	0.746	1.000
0.5	0.1	6.9	13.8	0.8	0.828	1.000
0.5	0.1	6.9	13.8	0.95	0.879	1.000
0.5	0.1	13.8	6.9	0.5	0.823	0.999
0.5	0.1	13.8	6.9	0.65	0.942	1.000
0.5	0.1	13.8	6.9	0.8	0.990	0.998
0.5	0.1	13.8	6.9	0.95	0.997	0.966

Table 5.1: Sensitivity of O_1 and O_2 with respect to b_1, b_2, δ_1, δ_2 and H_1. (Example 1).

μ_1	O_1	O_2
0.25	0.995	0.983
0.3	0.995	0.981
0.35	0.995	0.980
0.4	0.995	0.979
0.45	0.987	0.970
0.5	0.980	0.963

Table 5.2: Sensitivity of O_1 and O_2 with respect to μ_1. (Example 1).

We first choose $h = 0.032$, $\lambda = 1/0.65$ and $\beta = 1/0.352$ in (5.26), in line with the parameters for coded voice used in [162]. Hence, the mean traffic rate of a source of class 2 (μ_2) is 0.011 Mbps. Note that traffic of class 1 is LRD (i.e., the autocorrelations are non-summable), whereas the traffic of class 2 is SRD.

Table 5.1 shows the values of the performance measures O_1 and O_2 for multiple combinations of b_1, b_2, δ_1, δ_2 and H_1. Note that $\delta_i = 13.8$ ($\delta_i = 6.9$) corresponds to an overflow probability of 10^{-6} (10^{-3}) for class i, $i = 1, 2$. Table 5.1 shows that either O_1 or O_2 (or both) is approximately equal to 1. Hence, this implies that most of R can be obtained by giving priority to class 1 or 2. In case of $O_1 \approx 1$ and $O_2 \approx 1$, it does not really matter which class to prioritize, in the sense that the realizable region is almost completely obtained by applying one of the priority strategies.

One can expect that O_1 and O_2 are sensitive to the traffic characteristics. In order to investigate this, we performed several experiments. Table 5.2 shows the values of O_1 and O_2 for different values of μ_1, given that $b_1 = 0.1$, $b_2 = 0.5$, $\delta_1 = 6.9$, $\delta_2 = 13.8$, $\mu_1 = 0.2$, $h = 0.032$, $\lambda = 1/0.65$ and $\beta = 1/0.352$. The results show that O_1 and O_2 are only mildly affected by μ_1.

Subsequently, we replace λ and β by $\alpha\lambda$ and $\alpha\beta$, respectively, in (5.26), with

H_1	α	O_1	O_2
0.5	2	1.000	0.846
0.5	4	1.000	0.837
0.5	8	1.000	0.836
0.5	16	1.000	0.838
0.65	2	0.999	0.931
0.65	4	0.999	0.928
0.65	8	0.999	0.933
0.65	16	0.999	0.938
0.8	2	0.993	0.974
0.8	4	0.993	0.977
0.8	8	0.992	0.985
0.8	16	0.993	0.992
0.95	2	0.965	0.989
0.95	4	0.959	0.993
0.95	8	0.955	0.998
0.95	16	0.952	1.000

Table 5.3: Sensitivity of O_1 and O_2 with respect to H_1 and α. (Example 1).

$\alpha > 0$. Note that by increasing α, one accelerates the fluctuation-level of the on-off sources, while keeping the mean traffic rate μ_2 constant. Table 5.3 shows the values of O_1 and O_2 for multiple combinations of H_1 and α, assuming that $b_1 = 0.1$, $b_2 = 0.5$, $\delta_1 = 6.9$, $\delta_2 = 13.8$, $\mu_1 = 0.2$, $h = 0.032$, $\lambda = 1/0.65$ and $\beta = 1/0.352$. Table 5.3 shows that the values of O_1 and O_2 are hardly sensitive to the value of α, but very sensitive to H_1. It seems that class 1 is dominating, which can be explained from the fact that traffic of class 1 is LRD, whereas traffic of class 2 is SRD.

In addition to the parameter values presented in Tables 5.1-5.3, we have considered many other parameter values for the B_is, δ_is, μ_is, H_1, α, and c. The result that priority strategies cover nearly the entire realizable region appears to remain valid.

5.5.2 Example 2

In this example we let the two traffic classes share a total capacity of 100 Mbps. Both traffic classes consist of data traffic, where the first class has a high access rate and the second class has a somewhat lower access rate. Recall that the data rate of the class-i user access-channel in a network is known as the access rate of the class-i user, $i = 1, 2$. The speed of the access-channel determines how fast (or the maximum rate) the class-i user can inject data into a network. The variance functions are given by $0.5625t^{2H_1}$ and $0.0025t^{H_2}$, so both classes are modeled as fractional Brownian motions.

Table 5.4 shows the values of O_1 and O_2 for different combinations of b_1, b_2, δ_1, δ_2 and H_1, assuming that $H_2 = 0.8$, $\mu_1 = 3$ and $\mu_2 = 0.2$. Table 5.5 shows the values of O_1 and O_2 for different combinations of μ_1 and μ_2, assuming that $b_1 = 1$, $b_2 = 4$, $\delta_1 = 6.9$, $\delta_2 = 18.4$, $H_1 = 0.8$ and $H_2 = 0.65$. Finally, Table 5.6 shows the values of O_1 and O_2 for different combinations of H_1 and H_2, assuming that $b_1 = 1$, $b_2 = 4$,

b_1	b_2	δ_1	δ_2	H_1	O_1	O_2
1	4	6.9	18.4	0.5	0.992	0.876
1	4	6.9	18.4	0.65	0.973	0.980
1	4	6.9	18.4	0.8	0.936	1.000
1	4	6.9	18.4	0.95	0.895	1.000
1	4	18.4	6.9	0.5	1.000	0.601
1	4	18.4	6.9	0.65	1.000	0.716
1	4	18.4	6.9	0.8	1.000	0.807
1	4	18.4	6.9	0.95	1.000	0.865
4	1	6.9	18.4	0.5	0.402	1.000
4	1	6.9	18.4	0.65	0.647	1.000
4	1	6.9	18.4	0.8	0.783	1.000
4	1	6.9	18.4	0.95	0.853	1.000
4	1	18.4	6.9	0.5	0.842	0.968
4	1	18.4	6.9	0.65	0.976	0.968
4	1	18.4	6.9	0.8	0.998	0.947
4	1	18.4	6.9	0.95	1.000	0.900

Table 5.4: Sensitivity of O_1 and O_2 with respect to b_1, b_2, δ_1, δ_2 and H_1. (Example 2).

$\delta_1 = 6.9$, $\delta_2 = 18.4$, $\mu_1 = 3$ and $\mu_2 = 0.2$.

Tables 5.4-5.6 show again that R is nearly fully covered by $S(0)$ and/or $S(1)$. We have experimented with other parameter values, and observed that this claim was still valid in virtually all situations considered.

5.5.3 Discussion

In Section 5.4.3 we showed that, in case of Brownian inputs, R is accurately approximated by $S(1)$ if $b \leq d$, and by $S(0)$ otherwise. Therefore, if the ratio of the buffer thresholds is less than or equal to the ratio of the (exponential) decay rates of the overflow probabilities, then one should select $(\phi_1, \phi_2) = (1, 0)$, otherwise $(\phi_1, \phi_2) = (0, 1)$. That is, if $b \leq d$ ($b > d$) then one should prioritize class 1 (2). Interestingly, this criterion does not involve the characteristics of the sources. The numerical analysis presented in this section (as well as the additional numerical experiments that we performed) suggest that for other Gaussian sources there is a similar criterion. However, it is in general not given by $b \leq d$ ($b > d$) as is the case for Brownian inputs; it seems that the traffic characteristics of the two classes should be taken into account as well, as illustrated in Tables 5.1-5.6.

In the scenario that one class has bursty traffic and loose QoS requirements, whereas for the other class it is the reverse (smooth traffic and stringent QoS requirements), we can give an argument that may informally explain why nearly the entire realizable region is achievable by strict priority scheduling strategies. In that case the buffer asymptotics of the bursty traffic class will not be affected by the weights (may be even completely insensitive), as long as the traffic intensity of the smooth traffic class (defined as the ratio of the aggregate input rate of the smooth traffic class to the service rate c) does not exceed its weight. The latter will necessar-

μ_1	μ_2	O_1	O_2
3	0.5	0.940	1.000
3	1	0.940	1.000
3	2	0.944	1.000
3	3	0.941	1.000
0.5	0.2	0.854	0.998
2	0.2	0.926	1.000
5	0.2	0.954	0.999
8	0.2	0.952	0.987

Table 5.5: Sensitivity of O_1 and O_2 with respect to μ_1 and μ_2. (Example 2).

H_1	H_2	O_1	O_2
0.5	0.5	0.990	0.855
0.5	0.65	0.990	0.869
0.5	0.8	0.992	0.876
0.5	0.95	0.990	0.877
0.65	0.5	0.971	0.963
0.65	0.65	0.974	0.976
0.65	0.8	0.973	0.980
0.65	0.95	0.971	0.983
0.8	0.5	0.944	0.997
0.8	0.65	0.941	1.000
0.8	0.8	0.936	1.000
0.8	0.95	0.931	1.000
0.95	0.5	0.900	0.999
0.95	0.65	0.898	1.000
0.95	0.8	0.895	1.000
0.95	0.95	0.890	1.000

Table 5.6: Sensitivity of O_1 and O_2 with respect to H_1 and H_2. (Example 2).

ily hold, as otherwise the smooth traffic class would be negatively influenced by the bursty traffic class. This insensitivity implies that there is little lost by simply giving strict priority to the smooth traffic class. In other scenarios there does not seem to be a clear intuitive explanation.

Part II

Flow-level models for bandwidth-sharing networks

Chapter 6

Importance Sampling in rate-sharing networks

In Part I we focused on bandwidth sharing as a result of explicit scheduling in network nodes. In Part II we turn to the case that bandwidth sharing is a consequence of the end-to-end rate control by end-users. The difference between these two cases is that Part I deals with bandwidth sharing among applications on small time scales, whereas Part II considers sharing among routes on somewhat larger time scales. As mentioned in Chapter 1, the latter scenario is well represented by flow-level models, as we have to take the random nature of flows into account, as opposed to Part I, where the number of flows was assumed to be fixed.

Over the past several years the Processor-Sharing (PS) discipline has been widely used for evaluating the flow-level performance of elastic data transfers competing for bandwidth on a single bottleneck link. In a multi-link setting, bandwidth-sharing networks as considered in [135] provide a natural extension for modeling the dynamic interaction among competing elastic flows.

It is well-known that the queue length distribution in a single-server PS system with Poisson arrivals has a simple geometric distribution that only depends on the service requirement distribution through its mean. In contrast, the distribution of the number of active users in bandwidth-sharing networks with several nodes has remained generally intractable, even for exponentially distributed service requirements. The crucial result that the wide family of so-called Alpha-Fair Sharing (AFS) policies, as introduced in [140], achieve stability under the simple condition that no individual link is overloaded, was established in [23]. The family of AFS policies covers several common notions of fairness as special cases, such as max-min fairness ($\alpha \to \infty$), proportional fairness ($\alpha \to 1$), and maximum throughput ($\alpha \downarrow 0$). In [146] it has also been shown that the case $\alpha = 2$, with additional class weights set inversely proportional to the respective RTTs, provides a reasonable modeling abstraction for the bandwidth sharing realized by TCP in the Internet. We refer to Chapter 1 for more

details.

In this chapter we consider a network operating under an AFS policy. Since the service rate allocated to a flow is restricted in practice [28], we impose class-dependent access-link rate limitations, similar as in [14]. Assuming Poisson arrivals and exponentially distributed service requirements for each class, the dynamics of the user population may be described by a Markov process.

An essential requirement of modern bandwidth-sharing networks is their capability of providing a variety of Quality-of-Service (QoS) guarantees, where QoS is usually expressed in term of constraints on a set of performance measures, such as mean transfer delays, but also the probability that there are many flows (per class) active in the network. Typically, such a probability is required to be below some small threshold, as this prevents flows from experiencing large delays. Motivated by this, we analyze in this chapter the probability that, given that the network is in some specific state n_0 at time 0, the network is in some set of states A after some predefined time T. In particular, we assume that the underlying event is rare, i.e., this probability is small. As in general no explicit expressions are known for the probability of interest, an attractive approach may be to resort to *Monte-Carlo* (MC) *simulation*. In general, the number of runs needed to obtain an estimate with predefined accuracy and confidence, is inversely proportional to the probability to be estimated [80], implying that MC simulation is impractical due to the rarity of the event under consideration. A natural method to accelerate the simulation is to use *Importance Sampling* (IS). The idea underlying IS is to simulate the system with a new set of input probability distributions, i.e., new interarrival and service time distributions, such that the rare event becomes more likely, and then to correct the simulation output with appropriate likelihood ratios, in order to obtain an unbiased estimate.

To obtain appropriate new input distributions we first identify the most probable path (MPP) for the event to occur. Informally speaking, given that this rare event occurs, with overwhelming probability it will happen by a path close to this MPP. For the M/M/1-PS queue the MPP is already known [161], whereas this is not the case for a general AFS network topology. We develop an approach for finding the MPP, which exploits the large deviations results of [161]. The underlying idea is that locally the flow-level dynamics of a particular class in the network can be approximated as a M/M/1-PS queue. It is noted that, in contrast to the M/M/1-PS queue where the most likely path has a linear shape, the MPP has a non-linear shape in case of a general AFS network topology. The path is then subsequently translated into new input distributions, that are such that the event under consideration occurs by realizations close to this MPP.

Extensive numerical experiments indicate that the above approach is quite effective: we are able to estimate probabilities as small as 10^{-13} quickly, whereas 10^{-8} up to 10^{-4} is typically the range of interest. It is emphasized that we do not prove that our IS technique is *asymptotically optimal* or *asymptotically efficient* [41]. The

numerical experiments, however, suggest that the IS scheme is close to asymptotically optimal.

The remainder of the chapter is organized as follows. In Section 6.1 we first provide a detailed model description, discuss the use of IS, and present a key large deviations theorem. Section 6.2 deals with the M/M/1-PS queue, which is in fact a special case of our network. In Section 6.3 we derive (that is, approximate, but the approximation can be made arbitrarily close) the MPP for a rare event to occur in a general AFS network topology, by exploiting the results for the M/M/1-PS queue. Section 6.4 shows how one can translate this MPP into new input distributions that can be incorporated in an IS algorithm. The pseudocode of the IS algorithm is presented in the Appendix. Section 6.5 examines the performance of the IS algorithm for two special networks, and shows that the IS scheme performs well. Finally, Section 6.6 concludes with some final observations.

6.1 Preliminaries

In this section we first describe our queueing model. Next we discuss IS, a simulation technique designed for estimating rare event probabilities. Finally, we briefly discuss some large deviations results, which are needed in the analysis.

6.1.1 Queueing model

We consider a network consisting of L nodes, where node j has capacity c_j, $j = 1, \ldots, L$. There are M classes of users in the network, where each class corresponds to a specific route in the network. We assume that class-i users arrive according to a Poisson process of rate λ_i, and have independent and exponentially distributed service requirements with mean μ_i^{-1}, $i = 1, \ldots, M$. The traffic load of class i is then $\rho_i := \lambda_i/\mu_i$, $i = 1, \ldots, M$. The arrival processes and service requirements are all assumed to be independent. If a user requires service at multiple nodes, then it must be served at all nodes simultaneously. Let $S(j)$ denote the set of classes that require service at node j, $j = 1, \ldots, L$. Finally, let $N(t) = (N_1(t), \ldots, N_M(t)) \in \mathbb{N}_0^M$ be a vector denoting the state of the network at time $t \geq 0$, with $N_i(t)$ representing the number of class-i users at time $t \geq 0$.

The network operates under the AFS policy, as introduced in [140]. When the network is in state $n = (n_1, \ldots, n_M) \in \mathbb{N}_0^M \setminus \{\vec{0}\}$, the service rate x_i^* allocated to each of the class-i users is obtained by solving the optimization problem presented in Section 1.4.3.

Let $s_i(n) := n_i x_i^*$ denote the total service rate allocated to class i. Since the rate allocated to single flows is often restricted in practice, we assume that the effective total rate allocated to class-i users is [14]

$$d_i(n) := \min\{s_i(n), n_i r_i\},$$

where r_i can be thought of as the access-link rate limitation for a class-i flow, $i = 1, \ldots, M$.

Remark: Above we first determined the AFS allocation, and then truncated the resulting rates at the access-link rates r_i, $i = 1, \ldots, M$. Note that if $n_i r_i < s_i(n)$ for some class i, then the excess rate $s_i(n) - n_i r_i$ does not become available to the other classes. A way to circumvent this problem is to incorporate the restrictions $x_i \leq r_i$, $i = 1, \ldots, M$, directly in the optimization problem (1.2). In this chapter, however, we do not choose to use this latter approach; it is not in the scope of this chapter to verify which of these alternatives is closest to reality. It turns out that in general our approach allows fairly explicit analysis, whereas this is considerably harder under the alternative method, see also [14].

It is easily verified that $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \nu_i(n), \quad i = 1, \ldots, M,$$

where $\nu_i(n) := \mu_i d_i(n)$. We note that $N(t)$ is in fact an M-dimensional birth-death process. Given that $r_i \geq c_i$, $i = 1, \ldots, M$, i.e., given that there are *no* access-link rate limitations, in [23] the appealing result was shown that $N(t)$ is an ergodic Markov process if

$$\sum_{i \in S(j)} \rho_i < c_j, \quad j = 1, \ldots, L. \tag{6.1}$$

Since the 'down' rates of our system differ only for a finite number of states from those in a similar system without rate limitations, it follows from Proposition 1 in [111] that $N(t)$ is ergodic for all values of $r_i > 0$, $i = 1, \ldots, M$, given that (6.1) holds. We emphasize that in general no explicit expressions are known for the steady-state distribution of $N(t)$.

In this chapter our goal is to estimate

$$P := \mathbb{P}(N(T) \in A | N(0) = n_0),$$

i.e., the probability that, given that network is in state n_0 at time 0, the state of the network at time $T > 0$ is contained in set A. For example, here n_0 might be a state around which the network operates most of the time, and A might be an 'overflow set':

$$\left\{ (x_1, \ldots, x_M) \in \mathbb{N}_0^M \,\middle|\, \sum_{i=1}^{M} x_i > b \right\},$$

where $b \geq 0$ is a scalar.

6.1.2 Variance-reduction technique

As in general no analytical expression for P is known, a natural approach to obtain an estimate of P is to perform simulation experiments. Let $\Omega = \{f_i, i = 1, 2, \dots\}$ be the set of all paths f in the evolution of the system, given that the system is in state n_0 at time $t = 0$, i.e., $f(0) = n_0$. Let 1_E be an indicator of the event E, and $p(f)$ the probability measure of the sample path f. Then we obtain that

$$P = \int_\Omega 1_{f(T)\in A}\, p(f)\mathrm{d}f = \mathbb{E}_p\left(1_{f(T)\in A}\right), \tag{6.2}$$

where the subscript p indicates sampling from the measure p. An unbiased estimate of (6.2) can be obtained by performing MC simulation, i.e., we run R independent simulations, with the system starting in state n_0, and we determine

$$P_{\mathrm{MC}} := \frac{1}{R}\sum_{i=1}^{R} 1_{f_i(T)\in A},$$

where f_i is the path obtained in the ith run. In case n_0 and A are such that $f(T) \in A$ occurs relatively often, we can accurately estimate P in a relatively small amount of time by P_{MC}. The number of runs needed to obtain an estimate with predefined accuracy and confidence, is in general inversely proportional to the probability to be estimated, see e.g. [80].

If n_0 and A are such that $f(T) \in A$ is a rare event, then the above properties entail that we need a large number of simulations to provide an accurate statistical estimate of P. In this case the simulation can be accelerated by using IS. The idea underlying IS is to simulate the system with a new set of input probability distributions, such that the rare event becomes more likely. To this end, let us consider a new probability measure p'. Then, (6.2) is equivalent to

$$\begin{aligned} P &= \int_\Omega 1_{f(T)\in A}\frac{p(f)}{p'(f)}p'(f)\mathrm{d}f \\ &= \int_\Omega 1_{f(T)\in A}L(f)p'(f)\mathrm{d}f \\ &= \mathbb{E}_{p'}\left(1_{f(T)\in A}L(f)\right), \end{aligned} \tag{6.3}$$

where $L(f) := p(f)/p'(f)$ is called the *likelihood ratio*. Note that (6.3) is valid for any measure $p'(\cdot)$, given that $p'(f) > 0$ for all f that are such that $f(T) \in A$. Hence, an unbiased IS estimator is given by

$$P_{\mathrm{IS}} := \frac{1}{R}\sum_{i=1}^{R} 1_{f_i(T)\in A}L(f_i),$$

where f_i is now simulated under the measure p', with $f_i(0) = n_0$, $i = 1, \dots, R$.

Assuming that $L(f)$ can be found, the simulation can be accelerated considerably if p' is properly chosen, in the sense that the number of runs needed to obtain an accurate statistical estimate of P using $P_{\rm IS}$, is less than the number of runs required in case of MC simulation. Hence, IS can be seen as a variance-reduction technique. We note, however, that not every choice of p' will reduce the variance. In fact, if p' is badly chosen, then this may increase the variance, or even make it infinite.

In this chapter we assume that n_0 and A are such that $f(T) \in A$ is a rare event. As mentioned above, in this case MC simulation is inefficient, and one may resort to IS to obtain an estimate of P. We derive an IS scheme that considerably speeds up the simulation. This scheme is based on sample-path large deviations results, see e.g. [161].

6.1.3 Large deviations

In this subsection we present large deviations results of [161], which will be needed in the next sections.

Let $X(t)$ be a Markovian jump process with state space $\mathbb{R}^d$ and with transition rates:

$$q(x, x + v_i) = \psi_i(x),$$

where v_i is a vector in $\mathbb{R}^d$ and $\psi_i(x)$ is the rate of the jump in that direction when the state is x, $i = 1, \ldots, l$. Also, let $\overline{X}^k(t) := X(kt)/k$, $t \geq 0$, $k \geq 1$, be the fluid scaled process, which is obtained by making the jumps smaller, but faster. Define the 'local' rate function

$$\ell(x, y) := \sup_{\theta} \left(\langle \theta, y \rangle - \sum_{i=1}^{l} \psi_i(x) \left(e^{\langle \theta, v_i \rangle} - 1 \right) \right),$$

where x, y and θ are in $\mathbb{R}^d$, and $\langle \cdot, \cdot \rangle$ denotes the usual inner product: $\langle a, b \rangle := \sum_{i=1}^{d} a_i b_i$. Finally, define the rate function

$$I_T(f) := \begin{cases} \int_0^T \ell(f(s), f'(s)) \mathrm{d}s & \text{if } f \text{ is absolutely continuous;} \\ \infty & \text{otherwise,} \end{cases}$$

where f is in $\mathbb{R}^d$, and f' is the derivative of f. The following sample-path large deviations principle (LDP) now holds (see Theorem 5.1 in [161]).

Theorem 6.1.1 *For any well-defined x_0 and set F,*

$$-\lim_{k \to \infty} \frac{1}{k} \log \mathbb{P}\left(\overline{X}^k(\cdot) \in F | \overline{X}^k(0) = x_0 \right) = \inf_{f : f \in F, f(0) = x_0} I_T(f).$$

Remark: Intentionally, Theorem 6.1.1 has been formulated in a slightly imprecise manner. In fact, the LDP consists of an upper and lower bound, which apply to closed and open sets, respectively, see Theorem 5.1 in [161]. However, for the purpose of this chapter, it is sufficient to state the theorem as above. For more details we refer to Chapter 5 of [161].

Let us write $g(x) \sim h(x)$ when $g(x)/h(x) \to 1$ if $x \to \infty$. Then it follows from the above that

$$\mathbb{P}\left(\overline{X}^k(\cdot) \in F | \overline{X}^k(0) = x_0\right) \sim g(k, F, x_0) e^{-kI_T(f^*)}, \quad k \to \infty,$$

where f^* is the optimizing path in Theorem 6.1.1, and $g(k, F, x_0)$ is a subexponential function, i.e.,

$$\lim_{k\to\infty} \frac{\log g(k, F, x_0)}{k} = 0.$$

From the above it follows that Theorem 6.1.1 only gives the logarithmic asymptotics. Therefore, in general Theorem 6.1.1 does not provide any information on the function $g(k, F, x_0)$, which implies that we can only use it to obtain a rough estimate of $\mathbb{P}\left(\overline{X}^k(\cdot) \in F | \overline{X}^k(0) = x_0\right)$.

In the next section we apply Theorem 6.1.1 to the so-called free M/M/1-PS process.

6.2 Free M/M/1-PS process

We first assume that $X(t)$ corresponds to the free M/M/1-PS process, i.e., the M/M/1-PS queue that is not reflected at 0, meaning that the state space of $X(t)$ is $\mathbb{Z}$ (whereas the state space of a M/M/1-PS queue is $\mathbb{N}_0$). We note that the queue length dynamics of the M/M/1-PS queue coincide with those of the M/M/1-First-In First-Out (FIFO) queue, implying that both have the same steady-state queue length distribution. Hence, the results derived in this section in fact hold for the free M/M/1-FIFO process as well.

In this section we treat the free M/M/1-PS process, because this plays a key role in the analysis of a general AFS network topology, as we will see in Section 6.3. This may sound surprising, as the down rates corresponding to free M/M/1-PS process are constant, whereas the down rates corresponding to a general AFS network topology are state-dependent. The idea underlying this analysis is that we can locally approximate the flow-level dynamics of a particular class in a general AFS network topology by a free M/M/1-PS process with class-specific arrival and service rates, which will be exploited in the next sections to obtain an estimate of P.

Since $X(t)$ corresponds to the free M/M/1-PS process, we have that $X(t) = X_{\text{up}}(t) - X_{\text{down}}(t)$, where $X_{\text{up}}(t)$ is a Poisson process of rate λ and $X_{\text{down}}(t)$ is an

independent Poisson process of rate μ. Assume that $\lambda < \mu$, so that $X(t)$ has a negative drift. The transition structure of $X(t)$ is then, in the terminology of Section 6.1.3,

$$\begin{aligned} v_1 &= +1; \quad \psi_1(x) = \lambda; \\ v_2 &= -1; \quad \psi_2(x) = \mu, \end{aligned}$$

with $x \in \mathbb{Z}$. Then,

$$\ell(x, y) = \ell(y) = \sup_{\theta} \left\{ \theta y - \lambda \left(e^{\theta} - 1 \right) - \mu \left(e^{-\theta} - 1 \right) \right\},$$

i.e., the local rate function is independent of the current state x. Straightforward calculus shows that the optimizer satisfies

$$e^{\theta^*} = \frac{y + \sqrt{y^2 + 4\lambda\mu}}{2\mu},$$

which yields

$$\begin{aligned} \ell(y) &= y \log \left(\frac{y + \sqrt{y^2 + 4\lambda\mu}}{2\lambda} \right) + \lambda + \mu - \sqrt{y^2 + 4\lambda\mu} \\ &=: \ell(y|\lambda, \mu). \end{aligned}$$

We now focus on the overflow probability

$$\mathbb{P}(\overline{X}^k(T) > z | \overline{X}^k(0) = z_0),$$

with $z > z_0$. Using Theorem 6.1.1, we have that

$$\mathbb{P}(\overline{X}^k(T) > z | \overline{X}^k(0) = z_0) \approx e^{-kI^*},$$

where

$$I^* := \inf_{f : f \in G, f(0) = z_0} I_T(f), \text{ with } G := \{f | f(T) > z\}.$$

In Lemma 5.16 of [161] it is shown that the MPP, i.e., the path f^* in the set G that minimizes I_T, is a straight line from z_0 to z in the interval $[0, T]$, with cost

$$\begin{aligned} I^* &= I_T(f^*) = T \times \ell \left(\frac{z - z_0}{T} \middle| \lambda, \mu \right) \\ &= T \left(\frac{z - z_0}{T} \log \left(\frac{z - z_0}{2T\lambda} + \frac{1}{2\lambda} \sqrt{\frac{(z - z_0)^2}{T^2} + 4\lambda\mu} \right) \right. \\ &\quad \left. + \lambda + \mu - \sqrt{\frac{(z - z_0)^2}{T^2} + 4\lambda\mu} \right) \\ &=: \mathbb{C}(z - z_0, T | \lambda, \mu). \end{aligned} \tag{6.4}$$

Below we try to provide some additional interpretation for (6.4). First note that the cost of a Poisson process of rate λ behaving like a Poisson process of rate λ^* is, during one unit of time,

$$\tilde{I}(\lambda^*|\lambda) := \lambda^* \log\left(\frac{\lambda^*}{\lambda}\right) + \lambda - \lambda^*,$$

see page 20 of [161]. Here $\tilde{I}(\lambda^*|\lambda)$ is the Legendre transform of the logarithmic Moment Generating Function (MGF) of a random variable that has a Poisson distribution with mean λ. Clearly, $\tilde{I}(\mu^*|\mu)$ follows in the same way. Observe that indeed $\tilde{I}(p|p) = 0$, $p = \lambda, \mu$, as required.

In order to make sure that it becomes likely that $X(T) > z$, given that $X(0) = z_0$, we should have that $X_{\rm up}$ ($X_{\rm down}$) behaves as a different Poisson process of rate λ^* (μ^*), where $(\lambda^* - \mu^*)T > z - z_0$. We thus get the minimization problem:

$$T \min_{\lambda^*,\mu^*} \left\{ \tilde{I}(\lambda^*|\lambda) + \tilde{I}(\mu^*|\mu) \right\},$$

over all λ^*, μ^* such that $(\lambda^* - \mu^*)T > z - z_0$. Straightforward calculations yield that the optimizers are

$$\begin{aligned} \lambda^* &= \frac{z - z_0}{2T} + \frac{1}{2}\sqrt{\frac{(z-z_0)^2}{T^2} + 4\lambda\mu}; \\ \mu^* &= -\frac{z - z_0}{2T} + \frac{1}{2}\sqrt{\frac{(z-z_0)^2}{T^2} + 4\lambda\mu}, \end{aligned} \tag{6.5}$$

and the corresponding objective function value indeed equals (6.4).

6.3 Most probable path

In the previous section we obtained an approximation for the overflow probability in the M/M/1-PS queue (where we assumed that there was no reflection at 0). In this section we use the same ideas to derive an approximation for P in a general AFS network topology.

We first consider the cost $\mathbb{K}(f, T)$ of a path f, with $f(0) = n_0$, in the interval $[0, T]$. We find that

$$\mathbb{K}(f, T) = \sum_{i=1}^{M} \int_0^T \ell\left(f_i'(t)|\lambda_i, \nu_i(f(t))\right) \mathrm{d}t.$$

From the logarithmic asymptotics stated in Theorem 6.1.1 it then follows that the following approximation applies:

$$\begin{aligned} P &= \mathbb{P}(N(T) \in A | N(0) = n_0) \\ &\approx \exp\left(- \inf_{f: f(T) \in A, f(0) = n_0} \mathbb{K}(f, T) \right). \end{aligned} \tag{6.6}$$

Let f^* denote the path that minimizes the cost, i.e., the MPP. Since the down rates in our model are state-dependent, in contrast to what is the case for the free M/M/1-PS process, the MPP in general has a non-linear shape. In fact, in general no closed-form expression is available for the path that minimizes $\mathbb{K}(f,T)$. Equation (6.6) suggests that we should try to find an accurate approximation of f^* to obtain an estimate of P, which is done below.

Divide T into m (which is typically a large number) subintervals of length $\Delta_m := T/m$. Consider the contribution to a path of the k-th subinterval, i.e., the interval $[k\Delta_m,(k+1)\Delta_m)$, for $k=0,\ldots,m-1$, and assume that the down rates are $\nu_i(f(k\triangle_m))$, $i=1,\ldots,M$, in this subinterval. Then the cost of this time interval, related to class i are given by

$$\mathbb{C}(f_i((k+1)\Delta_m)-f_i(k\Delta_m),\Delta_m|\lambda_i,\nu_i(f(k\Delta_m)).$$

Hence, we find that the total cost $\mathbb{K}_m(f,T)$ are

$$\sum_{i=1}^{M}\sum_{k=0}^{m-1}\mathbb{C}(f_i((k+1)\Delta_m)-f_i(k\Delta_m),\Delta_m|\lambda_i,\nu_i(f(k\Delta_m))).$$

Note that the higher the value of m, the more accurate the approximation will be, i.e.,

$$\lim_{m\to\infty}\mathbb{K}_m(f,T)=\mathbb{K}(f,T).$$

Using the above, we can approximate $\mathbb{K}(f,T)$, for given $m\in\mathbb{N}$, by $\mathbb{K}_m(f,T)$. Also, the path that minimizes $\mathbb{K}_m(f,T)$ can be regarded as an approximation of f^*. In order to obtain this approximating path, optimization should be performed over all $f_i(j\Delta_m)$, $i=1,\ldots,M$, $j=0,\ldots,m$, i.e., $(m+1)M$ entries, given that $f(0)=n_0$ and $f(m\Delta_m)=f(T)\in A$.

Approximation (6.6) turns out not to be very accurate in general. Clearly, this is no surprise, as in Section 6.1.3 we already argued that Theorem 6.1.1 just gives the logarithmic asymptotics, and that we therefore have only a rough estimate of P.

6.4 New input distributions

In the previous section we derived an approximation for P which required the calculation of an optimizing path. This path can be regarded as an approximation for the most likely way for the event to happen. That is, given that the event occurs, with overwhelming probability $N(T)\in A$ is reached by a path close to this optimizing path. In this section we show how we can exploit the results of Section 6.3 to develop a methodology for obtaining an accurate estimate of P.

Assume that we have (an accurate approximation of) the MPP

$$f^*:=\arg\inf_{f:f(T)\in A,f(0)=n_0}\mathbb{K}(f,T),$$

as discussed in the previous section. Suggested by (6.5), the following change-of-measure at time t corresponds to f^*:

$$\lambda_i^*(t) := \frac{1}{2}(f_i^*)'(t) + \frac{1}{2}\sqrt{((f_i^*)'(t))^2 + 4\lambda_i\nu_i(f^*(t))};$$

$$\nu_i^*(t) := -\frac{1}{2}(f_i^*)'(t) + \frac{1}{2}\sqrt{((f_i^*)'(t))^2 + 4\lambda_i\nu_i(f^*(t))},$$

$i = 1, \ldots, M$. When, at time $t \geq 0$, the process is simulated with arrival rates $\lambda^*(t)$ and departure rates $\nu^*(t)$, given that the process starts at n_0 at $t = 0$, it is not hard to see that the ith coordinate of the expected position of the process at time t is

$$\begin{aligned} n_{0,i} + \int_0^t \lambda_i^*(s)\mathrm{d}s - \int_0^t \nu_i^*(s)\mathrm{d}s &= f_i^*(0) + \int_0^t (f_i^*)'(s)\mathrm{d}s \\ &= f_i^*(t), \end{aligned}$$

$i = 1, \ldots, M$, i.e., the process has the 'correct' expected position, under this change-of-measure.

In the Appendix we present an IS scheme that can be used to obtain an estimate of P. The basic idea underlying this scheme is to simulate the model with rates $\lambda_i^*(t)$ and $\nu_i^*(t)$, $i = 1, \ldots, M$. Typically, we only know these rates at $m+1$ time points, as in general the MPP is not explicitly known, but approximated, see Section 6.3. However, if one assumes the rates to be constant between two consecutive time points, i.e., in a subinterval, then each class essentially behaves as a free M/M/1-PS process with class-specific arrival and service rate in this subinterval, which is easy to simulate. For more details we refer to the Appendix.

In the next section we show that, compared to MC simulation, this scheme can considerably speed up the simulation, given that the underlying event is rare. That is, the number of runs that are needed to achieve some given level of confidence with the IS scheme, is substantially less than the number of runs needed with MC simulation.

6.5 Simulation results

In this section the performance of the IS algorithm is examined in case of a single-node network (shared by multiple traffic classes) and a linear network, respectively. These are the two simplest networks, and therefore of particular interest to gain insight. We have performed extensive simulation experiments for each of these two networks, and the results are presented below. We mention that, besides the results reported in this section, we have considered many other examples, in which usually a substantial speed-up is achieved

6.5.1 Single-node network

We first consider a single-node network with capacity c, where capacity is shared between M classes. In order to obtain the AFS allocation we have to solve the following optimization problem for state $n \in \mathbb{N}_0^M \setminus \{\vec{0}\}$:

$$\begin{array}{rl} \max & \sum_{i=1}^M n_i U_i(x_i) \\ \text{subject to} & \sum_{i=1}^M n_i x_i \leq c \\ \text{over} & x_i \geq 0, \quad i = 1, \ldots, M, \end{array}$$

where $U_i(x_i)$ is as defined in (1.3). It is a straightforward exercise to show that the optimizers are such that

$$s_i(n) = n_i x_i^* = \frac{\kappa_i^{1/\alpha} n_i c}{\sum_{j=1}^M \kappa_j^{1/\alpha} n_j}, \quad i = 1, \ldots, M. \tag{6.7}$$

From (6.7) it follows that AFS in a single-node network corresponds to sharing in a DPS fashion, with relative weights $\kappa_i^{1/\alpha}$, $i = 1, \ldots, M$, see [63]. We find [14] that

$$d_i(n) = \min \left\{ \frac{\kappa_i^{1/\alpha} n_i c}{\sum_{j=1}^M \kappa_j^{1/\alpha} n_j}, n_i r_i \right\}, \quad i = 1, \ldots, M.$$

The steady-state distribution of $N(t)$ is only known in explicit form for some special cases, given that the stability condition $\sum_{i=1}^M \rho_i < c$ holds. In case $\kappa_i = \kappa$ and $r_i \geq c$, $i = 1, \ldots, M$, the steady-state distribution is given by Equation (1.7). The steady-state distribution is available as well in case $\kappa_i = \kappa$ and $r_i = r \leq c$, $i = 1, \ldots, M$.

The first part of the IS algorithm consists of finding a MPP. We have performed numerical experiments to gain insight in the typical shape of such a minimizing path. We consider the setting with $M = 2$, $\lambda_1 = 0.75$, $\lambda_2 = 1.5$, $\mu_1 = 2$, $\mu_2 = 4$, $\kappa_1^{1/\alpha} = 1/3$, $\kappa_2^{1/\alpha} = 2/3$, $r_1 = 0.9$, $r_2 = 0.8$, and $c = 1$, and we let T, n_0 and the set A vary. The results are depicted in Figure 6.1, and are obtained by using an optimization procedure in Mathematica 5.2. We solved the problem for $m = 2^p$, $p = 1, \ldots, 5$, and we used the minimizing path found for $m = 2^{q-1}$ as starting path in the optimization procedure for $m = 2^q$, $q = 2, \ldots, 5$ (for $m = 2$ we do not have a nice starting path). Hence, the depicted paths are associated with $m = 2^5 = 32$. We note that the above approach is much faster than solving the optimization problem directly for $m = 32$ (without an appropriate starting path). We observed that the optimization problem can be solved in a relatively small amount of time if $m \leq 32$. For higher values of m the obtained path is almost identical to the one obtained for $m = 32$, but the computation requires more time. In the first, second and third column of Figure 6.1 we depict $(f_1(i\Delta_{32}), i\Delta_{32})$, $(f_2(i\Delta_{32}), i\Delta_{32})$ and $(f_1(i\Delta_{32}), f_2(i\Delta_{32}))$, $i = 0, \ldots, 32$, respectively. Since we only know the minimizing paths at $m + 1 = 33$ time points, we linearly interpolate between consecutive points. We note that we have considered

many other scenarios besides the ones depicted in Figure 6.1. In these cases, the minimizing paths do not seem to be linear either.

Although the shapes of the MPPs corresponding to scenarios (a)-(c) are not always trivial, the shape of the path corresponding to scenario (d) perhaps requires some more explanation. In particular, the shape of the path corresponding to class 1 is surprising in this scenario: it first slightly decreases, and then it starts to increase. A possible explanation for this phenomenon may be the following. In [14] it was shown that there exists a unique point $n^* = (n_1^*, n_2^*)$ such that $\lambda_i = d_i(n^*)$, $i = 1, 2$. This is the equilibrium point of the so-called fluid limit: the system operates (most likely) most of the time around this point. The fluid limit is obtained by both speeding up the arrivals and service speed by a given factor, and then letting this factor go to infinity. It can be shown that the resulting normalized Markov process converges to a deterministic limit. From Proposition 2.1 in [14] it follows that $n_1^* = 0.5625$ and $n_2^* = 0.46875$ in scenarios (a)-(d). Recalling that the path starts in $n_0 = (3, 0)$ in scenario (d), we see that the MPP initially evolves in the direction of the fluid limit, but then changes its direction to make sure that $f_2(T) > 6$. It remains, however, hard to fully explain the shapes of the MPPs in general. One can imagine that the MPP from any n_0 to any set A is more or less linear if T is relatively small. In contrast, if T is relatively large, then one can expect that the MPP first drifts to n^*, and then changes its direction towards the set A, see e.g. [123]. We remark that the equilibrium point n^* depends critically on the access-link rates [14].

To quantify the performance of the proposed IS scheme we take the same parameter values as above, where we let T, n_0 and the set A vary. We consider three structures for A: (i) $\{f|f_1(T) > a\}$, (ii) $\{f|f_2(T) > a\}$ and (iii) $\{f|f_1(T)+f_2(T) > a\}$, with $a > 0$. The results are presented in Tables 6.1-6.4. These results (and also the ones in the next subsection) are obtained with Mathematica 5.2 and are tested on a personal computer with an AMD Athlon 64 3500+ processor (2.2 GHz). In the tables $\#_{\rm IS}$ ($\#_{\rm MC}$) denotes the number of runs needed with IS (MC) simulation to obtain a confidence of 95% and a relative efficiency (i.e., the ratio of the confidence interval half-length to the estimated value) of 10%, and $\tau_{\rm IS}$ ($\tau_{\rm MC}$) denotes the time needed with IS (MC simulation). Note that $\tau_{\rm IS}$ consists of two parts: (a) finding the optimal path and (b) performing the simulation with the new input distributions.

Table 6.1 compares IS with MC simulation. The MC estimator is obtained by simulating independent runs of the original model (starting in n_0) until time T, and subsequently determining the fraction of the runs that are such that $f(T) \in A$. The table shows that for a relatively large value of P (larger than 0.01), MC simulation yields an accurate estimate much faster than the IS scheme does. In contrast, for a relatively small value of P (smaller than 0.01), IS significantly outperforms MC simulation. Clearly, this is no surprise: the IS scheme presented in the Appendix is based on large deviations results, and therefore one expects this scheme to perform well in case the underlying event is rare, i.e., if P is relatively small.

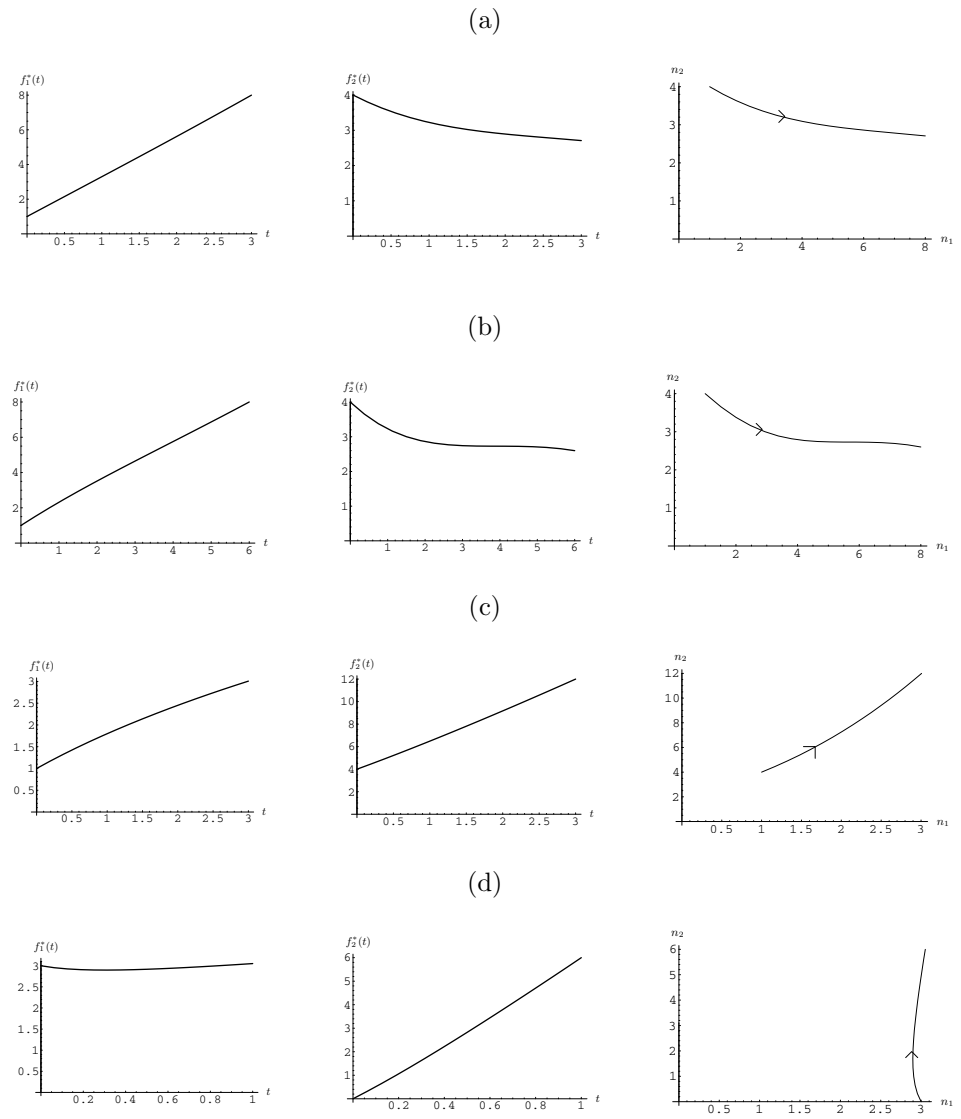

Figure 6.1: The minimizing paths of $\mathbb{K}_{32}(f, T)$ in four scenarios. Scenario (a): $T = 3$, $A = \{f | f_1(T) > 8\}$ and $n_0 = (1, 4)$. Scenario (b): $T = 6$, $A = \{f | f_1(T) > 8\}$ and $n_0 = (1, 4)$. Scenario (c): $T = 3$, $A = \{f | f_2(T) > 12\}$ and $n_0 = (1, 4)$. Scenario (d): $T = 1$, $A = \{f | f_2(T) > 6\}$ and $n_0 = (3, 0)$. The left panel shows $f_1^*(\cdot)$ as function of time t. The middle panel shows $f_2^*(\cdot)$ as function of time t. The right panel shows the parametric plot of $(f_1^*(\cdot), f_2^*(\cdot))$.

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}	P_{MC}	$\#_{\text{MC}}$	τ_{MC}
1	6	1	4	$6.1 \cdot 10^{-4}$	1559	12.4	$6.2 \cdot 10^{-4}$	622085	341.1
1	6	1	0	$3.0 \cdot 10^{-4}$	3312	16.4	$2.7 \cdot 10^{-4}$	1468866	523.4
3	6	1	4	$1.5 \cdot 10^{-2}$	2441	17.2	$1.4 \cdot 10^{-2}$	27518	35.9
3	6	1	0	$5.1 \cdot 10^{-3}$	22745	91.4	$4.9 \cdot 10^{-3}$	78851	77.7
6	6	1	4	$4.0 \cdot 10^{-2}$	12821	80.4	$3.7 \cdot 10^{-2}$	9397	23.4
6	6	1	0	$2.0 \cdot 10^{-2}$	74237	390.1	$1.8 \cdot 10^{-2}$	22397	44.9

Table 6.1: Simulation results for structure (i): comparison with MC simulation (times in seconds).

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	10	1	0	$3.3 \cdot 10^{-8}$	3509	18.9
1	14	1	0	$5.1 \cdot 10^{-13}$	4744	24.4
2	10	1	0	$2.9 \cdot 10^{-6}$	6762	33.9
2	14	1	0	$7.3 \cdot 10^{-10}$	8123	40.7
3	10	1	0	$2.6 \cdot 10^{-5}$	12439	59.2
3	14	1	0	$2.5 \cdot 10^{-8}$	12526	63.7
1	10	1	4	$4.2 \cdot 10^{-8}$	2861	17.5
1	14	1	4	$7.8 \cdot 10^{-13}$	3349	20.6
2	10	1	4	$6.0 \cdot 10^{-6}$	2680	19.5
2	14	1	4	$1.2 \cdot 10^{-9}$	4686	28.5
3	10	1	4	$6.2 \cdot 10^{-5}$	4173	25.8
3	14	1	4	$5.6 \cdot 10^{-8}$	5847	37.4

Table 6.2: Simulation results for structure (i): rare events (times in seconds).

Tables 6.2-6.4 show the performance of our scheme in case of rare events. As mentioned in Section 6.1.2, in this case MC simulation is inefficient. Therefore, we have decided not to compare the performance of the IS scheme with that of the MC simulation. These tables show that our scheme works remarkably well for rare events: we are able to estimate probabilities as small as 10^{-13} in a fast way.

The results also show that the performance of the IS scheme decreases as T increases (for fixed other model parameters), i.e., more runs are needed to achieve the required efficiency. This can be explained as follows. As T increases and m (the number of subintervals) remains constant, the approximation of the minimizing path becomes less accurate, and therefore the performance of the IS algorithm is also negatively affected.

We also empirically observed that, for fixed arbitrarily chosen n_0 and T, the ratio

$$\frac{\log \mathbb{E}_{p'} \left(1_{f(T) \in A \cdot k} L^2(f)\right)}{\log \mathbb{E}_{p'} \left(1_{f(T) \in A \cdot k} L(f)\right)}, \tag{6.8}$$

is close to (but smaller than) 2 for large values of k, where p' is the IS distribution and $A \cdot k := \{y : y/k \in A\}$. It is noted that one can estimate both denominator and numerator in (6.8) by using the simulation output. The above suggests that our IS scheme is nearly asymptotically optimal [41], which however seems difficult to prove.

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	10	1	0	$5.4 \cdot 10^{-7}$	2702	15.3
1	14	1	0	$1.0 \cdot 10^{-10}$	3624	21.5
2	10	1	0	$3.1 \cdot 10^{-5}$	8958	40.4
2	14	1	0	$8.4 \cdot 10^{-8}$	4756	28.6
3	10	1	0	$1.6 \cdot 10^{-4}$	10423	51.0
3	14	1	0	$1.7 \cdot 10^{-6}$	18663	94.6
1	16	1	4	$5.5 \cdot 10^{-9}$	2776	16.2
1	20	1	4	$7.2 \cdot 10^{-13}$	3580	21.1
2	16	1	4	$1.1 \cdot 10^{-6}$	2613	17.6
2	20	1	4	$1.7 \cdot 10^{-9}$	3792	24.4
3	16	1	4	$8.0 \cdot 10^{-6}$	3517	24.6
3	20	1	4	$6.2 \cdot 10^{-8}$	4152	25.9

Table 6.3: Simulation results for structure (*ii*): rare events (times in seconds).

T	a	$n_{0,1}$	$n_{0,2}$	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	20	1	0	$2.3 \cdot 10^{-13}$	4396	67.6
2	25	1	0	$5.1 \cdot 10^{-13}$	6605	80.8
3	30	1	0	$1.2 \cdot 10^{-13}$	12017	107.1
1	20	1	4	$9.6 \cdot 10^{-10}$	3281	52.1
2	25	1	4	$3.5 \cdot 10^{-10}$	5156	60.4
3	30	1	4	$4.0 \cdot 10^{-11}$	8483	86.2

Table 6.4: Simulation results for structure (*iii*): rare events (times in seconds).

6.5.2 Linear network

We next consider a linear network that consists of L nodes, where node i has capacity c_i. There are $M = L + 1$ classes of users: each class corresponds to a specific route in the network. Class-i users require service at node i only, $i = 1, \ldots, L$, whereas class-$(L+1)$ users require service at all L nodes simultaneously, see Figure 1.7 for an illustration. For ease of notation relabel class-$(L+1)$ users as class-0 users. In order to obtain the AFS allocation we have to solve the following optimization problem for state $n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}$:

$$\begin{aligned} \text{max} \quad & \textstyle\sum_{i=0}^{L} n_i U_i(x_i) \\ \text{subject to} \quad & n_0 x_0 + n_i x_i \leq c_i, \quad i = 1, \ldots, L \\ \text{over} \quad & x_i \geq 0, \quad i = 0, \ldots, L, \end{aligned} \tag{6.9}$$

where $U_i(x_i)$ is as defined in (1.3). Only in case $c_i = c$, $i = 1, \ldots, L$, i.e., if all nodes have the same capacity, there exist explicit expressions for the optimizing x_i^*s. In that case the optimizers are such that [23]

$$s_0(n) = n_0 x_0^* = \frac{(\kappa_0 n_0^\alpha)^{1/\alpha} c}{(\kappa_0 n_0^\alpha)^{1/\alpha} + (\sum_{j=1}^{L} \kappa_j n_j^\alpha)^{1/\alpha}};$$

$$s_i(n) = n_i x_i^* = (c - s_0(n)) 1_{n_i > 0}, \quad i = 1, \ldots, L.$$

T	a_1	a_2	P_{IS}	$\#_{\text{IS}}$	τ_{IS}	P_{MC}	$\#_{\text{MC}}$	τ_{MC}
1	6	6	$6.1 \cdot 10^{-2}$	2308	126.9	$6.5 \cdot 10^{-2}$	5527	8.8
1	8	8	$7.8 \cdot 10^{-3}$	2102	140.3	$7.7 \cdot 10^{-3}$	49912	78.5
2	8	8	$4.6 \cdot 10^{-2}$	5045	145.2	$4.6 \cdot 10^{-2}$	7908	23.4
2	10	10	$8.5 \cdot 10^{-3}$	4573	172.5	$8.9 \cdot 10^{-3}$	40859	121.3
3	10	10	$2.8 \cdot 10^{-2}$	9722	274.2	$3.0 \cdot 10^{-2}$	12010	52.6
3	12	12	$6.7 \cdot 10^{-3}$	19131	306.3	$6.8 \cdot 10^{-3}$	53159	232.3

Table 6.5: Simulation results for the linear network: comparison with MC simulation (times in seconds).

T	a_1	a_2	P_{IS}	$\#_{\text{IS}}$	τ_{IS}
1	15	15	$1.6 \cdot 10^{-7}$	6481	191.3
1	20	20	$7.0 \cdot 10^{-12}$	10782	199.7
2	20	20	$3.7 \cdot 10^{-8}$	10994	175.1
2	25	25	$1.1 \cdot 10^{-11}$	19255	272.7
3	25	25	$3.0 \cdot 10^{-9}$	19326	312.0
3	30	30	$2.0 \cdot 10^{-12}$	48310	631.5

Table 6.6: Simulation results for the linear network: rare events (times in seconds).

Therefore, we find that $d_0(n)$ equals

$$\min\left\{\frac{(\kappa_0 n_0^\alpha)^{1/\alpha} c}{(\kappa_0 n_0^\alpha)^{1/\alpha} + (\sum_{j=1}^L \kappa_j n_j^\alpha)^{1/\alpha}}, n_0 r_0\right\},$$

and $d_i(n)$, for $i = 1, \ldots, L$, equals

$$\min\left\{\frac{(\sum_{j=1}^L \kappa_j n_j^\alpha)^{1/\alpha} c}{(\kappa_0 n_0^\alpha)^{1/\alpha} + (\sum_{j=1}^L \kappa_j n_j^\alpha)^{1/\alpha}}, n_i r_i\right\}.$$

The steady-state distribution of $N(t)$ is only known in explicit form if $\alpha = 1$, $\kappa_i = \kappa$, $c_j = c$, and $r_i \geq c$, $i = 0, \ldots, L$, $j = 1, \ldots, L$, given that the stability condition $\max_{1 \leq i \leq L} \rho_0 + \rho_i < c$ holds, see Theorem 7.2.1.

We test the performance of our IS scheme in case $L = 2$, $\lambda_0 = 2$, $\lambda_1 = 1$, $\lambda_2 = 1.75$, $\mu_0 = 5$, $\mu_1 = 2$, $\mu_2 = 4$, $r_0 = 0.8$, $r_1 = 0.6$, $r_2 = 0.3$, $\kappa_0 = 0.5$, $\kappa_1 = 2$, $\kappa_2 = 1$, $\alpha = 1$, and starting state $(1, 1, 2)$. Furthermore, we assume $c_i = c = 1$, $i = 1, 2$, so that we have a closed-form expression for $d_i(n)$, $i = 0, 1, 2$, and we let T and A vary. We assume that the structure of A is of the form $\{f | f_0(T) + f_1(T) > a_1, f_0(T) + f_2(T) > a_2\}$, with $a_1, a_2 > 0$. The results are given in Tables 6.5 and 6.6.

The results again show that the rare event probabilities can be estimated rather efficiently. Compared to the single-node network, it now takes much more time to find the MPP (which in general has a non-linear shape), as one needs to optimize over more entries.

6.6 Discussion

In this chapter we studied the transient behavior of the process $N(t)$. A topic for further research is the derivation of an approximation of $\pi(A)$, where $\pi(\cdot)$ denotes the steady-state distribution of $N(t)$. Using regenerative arguments, one can obtain $\pi(A)$ by dividing the expected time that the process spends in set A during a cycle from n_0 to n_0, by the associated expected cycle time, see e.g. Corollary 1.4 in [12]. One may use specific measures to estimate both numerator and denominator, so-called *measure specific dynamic IS*, see e.g. [67]. Dynamic refers to the fact that per run the IS is turned on until the event of interest occurs and turned off thereafter.

Appendix

Below we present the pseudocode of an IS scheme that can be used to estimate rare event probabilities.

IS Algorithm

```
Compute (or approximate) the minimizing path f*.
Divide T into m subintervals of length Δ_m := T/m.
FOR j = 1 TO R
  Ñ_i(0) ← n_{0,i}, i = 1,...,M.
  Set the likelihood ratio equal to 1:  L_j ← 1.
  FOR k = 1 TO m
    Ñ_i(kΔ_m) ← Ñ_i((k−1)Δ_m), i = 1,...,M.
    Simulate Arrivals of type i as Poisson process of rate λ*_i(kΔ_m).
    Simulate Departures of type i as Poisson process of rate ν*_i(kΔ_m).
    Thus K events are generated, with inter-event times t_1,...,t_K.
    FOR ℓ = 1 TO K
      IF Event(ℓ) = Arrival of type i
      THEN
        Update likelihood:
          L_j ← L_j × exp((λ*_i(kΔ_m) − λ_i)t_ℓ) × (λ_i/λ*_i(kΔ_m)).
        Ñ_i(kΔ_m) ← Ñ_i(kΔ_m) + 1.
      IF Event(ℓ) = Departure of type i AND Ñ_i(kΔ_m) > 0
      THEN
        Update likelihood:
          L_j ← L_j × exp((ν*_i(kΔ_m) − ν_i(Ñ(kΔ_m)))t_ℓ)
                ×(ν_i(Ñ(kΔ_m))/ν*_i(kΔ_m)).
        Ñ_i(kΔ_m) ← Ñ_i(kΔ_m) − 1.
      IF Event(ℓ) = Departure of type i AND Ñ_i(kΔ_m) = 0
      THEN
        Set the likelihood ratio equal to 0:  L_j ← 0.
        Abort current simulation run and proceed with the next run.
```

```
    END
    Set t_K equal to 0 when K = 0.
    FOR i = 1 TO M
      Update likelihood:
        L_j ← L_j × exp((λ*_i(kΔ_m) − λ_i)(Δ_m − t_K))
              × exp((ν*_i(kΔ_m) − ν_i(Ñ(kΔ_m)))(Δ_m − t_K)).
    END
  END
  Put 1_j ← 1 if N(mΔ_m) ∈ A, and 0 else.
END
Estimator P_IS ← R^{-1} · Σ_{j=1}^{R} 1_j L_j.
```

Justification of the IS algorithm: We simulate the process $\tilde{N}(t) = (\tilde{N}_1(t), \ldots, \tilde{N}_M(t))$ during a time period of T units, given that $\tilde{N}(0) = n_0$, where

$$\tilde{N}_i(t) := \tilde{N}_{i,\mathrm{up}}(t) - \tilde{N}_{i,\mathrm{down}}(t), \quad i = 1, \ldots, M,$$

with $\tilde{N}_{i,\mathrm{up}}(t)$ being a Poisson process of rate $\lambda_i^*(k\Delta_m)$ and $\tilde{N}_{i,\mathrm{down}}(t)$ being a Poisson process of rate $\nu_i^*(k\Delta_m)$ if $t \in [(k-1)\Delta_m, k\Delta_m)$, $k = 1, \ldots, m$. Clearly, this corresponds to the process described in Section 6.1.1, but with different input distributions and with a different state space, as the state space of $\tilde{N}(t)$ is $\mathbb{Z}^M$, whereas that of $N(t)$ is $\mathbb{N}_0^M \subset \mathbb{Z}^M$. It follows from Section 6.1.2 that we can obtain an unbiased IS estimator of P by simulating $\tilde{N}(t)$ and by keeping track of the likelihood ratio in each run.

In the algorithm we use that the interarrival times are exponentially distributed with mean $1/\lambda_i^*(k\Delta_m)$ $(1/\lambda_i)$ under the new (old) measure if $t \in [(k-1)\Delta_m, k\Delta_m)$. Also, we exploit that the service requirements are exponentially distributed with mean $1/\nu_i^*(k\Delta_m)$ $(1/\nu_i(\tilde{N}(t))$, with $\tilde{N}_i(t) > 0)$ under the new (old) measure, if $t \in [(k-1)\Delta_m, k\Delta_m)$. Clearly, if $\tilde{N}_i(t) = 0$ and a departure of class i occurs, then we reach a state that is infeasible in our model (that is, under the original probability measure), so that we set L equal to zero when this occurs. Since the likelihood ratio will stay zero once it has reached zero, one can abort the current simulation run. By simulating R independent runs, adding all the likelihood ratios at time $m\Delta_m = T$ of the runs that are such that $\tilde{N}(T) \in A$, and dividing this sum by R, we obtain an unbiased estimator of P.

Remark: The obvious advantage of the above algorithm is that the change-of-measure has to be computed just once, and can be applied in all runs. The drawback is that there is no control *within* the run: if the process happens to deviate from the minimizing path, it is not directed back towards this path. These considerations may lead to the following approach. Denote by $f^*(\cdot|n_0, A, T)$ the minimizing path corresponding to the probability P. Define

$$g(s|t) := f^*(s|\tilde{N}(t), A, T - t),$$

i.e., suppose that we find ourself in state $\tilde{N}(t)$ at time t, and we wish to reach the set A at time T, then $g(s|t)$ defines the most likely position at time $s+t$. Note that this implies that $g(s|0) = f^*(s)$. This suggests that one should use the rates

$$\tilde{\lambda}_i(t) := \frac{1}{2}g_i'(0|t) + \frac{1}{2}\sqrt{(g_i'(0|t))^2 + 4\lambda_i\nu_i(\tilde{N}(t))};$$

$$\tilde{\nu}_i(t) := -\frac{1}{2}g_i'(0|t) + \frac{1}{2}\sqrt{(g_i'(0|t))^2 + 4\lambda_i\nu_i(\tilde{N}(t))},$$

$i = 1, \ldots, M$. It can be checked that also for these rates the expected position at time t is $f^*(t)$, but the difference with the first algorithm is that the process evolution is better controlled, cf. [21, 55, 57]. In practice the interval $[0, T]$ is again split into m subintervals, and the rates $\tilde{\lambda}_i(k\Delta_m)$ and $\tilde{\nu}_i(k\Delta_m)$ are used in the k-th interval. Unfortunately, this approach is very time-consuming, as it requires the calculation of a minimizing path in each of the m subintervals.

Chapter 7

Flow-level performance of linear networks

Recall that a wide family of AFS policies achieve stability under the simple condition that no individual link is overloaded, given Poisson arrivals and exponentially distributed service requirements [23]. These stability results imply that flow-level performance measures such as expected transfer delays are finite provided that no individual link is overloaded. However, the derivation of the exact transfer delays and actual user throughputs has proven largely elusive, except in the special case of an unweighted proportional fair bandwidth-sharing policy in certain topologies, such as linear networks. In particular, it is not well understood how the flow-level performance measures depend on the specific choice of the fairness coefficient α and the possible additional weight factors associated with the various classes.

In order to gain further insight in the latter issues, in this chapter we develop approximations for the mean number of users in linear networks operating under AFS policies. The approximations are based on the assumption that one or two of the nodes experience heavy-traffic conditions. In case of just a single 'bottleneck' node, we exploit the fact that this node approximately behaves as a two-class DPS queue. The mean number of users can thus be calculated from the results of [63]. In the case that there are two nodes critically loaded, we rely on the following two observations. First, the heavy-traffic results of [85, 91] show that with equal class weights, the joint workload process is asymptotically independent of the fairness coefficient α. Second, the joint workload process for a proportional fair policy can be exactly computed from the known distribution of the number of users [135]. Combining these two observations, we obtain simple explicit estimates for the workloads at the two bottleneck nodes, which we also numerically validate. We then develop various approximation methods by using the latter estimates in conjunction with characterizations of invariant states from [85, 91] that relate the number of users of the various classes to the workloads at the various nodes.

The remainder of this chapter is organized as follows. In Section 7.1 we provide a detailed model description and discuss some preliminaries. In Section 7.2 we present some results for the known distribution of the user population for a proportional fair policy, and use these to obtain the Laplace-Stieltjes Transform (LST) of the joint workload process at the various nodes. Section 7.3 reviews the heavy-traffic results of [85, 91], which provide the basis for the approximations that we develop subsequently. In Section 7.4 we focus on the case of a single bottleneck node, and exploit the fact that this node approximately behaves as a two-class DPS model to obtain approximations for the mean number of users. Next, in Section 7.5 we turn the attention to a scenario with two bottleneck nodes, and invoke the principle that the joint workload process can be approximated by the known behavior for a proportional fair policy, provided all classes have equal weights. In conjunction with a few equivalent characterizations of invariant states from [85, 91], the latter principle is then leveraged in Section 7.6 to devise various approximation methods. In Sections 7.7 and 7.8 we discuss various model extensions.

7.1 Queueing model

We consider a linear network as depicted in Figure 1.7. The network consists of L nodes, each with unit service rate. There are $L+1$ classes of users: each class corresponds to a specific route in the network. Class-i users require service at node i only, $i = 1, \ldots, L$, whereas class-0 users require service at all L nodes simultaneously.

We assume that class-i users arrive according to a Poisson process of rate λ_i, and have exponentially distributed service requirements with mean μ_i^{-1}, $i = 0, \ldots, L$. The arrival processes are all independent. The traffic load of class i is then $\rho_i = \lambda_i \mu_i^{-1}$. Note that the traffic load at node i is given by $\rho_0 + \rho_i$, $i = 1, \ldots, L$. Let $n = (n_0, n_1, \ldots, n_L)$ be the state of the network, with n_i representing the number of class-i users.

The network operates under an AFS policy. When the network is in state $n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}$, the service rate x_i^* allocated to each of the class-i users is obtained by solving the optimization problem (6.9). Let $s_i(n) := n_i x_i^*$ denote the total service rate allocated to class i. In [23] it was shown that, for $i = 1, \ldots, L$,

$$s_0(n) = \frac{(\kappa_0 n_0^\alpha)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + (\sum_{j=1}^L \kappa_j n_j^\alpha)^{1/\alpha}}; \quad s_i(n) = (1 - s_0(n)) 1_{n_i > 0}, \tag{7.1}$$

if $n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}$, where $1_{n_i>0} = 1$ if $n_i > 0$, and 0 otherwise.

Let $N(t)$ denote the state of the network at time t. Then $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i s_i(n), \quad i = 0, \ldots, L,$$

where e_i denotes the $(i+1)$th unit vector in $\mathbb{R}^{L+1}$. Evidently, $\rho_0+\rho_i < 1$, $i = 1, \ldots, L$, is a necessary condition for the process $N(t)$ to be ergodic. In [23] it was shown that this condition is in fact also sufficient for every $\alpha \in (0, \infty)$.

In general there are no closed-form expressions available for the steady-state distribution of $N(t)$. However, for the case $\alpha = 1$ and $\kappa_i = \kappa$ an explicit expression has been derived in [135], as will be presented in the next section.

7.2 Unweighted proportional fairness

In this section we consider the case $\alpha = 1$ and $\kappa_i = \kappa$, $i = 0, \ldots, L$. The following theorem appeared in slightly different form in [135].

Theorem 7.2.1 *Under the stability condition* $\max_{1\le i\le L} \rho_0+\rho_i < 1$*, the process* $N(t)$ *is reversible, with steady-state distribution given by*

$$\pi(n) = C^{-1} \begin{pmatrix} \sum_{i=0}^{L} n_i \\ n_0 \end{pmatrix} \prod_{i=0}^{L} \rho_i^{n_i}, \tag{7.2}$$

where the normalization constant C *equals*

$$C = \frac{(1-\rho_0)^{L-1}}{\prod_{i=1}^{L}(1-\rho_0-\rho_i)}. \tag{7.3}$$

The mean number of class-0 users in steady state is given by

$$\mathbb{E}(N_0) = \frac{\rho_0}{1-\rho_0}\left(1 + \sum_{i=1}^{L} \frac{\rho_i}{1-\rho_0-\rho_i}\right)$$

and for $i = 1, \ldots, L$,

$$\mathbb{E}(N_i) = \frac{\rho_i}{1-\rho_0-\rho_i}.$$

Let $W_i(t)$ denote the workload, i.e., the unfinished amount of work at node i at time t, $i = 1, \ldots, L$. Thus $W_i(t)$ consists of the remaining service requirements of all class-0 and class-i users at time t. Theorem 7.2.1 enables us to derive the LST of the joint distribution of $W(t) = (W_1(t), \ldots, W_L(t))$ in steady state.

Theorem 7.2.2 *Under the stability condition* $\max_{1\le i\le L} \rho_0+\rho_i < 1$*, the LST of* $W(t)$ *in steady state is given by*

$$\tilde{W}(z) \equiv \tilde{W}(z_1, \ldots, z_L) = \left(\frac{1 - \frac{\lambda_0}{\mu_0+\sum_{j=1}^{L} z_j}}{1-\rho_0}\right)^{L-1} \prod_{i=1}^{L} \frac{1-\rho_0-\rho_i}{1 - \frac{\lambda_0}{\mu_0+\sum_{j=1}^{L} z_j} - \frac{\lambda_i}{\mu_i+z_i}}. \tag{7.4}$$

Proof: Due to the memoryless property of the exponential distribution, the residual service requirement of a class-i user is also exponentially distributed with mean μ_i^{-1}, $i = 0, \ldots, L$. Therefore $W_i(t)$ is distributed as $\sum_{j=1}^{N_0(t)} B_{0,j} + \sum_{j=1}^{N_i(t)} B_{i,j}$, where $B_{i,j}$ are i.i.d. copies of an exponentially distributed variable with mean μ_i^{-1}, $i = 1, \ldots, L$. Now

$$\tilde{W}(z) = \mathbb{E}\left(e^{-\sum_{i=1}^{L} z_i W_i}\right) = \mathbb{E}\left(e^{-\sum_{i=1}^{L} z_i \sum_{j=1}^{N_0} B_{0,j} - \sum_{i=1}^{L}\left(z_i \sum_{j=1}^{N_i} B_{i,j}\right)}\right).$$

Conditioning on the values of N_i, $i = 0, \ldots, L$, we obtain that $\tilde{W}(z)$ equals

$$\sum_{n_0=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \pi(n) \mathbb{E}\left(e^{-\sum_{i=1}^{L} z_i \sum_{j=1}^{n_0} B_{0,j} - \sum_{i=1}^{L}\left(z_i \sum_{j=1}^{n_i} B_{i,j}\right)}\right)$$

$$= \sum_{n_0=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \pi(n) \left(\frac{\mu_0}{\mu_0 + \sum_{i=1}^{L} z_i}\right)^{n_0} \prod_{i=1}^{L} \left(\frac{\mu_i}{\mu_i + z_i}\right)^{n_i}.$$

Substituting (7.2) and invoking that $\rho_i = \lambda_i \mu_i^{-1}$, we obtain that $\tilde{W}(z)$ is equivalent to

$$C^{-1} \prod_{i=1}^{L} \sum_{n_i=0}^{\infty} \left(\frac{\rho_i \mu_i}{\mu_i + z_i}\right)^{n_i} \sum_{n_0=0}^{\infty} \binom{\sum_{j=0}^{L} n_j}{n_0} \left(\frac{\rho_0 \mu_0}{\mu_0 + \sum_{j=1}^{L} z_j}\right)^{n_0}$$

$$= C^{-1} \prod_{i=1}^{L} \sum_{n_i=0}^{\infty} \left(\frac{\lambda_i}{\mu_i + z_i}\right)^{n_i} \left(1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}\right)^{-1-\sum_{j=1}^{L} n_j}$$

$$= C^{-1} \frac{1}{1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}} \prod_{i=1}^{L} \sum_{n_i=0}^{\infty} \left(\frac{\frac{\lambda_i}{\mu_i + z_i}}{1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}}\right)^{n_i}$$

$$= C^{-1} \frac{1}{1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}} \prod_{i=1}^{L} \frac{1}{1 - \frac{\frac{\lambda_i}{\mu_i + z_i}}{1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}}}$$

$$= \frac{1}{\left(1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}\right)(1 - \rho_0)^{L-1}} \prod_{i=1}^{L} \frac{1 - \rho_0 - \rho_i}{\left(1 - \frac{\frac{\lambda_i}{\mu_i + z_i}}{1 - \frac{\lambda_0}{\mu_0 + \sum_{j=1}^{L} z_j}}\right)}. \tag{7.5}$$

The second equality above follows by applying the negative binomial formula:

$$(1 - x)^{-d} = \sum_{n=0}^{\infty} \binom{d - 1 + n}{n} x^n.$$

The final equality follows by substituting (7.3). Rearranging (7.5) finally gives (7.4), and completes the proof. □

Remark: We now provide some interpretation for the expression for $\tilde{W}(z)$ given in Theorem 7.2.2. Consider an M/H$_2$/1 queue with arrival rate $\tilde{\lambda}_0 + \lambda_i$ and service requirements that are exponentially distributed with mean $1/\tilde{\mu}_0$ $(1/\mu_i)$ with probability $\frac{\tilde{\lambda}_0}{\tilde{\lambda}_0+\lambda_i}$ $\left(\frac{\lambda_i}{\tilde{\lambda}_0+\lambda_i}\right)$, where $\tilde{\lambda}_0 := \lambda_0/L$ and $\tilde{\mu}_0 := \mu_0/L$. The LST of the workload $V_i(t)$ in steady state of this M/H$_2$/1 queue is given by the well-known Pollaczek-Khinchin formula

$$\tilde{V}_i(z_i) = \frac{(1-\rho_0-\rho_i)z_i}{(\tilde{\lambda}_0+\lambda_i)\tilde{B}(z_i) + z_i - (\tilde{\lambda}_0+\lambda_i)},$$

where

$$\tilde{B}(z_i) := \frac{\tilde{\lambda}_0}{\tilde{\lambda}_0+\lambda_i}\frac{\tilde{\mu}_0}{\tilde{\mu}_0+z_i} + \frac{\lambda_i}{\tilde{\lambda}_0+\lambda_i}\frac{\mu_i}{\mu_i+z_i}.$$

Substituting $\tilde{B}(z_i)$ we find

$$\tilde{V}_i(z_i) = \frac{1-\rho_0-\rho_i}{1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z_i}-\frac{\lambda_i}{\mu_i+z_i}}.$$

Let us assume we have L of these M/H$_2$/1 queues, all independent, indexed by i, $i = 1, \ldots, L$. Then the joint LST of the workload $V(t)$ is given by

$$\tilde{V}(z) \equiv \tilde{V}(z_1, \ldots, z_L) = \prod_{i=1}^{L} \tilde{V}_i(z_i) = \prod_{i=1}^{L} \frac{1-\rho_0-\rho_i}{1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z_i}-\frac{\lambda_i}{\mu_i+z_i}}. \tag{7.6}$$

Comparing (7.6) with (7.4) indeed shows some similar terms. Obviously, the two expressions cannot be expected to be identical, because the linear network is different from L independent M/H$_2$/1 queues. Taking $z_i = z$, $i = 1, \ldots, L$, (7.4) can however be rewritten as

$$\tilde{W}(z, \ldots, z) = \left(\frac{1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z}}{1-\rho_0}\right)^{L-1} \prod_{i=1}^{L} \frac{1-\rho_0-\rho_i}{1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z}-\frac{\lambda_i}{\mu_i+z}}.$$

The above provides some interpretation for the LST (7.4). It says that $\tilde{V}(z, \ldots, z) = \tilde{W}(z, \ldots, z)\tilde{U}(z)$, where

$$\tilde{U}(z) := \left(\frac{1-\rho_0}{1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z}}\right)^{L-1}$$

is a term that accounts for the dependence and interaction among the L M/H$_2$/1 queues. Note that the LST of the workload $S(t)$ in steady state in an M/M/1 queue with arrival rate $\tilde{\lambda}_0$ and service rate $\tilde{\mu}_0$ is given by $\tilde{S}(z) = (1-\rho_0)/\left(1-\frac{\tilde{\lambda}_0}{\tilde{\mu}_0+z}\right)$.

Hence, $\tilde{U}(z)$ is the LST of the sum of the workloads in $L-1$ of these M/M/1 queues (all independent). The above shows that

$$\sum_{i=1}^{L} W_i + \sum_{i=1}^{L-1} U_i \stackrel{\mathrm{d}}{=} \sum_{i=1}^{L} V_i,$$

where U_i, $i = 1, \ldots, L-1$ are i.i.d. copies of U, and $\stackrel{\mathrm{d}}{=}$ indicates that both sides are equal in distribution.

If $\alpha \neq 1$ or $\kappa_i \neq \kappa$, then there are no explicit expressions available for the steady-state distribution of $N(t)$.

7.3 Fluid and diffusion models

In this section we discuss the heavy-traffic results of [85, 91], which provide the basis for the approximations developed in Sections 7.5 and 7.6. Define the following fluid scaled processes:

$$\overline{N}^k(t) := N(kt)/k \quad \text{and} \quad \overline{W}^k(t) := W(kt)/k,$$

where $W_i(t) = N_0(t)/\mu_0 + N_i(t)/\mu_i$, $i = 1, \ldots, L$. The fluid model can then be obtained from the original model by letting $k \to \infty$. For ease of notation, let $\overline{N}^\infty(t)$ be denoted by $\overline{N}(t)$, and $\overline{W}^\infty(t)$ by $\overline{W}(\overline{N}(t))$. Define

$$s_0(t) := \frac{\left(\kappa_0 \overline{N}_0(t)^\alpha\right)^{1/\alpha}}{\left(\kappa_0 \overline{N}_0(t)^\alpha\right)^{1/\alpha} + (\sum_{l=1}^{L} \kappa_l \overline{N}_l(t)^\alpha)^{1/\alpha}}; \quad s_i(t) := (1 - s_0(t)) 1_i(t),$$

for $i = 1, \ldots, L$, where $1_i(t) = 1$ if $\overline{N}_i(t) > 0$, and 0 otherwise, i.e., $s_i(t)$ denotes the total service rate allocated to class i at time t, $i = 0, \ldots, L$. Then the evolution of the workload process can be described as follows:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\overline{N}_i(t) &= \lambda_i - \mu_i s_i(t), \qquad \text{for} \quad i = 0, \ldots, L; \\ \overline{N}_i(t) &\geq 0, \qquad \text{for} \quad i = 0, \ldots, L. \end{aligned}$$

A fluid model solution is an absolutely continuous function $\overline{N} : [0, \infty) \to \mathbb{R}_+^{L+1}$, such that at each regular point t for $\overline{N}(\cdot)$ (i.e., a value of t at which each component of $\overline{N}(\cdot)$ is differentiable), we have that for $i = 0, \ldots, L$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{N}_i(t) = \begin{cases} \lambda_i - \mu_i s_i(t) & \text{if } \overline{N}_i(t) > 0; \\ 0 & \text{if } \overline{N}_i(t) = 0, \end{cases}$$

and for $i = 1, \ldots, L$,

$$s_0(t) 1_0(t) + \rho_0(1 - 1_0(t)) + s_i(t) 1_i(t) + \rho_i(1 - 1_i(t)) \leq 1.$$

A state $\overline{N}$ is called invariant if there is a fluid model solution such that $\overline{N}(t) = \overline{N}$ for all $t \geq 0$. Let $J := \{j \in \{1, \ldots, L\} : \rho_0 + \rho_j = 1\} \neq \emptyset$ be the set of nodes that are critically loaded.

The following theorem appeared in slightly different form in [91].

Theorem 7.3.1 *The following statements are equivalent:*
(i) $\overline{N}$ *is an invariant state;*
(ii) $s_i(t) = \rho_i$ *for all* i *such that* $\overline{N}_i > 0$*;*
(iii) There is a $q \in \mathbb{R}_+^L$ *such that*

$$\overline{N}_0 = \rho_0 \left(\frac{\sum_{j \in J} q_j}{\kappa_0} \right)^{1/\alpha},$$

for $i \in J$,

$$\overline{N}_i = \rho_i \left(\frac{q_i}{\kappa_i} \right)^{1/\alpha},$$

and for $i \notin J$, $\overline{N}_i = 0$*;*
(iv) $\overline{N} = \triangle(\overline{W}(\overline{N}))$*, where* $\triangle(x)$ *is the unique value of* $\overline{N} \in \mathbb{R}_+^{L+1}$ *that solves the optimization problem:*

$$\begin{array}{rc} \min & F(\overline{N}) = \frac{1}{\alpha+1} \sum_{i=0}^{L} \lambda_i \kappa_i \mu_i^{\alpha-1} \left(\frac{\overline{N}_i}{\lambda_i} \right)^{\alpha+1} \\ \text{subject to} & \overline{N}_0/\mu_0 + \overline{N}_i/\mu_i \geq x_i, \quad i \in J \\ \text{over} & \overline{N}_i \geq 0, \quad i = 0, \ldots, L. \end{array}$$

In the remainder of this section we assume that there are $L = 2$ nodes, and that $\kappa_0 = \kappa_1 = \kappa_2 = \kappa$. Furthermore, we assume heavy-traffic conditions at both nodes, i.e., $J = \{1, 2\}$. Define the diffusion scaled processes:

$$\hat{N}^k(t) := N(k^2 t)/k \text{ and } \hat{W}^k(t) := W(k^2 t)/k,$$

where $W_i(t) = N_0(t)/\mu_0 + N_i(t)/\mu_i$, $i = 1, 2$, as before. In [85] the authors show (under the assumptions mentioned above) that $\hat{W}^k(t)$ converges in distribution to a continuous process $\breve{W}(t)$ as $k \to \infty$. The process $\breve{W}(t)$ is a so-called Semimartingale Reflecting Brownian Motion (SRBM) that lives in the cone

$$\left\{ w : w_i = \frac{\rho_0}{\mu_0} \left(\frac{q_1 + q_2}{\kappa} \right)^{1/\alpha} + \frac{\rho_i}{\mu_i} \left(\frac{q_i}{\kappa} \right)^{1/\alpha},\ q_1, q_2 \geq 0,\ i = 1, 2 \right\}.$$

In [85] it was shown that for all $\alpha \in (0, \infty)$ this is the same as the cone

$$\left\{ (w_1, w_2) : w_1 \geq 0, \quad w_1 \frac{\rho_0/\mu_0}{(1-\rho_0)/\mu_1 + \rho_0/\mu_0} \leq w_2 \leq w_1 \frac{(1-\rho_0)/\mu_2 + \rho_0/\mu_0}{\rho_0/\mu_0} \right\},$$

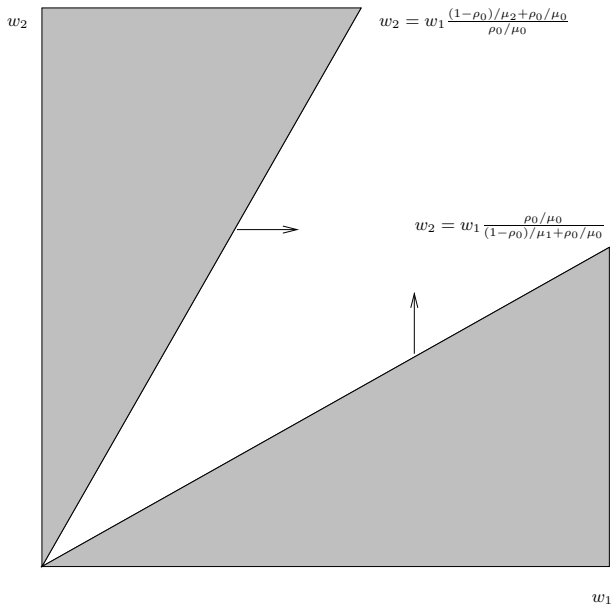

Figure 7.1: The workload cone.

as depicted in Figure 7.1. The state space is an infinite two-dimensional wedge, and the process behaves in the interior of the wedge like a two-dimensional Brownian motion with zero drift and covariance matrix

$$\begin{pmatrix} 2\left(\frac{\rho_0}{\mu_0}+\frac{\rho_1}{\mu_1}\right) & 2\frac{\rho_0}{\mu_0} \\ 2\frac{\rho_0}{\mu_0} & 2\left(\frac{\rho_0}{\mu_0}+\frac{\rho_2}{\mu_2}\right) \end{pmatrix}.$$

The process reflects instantaneously at the boundary of the wedge, the angle of reflection being constant along each side. Vertical (horizontal) reflection on the bounding face $w_2 = w_1 \frac{\rho_0/\mu_0}{(1-\rho_0)/\mu_1+\rho_0/\mu_0}$ $\left(w_2 = w_1 \frac{(1-\rho_0)/\mu_2+\rho_0/\mu_0}{\rho_0/\mu_0}\right)$ can be interpreted as a manifestation of so-called entrainment: congestion at node 1 (node 2) prevents node 2 (node 1) from utilizing the full service rate. In [168] it was shown that the process is transient in the cone, i.e., no steady-state distribution exists.

7.4 Single bottleneck node

In this section we propose a method for approximating $\mathbb{E}N_i$, $i = 0, \ldots, L$, in case of a single bottleneck node, i.e., $|J| = 1$. In case just a single node, say z, $z \in \{1, \ldots, L\}$, is critically loaded, statement (iii) of Theorem 7.3.1 suggests that the number of class-i users, $i = 1, \ldots, L$, $i \neq z$, will be negligible compared to the number of

class-0 and class-z users. Hence, the service rates allocated to the various classes will be predominantly determined by the number of class-0 and class-z users, and approximately equal

$$s_0(n) = \frac{(\kappa_0 n_0^\alpha)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + \left(\sum_{j=1}^L \kappa_j n_j^\alpha\right)^{1/\alpha}} \approx \frac{(\kappa_0 n_0^\alpha)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + (\kappa_z n_z^\alpha)^{1/\alpha}} = \frac{\kappa_0^* n_0}{\kappa_0^* n_0 + \kappa_z^* n_z};$$

$$s_i(n) = \frac{\left(\sum_{j=1}^L \kappa_j n_j^\alpha\right)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + \left(\sum_{j=1}^L \kappa_j n_j^\alpha\right)^{1/\alpha}} \approx \frac{\kappa_z^* n_z}{\kappa_0^* n_0 + \kappa_z^* n_z}, \quad i = 1, \ldots, L,$$

where $\kappa_0^* = \kappa_0^{1/\alpha}$ and $\kappa_z^* = \kappa_z^{1/\alpha}$. Thus, node z roughly behaves as a DPS model with relative weights κ_0^* and κ_z^* for classes 0 and z, respectively. The results of [63] then imply that $\mathbb{E}N_0$ and $\mathbb{E}N_z$ satisfy the set of linear equations

$$\mathbb{E}N_0 - \rho_0 \mathbb{E}N_0 - \kappa_z^* \frac{\lambda_z \mathbb{E}N_0 + \lambda_0 \mathbb{E}N_z}{\kappa_0^* \mu_0 + \kappa_z^* \mu_z} \approx \rho_0;$$

$$\mathbb{E}N_z - \rho_z \mathbb{E}N_z - \kappa_0^* \frac{\lambda_z \mathbb{E}N_0 + \lambda_0 \mathbb{E}N_z}{\kappa_0^* \mu_0 + \kappa_z^* \mu_z} \approx \rho_z,$$

from which we deduce that

$$\mathbb{E}N_0 \approx \frac{\rho_0}{1-\rho_0-\rho_z}\left(1 + \frac{\mu_0 \rho_z(\kappa_z^* - \kappa_0^*)}{\kappa_0^* \mu_0(1-\rho_0) + \kappa_z^* \mu_z(1-\rho_z)}\right);$$

$$\mathbb{E}N_z \approx \frac{\rho_z}{1-\rho_0-\rho_z}\left(1 + \frac{\mu_z \rho_0(\kappa_0^* - \kappa_z^*)}{\kappa_0^* \mu_0(1-\rho_0) + \kappa_z^* \mu_z(1-\rho_z)}\right).$$

Let $\mathbb{E}\tilde{N}_i$ denote the approximation for $\mathbb{E}N_i$, $i = 0, z$. Then

$$\overline{s} := \frac{\kappa_z^* \mathbb{E}\tilde{N}_z}{\kappa_0^* \mathbb{E}\tilde{N}_0 + \kappa_z^* \mathbb{E}\tilde{N}_z}$$

can be regarded as an approximation for the service rate allocated to classes $i = 1, \ldots, L$, $i \neq z$. The number of class-i users, $i = 1, \ldots, L$, $i \neq z$, will approximately behave as in an M/M/1 queue with arrival rate λ_i and service rate $\mu_i \overline{s}$. This gives the approximation

$$\mathbb{E}N_i \approx \frac{\rho_i}{\overline{s} - \rho_i}, \quad i = 1, \ldots, L, \quad i \neq z.$$

Note that the values of κ_i, $i = 1, \ldots, L$, $i \neq z$, do not appear in this approximation. This suggests that the weights of classes that do not traverse the bottleneck node, will tend to have limited impact on the flow-level performance.

We now discuss the numerical experiments that we conducted to examine the accuracy of the above-described method. We first test this approach for a linear

ρ_2	$\mathbb{E}N_0^{exact}$	$\mathbb{E}N_0^{method}$	$\mathbb{E}N_1^{exact}$	$\mathbb{E}N_1^{method}$	$\mathbb{E}N_2^{exact}$	$\mathbb{E}N_2^{method}$
0.1	60.50	60.00	39.00	39.00	0.33	0.34
0.2	61.50	60.00	39.00	39.00	1.00	1.03
0.3	64.50	60.00	39.00	39.00	3.00	3.19

Table 7.1: Results for $\alpha = 1$ and $\kappa_0 = \kappa_1 = \kappa_2 = \kappa$.

ρ_2	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{method}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_1^{method}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_2^{method}$
0.1	1	27.88	28.24	69.12	70.76	0.35	0.35
0.2	1	27.54	28.24	67.25	70.76	1.06	1.08
0.3	1	34.51	28.24	80.19	70.76	3.34	3.52
0.1	2	49.76	43.40	63.75	55.60	0.36	0.34
0.2	2	45.08	43.40	57.55	55.60	1.08	1.05
0.3	2	41.59	43.40	52.34	55.60	3.46	3.32
0.1	5	61.35	53.43	52.03	45.57	0.35	0.34
0.2	5	54.37	53.43	46.40	45.57	1.08	1.04
0.3	5	50.23	53.43	42.52	45.57	3.47	3.24
0.1	∞	60.09	60.00	39.20	39.00	0.36	0.34
0.2	∞	60.68	60.00	39.52	39.00	1.08	1.03
0.3	∞	63.72	60.00	40.99	39.00	3.52	3.19

Table 7.2: Results for Scenario 1.

ρ_2	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{method}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_1^{method}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_2^{method}$
0.1	1	35.72	30.91	89.57	77.78	0.34	0.35
0.2	1	26.21	30.91	63.59	77.78	1.06	1.07
0.3	1	34.84	30.91	81.73	77.78	3.37	3.48
0.1	2	45.85	45.60	58.73	58.34	0.35	0.34
0.2	2	50.17	45.60	63.96	58.34	1.09	1.04
0.3	2	58.57	45.50	74.97	58.34	3.51	3.31
0.1	5	54.47	54.43	46.59	46.43	0.36	0.34
0.2	5	63.51	54.43	53.81	46.43	1.09	1.04
0.3	5	55.19	54.43	46.88	46.43	3.59	3.24
0.1	∞	44.16	60.00	28.72	39.00	0.36	0.34
0.2	∞	78.66	60.00	50.98	39.00	1.12	1.03
0.3	∞	53.80	60.00	34.60	39.00	3.72	3.19

Table 7.3: Results for Scenario 2.

network with $L = 2$ nodes, $\alpha = 1$, and $\kappa_i = \kappa$, $i = 0, 1, 2$, for which we have exact expressions for $\mathbb{E}N_i$, $i = 0, 1, 2$, see Theorem 7.2.1. We fix $\rho_0 = 0.6$ and $\rho_1 = 0.39$, so that node 1 is highly loaded ($z = 1$), and vary the value of ρ_2. Note that in case of equal weights, the approximations only depend on the traffic characteristics through the class loads, and not on the specific values of the λ_is and μ_is. The results are presented in Table 7.1, and indicate that the approximations are remarkably accurate. As could be expected, the smaller the value of ρ_2, the better the approximations.

In case $\alpha \neq 1$ or $\kappa_i \neq \kappa$, there are no exact expressions available for $\mathbb{E}N_i$, $i = 0, 1, 2$, and we need to resort to simulation experiments to investigate the accuracy of the approximations. Throughout this chapter, the simulation numbers are obtained as

averages over 10000 busy periods. We choose the same setting as above, but with $\kappa_0 = 2$, $\kappa_1 = 0.5$ and $\kappa_2 = 1$. In this case the approximations do depend on the specific values of the μ_is. We consider two scenarios: in Scenario 1 we take $\mu_0 = \mu_1 = \mu_2 = 1$, while in Scenario 2 we set $\mu_0 = 0.75$, $\mu_1 = 1$ and $\mu_2 = 1.5$. The results are presented in Tables 7.2 and 7.3.

Note that the approximations for $\mathbb{E}N_0$ and $\mathbb{E}N_1$ do not depend on the presence of class-2 users, and are in particular independent of the value of ρ_2. Further observe that if $\alpha \to \infty$, then $\kappa_0^*, \kappa_1^* \to 1$, and as a consequence $\mathbb{E}N_i \approx \rho_i/(1-\rho_0-\rho_i)$, $i = 0, 1$. The results are surprisingly accurate, even if node 2 is also relatively highly loaded ($\rho_0 + \rho_2 = 0.9$). Note that $\mathbb{E}N_2^{sim}$ is increasing in ρ_2, as could be expected. The influence of ρ_2 on $\mathbb{E}N_0^{sim}$ and $\mathbb{E}N_1^{sim}$ is more subtle, as closer inspection of Tables 7.2 and 7.3 demonstrates. It might be natural to expect that increasing ρ_2 would also have an adverse impact on $\mathbb{E}N_0^{sim}$ and $\mathbb{E}N_1^{sim}$. As the value of ρ_2 and the number of class-2 users increases, however, the service rate $s_0(n)$ will decrease, whereas the service rate $s_2(n)$ will increase. The resulting increase in the number of class-0 users will have the counteracting effect of decreasing $s_2(n)$, and conversely the expected decrease in the number of class-2 users will have the opposite effect of increasing $s_0(n)$. Because of these interacting effects, the net impact basically remains unpredictable, and as Tables 7.2 and 7.3 reveal, $\mathbb{E}N_0^{sim}$ and $\mathbb{E}N_1^{sim}$ do not necessarily change in a monotone manner as the value of ρ_2 increases.

7.5 Two bottleneck nodes and equal weights: workload invariance

In this section we consider the scenario that there are two nodes critically loaded, i.e., $|J| = 2$. Since the nodes can be indexed arbitrarily, we may assume without loss of generality that $J = \{1, 2\}$. Also, suppose that $\kappa_i = \kappa$, $i = 0, \ldots, L$.

Let $W(t)$ be the workload process associated with the two bottleneck nodes. The results from [85, 91] as reviewed in Section 7.3 indicate that the behavior of $W(t)$ is asymptotically independent of the value of α. In particular, this suggests that the behavior of the workload process can be approximated by the known distribution for $\alpha = 1$. In order to examine this hypothesis, we calculated the mean workload (using Theorem 7.2.2)

$$\mathbb{E}W_i^{exact}(\alpha = 1) \equiv \mathbb{E}W_i^{exact}(1) = \frac{\lambda_i/\mu_i^2}{1-\rho_0-\rho_i} + \frac{\lambda_0/\mu_0^2}{1-\rho_0}\left[\left(1 + \sum_{j=1}^{L} \frac{\rho_j}{1-\rho_0-\rho_j}\right)\right., \quad (7.7)$$

with $i = 1, 2$, and compared it with simulation for the case of $L = 2$ nodes, $\rho_0 + \rho_1 = \rho_0 + \rho_2 = 0.99$, and $\mu_i = \kappa_i = 1$, $i = 0, 1, 2$. We also considered the asymmetric case $\rho_0 + \rho_1 = \rho_0 + \rho_2 = 0.99$, $\kappa_i = 1$, $i = 0, 1, 2$, $\mu_0 = 0.75$, $\mu_1 = 1$ and $\mu_2 = 1.5$.

ρ_0	$\rho_1 = \rho_2$	α	X_1	X_2
0.3	0.69	1	0.001	0.019
0.5	0.49	1	0.006	0.020
0.7	0.29	1	-0.015	-0.024
0.3	0.69	2	-0.042	-0.047
0.5	0.49	2	-0.041	-0.056
0.7	0.29	2	-0.039	-0.040
0.3	0.69	5	-0.027	-0.065
0.5	0.49	5	-0.005	0.003
0.7	0.29	5	-0.058	-0.069
0.3	0.69	∞	-0.007	0.011
0.5	0.49	∞	-0.043	-0.063
0.7	0.29	∞	-0.061	-0.055

ρ_0	$\rho_1 = \rho_2$	α	X_1	X_2
0.3	0.69	1	0.006	-0.009
0.5	0.49	1	-0.046	-0.033
0.7	0.29	1	0.048	0.042
0.3	0.69	2	-0.065	-0.077
0.5	0.49	2	-0.025	-0.038
0.7	0.29	2	-0.039	-0.049
0.3	0.69	5	-0.040	-0.036
0.5	0.49	5	-0.055	-0.057
0.7	0.29	5	-0.037	-0.035
0.3	0.69	∞	-0.028	-0.022
0.5	0.49	∞	-0.048	-0.076
0.7	0.29	∞	-0.003	-0.009

Table 7.4: Testing whether $W(t)$ is independent of α. Left (Right): the symmetric (asymmetric) case.

Define

$$X_i := \mathbb{E}W_i^{sim}(\alpha)/\mathbb{E}W_i^{exact}(1) - 1, \quad i = 1, 2.$$

The results, summarized in Table 7.4, indicate that the mean workload for $\alpha = 1$ indeed provides a reasonably accurate approximation for a wide range of α values. Note that X_i should be equal to 0 for all cases with $\alpha = 1$. In most cases with $\alpha > 1$, $\mathbb{E}W_i^{exact}(1)$ is larger than $\mathbb{E}W_i^{sim}(\alpha)$, and thus seems to yield a conservative approximation. Below we provide an explanation for this observation. In preparation for that, we first present the following proposition.

Proposition 7.5.1 *For fixed $n = (n_0, \ldots, n_L)$ and $\kappa_i = \kappa$, $i = 0, \ldots, L$, the service rate $s_0(n)$ allocated to class-0 users is increasing in α.*

Proof: For fixed $n = (n_0, \ldots, n_L)$ and $\kappa_i = \kappa$, $i = 0, \ldots, L$, we obtain from (7.1) that

$$s_0(n) = \frac{n_0}{n_0 + (\sum_{j=1}^{L} n_j^{\alpha})^{1/\alpha}}. \tag{7.8}$$

Equivalently, we have to prove that $(\sum_{j=1}^{L} n_j^{\alpha})^{1/\alpha}$ is decreasing in α. First note that

$$n_1^{\alpha r} + \cdots + n_L^{\alpha r} < (n_1^{\alpha} + \cdots + n_L^{\alpha})^r$$

for all $r > 1$. Therefore,

$$\begin{aligned}\left(\sum_{i=1}^{L} n_i^{\beta}\right)^{1/\beta} = \left(\sum_{i=1}^{L} n_i^{\alpha r}\right)^{1/\alpha r} &= (n_1^{\alpha r} + \cdots + n_L^{\alpha r})^{1/\alpha r} \\ &< (n_1^{\alpha} + \cdots + n_L^{\alpha})^{r/\alpha r} = \left(\sum_{i=1}^{L} n_i^{\alpha}\right)^{1/\alpha},\end{aligned}$$

for all $\beta > \alpha$, which proves the stated. □

Now observe that the workload at each of the nodes is minimized (sample-path-wise in fact) when class 0 receives priority over classes 1 and 2. Since the capacity allocated to class-0 users is increasing in α, it is thus plausible that more generally the mean workload $\mathbb{E}W_i^{exact}(\alpha)$ decreases as function of α, which implies that X_i is smaller than 0 for $\alpha > 1$, $i = 1, 2$. This provides an explanation for the negative values in Table 7.4. Below we show that the latter property can in fact be rigorously proved using Proposition 7.5.1 and stochastic coupling arguments.

Denote by $r_i(t)$ the instantaneous service rate allocated to class i at time t, i.e., $r_i(t) = s_i(N(t))$ if $N_i(t) > 0$, and otherwise $r_i(t) = 0$, $i = 0, \ldots, L$. Denote by $R_i(t) := \int_{u=0}^{t} r_i(u)\mathrm{d}u$ the cumulative amount of service received by class i during the time interval $[0, t]$, $i = 0, \ldots, L$. Denote by $B_{i,n}$ the service requirement of the n-th arriving class-i user, $i = 0, \ldots, L$. Denote by $C_i(s) := \sup\{n : \sum_{m=1}^{n} B_{i,n} < s\}$ the number of class-i service completions as function of the amount of service received by class i, $i = 0, \ldots, L$, assuming a FIFO service discipline. Thus $D_i(t) = C_i(R_i(t))$ represents the number of class-i service completions during the time interval $[0, t]$, $i = 0, \ldots, L$. Denote by $A_i(t)$ the number of class-i users arriving during the time interval $[0, t]$, $i = 0, \ldots, L$. Denote by $Q_i(t) := \sum_{m=1}^{A_i(t)} B_{i,m}$ the amount of class-i work arriving during the time interval $[0, t]$, $i = 0, \ldots, L$. Denote by $V_i(t)$ the amount of class-i work at time t, $i = 0, \ldots, L$.

Since the service requirements are exponentially distributed, the stochastic behavior of the network does not depend on the service discipline within classes, as long as that discipline is not based on any knowledge of the actual realizations of the service requirements. We may therefore assume that the service discipline within classes is FIFO.

Consider the behavior of the network under two AFS policies with parameters β and γ for the same realizations of the arrival processes and service requirements. We attach β and γ as superscripts to the various quantities associated with the two policies.

Proposition 7.5.2 *Suppose that the system is empty at time $t = 0$. If $\beta \leq \gamma$, then $W_i^{\beta}(t) \geq W_i^{\gamma}(t)$ for all $t \geq 0$, $i = 1, \ldots, L$.*

Proof: Below we will prove that if $\beta \leq \gamma$, then (i) $N_0^{\beta}(t) \geq N_0^{\gamma}(t)$, (ii) $R_0^{\beta}(t) \leq R_0^{\gamma}(t)$, and (iii) $R_0^{\beta}(t) + R_i^{\beta}(t) \leq R_0^{\gamma}(t) + R_i^{\gamma}(t)$ for all $t \geq 0$, $i = 1, \ldots, L$. Note that $V_i^j(t) = Q_i^j(t) - R_i^j(t)$, $i = 0, \ldots, L$, $j = \beta, \gamma$, so that $R_0^{\beta}(t) + R_i^{\beta}(t) \leq R_0^{\gamma}(t) + R_i^{\gamma}(t)$ implies that $W_0^{\beta}(t) = V_0^{\beta}(t) + V_i^{\beta}(t) \geq V_0^{\gamma}(t) + V_i^{\gamma}(t) = W_0^{\gamma}(t)$ for all $t \geq 0$, $i = 1, \ldots, L$.

First note that $N_i^j(t) = A_i^j(t) - D_i^j(t)$, with $D_i^j(t) = C_i^j(R_i^j(t))$, $i = 0, \ldots, L$, $j = \beta, \gamma$, i.e., inequality (i) follows from (ii), and it suffices to prove that inequalities

(ii) and (iii) hold. Below we assume that inequality (ii) or (iii) does not hold, and we show that this results in a contradiction. Let $u > 0$ be the first time epoch at which one of the two inequalities is violated. First assume that inequality (ii) is the first one to be violated, i.e., $R_0^\beta(u) = R_0^\gamma(u)$ and $r_0^\beta(u) > r_0^\gamma(u)$ (with strict inequality), but $R_0^\beta(u) + R_i^\beta(u) \leq R_0^\gamma(u) + R_i^\gamma(u)$, $i = 1, \ldots, L$. Clearly, then $N_0^\beta(u) = N_0^\gamma(u)$, and, using Proposition 7.5.1, it follows that $N_j^\beta(u) < N_j^\gamma(u)$ for some $j = 1, \ldots, L$, because otherwise we would have $r_0^\beta(u) \leq r_0^\gamma(u)$. This implies that $R_j^\beta(u) > R_j^\gamma(u)$, and thus $R_0^\beta(u) + R_j^\beta(u) > R_0^\gamma(u) + R_j^\gamma(u)$, which contradicts the initial assumption. Next, assume that inequality (iii) is the first one to be violated, i.e., $R_0^\beta(u)+R_j^\beta(u) = R_0^\gamma(u)+R_j^\gamma(u)$ and $r_0^\beta(u) + r_j^\beta(u) > r_0^\gamma(u) + r_j^\gamma(u)$ for some $j = 1, \ldots, L$, but $R_0^\beta(u) \leq R_0^\gamma(u)$. It follows that $N_j^\gamma(u) = 0$, because otherwise we would have $r_0^\gamma(u) + r_j^\gamma(u) = 1 \geq r_0^\beta(u) + r_j^\beta(u)$. This implies that $R_j^\beta(u) \leq R_j^\gamma(u)$, and thus $R_0^\beta(u) = R_0^\gamma(u)$, $R_j^\beta(u) = R_j^\gamma(u)$ (as $R_0^\beta(u) + R_j^\beta(u) = R_0^\gamma(u) + R_j^\gamma(u)$ and $R_0^\beta(u) \leq R_0^\gamma(u)$ by assumption), and $R_i^\beta(u) \leq R_i^\gamma(u)$ for all $i = 1, \ldots, L$, $i \neq j$, as well. Consequently, $N_0^\beta(u) = N_0^\gamma(u)$, $N_j^\beta(u) = N_j^\gamma(u) = 0$ and $N_i^\beta(u) \geq N_i^\gamma(u)$ for all $i = 1, \ldots, L$, $i \neq j$. This means that $r_0^\beta(u) \leq r_0^\gamma(u)$, and thus, since $r_j^\beta(u) = r_j^\gamma(u) = 0$, $r_0^\beta(u) + r_j^\beta(u) \leq r_0^\beta(u) = r_0^\gamma(u) + r_j^\gamma(u)$, which contradicts the initial assumption. Hence, we have proven that if $\beta \leq \gamma$, then inequalities (i), (ii) and (iii) hold, and therefore this proves the stated. □

7.6 Two bottleneck nodes and equal weights: approximations

In this section we develop three methods for approximating $\mathbb{E}N_i$, $i = 0, 1, 2$. Recall that we suppose that $J = \{1, 2\}$ and $\kappa_i = \kappa$, $i = 0, \ldots, L$. The various methods differ in some technical details, but they all rely on the insights from the heavy-traffic results as reviewed in Section 7.3. In Section 7.6.4 we present approximations for $\mathbb{E}N_i$, $i = 3, \ldots, L$.

7.6.1 Method 1

The numerical results presented in the previous section indicate that $\mathbb{E}W_i^{exact}(\alpha)$ is nearly constant in $\alpha \in (0, \infty)$, provided that the load at nodes 1 and 2 is sufficiently high. In particular, it is approximately equal to the known value for $\alpha = 1$ as given by (7.7). Further observe that $\mathbb{E}W_i^{exact}(\alpha) = \mathbb{E}N_0/\mu_0 + \mathbb{E}N_i/\mu_i$, $i = 1, \ldots, L$. Thus, we obtain

$$\mathbb{E}N_0/\mu_0 + \mathbb{E}N_i/\mu_i \approx \frac{\lambda_i/\mu_i^2}{1-\rho_0-\rho_i} + \frac{\lambda_0/\mu_0^2}{1-\rho_0}\left(1 + \sum_{j=1}^{L} \frac{\rho_j}{1-\rho_0-\rho_j}\right), \quad i = 1, 2, \tag{7.9}$$

i.e., a set of two approximately linear equations with three unknowns. If we can find one additional constraint, then we should be able to determine $\mathbb{E}N_i$, $i = 0, 1, 2$ (as

long as the resulting system of equations is non-singular).

Now observe that Theorem 7.3.1 shows that an invariant state $\overline{N}$ in the fluid model can be expressed as

$$\overline{N}_0 = \rho_0 \left(\frac{q_1 + q_2}{\kappa_0} \right)^{1/\alpha}; \quad \overline{N}_i = \rho_i \left(\frac{q_i}{\kappa_i} \right)^{1/\alpha}, \quad i = 1, 2, \quad q \in \mathbb{R}_+^2 .$$

This suggests the following approximation for $(\mathbb{E}N_0, \mathbb{E}N_1, \mathbb{E}N_2)$:

$$\int_{q_1=0}^{\infty} \int_{q_2=0}^{\infty} \left(\rho_0 \left(\frac{q_1 + q_2}{\kappa} \right)^{1/\alpha}, \rho_1 \left(\frac{q_1}{\kappa} \right)^{1/\alpha}, \rho_2 \left(\frac{q_2}{\kappa} \right)^{1/\alpha} \right) \mathrm{d}\mathbb{P}\left(Q_1 < q_1, Q_2 < q_2\right),$$

which is equivalent to

$$\frac{1}{\kappa^{1/\alpha}} \left(\rho_0 \mathbb{E}\left((Q_1 + Q_2)^{1/\alpha} \right), \rho_1 \mathbb{E}\left(Q_1^{1/\alpha} \right), \rho_2 \mathbb{E}\left(Q_2^{1/\alpha} \right) \right).$$

Using the additional approximation

$$(\mathbb{E}N_0, \mathbb{E}N_1, \mathbb{E}N_2) \approx \frac{\gamma}{\kappa^{1/\alpha}} \left(\rho_0 \left(\mathbb{E}Q_1 + \mathbb{E}Q_2 \right)^{1/\alpha}, \rho_1 \left(\mathbb{E}Q_1 \right)^{1/\alpha}, \rho_2 \left(\mathbb{E}Q_2 \right)^{1/\alpha} \right), \quad (7.10)$$

with γ some multiplicative constant, and substituting (7.10) in (7.9) then yields a system of two equations with two unknowns. Numerically solving this system yields $\gamma^\alpha \mathbb{E}Q_i$, $i = 1, 2$, from which we can obtain $\mathbb{E}N_i$, $i = 0, 1, 2$, using (7.10). Note that

$$\mathbb{E}\left((Q_1 + Q_2)^{1/\alpha} \right) \leq \mathbb{E}\left(Q_1^{1/\alpha} \right) + \mathbb{E}\left(Q_2^{1/\alpha} \right)$$

if $\alpha \in (1, \infty)$, which would provide an upper bound for $\mathbb{E}N_0$ relative to $\mathbb{E}N_i$, $i = 1, 2$. Likewise, if $\alpha \in (0, 1)$, then this would give a lower bound for $\mathbb{E}N_0$ relative to $\mathbb{E}N_i$, $i = 1, 2$.

We tested this approach by comparing the results with simulation figures. We took the same simulation parameters as in the previous section. The results are presented in Tables 7.5 and 7.6. Throughout this chapter, $\mathbb{E}N_i^{Mj}$ denotes the approximation of $\mathbb{E}N_i$ that is obtained by using Method j. Note that in Table 7.5 we have $\mathbb{E}N_1^{M1} = \mathbb{E}N_2^{M1}$ by symmetry. The tables indicate that Method 1 gives reasonably accurate estimates for $\mathbb{E}N_i$, particularly $\mathbb{E}N_0$. Note that Method 1 is fast as well: it suffices to solve a system of two equations with two unknowns.

7.6.2 Method 2

We now discuss a second method for approximating $\mathbb{E}N_i$, $i = 0, 1, 2$. Again, we start from Equation (7.9) as in Method 1. The difference with Method 1 is that we now use statement (iv) (instead of (iii)) of Theorem 7.3.1. Statement (iv) implies that a

ρ_0	$\rho_1 = \rho_2$	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{M1}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_1^{M1}$=$\mathbb{E}N_2^{M1}$
0.3	0.69	1	60.20	59.80	68.58	70.81	68.77
0.5	0.49	1	100.27	99.33	48.61	50.66	48.67
0.7	0.29	1	135.01	138.07	29.19	27.67	28.60
0.3	0.69	2	50.21	48.95	72.98	72.36	79.62
0.5	0.49	2	86.98	87.42	54.90	52.79	60.58
0.7	0.29	2	126.62	128.91	33.59	33.38	37.76
0.3	0.69	5	47.86	42.83	77.25	72.40	85.75
0.5	0.49	5	88.24	79.86	58.97	60.25	68.14
0.7	0.29	5	120.76	122.49	36.22	34.38	44.18
0.3	0.69	∞	49.75	39.02	77.94	80.25	89.55
0.5	0.49	∞	82.62	74.83	59.02	56.06	73.17
0.7	0.29	∞	120.81	117.92	35.64	36.67	48.75

Table 7.5: Results for Method 1: the symmetric case.

ρ_0	$\rho_1 = \rho_2$	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{M1}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_1^{M1}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_2^{M1}$
0.3	0.69	1	59.40	59.75	70.06	68.77	67.58	68.65
0.5	0.49	1	95.19	99.22	45.80	48.70	48.46	48.55
0.7	0.29	1	143.78	137.93	31.01	28.66	29.58	28.48
0.3	0.69	2	50.17	51.12	71.83	80.27	73.27	85.91
0.5	0.49	2	90.58	90.72	55.61	60.04	56.49	65.56
0.7	0.29	2	127.79	131.91	33.86	36.68	33.78	40.51
0.3	0.69	5	49.89	45.99	75.98	87.11	81.50	96.17
0.5	0.49	5	85.94	85.16	56.51	67.46	61.03	76.68
0.7	0.29	5	127.15	127.71	35.20	42.27	39.39	48.90
0.3	0.69	∞	50.20	43.75	77.31	90.08	83.68	100.64
0.5	0.49	∞	84.83	82.88	59.16	70.49	58.67	81.23
0.7	0.29	∞	130.52	126.06	37.72	44.48	40.41	52.22

Table 7.6: Results for Method 1: the asymmetric case.

workload vector $w = (w_1, w_2)$ uniquely determines a state vector n that solves the optimization problem:

$$\begin{aligned} \min \quad & F(n_0, n_1, \ldots, n_L) = \tfrac{1}{\alpha+1} \textstyle\sum_{i=0}^{L} \lambda_i \kappa_i \mu_i^{\alpha-1} \left(\tfrac{n_i}{\lambda_i}\right)^{\alpha+1} \\ \text{subject to} \quad & n_0/\mu_0 + n_i/\mu_i \geq w_i, \quad i = 1, 2 \\ \text{over} \quad & n_i \geq 0, \quad i = 0, \ldots, L. \end{aligned} \tag{7.11}$$

The method now works as follows. We determine the vector $(\mathbb{E}N_0, \mathbb{E}N_1, \ldots, \mathbb{E}N_L)$ that minimizes the function $F(\mathbb{E}N_0, \mathbb{E}N_1, \ldots, \mathbb{E}N_L)$ subject to the constraints in (7.9). Note that $\mathbb{E}N_i = 0$, $i = 3, \ldots, L$.

As it turns out, Methods 1 and 2 result in similar approximations for $\mathbb{E}N_i$, $i = 0, 1, 2$. This is not too surprising: the only difference between the methods is that we use statement (*iii*) in one case, and (*iv*) in the other. However, statements (*iii*) and (*iv*) are in fact equivalent in case of heavy traffic, so both methods should roughly agree when the load is sufficiently high.

Remark: Method 2 uses the *mean* workloads to approximate the mean number of

users. However, we can potentially improve the accuracy of the approximation if we use the *distribution* of the workloads, which is also asymptotically independent of α in heavy traffic. The resulting approximation is then given by

$$\mathbb{E}N_i = \sum_{n\geq 0} \triangle_i(w(n))\pi(n), \quad i = 0, 1, 2,$$

where $w_i(n) = n_0/\mu_0 + n_i/\mu_i$, $i = 1, 2$, $\triangle(x)$ is as in Theorem 7.3.1, and $\pi(n)$ is given by (7.2). This will typically result in a different approximation for $\mathbb{E}N_i$ than Method 2, since the optimization problem (7.11) is non-linear. The disadvantage is that it is very time-consuming.

7.6.3 Method 3

This method is similar to both previous methods, i.e., we again start from Equation (7.9) to obtain a set of two equations with three unknowns $\mathbb{E}N_i$, $i = 0, 1, 2$. Statement (ii) of Theorem 7.3.1 provides an additional equation, which allows us to numerically solve the above system of equations. First note from (7.8) that

$$\sum_{n\geq 0} \frac{n_0}{n_0 + (\sum_{l=1}^{L} n_l^{\alpha})^{1/\alpha}} \tilde{\pi}(n) = \sum_{n\geq 0} s_0(n)\tilde{\pi}(n) = \rho_0,$$

where $\tilde{\pi}(n)$ is the steady-state distribution of $N(t)$ in case $\alpha \in (0, \infty)/\{1\}$. The additional equation is then obtained by replacing the latter equation by the approximation

$$\frac{\mathbb{E}N_0}{\mathbb{E}N_0 + (\sum_{l=1}^{L} \mathbb{E}N_l^{\alpha})^{1/\alpha}} \approx \frac{\mathbb{E}N_0}{\mathbb{E}N_0 + (\mathbb{E}N_1^{\alpha} + \mathbb{E}N_2^{\alpha})^{1/\alpha}} = \rho_0.$$

We numerically solved the above system of equations for both the symmetric and asymmetric scenarios considered in the previous section. The results are presented in Tables 7.7 and 7.8. Note that the approximations obtained from Method 3 slightly differ from those of Methods 1 and 2. This may be explained from the fact that statement (ii) of Theorem 7.3.1 (for $i = 0$) is only partly satisfied.

7.6.4 Approximation for non-bottleneck nodes

In the previous subsections we presented three methods for approximating the mean number of users at the bottleneck nodes. We now provide an approximation for the number of users at the remaining nodes, i.e., $\mathbb{E}N_i$, $i = 3, \ldots, L$. The method is similar in nature as the one presented in Section 7.4 for the case of a single bottleneck node. Let $\mathbb{E}\tilde{N}_i$ denote the approximations obtained for $\mathbb{E}N_i$, $i = 0, 1, 2$. In view of (7.8), define

$$\overline{s}_0 := \frac{\mathbb{E}\tilde{N}_0}{\mathbb{E}\tilde{N}_0 + \left(\mathbb{E}\tilde{N}_1^{\alpha} + \mathbb{E}\tilde{N}_2^{\alpha}\right)^{1/\alpha}}$$

ρ_0	$\rho_1=\rho_2$	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{M3}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_1^{M3}$=$\mathbb{E}N_2^{M3}$
0.3	0.69	1	60.20	59.34	68.58	70.81	69.23
0.5	0.49	1	100.27	98.67	48.61	50.66	49.33
0.7	0.29	1	135.01	137.26	29.19	27.67	29.41
0.3	0.69	2	50.21	48.52	72.98	72.36	80.05
0.5	0.49	2	86.98	86.70	54.90	52.79	61.30
0.7	0.29	2	126.62	127.91	33.59	33.38	38.76
0.3	0.69	5	47.86	42.41	77.25	72.40	86.16
0.5	0.49	5	88.24	79.12	58.97	60.25	68.88
0.7	0.29	5	120.76	121.38	36.22	34.38	45.29
0.3	0.69	∞	49.75	38.63	77.94	80.25	89.94
0.5	0.49	∞	82.62	74.08	59.02	56.06	73.92
0.7	0.29	∞	120.81	116.74	35.64	36.67	49.93

Table 7.7: Results for Method 3: the symmetric case.

ρ_0	$\rho_1=\rho_2$	α	$\mathbb{E}N_0^{sim}$	$\mathbb{E}N_0^{M3}$	$\mathbb{E}N_1^{sim}$	$\mathbb{E}N_1^{M3}$	$\mathbb{E}N_2^{sim}$	$\mathbb{E}N_2^{M3}$
0.3	0.69	1	59.40	59.40	70.06	69.24	67.58	69.35
0.5	0.49	1	95.19	98.77	45.80	49.31	48.46	49.46
0.7	0.29	1	143.78	137.40	31.01	29.35	29.58	29.53
0.3	0.69	2	50.17	50.76	71.83	80.75	73.27	86.63
0.5	0.49	2	90.58	90.18	55.61	60.76	56.49	66.64
0.7	0.29	2	127.79	131.24	33.86	37.57	33.78	41.86
0.3	0.69	5	49.89	45.63	75.98	87.58	81.50	96.88
0.5	0.49	5	85.94	84.59	56.51	68.21	61.03	77.82
0.7	0.29	5	127.15	126.97	35.20	43.27	39.39	50.40
0.3	0.69	∞	50.20	43.42	77.31	90.54	83.68	101.31
0.5	0.49	∞	84.83	82.33	59.16	71.22	58.67	82.33
0.7	0.29	∞	130.52	125.31	37.72	45.37	40.41	53.71

Table 7.8: Results for Method 3: the asymmetric case.

as an approximation for the service rate allocated to class 0. As before, the number of class-i users, $i=3,\ldots,L$, will roughly behave as in an M/M/1 queue with arrival rate λ_i and service rate $\mu_i(1-\overline{s}_0)$. This gives the approximation

$$\mathbb{E}N_i \approx \frac{\rho_i}{1-\overline{s}_0-\rho_i}, \quad i=3,\ldots,L.$$

Remark: For the linear network (see Figure 1.7) it was shown in [27] that BFS is equivalent to unweighted proportional fairness, i.e., $\alpha=1$ and $\kappa_i=\kappa$, $i=0,\ldots,L$. Note that the steady-state distribution in Theorem 7.2.1 indeed only depends on the loads, and not on any higher-order traffic characteristics. In [24] it was shown that BFS provides a good approximation for unweighted proportional fairness and unweighted max-min fairness. The results of this section, though, illustrate that the accuracy of the BFS approximation for unweighted max-min fairness degrades in heavy-traffic conditions.

7.7 Unequal service rates

In the previous sections we assumed that each of the nodes had unit service rate. In this section we assume that node i has service rate c_i, $i = 1, \ldots, L$, and indicate how the results of the previous sections can be generalized.

In case the service rates are not all equal, it can be verified that there exists no closed-form expression for the AFS allocation $s_i(n)$, $i = 0, \ldots, L$. The following proposition presents bounds on $s_i(n)$. First define

$$c_{\min} := \min_{i=1,\ldots,L} c_i; \quad c_{\max} := \max_{i=1,\ldots,L} c_i;$$

$$\underline{s_0}(n) := \frac{(\kappa_0 n_0^\alpha)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + \left(\sum_{j=1}^L \kappa_j \left(\frac{n_j c_{\min}}{c_j}\right)^\alpha\right)^{1/\alpha}} c_{\min},$$

and

$$\overline{s_0}(n) := \frac{(\kappa_0 n_0^\alpha)^{1/\alpha}}{(\kappa_0 n_0^\alpha)^{1/\alpha} + \left(\sum_{j=1}^L \kappa_j \left(\frac{n_j c_{\max}}{c_j}\right)^\alpha\right)^{1/\alpha}} c_{\max}.$$

Proposition 7.7.1 *If $n \neq 0$ then, for $i = 1, \ldots, L$,*

$$\underline{s_0}(n) \leq s_0(n) \leq \overline{s_0}(n); \quad (c_i - \overline{s_0}(n))\, 1_{n_i>0} \leq s_i(n) \leq \left(c_i - \underline{s_0}(n)\right) 1_{n_i>0}.$$

Proof: In order to obtain $s_i(n)$, we first need to solve the optimization problem (6.9). If $n_0 > 0$, then it is straightforward to show that the optimizer x_0^* satisfies $f(x_0^*) = g(x_0^*)$, where

$$f(x_0) := \kappa_0 x_0^{-\alpha}; \quad g(x_0) := \sum_{j=1}^L \kappa_j n_j^\alpha (c_j - n_0 x_0)^{-\alpha}.$$

As mentioned above, in general there does not exist a closed-form expression for x_0^* that satisfies $f(x_0^*) = g(x_0^*)$. However, note that

$$g(x_0) \geq \sum_{j=1}^L \kappa_j n_j^\alpha \left(c_j - \frac{c_j}{c_{\max}} n_0 x_0\right)^{-\alpha} = \sum_{j=1}^L \kappa_j n_j^\alpha \left(\frac{c_j}{c_{\max}} (c_{\max} - n_0 x_0)\right)^{-\alpha} =: \underline{g}(x_0).$$

Also, we have that

$$g(x_0) \leq \sum_{j=1}^L \kappa_j n_j^\alpha \left(c_j - \frac{c_j}{c_{\min}} n_0 x_0\right)^{-\alpha} = \sum_{j=1}^L \kappa_j n_j^\alpha \left(\frac{c_j}{c_{\min}} (c_{\min} - n_0 x_0)\right)^{-\alpha} =: \overline{g}(x_0).$$

The value of x_0 for which $n_0 f(x_0)$ equals $n_0 \underline{g}(x_0)$ is $\overline{s_0}(x)$, and the value of x_0 for which $n_0 f(x_0)$ equals $n_0 \overline{g}(x_0)$ is $\underline{s_0}(x)$, i.e., we find that $\underline{s_0}(n) \leq s_0(n) = n_0 x_0^* \leq \overline{s_0}(n)$

if $n_0 > 0$. Next use that $s_i(n) = c_i - s_0(n)$ if $n_i > 0$, and the bounds on $s_0(n)$, to find bounds on $s_i(n)$, $i = 1, \ldots, L$. Clearly, if $n_i = 0$, then by definition $s_i(n) = 0$, $i = 0, \ldots, L$, which is also supported by the bounds. □

Proposition 7.7.1 shows that the bounds are tight if $c_i = c$, $i = 1, \ldots, L$, i.e., if each node has service rate c. By setting $\rho_i = \lambda_i/(\mu_i c_{\min})$ or $\rho_i = \lambda_i/(\mu_i c_{\max})$, $i = 0, \ldots, L$, we may use the same techniques of the previous sections to derive approximations for the mean number of users of each class, given that one or two of the nodes are critically loaded. Clearly, the smaller the difference between $c_{\max}$ and $c_{\min}$, the better the approximations will be.

7.8 Discussion

In Section 7.6 we devised approximations for the mean number of users, based on the assumption that two of the nodes operate under heavy-traffic conditions and that all classes have equal weights. It is substantially more difficult to handle the cases in which there are 1) two nodes critically loaded and not all class weights are equal, or 2) more than two bottleneck nodes. Although the mean number of users can still be related to the mean workloads in these scenarios, the joint workload process at these nodes is no longer independent of the fairness coefficient α. In addition, even for a weighted proportional fair policy the workload distribution is no longer known. Hence, we cannot apply the three methods presented in Section 7.6 for approximating the mean number of users.

One option to obtain conservative estimates in case 2) would be to use the property that the workload for an unweighted AFS policy, with α larger than one, is smaller than for an unweighted proportional fair policy as mentioned in Section 7.5. Alternatively, as in Section 7.3, we can approximate the workload process by an SRBM living in a cone that now does depend on the fairness coefficient α. Subsequently, we can derive the steady-state distribution of the process, thus having an approximation for the mean workloads. If we succeeded in this, then we could obtain approximations for the mean number of users by applying one of the three methods. However, it turns out to be extremely hard to derive the steady-state distribution of an SRBM living in a multi-dimensional cone, see [76]. The latter suggests that it is also hard to determine the steady-state distribution of the approximation for the workload process, if possible at all.

Chapter 8

Flow-level performance of traffic-splitting networks

In the previous two chapters we assumed that each class of users corresponded to a unique route in the network. In this chapter we consider the case that some class of users has multiple alternative paths through the network.

The performance of communication networks can be improved when the service demands are efficiently divided among the available resources, so-called *load balancing*. One can apply either *static* or *dynamic* load balancing. In the former case the balancing is not affected by the state of the network, whereas in the latter case it does depend on the system state. It is clear that better performance can be achieved when using dynamic load balancing, but it is often hard to find the optimal load balancing policy. Even for simple systems such a dynamic load balancing problem has non-trivial solutions [173].

In this chapter we analyze load balancing in data networks carrying elastic traffic, as considered by [135]. Transfers in such networks can be represented by flows. We may distinguish between load balancing at the flow-level or the packet-level, depending on whether an arriving flow is entirely directed to a specific route (that it uses until the flow is finished) or a flow can be split between several routes, respectively. This chapter deals with packet-level load balancing, i.e., we assume that packets of a flow can be divided among several routes.

We analyze a network in which, besides classes of users that use specific routes, one class of users can split its traffic over several routes. This particular network is useful for analyzing the performance and potential gains of load balancing at the packet-level. In addition, this system allows for rather explicit results.

We assume that packet-level load balancing is based on an AFS policy. Under this policy, the above network can be shown to have multiple possible behaviors. In particular, we show that packet-level load balancing based on AFS implies that some classes of users, depending on the state of the network, share capacity according to

some DPS model, whereas each of the remaining classes of users behaves as in a single-class single-node model.

The flow-level performance of the above network is compared to that of a similar network, where packet-level load balancing is based on BFS, so-called *insensitive* load balancing at the packet-level. The term 'insensitive' refers to the fact that the corresponding steady-state distribution depends on the traffic characteristics through the traffic intensity only.

Assuming Poisson arrivals and exponentially distributed service requirements, the dynamics of the flow population may be described by a Markov process under both packet-level load balancing policies. We derive closed-form expressions for the mean number of users of each class under insensitive load balancing. Extensive simulation experiments show that these are also quite accurate approximations for the ones in a similar network where load balancing is based on AFS, for which no explicit expressions are available.

The remainder of this chapter is organized as follows. In Section 8.1 we first provide a detailed model description, and introduce BFS and AFS. In the next section we consider the model for a fixed flow population, and characterize how bandwidth is allocated under both policies. In Section 8.3 we consider the model at large timescales, so that the state of the network varies. We derive explicit expressions for the mean number of users under BFS, and show by conducting extensive simulation experiments that these provide accurate approximations for the ones under AFS. In the next section we examine the performance gain that one can achieve for both policies by using packet-level load balancing instead of static or flow-level load balancing.

8.1 Queueing model

We consider the network as depicted in Figure 8.1. The network consists of L nodes, where node i has service rate c_i, $i = 1, \ldots, L$. There are $L + 1$ classes of users. Class i requires service at node i, $i = 1, \ldots, L$, whereas class 0 can be served at all nodes at the same time, i.e., class-0 users can split their traffic.

We assume that class-i users arrive according to a Poisson process of rate λ_i, and have exponentially distributed service requirements with mean μ_i^{-1}, $i = 0, \ldots, L$. The arrival processes are all independent. The traffic load of class i is then $\rho_i = \lambda_i \mu_i^{-1}$. Let $n = (n_0, \ldots, n_L)$ denote the state of the network, with n_i representing the number of class-i users.

8.1.1 BFS

We first assume that the bandwidth is shared according to BFS, see Chapter 1. Let $\phi_i(n)$ denote the service rate allocated to class i, $i = 0, \ldots, L$, under BFS, when the network is in state n (here $\phi_0(n) = \sum_{i=1}^{L} \phi_{0i}(n)$). These service rates have to satisfy

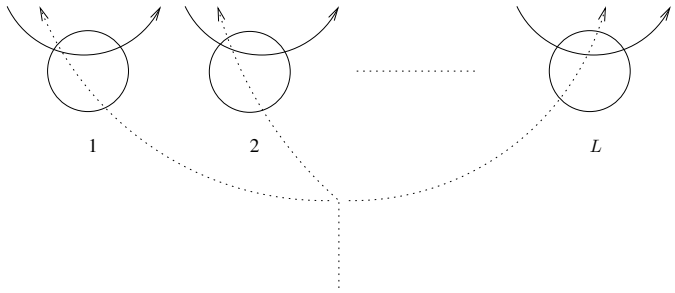

Figure 8.1: The bandwidth-sharing network.

the balance conditions

$$\frac{\phi_i(n-e_j)}{\phi_i(n)} = \frac{\phi_j(n-e_i)}{\phi_j(n)} \quad \forall i,j = 0,\ldots,L, \quad n_i, n_j > 0, \tag{8.1}$$

where e_i denotes the $(i+1)$th unit vector in $\mathbb{R}^{L+1}$. All BFS rates can be expressed in terms of a unique balance function $\Phi(\cdot)$, so that $\Phi(0) = 1$ and

$$\phi_i(n) = \frac{\Phi(n-e_i)}{\Phi(n)} \quad \forall n : n_i > 0, \quad i = 0,\ldots,L. \tag{8.2}$$

Hence, characterization of $\Phi(n)$ implies that $\phi_i(n)$ is characterized as well. Define $\Phi(n) = 0$ if $n \notin \mathbb{N}_0^{L+1}$. In order to obtain $\Phi(n)$, we need to solve the following maximization problem for each $n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}$:

$$\begin{aligned} \max \quad & \Phi(n)^{-1} \\ \text{subject to} \quad & \textstyle\sum_{j=1}^{L} \phi_{0j}(n) = \frac{\Phi(n-e_0)}{\Phi(n)} \\ & \phi_i(n) = \tfrac{\Phi(n-e_i)}{\Phi(n)}, \quad i = 1,\ldots,L \\ & \phi_{0i}(n) + \phi_i(n) \le c_i, \quad i = 1,\ldots,L \\ \text{over} \quad & \phi_{0i}(n), \phi_i(n) \ge 0, \quad i = 1,\ldots,L. \end{aligned} \tag{8.3}$$

It is clear that $\Phi(n)$ can be obtained recursively: $\Phi(n-e_i)$ is required to determine $\Phi(n)$, $i = 0,\ldots,L$. Also note that (8.3) is a simple linear programming (LP) problem, which can be solved using standard LP algorithms. In Section 8.2.1, however, we solve (8.3) by rewriting the LP problem in terms of a related network.

8.1.2 AFS

We next assume that the network operates under an AFS policy, as introduced in [140]. When the network is in state $n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}$, the service rate x_i^* allocated to each of

the class-i users is obtained by solving the following optimization problem:

$$\begin{array}{rll} \max & & F(x) \\ \text{subject to} & n_0 x_{0i} + n_i x_i \le c_i, & i = 1, \ldots, L \\ \text{over} & x_{0i}, x_i \ge 0, & i = 1, \ldots, L, \end{array} \tag{8.4}$$

where the objective function $F(x)$ is defined by

$$F(x) := \begin{cases} n_0 \kappa_0 \frac{(\sum_{i=1}^L x_{0i})^{1-\alpha}}{1-\alpha} + \sum_{i=1}^L n_i \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in (0,\infty)\backslash\{1\}; \\ n_0 \kappa_0 \log(\sum_{i=1}^L x_{0i}) + \sum_{i=1}^L n_i \kappa_i \log(x_i) & \text{if } \alpha = 1. \end{cases}$$

The κ_is are non-negative class weights, and $\alpha \in (0,\infty)$ may as before be interpreted as a fairness coefficient. The value of x_{0i}^* denotes how much capacity is allocated to path i (that requires service at node i) of class 0. Here $x_0^* = \sum_{i=1}^L x_{0i}^*$ denotes how much capacity is assigned to a single class-0 user in the network. Let $s_i(n) := n_i x_i^*$ denote the total service rate allocated to class i, $i = 0, \ldots, L$.

8.2 Static setting

In this section we consider the model for a fixed flow population, i.e., the state $n \in \mathbb{N}_0^{L+1}\backslash\{0\}$ is fixed, and we derive how bandwidth is shared between the various classes in case of BFS and AFS, respectively. We first show that the network depicted in Figure 8.1 is equivalent to another network. In order to do so, let us first introduce the notion of the capacity set.

The allocations $\phi(n) = (\phi_0(n), \ldots, \phi_L(n))$ and $s(n) = (s_0(n), \ldots, s_L(n))$ are clearly constrained by the capacity set $\mathcal{C} \subseteq \mathbb{R}_+^{L+1}$:

$$\mathcal{C} := \left\{ x \ge 0 : \exists a_1, \ldots, a_L \ge 0, \quad \sum_{j=1}^L a_j = 1, \quad a_i x_0 + x_i \le c_i, \quad i = 1, \ldots, L \right\},$$

i.e., $\phi(n) \in \mathcal{C}$ and $s(n) \in \mathcal{C}$ for all $n \in \mathbb{N}_0^{L+1}$. It is straightforward to show that the capacity set $\mathcal{C}$ can also be expressed as

$$\tilde{\mathcal{C}} := \left\{ x \ge 0 : \sum_{j=0}^L x_j \le \sum_{j=1}^L c_j, \quad x_i \le c_i, \quad i = 1, \ldots, L \right\},$$

i.e., $\mathcal{C} = \tilde{\mathcal{C}}$. Since $\tilde{\mathcal{C}}$ is the capacity set corresponding to the tree network depicted in Figure 8.2, it follows that the networks depicted in Figures 8.1 and 8.2 are in fact equivalent. The tree has a common link with capacity $c_1 + \cdots + c_L$, and $L+1$ branches with capacities ∞, $c_1, \ldots, c_L$, respectively. In this network class-i users require service at the node with service rate c_i and at the common link, $i = 1, \ldots, L$,

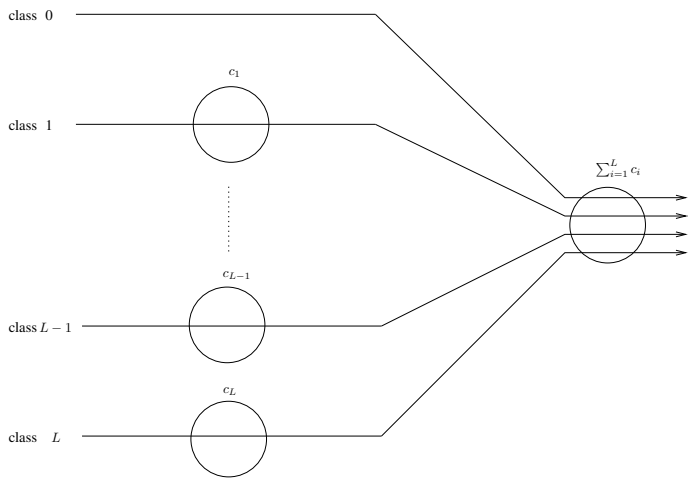

Figure 8.2: Tree network

whereas class-0 users only require service at the common link. Note that each class of users corresponds to a specific route in the tree network.

As a side remark we mention that in general it is not true that a network (where some classes of users can split their traffic over several routes at the same time) can be converted into a tree network. In fact, if we extend the model depicted in Figure 8.1 by adding a class of users that requires service at all L nodes simultaneously, then it is no longer possible to represent the network as a tree network. However, we note that in general one may still be able to convert a traffic-splitting network into some other network (with dummy nodes) without traffic splitting.

8.2.1 BFS

In this subsection we derive the BFS allocation by solving the problem (8.3). Since the models depicted in Figures 8.1 and 8.2 are equivalent, it follows that the balance function $\tilde{\Phi}(\cdot)$ corresponding to the tree network coincides with $\Phi(\cdot)$, i.e., $\tilde{\Phi}(\cdot) = \Phi(\cdot)$, see [24]. In the following lemma we present the solution of the problem (8.3).

Lemma 8.2.1 *The BFS function $\Phi(n)$ satisfies, with $\Phi(0) = 1$,*

$$\Phi(n) = \max\left\{ \frac{\Phi(n-e_1)}{c_1}, \ldots, \frac{\Phi(n-e_L)}{c_L}, \frac{\sum_{i=0}^{L} \Phi(n-e_i)}{\sum_{i=1}^{L} c_i} \right\}, \quad n \in \mathbb{N}_0^{L+1} \backslash \{\vec{0}\}. \tag{8.5}$$

Proof: From the above it follows that we can obtain $\Phi(\cdot)$ by determining $\tilde{\Phi}(\cdot)$, as $\Phi(\cdot) = \tilde{\Phi}(\cdot)$. Subsequently, $\tilde{\Phi}(\cdot)$ is obtained by using Equation (2) in [27]. □

We note that Lemma 8.2.1 is in agreement with Equation (19) in [108]. From Lemma 8.2.1 it follows that $\Phi(n)$ can be obtained recursively. The total service rate allocated to class i, $i = 0, \ldots, L$, in each state $n \in \mathbb{N}_0^{L+1}$ can be obtained using Lemma 8.2.1 and (8.2).

8.2.2 AFS

In this subsection we focus on the AFS allocation, which is obtained by solving the problem (8.4). Similar to the previous subsection, we can obtain the AFS allocation $s(n)$ by determining the AFS allocation $\tilde{s}(n)$ in the tree network, as both networks are the same, implying that $s(n) = \tilde{s}(n)$. In order to obtain $\tilde{s}(n)$ we need to solve the following maximization problem:

$$\begin{array}{rl} \max & H(x) \\ \text{subject to} & \sum_{i=0}^{L} n_i x_i \leq \sum_{i=1}^{L} c_i \\ & n_i x_i \leq c_i, \quad i = 1, \ldots, L \\ \text{over} & x_i \geq 0, \quad i = 0, \ldots, L, \end{array} \tag{8.6}$$

where the objective function $H(x)$ is defined by

$$H(x) := \begin{cases} \sum_{i=0}^{L} n_i \kappa_i \frac{x_i^{1-\alpha}}{1-\alpha} & \text{if } \alpha \in (0,\infty)\backslash\{1\}; \\ \sum_{i=0}^{L} n_i \kappa_i \log(x_i) & \text{if } \alpha = 1. \end{cases}$$

Below we show that (8.6) is solvable, but the optimal solution strongly depends on the state $n \in \mathbb{N}_0^{L+1}\backslash\{\vec{0}\}$. We present a simple algorithm for obtaining the AFS allocation.

Lemma 8.2.2 *The AFS allocation $s(n)$ can be obtained with the following algorithm:*

Set Stop:=False
Set $S := \{0, \ldots, L\}$
WHILE Stop=False DO
  Determine the $|S|$-class DPS rates: $s_i(n) := \frac{n_i \kappa_i^{1/\alpha} \sum_{j \in S\backslash\{0\}} c_j}{\sum_{j \in S} n_j \kappa_j^{1/\alpha}}$, $i \in S$
  IF $s_i(n) \leq c_i$ for all $i \in S\backslash\{0\}$ THEN set Stop:=True
  ELSE
    Take any $i^* \in S\backslash\{0\}$ such that $s_{i^*}(n) > c_i$
    Set $S := S\backslash\{i^*\}$
    Set $s_{i^*}(n) := c_i$
  END
END

Proof: First consider the Karush-Kuhn-Tucker (KKT) [104] necessary conditions for the problem (8.6). If x is an optimal solution to the problem (8.6), then there exist constants $p_i \geq 0$, $i = 0, \ldots, L$, such that,

$$\frac{n_0 \kappa_0}{x_0^\alpha} - n_0 p_0; \tag{8.7}$$

$$\frac{n_i \kappa_i}{x_i^\alpha} - n_i(p_0 + p_i), \quad i = 1, \ldots, L; \tag{8.8}$$

$$p_0 \left(\sum_{i=1}^{L} c_i - \sum_{i=0}^{L} n_i x_i \right) = 0; \tag{8.9}$$

$$p_i \left(c_i - n_i x_i \right) = 0, \quad i = 1, \ldots, L. \tag{8.10}$$

Note that (8.7) and (8.8) hold for any $\alpha \in (0, \infty)$. Solving (8.7)-(8.10) for $(x_0, \ldots, x_L)$ and $(p_0, \ldots, p_L)$ yields $\sum_{q=1}^{L} \frac{L!}{q!(L-q)!} = 2^L - 1$ possible solutions, however, depending on the state of the network n, only one of the $2^L - 1$ solutions, x^*, is such that $p_i \geq 0$, $i = 0, \ldots, L$, i.e., this is the optimal solution for (8.6). For each of the other solutions there exists at least one Lagrange multiplier that is negative, implying that these solutions cannot be optimal. Note that the existence of a unique optimal solution x^* for (8.6) also follows as $H(x)$ is strictly concave and the constraints are linear. Straightforward calculus shows that the corresponding AFS allocation $\tilde{s}_i(n) = s_i(n) = n_i x_i^*$, $i = 0, \ldots, L$, can be obtained by the above algorithm. The algorithm reflects that $2^L - 1$ solutions exist for (8.7)-(8.10), but it also shows that only one of these solutions, x^*, is found after termination of the algorithm. The Lagrange multipliers corresponding to x^* are such that $p_i = 0$ if $i \in S \backslash \{0\}$, and $p_i > 0$ if $i \notin S \backslash \{0\}$, where S is the set obtained after termination of the algorithm. Furthermore, $p_0 = 0$ if $n_0 = 0$ and if there exists an i such that $n_i = 0$, $i = 1, \ldots, L$, otherwise $p_0 > 0$. □

8.3 Flow-level dynamics

In the previous section we considered the model for a fixed flow population, and we derived expressions for the BFS and AFS allocations in each state of the network. In this section we analyze the model at sufficiently large time scales. In this case we also have to take the random nature of the traffic into account, i.e., the state of the network n varies at large time scales.

8.3.1 BFS

Let $N(t) = (N_0(t), \ldots, N_L(t))$ denote the state of the network at time t. Since we assumed Poisson arrivals and exponentially distributed service requirements, $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i \phi_i(n), \quad i = 0, \ldots, L,$$

in case of BFS. In [24] it was shown that the process $N(t)$ is stable if there exists $(\tilde{\rho}_{01}, \ldots, \tilde{\rho}_{0L})$ such that

$$\sum_{i=1}^{L} \tilde{\rho}_{0i} = \rho_0 \quad \text{and} \quad \tilde{\rho_{0i}} + \rho_i < c_i, \quad i = 1, \ldots, L,$$

or equivalently, if

$$\sum_{i=0}^{L} \rho_i < \sum_{j=1}^{L} c_j \quad \text{and} \quad \rho_i < c_i, \quad i = 1, \ldots, L. \tag{8.11}$$

It may be verified from (8.1) that the steady-state queue length distribution is given by

$$\pi(n) = \frac{1}{G(\rho)} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}, \quad n \in \mathbb{N}_0^{L+1}, \tag{8.12}$$

where the normalization constant $G(\rho)$ equals

$$G(\rho) = G(\rho_0, \ldots, \rho_L) = \sum_{n_0=0}^{\infty} \cdots \sum_{n_L=0}^{\infty} \Phi(n) \prod_{i=0}^{L} \rho_i^{n_i}.$$

As a side remark we mention that (8.12) in fact holds for much more general traffic characteristics, see [25] for a more detailed treatment.

When applying Little's formula we find that

$$\mathbb{E}N_i^{BF} = \rho_i \frac{\frac{\partial G(\rho)}{\partial \rho_i}}{G(\rho)} = \rho_i \frac{\partial \log G(\rho)}{\partial \rho_i}, \quad i = 0, \ldots, L, \tag{8.13}$$

i.e., characterization of $G(\rho)$ implies that $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$, is known as well.

By exploiting the results of [29] on tree networks we can determine $G(\rho)$, and it can be verified that this results in

$$G(\rho) = \frac{1}{1 - \frac{\sum_{i=0}^{L} \rho_i}{\sum_{i=1}^{L} c_i}} \frac{1 - \frac{\sum_{i=1}^{L} \rho_i}{\sum_{i=1}^{L} c_i}}{\prod_{i=1}^{L} \left(1 - \frac{\rho_i}{c_i}\right)}. \tag{8.14}$$

Then by using (8.13) we can obtain a closed-form expression for $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$. The expression for $\mathbb{E}N_i^{BF}$, $i = 1, \ldots, L$, is in general quite complicated, in contrast to the expression for the mean number of class-0 users, which is given by

$$\mathbb{E}N_0^{BF} = \frac{\rho_0}{\sum_{i=1}^{L} c_i - \sum_{i=0}^{L} \rho_i}.$$

From (8.14) it follows that $\mathbb{E}N_i^{BF}$, $i = 0, \ldots, L$, is finite if the stability condition (8.11) holds.

8.3.2 AFS

As before, let $N(t) = (N_0(t), \ldots, N_L(t))$ denote the state of the network at time t. In case of AFS $N(t)$ is a Markov process with transition rates:

$$q(n, n + e_i) = \lambda_i; \quad q(n, n - e_i) = \mu_i s_i(n), \quad i = 0, \ldots, L.$$

Since our network is equivalent to the tree network depicted in Figure 8.2, it follows from Theorem 1 in [23] that the process $N(t)$ is stable if (8.11) holds.

Lemma 8.2.2 shows that, depending on the state of the network $n \in \mathbb{N}_0^{L+1}$, the network has $2^L - 1$ possible behaviors. This illustrates the complication of finding closed-form expressions for the mean number of users of each class. In fact, so far no expressions for the mean number of users are available in case of AFS. To gain some insight, we derive in this section approximations for the mean number of users of each class, i.e., $\mathbb{E}N_i^{AF}$, $i = 0, \ldots, L$. The approximations are validated by means of simulation experiments. We consider the case where the network consists of $L = 2$ nodes, but we note that the approximations can be extended to the case $L > 2$ in a similar fashion.

Using Lemma 8.2.2 in Section 8.2.2, it follows that the network, depending on the state n, has three possible behaviors: (i) if

$$n_1 > \frac{c_1}{c_2}\left(\left(\frac{\kappa_2}{\kappa_1}\right)^{1/\alpha} n_2 + \left(\frac{\kappa_0}{\kappa_1}\right)^{1/\alpha} n_0\right),$$

then classes 0 and 2 behave as in a two-class DPS model with capacity c_2 and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 2$, whereas class 1 behaves as an M/M/1 queue with arrival rate λ_1 and service rate $\mu_1 c_1$; (ii) if

$$n_1 < \frac{c_1}{c_2}\left(\frac{\kappa_2}{\kappa_1}\right)^{1/\alpha} n_2 - \left(\frac{\kappa_0}{\kappa_1}\right)^{1/\alpha} n_0,$$

then classes 0 and 1 behave as in a two-class DPS model with capacity c_1 and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 1$, whereas class 2 behaves as an M/M/1 queue with arrival rate λ_2 and service rate $\mu_2 c_2$; (iii) otherwise the network will behave as in a three-class DPS model with capacity $c_1 + c_2$ and relative weights $\kappa_i^{1/\alpha}$, $i = 0, 1, 2$.

If the network were to behave as (i) all the time and if $\rho_1 < c_1$ and $\rho_0 + \rho_2 < c_2$ (stability conditions), then by exploiting the results of [63] we would obtain

$$\mathbb{E}N_0^{(i)} = \frac{\rho_0}{c_2 - \rho_0 - \rho_2}\left(1 + \frac{\mu_0\rho_2\left(\kappa_2^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(c_2 - \rho_0) + \kappa_2^{1/\alpha}\mu_2(c_2 - \rho_2)}\right);$$

$$\mathbb{E}N_1^{(i)} = \frac{\rho_1}{c_1 - \rho_1};$$

$$\mathbb{E}N_2^{(i)} = \frac{\rho_2}{c_2-\rho_0-\rho_2}\left(1+\frac{\mu_2\rho_0\left(\kappa_0^{1/\alpha}-\kappa_2^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(c_2-\rho_0)+\kappa_2^{1/\alpha}\mu_2(c_2-\rho_2)}\right).$$

Likewise, when the network behaves as (ii) and if $\rho_2 < c_2$ and $\rho_0+\rho_1 < c_1$ (stability conditions), we find

$$\mathbb{E}N_0^{(ii)} = \frac{\rho_0}{c_1-\rho_0-\rho_1}\left(1+\frac{\mu_0\rho_1\left(\kappa_1^{1/\alpha}-\kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(c_1-\rho_0)+\kappa_1^{1/\alpha}\mu_1(c_1-\rho_1)}\right);$$

$$\mathbb{E}N_1^{(ii)} = \frac{\rho_1}{c_1-\rho_0-\rho_1}\left(1+\frac{\mu_1\rho_0\left(\kappa_0^{1/\alpha}-\kappa_1^{1/\alpha}\right)}{\kappa_0^{1/\alpha}\mu_0(c_1-\rho_0)+\kappa_1^{1/\alpha}\mu_1(c_1-\rho_1)}\right);$$

$$\mathbb{E}N_2^{(ii)} = \frac{\rho_2}{c_2-\rho_2}.$$

If the network behaves as a three-class DPS model, i.e., as (iii), and if $\rho_0+\rho_1+\rho_2 < c_1+c_2$ (stability condition), then one can obtain the mean number of users of each class by solving the following set of linear equations for $\mathbb{E}N_i^{(iii)}$, $i=0,1,2$:

$$(c_1+c_2)\mathbb{E}N_i^{(iii)} - \lambda\sum_{j=0}^{2}\kappa_j^{1/\alpha}\frac{\frac{\lambda_j}{\lambda}\mathbb{E}N_i^{(iii)}+\frac{\lambda_i}{\lambda}\mathbb{E}N_j^{(iii)}}{\kappa_j^{1/\alpha}\mu_j+\kappa_i^{1/\alpha}\mu_i} = \rho_i, \quad i=0,1,2,$$

where $\lambda := \lambda_0+\lambda_1+\lambda_2$, see [63]. In this case there also exists a closed-form expression for $\mathbb{E}N_i^{(iii)}$, $i=0,1,2$, but it is complicated.

We propose the following approximation: $\mathbb{E}N_i^{AF} \approx \mathbb{E}N_i^{AP}$, $i=0,1,2$, where

$$\mathbb{E}N_0^{AP} := \mathbb{E}N_0^{(iii)}; \quad \mathbb{E}N_1^{AP} := \max\{\mathbb{E}N_1^{(i)}, \mathbb{E}N_1^{(iii)}\}; \quad \mathbb{E}N_2^{AP} := \max\{\mathbb{E}N_2^{(ii)}, \mathbb{E}N_2^{(iii)}\}.$$

It can be verified that $\mathbb{E}N_0^{AP}$ is bounded if $\rho_0+\rho_1+\rho_2 < c_1+c_2$, $\mathbb{E}N_1^{AP}$ is bounded if $\rho_1 < c_1$ and $\rho_0+\rho_1+\rho_2 < c_1+c_2$, and $\mathbb{E}N_2^{AP}$ is bounded if $\rho_2 < c_2$ and $\rho_0+\rho_1+\rho_2 < c_1+c_2$. Hence, $\mathbb{E}N_i^{AP}$, $i=0,1,2$, is bounded if (8.11) holds, i.e., if the process $N(t)$ is stable.

In [24] it was argued that the performance of a network under proportional fairness ($\alpha=1$) and max-min fairness ($\alpha\to\infty$) is closely approximated by that under BFS. Therefore, we also propose the following approximation: $\mathbb{E}N_i^{AF} \approx \mathbb{E}N_i^{BF}$, $i=0,1,2$. The value of $\mathbb{E}N_i^{BF}$, $i=0,1,2$, can be obtained using (8.13), and is independent of the value of α.

To examine the accuracy of the above approximations we have performed simulation experiments. We consider the setting with $c_1=c_2=1$, and we take $\lambda_i=\gamma$, $\mu_i=1$, $i=0,1,2$, such that $\rho_0=\rho_1=\rho_2=\gamma$. We first consider scenario I, where $\kappa_i=1$, $i=0,1,2$. Subsequently, we consider scenario II, where $\kappa_0=5$, $\kappa_1=1$ and $\kappa_2=2$. In scenario II we let the traffic load γ and the AFS coefficient α vary,

γ	EN_0^{AF}	EN_1^{AF}	EN_2^{AF}	EN_0^{AP}	EN_1^{AP}	EN_2^{AP}	EN_0^{BF}	EN_1^{BF}	EN_2^{BF}
0.1	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	0.30	0.46	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	0.55	0.77	0.77	0.50	0.67	0.67	0.50	0.83	0.83
0.5	1.10	1.39	1.39	1.00	1.00	1.00	1.00	1.50	1.50
0.6	3.17	3.48	3.48	3.00	3.00	3.00	3.00	3.75	3.75

Table 8.1: Simulation results for scenario I.

γ	α	EN_0^{AF}	EN_1^{AF}	EN_2^{AF}	EN_0^{AP}	EN_1^{AP}	EN_2^{AP}	EN_0^{BF}	EN_1^{BF}	EN_2^{BF}
0.1	1	0.06	0.12	0.12	0.06	0.11	0.11	0.06	0.11	0.11
0.2	1	0.13	0.28	0.27	0.12	0.25	0.25	0.14	0.27	0.27
0.3	1	0.23	0.54	0.49	0.22	0.43	0.43	0.27	0.49	0.49
0.4	1	0.39	0.97	0.83	0.35	0.67	0.67	0.50	0.83	0.83
0.5	1	0.68	1.95	1.46	0.59	1.43	1.00	1.00	1.50	1.50
0.6	1	1.55	5.93	3.47	1.38	4.82	2.80	3.00	3.75	3.75
0.1	2	0.06	0.12	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	2	0.14	0.27	0.26	0.13	0.25	0.25	0.14	0.27	0.27
0.3	2	0.26	0.50	0.48	0.24	0.43	0.43	0.27	0.49	0.49
0.4	2	0.47	0.88	0.81	0.42	0.67	0.67	0.50	0.83	0.83
0.5	2	0.87	1.71	1.44	0.77	1.23	1.00	1.00	1.50	1.50
0.6	2	2.35	4.81	3.66	2.06	3.95	2.98	3.00	3.75	3.75
0.1	5	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	5	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	5	0.28	0.48	0.46	0.26	0.43	0.43	0.27	0.49	0.49
0.4	5	0.52	0.82	0.78	0.46	0.67	0.67	0.50	0.83	0.83
0.5	5	1.00	1.51	1.40	0.90	1.09	1.00	1.00	1.50	1.50
0.6	5	2.84	3.95	3.61	2.60	3.38	3.01	3.00	3.75	3.75
0.1	∞	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	∞	0.15	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	∞	0.30	0.46	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	∞	0.55	0.77	0.77	0.50	0.67	0.67	0.50	0.83	0.83
0.5	∞	1.10	1.39	1.39	1.00	1.00	1.00	1.00	1.50	1.50
0.6	∞	3.17	3.48	3.48	3.00	3.00	3.00	3.00	3.75	3.75

Table 8.2: Simulation results for scenario II.

whereas in scenario I we only let γ vary, as it can be verified that $\mathbb{E}N_i^{AF}$ and $\mathbb{E}N_i^{AP}$, $i = 0, 1, 2$, are independent of the value of α in scenario I. To ensure stability we assume that $\gamma < 2/3$. The results are reported in Tables 8.1 and 8.2. Each reported simulation value in these (and other) tables is measured over $4{\cdot}10^6$ events, i.e., arrivals or departures.

Remark: We have also determined a 95% confidence interval (CI) for each listed simulation value in this chapter, but these are not presented. We note, however, that the relative efficiency, i.e., the ratio of the half-length of the CI to the reported simulation value, is less than 3% for all listed cases in Tables 8.1, 8.2, 8.5 and 8.6, and less than 10% for all listed cases in Tables 8.3 and 8.4.

Table 8.1 compares the value of $\mathbb{E}N_i^{AF}$ obtained by simulation with the approximations $\mathbb{E}N_i^{AP}$ and $\mathbb{E}N_i^{BF}$, $i = 0, 1, 2$, for scenario I. The results show that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$. Also, the table shows that $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{BF}$ and $\mathbb{E}N_i^{AF} \leq \mathbb{E}N_i^{BF}$, $i = 1, 2$. Overall we see that both approximations are accurate in case of equal class weights, especially for low traffic load.

Table 8.2 reports the results corresponding to scenario II, i.e., in case of unequal class weights. In this case $\mathbb{E}N_i^{AF}$ and $\mathbb{E}N_i^{AP}$ do depend on the value of α, as is shown

in the table. Again, we see that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$. For low traffic loads both approximations perform quite well, but for high traffic loads we see that the BFS approximation is less accurate than the other one.

Tables 8.1 and 8.2 show that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$, which may be explained as follows. First note that the rate allocated to class 1 is smaller than or equal to c_1 at all moments in time under AFS, whereas rate c_1 is continuously available to class 1 in (i). Clearly, this implies that $\mathbb{E}N_1^{AF} \geq \mathbb{E}N_1^{(i)}$. With similar reasoning, we find that $\mathbb{E}N_2^{AF} \geq \mathbb{E}N_2^{(ii)}$. Since class-$i$ users cannot be allocated more than c_i, $i = 1, 2$, under AFS, whereas in the three-class DPS model the upper bound is $c_1 + c_2$ for both classes, one may expect that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{(iii)}$, $i = 1, 2$. For any state $n \in \mathbb{N}_0^3 \backslash \{0\}$ it can be verified that the AFS allocation to class 0 is larger or equal than the one obtained in the three-class DPS model, so one would expect $\mathbb{E}N_0^{AF} \leq \mathbb{E}N_0^{(iii)}$ at first sight. However, recall that we argued that the number of users of classes 1 and 2 in the model operating under AFS will (on average) be larger than in the three-class DPS model, which causes that the total service allocated to class 0 in the model operating under AFS is less than or equal to that in the three-class DPS model, i.e., we may also expect $\mathbb{E}N_0^{AF} \geq \mathbb{E}N_0^{(iii)}$. The above reasoning indeed suggests that $\mathbb{E}N_i^{AF} \geq \mathbb{E}N_i^{AP}$, $i = 0, 1, 2$.

Fluid and quasi-stationary regimes

To test the performance of the two approximations in case of extreme parameter values, we now assume that the flow dynamics of the various classes occur on widely separate time scales, i.e., in fluid and quasi-stationary regimes.

Formally, let $\lambda_i^{(r)} := \lambda_i f_i(r)$ and $\mu_i^{(r)} := \mu_i f_i(r)$, where $f_i(r)$ represents the time scale associated with class i as function of r, $i = 0, \ldots, L$. Note that the traffic intensity of class i equals $\rho_i^{(r)} := \lambda_i^{(r)}/\mu_i^{(r)} = \rho_i$, $i = 0, \ldots, L$, so it is independent of r. Let $N_i^{(r)}$ be the number of class-i flows in the r-th system. Before analyzing the quality of the approximations, we first present the following useful proposition.

Proposition 8.3.1 *Assume that $L + 1$ classes of users share c units of capacity according to DPS, where class i has relative weight κ_i, $i = 0, \ldots, L$. If $f_{i-1}(r)/f_i(r) \to 0$ as $r \to \infty$, $i = 1, \ldots, L$, i.e., higher indexed classes operate on faster time scales, then*

$$\mathbb{E}N_i^{(\infty)} = \frac{\rho_i}{c - \sum_{j=i}^{L} \rho_j} + \sum_{j=0}^{i-1} \frac{\kappa_j}{\kappa_i} \frac{\rho_i \rho_j}{\left(c - \sum_{r=j}^{L} \rho_r\right)\left(c - \sum_{r=j+1}^{L} \rho_r\right)}, \quad i = 0, \ldots, L.$$

Proof: In [94] the above result was already proved for $L = 1$. For $L > 1$ the authors showed that $\mathbb{E}N_j^{(\infty)}$, $j = 1, \ldots, L$, could be obtained by determining $\mathbb{E}N_i^{(\infty)}$, $i = 0, \ldots, j - 1$, i.e., as a recursion. Straightforward calculus, however, shows that this recursion reduces to the above result. □

γ	$\mathbb{E}N_0^{AF}$	$\mathbb{E}N_1^{AF}$	$\mathbb{E}N_2^{AF}$	$\mathbb{E}N_0^{AP(\infty)}$	$\mathbb{E}N_1^{AP(\infty)}$	$\mathbb{E}N_2^{AP(\infty)}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	0.14	0.25	0.25	0.14	0.25	0.25	0.14	0.27	0.27
0.3	0.27	0.45	0.45	0.27	0.43	0.43	0.27	0.49	0.49
0.4	0.51	0.76	0.76	0.50	0.67	0.67	0.50	0.83	0.83
0.5	1.02	1.34	1.34	1.00	1.00	1.00	1.00	1.50	1.50
0.6	3.06	3.30	3.30	3.00	3.00	3.00	3.00	3.75	3.75

Table 8.3: Results for the fluid and quasi-stationary regimes (scenario I).

Let us return to the setting with $L = 2$ nodes and $L + 1 = 3$ classes of users. Proposition 8.3.1 allows us to obtain simple closed-form expressions for $\mathbb{E}_i^{AP}$, $i = 0, 1, 2$, when $r \to \infty$. Assuming that higher indexed classes operate on faster time scales and that the stability condition (8.11) holds, we find that

$$\mathbb{E}N_0^{AP(\infty)} := \frac{\rho_0}{\overline{c}_1 + \overline{c}_2 - \rho_0};$$

$$\mathbb{E}N_1^{AP(\infty)} := \max\left\{\frac{\rho_1}{\overline{c}_1}, \frac{\rho_1}{\overline{c}_1 + \overline{c}_2} + \frac{\kappa_0^{1/\alpha}\rho_0\rho_1}{\kappa_1^{1/\alpha}(\overline{c}_1 + \overline{c}_2 - \rho_0)(\overline{c}_1 + \overline{c}_2)}\right\},$$

and $\mathbb{E}N_2^{AP(\infty)}$ equals

$$\max\left\{\frac{\rho_2}{\overline{c}_2}, \frac{\rho_2}{c_1 + \overline{c}_2} + \frac{\kappa_0^{1/\alpha}\rho_0\rho_2}{\kappa_2^{1/\alpha}(\overline{c}_1 + \overline{c}_2 - \rho_0)(\overline{c}_1 + \overline{c}_2)} + \frac{\kappa_1^{1/\alpha}\rho_1\rho_2}{\kappa_2^{1/\alpha}(\overline{c}_1 + \overline{c}_2)(c_1 + \overline{c}_2)}\right\},$$

where $\overline{c}_i := c_i - \rho_i$, $i = 1, 2$. In case of equal class weights, $\kappa_i = \kappa$, $i = 0, 1, 2$, it is not hard to see that

$$\mathbb{E}N_0^{AP(\infty)} = \frac{\rho_0}{\overline{c}_1 + \overline{c}_2 - \rho_0};$$

$$\mathbb{E}N_1^{AP(\infty)} = \max\left\{\frac{\rho_1}{\overline{c}_1}, \frac{\rho_1}{\overline{c}_1 + \overline{c}_2 - \rho_0}\right\}; \quad \mathbb{E}N_2^{AP(\infty)} = \max\left\{\frac{\rho_2}{\overline{c}_2}, \frac{\rho_2}{\overline{c}_1 + \overline{c}_2 - \rho_0}\right\}.$$

Clearly, $\mathbb{E}N_i^{AP(\infty)}$, $i = 0, 1, 2$, strongly depends on the ordering of the classes with respect to the time scales. In case of other orderings than the one mentioned above, one can obtain expressions in a similar fashion.

The accuracy of the approximations in the fluid and quasi-stationary regimes is examined by performing simulation experiments. We take $c_1 = c_2 = 1$, $\lambda_0 = \gamma$, $\lambda_1 = 10\gamma$, $\lambda_2 = 100\gamma$, $\mu_0 = 1$, $\mu_1 = 10$, $\mu_2 = 100$, so that $\rho_i = \gamma$, $i = 0, 1, 2$, and thus assume that higher indexed classes operate on faster time scales.

Tables 8.3 and 8.4 report the results for scenarios I and II, respectively. Recall that $\mathbb{E}N_i^{AF}$ and $\mathbb{E}N_i^{AP(\infty)}$, $i = 0, 1, 2$, are independent of the value of α in scenario I, whereas they are sensitive to the value of α in scenario II. The tables show that in the fluid and quasi-stationary regimes the approximations are appropriate as well.

γ	α	$\mathrm{E}N_0^{AF}$	$\mathrm{E}N_1^{AF}$	$\mathrm{E}N_2^{AF}$	$\mathrm{E}N_0^{AP(\infty)}$	$\mathrm{E}N_1^{AP(\infty)}$	$\mathrm{E}N_2^{AP(\infty)}$	$\mathrm{E}N_0^{BF}$	$\mathrm{E}N_1^{BF}$	$\mathrm{E}N_2^{BF}$
0.1	1	0.06	0.12	0.12	0.06	0.11	0.11	0.06	0.11	0.11
0.2	1	0.14	0.31	0.28	0.14	0.25	0.25	0.14	0.27	0.27
0.3	1	0.26	0.63	0.52	0.27	0.51	0.43	0.27	0.49	0.49
0.4	1	0.45	1.23	0.92	0.50	1.17	0.71	0.50	0.83	0.83
0.5	1	0.89	2.85	1.82	1.00	3.00	1.67	1.00	1.50	1.50
0.6	1	2.49	10.28	5.44	3.00	12.00	6.21	3.00	3.75	3.75
0.1	2	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	2	0.14	0.27	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	2	0.27	0.51	0.48	0.27	0.43	0.43	0.27	0.49	0.49
0.4	2	0.49	0.93	0.83	0.50	0.71	0.67	0.50	0.83	0.83
0.5	2	1.03	1.94	1.58	1.00	1.62	1.24	1.00	1.50	1.50
0.6	2	2.69	5.53	4.17	3.00	5.78	4.21	3.00	3.75	3.75
0.1	5	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	5	0.14	0.26	0.26	0.14	0.25	0.25	0.14	0.27	0.27
0.3	5	0.27	0.47	0.46	0.27	0.43	0.43	0.27	0.49	0.49
0.4	5	0.52	0.82	0.79	0.50	0.67	0.67	0.50	0.83	0.83
0.5	5	1.00	1.51	1.42	1.00	1.19	1.08	1.00	1.50	1.50
0.6	5	2.86	4.06	3.65	3.00	3.85	3.41	3.00	3.75	3.75
0.1	∞	0.06	0.11	0.11	0.06	0.11	0.11	0.06	0.11	0.11
0.2	∞	0.14	0.25	0.25	0.14	0.25	0.25	0.14	0.27	0.27
0.3	∞	0.27	0.45	0.45	0.27	0.43	0.43	0.27	0.49	0.49
0.4	∞	0.51	0.76	0.76	0.50	0.67	0.67	0.50	0.83	0.83
0.5	∞	1.02	1.34	1.34	1.00	1.00	1.00	1.00	1.50	1.50
0.6	∞	3.06	3.30	3.30	3.00	3.00	3.00	3.00	3.75	3.75

Table 8.4: Results for the fluid and quasi-stationary regimes (scenario II).

8.4 Comparison with static and flow-level load balancing

In the previous sections we considered load balancing at the packet-level. In this section we quantify the gain that can be achieved by using packet-level load balancing instead of static or flow-level load balancing. We consider the same parameter values as in the previous section (without considering fluid and quasi-stationary regimes), and calculate the mean number of users of each class under static and flow-level load balancing, so that we can make a comparison with packet-level load balancing.

8.4.1 BFS

When static or flow-level load balancing is used, which is based on BFS, we need to keep track of the number of class-0 users at node i, $i = 1, 2$. Let n_{0i} denote the number of class-0 users at node i, $i = 1, 2$. Then the balance function is given by [22]

$$\Phi(n) = \frac{\binom{n_{01}+n_1}{n_1}\binom{n_{02}+n_2}{n_2}}{c_1^{n_1+n_{01}}c_2^{n_2+n_{02}}},$$

and we obtain

$$\phi_{0i}(n) = \frac{n_{0i}}{n_{0i}+n_i}c_i; \quad \phi_i(n) = \frac{n_i}{n_{0i}+n_i}c_i, \quad i = 1, 2.$$

Hence, at both nodes capacity is shared according to egalitarian PS.

Considering the symmetric parameter setting of the previous section, the optimal static load balancing policy is to route class-0 arrivals to node i, $i = 1, 2$, with probability $1/2$. Using the parameter values of the previous section, we thus find that

γ	$\mathbb{E}N_0^{BFst}$	$\mathbb{E}N_1^{BFst}$	$\mathbb{E}N_2^{BFst}$	$\mathbb{E}N_0^{BFfl}$	$\mathbb{E}N_1^{BFfl}$	$\mathbb{E}N_2^{BFfl}$	$\mathbb{E}N_0^{BF}$	$\mathbb{E}N_1^{BF}$	$\mathbb{E}N_2^{BF}$
0.1	0.12	0.12	0.12	0.11	0.12	0.12	0.06	0.11	0.11
0.2	0.29	0.29	0.29	0.25	0.27	0.27	0.14	0.27	0.27
0.3	0.55	0.55	0.55	0.46	0.50	0.50	0.27	0.49	0.49
0.4	1.00	1.00	1.00	0.82	0.87	0.87	0.50	0.83	0.83
0.5	2.00	2.00	2.00	1.59	1.64	1.64	1.00	1.50	1.50
0.6	6.00	6.00	6.00	5.15	5.27	5.27	3.00	3.75	3.75

Table 8.5: Results for static, flow-level and packet-level load balancing in case of BFS.

class-i (class-0) users arrive according to a Poisson process of rate γ ($\gamma/2$) at node i, and both class-0 and class-i users have exponentially distributed service requirements with mean 1, $i = 1, 2$. Recalling that $c_i = 1$, $i = 1, 2$, and using that capacity is shared according to PS at both nodes, it is a straightforward exercise to show that

$$\mathbb{E}N_i^{BFst} := \frac{\gamma}{1 - \frac{3}{2}\gamma}, \quad i = 0, 1, 2.$$

In Table 8.5 we report $\mathbb{E}N_i^{BFst}$, $i = 0, 1, 2$, for different values of the load γ.

In case of flow-level load balancing it is optimal (under the current setting) to route class-0 users to node 1 if $n_{01} + n_1 < n_{02} + n_2$, and to node 2 if $n_{01} + n_1 > n_{02} + n_2$. If $n_{01} + n_1 = n_{02} + n_2$ then an arriving class-0 user is sent to node i with probability $1/2$, $i = 1, 2$. In other words, an arriving class-0 user should join the shortest queue, see [164]. Since no explicit expressions are known for the mean number of users $\mathbb{E}N_i^{BFfl}$ of class i, $i = 0, 1, 2$, under flow-level load balancing, we have performed simulation experiments to obtain these values. The results are also reported in Table 8.5.

Table 8.5 shows that packet-level load balancing outperforms both static and flow-level load balancing, and flow-level load balancing is better than static load balancing, as was expected, i.e., $\mathbb{E}N_i^{BF} \leq \mathbb{E}N_i^{BFfl} \leq \mathbb{E}N_i^{BFst}$, $i = 0, 1, 2$. For low values of γ (low loads), the results are quite similar, but for higher loads the differences become more significant. We note that these results are in line with the findings of [109].

8.4.2 AFS

In case static or flow-level load balancing is executed through AFS, we also need to be aware of the number of class-0 users at nodes 1 and 2. In case n_i class-i users and n_{0i} class-0 users are present at node i, the allocated service rates are

$$s_i^*(n) = \frac{\kappa_i^{1/\alpha} n_i c_i}{\kappa_0^{1/\alpha} n_{0i} + \kappa_i^{1/\alpha} n_i}, \quad s_{0i}^*(n) = \frac{\kappa_0^{1/\alpha} n_{0i} c_i}{\kappa_0^{1/\alpha} n_{0i} + \kappa_i^{1/\alpha} n_i}, \quad i = 1, 2.$$

Hence, capacity is shared according to DPS with relative weights $\kappa_0^{1/\alpha}$ and $\kappa_i^{1/\alpha}$ at node i, $i = 1, 2$.

Again, due to the symmetric parameter values, in case of static load balancing it is optimal to route class-0 arrivals to node i, $i = 1, 2$, with probability $1/2$. Using

γ	α	$\mathrm{E}N_0^{AFst}$	$\mathrm{E}N_1^{AFst}$	$\mathrm{E}N_2^{AFst}$	$\mathrm{E}N_0^{AFfl}$	$\mathrm{E}N_1^{AFfl}$	$\mathrm{E}N_2^{AFfl}$	$\mathrm{E}N_0^{AF}$	$\mathrm{E}N_1^{AF}$	$\mathrm{E}N_2^{AF}$
0.1	1	0.11	0.12	0.12	0.10	0.12	0.12	0.06	0.12	0.12
0.2	1	0.25	0.31	0.30	0.23	0.28	0.28	0.13	0.28	0.27
0.3	1	0.44	0.61	0.59	0.40	0.54	0.53	0.23	0.54	0.49
0.4	1	0.71	1.17	1.12	0.64	0.98	0.94	0.39	0.97	0.83
0.5	1	1.21	2.47	2.32	1.09	1.97	1.85	0.68	1.95	1.46
0.6	1	2.90	7.85	7.26	2.81	6.68	6.21	1.55	5.93	3.47
0.1	2	0.11	0.12	0.12	0.11	0.12	0.12	0.06	0.12	0.11
0.2	2	0.27	0.30	0.29	0.24	0.27	0.27	0.14	0.27	0.26
0.3	2	0.48	0.58	0.57	0.43	0.53	0.51	0.26	0.50	0.48
0.4	2	0.83	1.10	1.06	0.71	0.94	0.91	0.47	0.88	0.81
0.5	2	1.54	2.28	2.17	1.30	1.83	1.78	0.87	1.71	1.44
0.6	2	4.17	7.13	6.69	4.03	6.43	6.09	2.35	4.81	3.66
0.1	5	0.12	0.12	0.12	0.11	0.12	0.11	0.06	0.11	0.11
0.2	5	0.28	0.29	0.29	0.25	0.27	0.27	0.15	0.26	0.26
0.3	5	0.52	0.56	0.56	0.44	0.51	0.50	0.28	0.48	0.46
0.4	5	0.93	1.04	1.03	0.78	0.92	0.90	0.52	0.82	0.78
0.5	5	1.80	2.12	2.07	1.51	1.77	1.73	1.00	1.51	1.40
0.6	5	5.21	6.50	6.29	4.46	5.20	5.07	2.84	3.95	3.61
0.1	∞	0.12	0.12	0.12	0.11	0.12	0.12	0.06	0.11	0.11
0.2	∞	0.29	0.29	0.29	0.25	0.27	0.27	0.15	0.26	0.26
0.3	∞	0.55	0.55	0.55	0.46	0.50	0.50	0.30	0.46	0.46
0.4	∞	1.00	1.00	1.00	0.82	0.87	0.87	0.55	0.77	0.77
0.5	∞	2.00	2.00	2.00	1.59	1.64	1.64	1.10	1.39	1.39
0.6	∞	6.00	6.00	6.00	5.15	5.27	5.27	3.17	3.48	3.48

Table 8.6: Results for static, flow-level and packet-level load balancing in case of AFS (scenario II).

the parameter values of the previous section, we thus find that class-i (class-0) users arrive according to a Poisson process of rate γ ($\gamma/2$) at node i, and both class-0 and class-i users have exponentially distributed service requirements with mean 1, $i = 1, 2$. Using that $c_i = 1$, $i = 1, 2$, and that capacity is shared according to DPS at both nodes, the results of [63] imply that

$$\mathbb{E}N_0^{AFst} := \frac{\frac{1}{2}\gamma}{1-\frac{3}{2}\gamma}\left(2 + \frac{\gamma\left(\kappa_1^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_1^{1/\alpha}(1-\gamma)} + \frac{\gamma\left(\kappa_2^{1/\alpha} - \kappa_0^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_2^{1/\alpha}(1-\gamma)}\right);$$

$$\mathbb{E}N_i^{AFst} := \frac{\gamma}{1-\frac{3}{2}\gamma}\left(1 + \frac{\frac{1}{2}\gamma\left(\kappa_0^{1/\alpha} - \kappa_i^{1/\alpha}\right)}{\kappa_0^{1/\alpha}(1-\frac{1}{2}\gamma) + \kappa_i^{1/\alpha}(1-\gamma)}\right), \quad i = 1, 2.$$

Note that $\mathbb{E}N_i^{AFst} = \mathbb{E}N_i^{BFst}$, $i = 0, 1, 2$, in case of equal class weights. Therefore, we only focus on scenario II, and these results are shown in Table 8.6.

The optimal flow-level load balancing policy is as before to join the shortest queue, see [164]. As no explicit expressions for the mean number of users $\mathbb{E}N_i^{AFfl}$ of class i, $i = 0, 1, 2$, are available under flow-level load balancing, we resort to simulation experiments to obtain these values. Note that $\mathbb{E}N_i^{AFfl} = \mathbb{E}N_i^{BFfl}$, $i = 0, 1, 2$, in case of equal class weights, so we only report the results corresponding to scenario II, see Table 8.6.

Table 8.6 shows that packet-level load balancing performs better than both static and flow-level load balancing: $\mathbb{E}N_i^{AF} \leq \mathbb{E}N_i^{AFfl} \leq \mathbb{E}N_i^{AFst}$, $i = 0, 1, 2$. Again, the results seem to vary more for high values of γ.

Bibliography

[1] S. Aalto, U. Ayesta, S. Borst, V. Misra, R. Núñez Queija (2007). Beyond Processor Sharing. *ACM SIGMETRICS Performance Evaluation Review*, 34 (4): 36-43.

[2] J. Abate, W. Whitt (1987). Transient behavior of regulated Brownian motion, I: starting at the origin. *Advances in Applied Probability*, 19: 560-598.

[3] J. Abate, W. Whitt (1987). Transient behavior of regulated Brownian motion, II: non-zero initial conditions. *Advances in Applied Probability*, 19: 599-631.

[4] J. Abate, W. Whitt (1988). The correlation functions of RBM and M/M/1. *Stochastic Models*, 4: 315-359.

[5] M. Abramowitz, I.A. Stegun (1970). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*. Dover, New York, USA.

[6] A. de Acosta (1994). Large deviations for vector-valued Lévy processes. *Stochastic Processes and their Applications*, 51: 75-115.

[7] R. Addie, P. Mannersalo, I. Norros (2002). Most probable paths and performance formulae for buffers with Gaussian input traffic. *European Transactions on Telecommunications*, 13: 183-196.

[8] R. Adler (1990). *An introduction to continuity, extrema, and related topics for general Gaussian processes*. IMS Lecture Notes-Monograph Series, 12.

[9] E. Altman, K. Avrachenkov, U. Ayesta (2006). A survey on Discriminatory Processor Sharing. *Queueing Systems*, 53: 53-63.

[10] E. Altman, T. Jimenez, D. Kofman (2004). DPS queues with stationary ergodic service times and the performance of TCP in overload. In: *Proceedings of the IEEE INFOCOM Conference*, Hong-Kong, China, 975-983.

[11] D. Anick, D. Mitra, M. Sondhi (1982). Stochastic theory of a data handling system with multiple resources. *Bell System Technical Journal*, 61: 1871-1894.

[12] S. Asmussen (2003). *Applied probability and queues*. Springer-Verlag, New York, USA.

[13] K. Avrachenkov, U. Ayesta, P. Brown, R. Núñez Queija (2005). Discriminatory Processor Sharing revisited. In: *Proceedings of the IEEE INFOCOM Conference*, Miami, USA, 784-795.

[14] U. Ayesta, M. Mandjes (2008). Bandwidth-sharing networks under a diffusion scaling. To appear in: *Annals of Operations Research*.

[15] R. Bahadur, S. Zabell (1979). Large deviations of the sample mean in general vector spaces. *Annals of Probability*, 7: 587-621.

[16] F. Baskett, K.M. Chandy, R.R. Muntz, F. Palacios-Gomez (1975). Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22: 248-260.

[17] R. Bekker, M. Mandjes (2007). A fluid model for a relay node in an ad-hoc network: the case of heavy-tailed input. *CWI Report*, PNA-E0703.

[18] S. Ben Fredj, T. Bonald, A. Proutière, G. Régnié, J.W. Roberts (2001). Statistical bandwidth sharing: a study of congestion at the flow-level. In: *Proceedings of the ACM SIGCOMM Conference*, San Diego, USA, 111-122.

[19] J. Bertoin (1996). *Lévy processes*. Cambridge University Press, Cambridge, UK.

[20] D. Bertsimas, I. Paschalidis, J. Tsitsiklis (1999). Large deviations analysis of the Generalized Processor Sharing policy. *Queueing Systems*, 32: 319-349.

[21] J. Blanchet, P.W. Glynn, J.C. Liu (2006). State-dependent importance sampling and large deviations. In: *Proceedings of the VALUETOOLS Conference, Pisa, Italy*.

[22] T. Bonald, M. Jonckheere, A. Proutière (2004). Insensitive load balancing. *ACM SIGMETRICS Performance Evaluation Review*, 32 (1): 367-377.

[23] T. Bonald, L. Massoulié (2001). Impact of fairness on Internet performance. *ACM SIGMETRICS Performance Evaluation Review*, 29 (1): 82-91.

[24] T. Bonald, L. Massoulié, A. Proutière, J. Virtamo (2006). A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems*, 53: 65-84.

[25] T. Bonald, A. Proutière (2003). Insensitive bandwidth sharing in data networks. *Queueing Systems*, 44: 69-100.

[26] T. Bonald, A. Proutière (2004). On performance bounds for balanced fairness. *Performance Evaluation*, 55: 25-50.

[27] T. Bonald, A. Proutière, J.W. Roberts, J. Virtamo (2003). Computational aspects of balanced fairness. In: *Proceedings of the 18th International Teletraffic Congress*, Berlin, Germany, 801-810.

[28] T. Bonald, J.W. Roberts (2003). Congestion at flow level and the impact of user behaviour. *Computer Networks*, 42: 521-536.

[29] T. Bonald, J. Virtamo (2004). Calculating the flow level performance of balanced fairness in tree networks. *Performance Evaluation*, 58: 1-14.

[30] T. Bonald, J. Virtamo (2005). A recursive formula for multirate systems with elastic traffic. *IEEE Communications Letters*, 9: 753-755.

[31] T. Bonald, J. Virtamo (2006). On light and heavy traffic approximations of balanced fairness. *ACM SIGMETRICS Performance Evaluation Review*, 34 (1): 109-120.

[32] S. Borst, O. Boxma, P. Jelenković (2000). Asymptotic behavior of Generalized Processor Sharing with long-tailed traffic sources. In: *Proceedings of the IEEE INFOCOM Conference*, Tel-Aviv, Israel, 912-921.

[33] S. Borst, O. Boxma, P. Jelenković. (2003). Reduced-load equivalence and induced burstiness in GPS queues with long-tailed traffic flows. *Queueing Systems*, 43: 273-306.

[34] S. Borst, M. Mandjes, M. van Uitert. (2002). GPS queues with heterogeneous traffic classes. In: *Proceedings of the IEEE INFOCOM Conference*, New York, USA, 74-83.

[35] S. Borst, M. Mandjes, M. van Uitert (2003). Generalized Processor Sharing queues with light-tailed and heavy-tailed input. *IEEE/ACM Transactions on Networking*, 11: 821-834.

[36] S. Borst, R. Núñez Queija, A.P. Zwart (2006). Sojourn time asymptotics in Processor-Sharing queues. *Queueing Systems*, 53: 31-51.

[37] S. Borst, D.T.M.B. van Ooteghem, A.P. Zwart (2005). Tail asymptotics for Discriminatory Processor-Sharing queues with heavy-tailed service requirements. *Performance Evaluation*, 61: 281-298.

[38] O. Boxma, V. Dumas (1998). Fluid queues with heavy-tailed activity period distributions. *Computer Communications*, 21: 1509-1529.

[39] M. Bramson (2005). Stability of networks for max-min fair routing. Presentation at the 13th INFORMS Applied Probability Conference, Ottawa, Canada.

[40] T. Bu, D. Towsley (2001). Fixed point approximation for TCP behaviour in an AQM network. *ACM SIGMETRICS Performance Evaluation Review*, 29 (1): 216-225.

[41] J. Bucklew (1990). *Large deviation techniques in decision, simulation and estimation*. Wiley, New York, USA.

[42] S.K. Cheung, J.L. van den Berg, R.J. Boucherie, R. Litjens, F. Roijers (2005). An analytical packet/flow-level modelling approach for wireless LANs with Quality-of-Service support. In: *Proceedings of the 19th International Teletraffic Congress*, Beijing, China, 1651-1662.

[43] J.W. Cohen (1974). Superimposed renewal processes and storage with gradual input. *Stochastic Processes and their Applications*, 2: 31-58.

[44] J.W. Cohen (1982). *The single server queue*. North-Holland, Amsterdam, The Netherlands.

[45] J.W. Cohen, O. Boxma (1983). *Boundary value problems*. North-Holland, Amsterdam, The Netherlands.

[46] M. Crovella, A. Bestavros (1997). Self-similarity in World Wide Web Traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5: 362-373.

[47] M. Crovella, M.S. Taqqu, A. Bestavros (1998). *Heavy tails in the World Wide Web. A practical guide to heavy tails*. Birkhäuser, Boston, USA.

[48] K. Dębicki, A. Dieker, T. Rolski (2007). Quasi-product form for Lévy-driven fluid networks. *Mathematics of Operations Research*, 32: 629-647.

[49] K. Dębicki, M. Mandjes (2007). A note on large-buffer asymptotics for Generalized Processor Sharing with Gaussian inputs. *Queueing Systems*, 55: 251-254.

[50] K. Dębicki, M. Mandjes, M. van Uitert (2007). A tandem queue with Lévy input: a new representation of the downstream queue length. *Probability in the Engineering and Informational Sciences*, 21: 83-107.

[51] K. Dębicki, M. van Uitert (2006). Large buffer asymptotics for Generalized Processor Sharing queues with Gaussian inputs. *Queueing Systems*, 54: 111-120.

[52] A. Dembo, O. Zeitouni (1998). *Large deviations techniques and applications*. Springer Verlag, New York.

[53] J.-D. Deuschel, D. Stroock (1989). *Large deviations*. Academic Press, London.

[54] A.B. Dieker (2006). *Extremes and fluid queues*. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands.

[55] A.B. Dieker, M. Mandjes (2006). Fast simulation of overflow probabilities in a queue with Gaussian input. *ACM Transactions on Modeling and Computer Simulation*, 16: 119-151.

[56] N. Dukkipati, J. Kuri, H. Jamadagni (2001). Optimal call admission control for Generalized Processor Sharing (GPS) schedulers. In: *Proceedings of the IEEE INFOCOM Conference*, Anchorage, USA, 468-477.

[57] P. Dupuis, H. Wang (2005). Dynamic importance sampling for uniformly recurrent Markov chains. *Annals of Applied Probability*, 15: 1-38.

[58] R. Egorova, S. Borst, A.P. Zwart (2007). Bandwidth-sharing networks in overload. *Performance Evaluation*, 64: 978-993.

[59] A. Elwalid, D. Mitra (1995). Analysis, approximations and admission control of a multi-service multiplexing system with priorities. In: *Proceedings of the IEEE INFOCOM Conference*, Boston, USA, 463-472.

[60] A. Elwalid, D. Mitra (1999). Design of Generalized Processor Sharing schedulers which statistically multiplex heterogeneous QoS classes. In: *Proceedings of the IEEE INFOCOM Conference*, New York, USA, 1220-1230.

[61] A. Es-Saghouani, M. Mandjes (2007). On the correlation structure of a Lévy-driven queue. *CWI Report*, PNA-R0711.

[62] G. Fayolle, A. de La Fortelle, J.M. Lasgouttes, L. Massoulié, J.W. Roberts (2001). Best-effort networks: modeling and performance analysis via large networks asymptotics. In: *Proceedings of the IEEE INFOCOM Conference*, Anchorage, USA, 709-716.

[63] G. Fayolle, I. Mitrani, R. Iasnogorodski (1980). Sharing a processor among many job classes. *Journal of the ACM*, 27: 519-532.

[64] C. Fraleigh, F. Tobagi, C. Diot (2003). Provisioning IP backbone networks to support latency sensitive traffic. In: *Proceedings of the IEEE INFOCOM Conference*, San Francisco, USA, 375-385.

[65] A. Ganesh, N. O'Connell, D. Wischik (2004). *Big queues*. Springer Lecture Notes in Mathematics, 1838.

[66] S. Giordano, M. Pagano, S. Tartarelli (2002). An importance sampling algorithm for the simulation of a GPS scheduler. *European Transactions on Telecommunication Systems*, special issue on rare event simulation, 13: 351-361.

[67] A. Goyal, P. Shahabuddin, P. Heidelberger, V. Nicola, P.W. Glynn (1992). A unified framework for simulating Markovian models of highly dependable systems. *IEEE Transactions on Computers*, 41: 36-51.

[68] S.A. Grishechkin (1992). On a relationship between Processor-Sharing queues and Crump-Mode-Jagers branching processes. *Advances in Applied Probability*, 24: 653-698.

[69] H.C. Gromoll, R.J. Williams (2008). Fluid limit of a network with fair bandwidth sharing and general document size distribution. To appear in: *Annals of Applied Probability*.

[70] H.C. Gromoll, R.J. Williams (2008). Fluid model for a data network with alpha-fair bandwidth sharing and general document size distributions: two examples of stability. To appear in: *IMS Festschrift volume in honor of Tom Kurtz*.

[71] F. Guillemin, A. Dupuis (1999). Simulation-based analysis of Weighted Fair Queueing algorithms for ATM networks. *Telecommunication Systems*, 12: 149-166.

[72] F. Guillemin, R. Mazumdar, A. Dupuis, J. Boyer (2003). Analysis of the fluid Weighted Fair Queueing system. *Journal of Applied Probability*, 40: 180-199.

[73] L. Guo, I. Matta (2001). The war between mice and elephants. In: *Proceedings of the 9th IEEE International Conference on Network Protocols*, Riverside, USA, 180-188.

[74] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley (2006). Multi-path TCP: a joint congestion control and routing scheme to exploit path diversity in the Internet. *IEEE/ACM Transactions on Networking*, 14: 1260-1271.

[75] J.M. Harrison (1985). *Brownian motion and stochastic flow systems*. Wiley, New York.

[76] J.M. Harrison, R.J. Williams (1987). Multidimensional reflected Brownian motions having exponential stationary distributions. *Annals of Probability*, 15: 115-137.

[77] J.M. Harrison, R.J. Williams (1992). Brownian models of feedforward queueing networks: quasireversibility and product form solutions. *Annals of Applied Probability*, 2: 263-293.

[78] M. Haviv, J. van der Wal (2008). Mean sojourn times for phase-type Discriminatory Processor Sharing systems. *European Journal of Operational Research*, 189: 375-386.

[79] D. Heath, S.I. Resnick, G. Samorodnitsky (1998). Heavy tails and long range dependence in on/off processes and associated fluid models. *Mathematics of Operations Research*, 23: 145-165.

[80] P. Heidelberger (1995). Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5: 43-85.

[81] D.P. Heyman, T.V. Lakshman, A.L. Neidhardt (1997). A new method for analysing feedback-based protocols with applications to engineering Web traffic over the Internet. *ACM SIGMETRICS Performance Evaluation Review*, 25 (1): 24-38.

[82] J. Hui (1988). Resource allocation for broadband networks. *IEEE Journal on Selected Areas in Communications*, 6: 1598-1608.

[83] V. Jacobson (1988). Congestion avoidance and control. In: *Proceedings of the ACM SIGCOMM Conference*, Stanford, USA, 314-329.

[84] M. Jonckheere, J. Virtamo (2005). Optimal insensitive routing and bandwidth sharing in simple data networks. *ACM SIGMETRICS Performance Evaluation Review*, 33 (1): 193-204.

[85] W. Kang, F.P. Kelly, N. Lee, R.J. Williams (2004). Fluid and Brownian approximations for an Internet congestion control model. In: *Proceedings of the 43rd IEEE Conference on Decision and Control*, The Bahamas, 3938-3943.

[86] O. Kella (1993). Parallel and tandem fluid networks with dependent Lévy inputs. *Annals of Applied Probability*, 3: 682-695.

[87] O. Kella, W. Whitt (1992). A tandem fluid network with Lévy input. *Queueing and Related Models*. U.N. Bhat and I.V. Basawa, editors. Oxford University Press, 112-128.

[88] F.P. Kelly (1997). Charging and rate control for elastic traffic (corrected version). *European Transactions on Telecommunications*, 8: 33-37.

[89] F.P. Kelly, A. Maulloo, D. Tan (1998). Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49: 237-252.

[90] F.P. Kelly, T. Voice (2005). Stability of end-to-end algorithms for joint routing and rate control. In: *Proceedings of the ACM SIGCOMM Conference*, Philadelphia, USA, 5-12.

[91] F.P. Kelly, R.J. Williams (2004). Fluid model for a network operating under a fair bandwidth-sharing policy. *Annals of Applied Probability*, 14: 1055-1083.

[92] D.G. Kendall (1953). Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov Chain. *Annals of Mathematical Statistics*, 24: 338-354.

[93] G. van Kessel, R. Núñez Queija, S. Borst (2004). Asymptotic regimes and approximations for Discriminatory Processor Sharing. *ACM SIGMETRICS Performance Evaluation Review*, 32 (2): 44-46.

[94] G. van Kessel, R. Núñez Queija, S. Borst (2005). Differentiated bandwidth sharing with disparate flow sizes. In: *Proceedings of the IEEE INFOCOM Conference*, Miami, USA, 2425-2435.

[95] P. Key, L. Massoulié, D. Towsley (2006). Combining multipath routing and congestion control for robustness. In: *Proceedings of the Conference on Information Sciences and Systems*, Princeton, USA, 345-350.

[96] P. Key, L. Massoulié, D. Towsley (2007). Path selection and multipath congestion control. In: *Proceedings of the IEEE INFOCOM Conference*, Anchorage, USA, 143-151.

[97] J. Kilpi, I. Norros (2002). Testing the Gaussian approximation of aggregate traffic. In: *Proceedings of the Internet Measurement Workshop*, Marseille, France, 49-61.

[98] J. Kim, B. Kim (2004). Sojourn time distribution in the M/M/1 queue with Discriminatory Processor-Sharing. *Performance Evaluation*, 58: 341-365.

[99] L. Kleinrock (1967). Time-shared systems: a theoretical treatment. *Journal of the ACM*, 14: 242-261.

[100] L. Kleinrock (1975). *Queueing systems*, Volume I. Wiley, New York, USA.

[101] L. Kleinrock (1976). *Queueing systems*, Volume II. Wiley, New York, USA.

[102] L. Kosten (1974). Stochastic theory of a multi-entry buffer, part 1. *Delft Progress Report*, Series F 1: 10-18.

[103] C. Kotopoulos, N. Likhanov, R. Mazumdar (2001). Asymptotic analysis of the GPS system fed by heterogeneous long-tailed sources. In: *Proceedings of the IEEE INFOCOM Conference*, Anchorage, USA, 299-308.

[104] H.W. Kuhn, A.W. Tucker (1951). Nonlinear programming. In: *Proceedings of the Berkeley Symposium*, Berkeley, USA, 481-492.

[105] K. Kumaran, G. Margrave, D. Mitra, K. Stanley (2000). Novel techniques for the design and control of Generalized Processor Sharing schedulers for multiple QoS classes. In: *Proceedings of the IEEE INFOCOM Conference*, Tel-Aviv, Israel, 932-941.

[106] J.F. Kurose, K.W. Ross (2003). *Computer networking: a top-down approach featuring the Internet*. Addison-Wesley, USA.

[107] R.J. La, V. Anantharam (2002). Utility-based rate control in the Internet for elastic traffic. *IEEE/ACM Transactions on Networking*, 10: 272-286.

[108] J. Leino, J. Virtamo (2005). Insensitive traffic splitting in data networks. In: *Proceedings of the 19th International Teletraffic Congress*, Beijing, China, 1355-1364.

[109] J. Leino, J. Virtamo (2006). Insensitive load balancing in data networks. *Computer Networks*, 50: 1059-1068.

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

[111] L. Leskelä (2006). Stabilization of an overloaded queueing network using measurement-based admission control. *Journal of Applied Probability*, 43: 231-244.

[112] P. Lieshout (2007). Traffic-splitting networks operating under alpha-fair sharing policies and balanced fairness. In: *Proceedings of the Workshop on IP QoS and Traffic Control*, Lisbon, Portugal, 17-24.

[113] P. Lieshout, S. Borst, M. Mandjes (2006). Heavy-traffic approximations for linear networks operating under alpha-fair bandwidth-sharing policies. In: *Proceedings of the VALUETOOLS Conference*, Pisa, Italy.

[114] P. Lieshout, M. Mandjes (2007). Tandem Brownian queues. *Mathematical Methods in Operations Research*, 66: 275-298.

[115] P. Lieshout, M. Mandjes (2007). Transient analysis of Brownian queues. *CWI Report*, PNA-R0705.

[116] P. Lieshout, M. Mandjes (2008). A note on the delay in Generalized Processor Sharing. *Operations Research Letters*, 36: 117-122.

[117] P. Lieshout, M. Mandjes (2008). Asymptotic analysis of Lévy-driven tandem queues. *CWI Report*, PNA-R0809.

[118] P. Lieshout, M. Mandjes (2008). Generalized Processor Sharing: characterization of the admissible region and selection of optimal weights. *Computers and Operations Research*, special issue on queues in practice, 35: 2497-2519.

[119] P. Lieshout, M. Mandjes (2008). Importance sampling in rate-sharing networks. In: *Proceedings of the SIMUTOOLS Conference*, Marseille, France.

[120] P. Lieshout, M. Mandjes, S. Borst (2006). GPS scheduling: selection of optimal weights and comparison with strict priorities. *ACM Performance Evaluation Review*, 34 (1): 75-86. (Best student paper award)

[121] X. Lin, N.B. Shroff (2006). Utility maximization for communication networks with multipath routing. *IEEE Transactions on Automatic Control*, 51: 766-781.

[122] K. Majewski (1998). Heavy traffic approximations of large deviations of feedforward queueing networks. *Queueing Systems*, 28: 125-155.

[123] M. Mandjes (1999). Rare event analysis of the state frequencies of a large number of Markov chains. *Stochastic Models*, 15: 577-592.

[124] M. Mandjes (2004). A note on the benefits of buffering. *Stochastic Models*, 20: 43-54.

[125] M. Mandjes (2004). Packet models revisited: tandem and priority systems. *Queueing Systems*, 47: 363-377.

[126] M. Mandjes (2005). Large deviations for complex buffer architectures: the short-range dependent case. *Stochastic Models*, 22: 99-128.

[127] M. Mandjes (2007). *Large deviations for Gaussian queues: modelling communication networks.* Wiley, Chichester, UK.

[128] M. Mandjes, P. Mannersalo, and I. Norros (2005). Priority queues with Gaussian input: a path-space approach to loss and delay asymptotics. In: *Proceedings of the 19th International Teletraffic Congress*, Beijing, China, 1135-1144.

[129] M. Mandjes, M. van Uitert (2005). Sample-path large deviations for Generalized Processor Sharing queues with Gaussian inputs. *Performance Evaluation*, 61: 225-256.

[130] M. Mandjes, M. van Uitert (2005). Sample-path large deviations for tandem and priority queues with Gaussian inputs. *Annals of Applied Probability*, 15: 1193-1226.

[131] P. Mannersalo, I. Norros (2001). Approximate formulae for Gaussian priority queues. In: *Proceedings of the 17th International Teletraffic Congress*, Salvador da Bahia, Brazil, 991-1002.

[132] P. Mannersalo, I. Norros (2002). GPS schedulers and Gaussian traffic. In: *Proceedings of the IEEE INFOCOM Conference*, New York, USA, 1660-1667.

[133] L. Massoulié (1999). Large deviations estimates for polling and Weighted Fair Queueing service systems. *Advances in Performance Analysis*, 2: 103-128.

[134] L. Massoulié (2007). Structural properties of proportional fairness: stability and insensitivity. *Annals of Applied Probability*, 17: 809-839.

[135] L. Massoulié, J.W. Roberts (2000). Bandwidth sharing and admission control for elastic traffic. *Telecommunication Systems*, 15: 185-201.

[136] L. Massoulié, J.W. Roberts (2002). Bandwidth sharing: objectives and algorithms. *IEEE/ACM Transactions on Networking*, 10: 320-328.

[137] H.P. McKean (1969). *Stochastic integrals*. Academic Press, New York, USA.

[138] R. van de Meent, M. Mandjes, A. Pras (2006). Gaussian traffic everywhere? In: *Proceedings of the IEEE International Conference on Communications*, Istanbul, Turkey, 573-578.

[139] V. Misra, W. Gong, D. Towsley (1999). Stochastic differential equation modeling and analysis of TCP-windowsize behavior. In: *Proceedings of the IFIP Performance Conference*, Istanbul, Turkey.

[140] J. Mo, J. Walrand (2000). Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8: 556-567.

[141] M. Nabe, M. Murata, M. Miyahara (1998). Analysis and modeling of World Wide Web traffic for capacity dimensioning of Internet access lines. *Performance Evaluation*, 34: 249-271.

[142] I. Norros (1999). Busy periods of fractional Brownian storage: a large deviations approach. *Advances in Performance Analysis*, 2: 1-20.

[143] R. Núñez Queija (2000). *Processor-Sharing models for integrated-service networks*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

[144] T.M. O'Donovan (1974). Direct solutions of M/G/1 Processor Sharing models. *Operations Research*, 22: 1232-1235.

[145] T. Osogami, M. Harchol-Balter, A. Scheller-Wolf (2005). Analysis of cycle stealing with switching times and thresholds. *Performance Evaluation*, 61: 347-369.

[146] J. Padhye, V. Firoiu, D. Towsley, J. Kurose (2000). Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8: 133-145.

[147] A. Panagakis, N. Dukkipati, I. Stavrakakis, J. Kuri (2004). Optimal call admission control on a single link with a GPS scheduler. *IEEE/ACM Transactions on Networking*, 12: 865-878.

[148] A. Parekh, R. Gallager (1993). A Generalized Processor Sharing approach to flow control in integrated services networks: the single node case. *IEEE/ACM Transactions on Networking*, 1: 344-357.

[149] A. Parekh, R. Gallager (1994). A Generalized Processor Sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2: 137-150.

[150] K. Park, W. Willinger (Editors) (2000). *Self-similar network traffic and performance evaluation*. Wiley, New York, USA.

[151] I. Paschalidis (1999). Class-specific Quality-of-Service guarantees in multimedia communication networks. *Automatica*, special issue on control methods for communication networks, 35: 1951-1969.

[152] V. Paxson, S. Floyd (1995). Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3: 226-244.

[153] F.M. Pereira, N.L.S. Fonseca, D.S. Arantes (2002). On the performance of Generalized Processor Sharing servers under long-range dependent traffic. *Computer Networks*, 40: 413-431.

[154] K.M. Rege, B. Sengupta (1996). Queue length distribution for the Discriminatory Processor-Sharing queue. *Operations Research*, 44: 653-657.

[155] E. Reich (1958). On the integrodifferential equation of Takács I. *Annals of Mathematical Statistics*, 29: 563–570.

[156] S.I. Resnick (1992). *Adventures in stochastic processes*. Birkhäuser, Boston, USA.

[157] J.W. Roberts (2001). Traffic theory and the Internet. *IEEE Communications Magazine*, 39 (1): 94-99.

[158] J.W. Roberts (2004). A survey on statistical bandwidth sharing. *Computer Networks*, 45: 319-332.

[159] J.W. Roberts, U. Mocci, J. Virtamo (1996). *Broadband network traffic - Final report of COST action 242*. Springer, Berlin, Germany.

[160] W.R.W. Scheinhardt (1998). *Markov-modulated and feedback fluid queue*. PhD thesis, University of Twente, Enschede, The Netherlands.

[161] A. Shwartz, A. Weiss (1995). *Large deviations for performance analysis: queues, communication and computing*. Chapman & Hall, London, UK.

[162] K. Sriram, W. Whitt (1986). Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, 4: 833-846.

[163] H.C. Tijms (1994). *Stochastic models: an algorithmic approach*. Wiley, Chichester, UK.

[164] D. Towsley, P.D. Sparaggis, C.G. Cassandras (1992). Optimal routing and buffer allocation for a class of finite capacity queueing systems. *IEEE Transactions on Automatic Control*, 37: 1446-1451.

[165] M. van Uitert (2003). *Generalized Processor Sharing queues*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

[166] M. van Uitert, S. Borst (2001). Generalised Processor Sharing networks fed by heavy-tailed traffic flows. In: *Proceedings of the IEEE INFOCOM Conference*, Anchorage, USA, 269-278.

[167] M. van Uitert, S. Borst (2002). A reduced-load equivalence for Generalised Processor Sharing networks with long-tailed traffic flows. *Queueing Systems*, 41: 123-163.

[168] S. Varadhan, R.J. Williams (1985). Brownian motion in a wedge with oblique reflection. *Communications on Pure and Applied Mathematics*, 38: 405-443.

[169] G. de Veciana, T.J. Lee, T. Konstantopoulos (2001). Stability and performance analysis of networks supporting elastic services. *IEEE/ACM Transactions on Networking*, 9: 2-14.

[170] T. Voice (2006). Stability of multi-path dual congestion control algorithms. In: *Proceedings of the VALUETOOLS Conference*, Pisa, Italy.

[171] J. Walrand, P. Varaiya (2001). *High-performance communication networks*. Morgan Kaufmann, San Francisco, USA.

[172] W.H. Wang, M. Palaniswami, S.H. Low (2003). Optimal flow control and routing in multi-path networks. *Performance Evaluation*, 52: 119-132.

[173] W. Whitt (1986). Deciding which queue to join: some counterexamples. *Operations Research*, 34: 226-244.

[174] W. Willinger, M.S. Taqqu, R. Sherman, D.V. Wilson (1997). Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5: 71-86.

[175] O. Yaron, M. Sidi (1994). Generalized Processor Sharing networks with exponentially bounded burstiness arrivals. *Journal of High Speed Networks*, 3: 375-387.

[176] H.Q. Ye (2003). Stability of data networks under an optimization-based bandwidth allocation. *IEEE Transaction on Automatic Control*, 48: 1238-1242.

[177] Z.-L. Zhang (1997). Large deviations and the Generalized Processor Sharing scheduling for a two-queue system. *Queueing Systems*, 26: 229-245.

[178] Z.-L. Zhang (1998). Large deviations and the Generalized Processor Sharing scheduling for a multiple-queue system. *Queueing Systems*, 28: 349-376.

[179] Z.-L. Zhang, D. Towsley, J. Kurose (1995). Statistical analysis of the Generalized Processor Sharing scheduling discipline. *IEEE Journal of Selected Areas in Communications*, 13: 1071-1080.

[180] A.P. Zwart (2001). *Queueing systems with heavy tails*. PhD thesis, Eindhoven University of Technology, Eindhoven, The Netherlands.

[181] A.P. Zwart, S. Borst, M. Mandjes (2004). Exact asymptotics for fluid queues fed by multiple heavy-tailed on-off sources. *Annals of Applied Probability*, 14: 903-957.

Summary

Modern communication networks aim to support a wide range of heterogeneous services, including data, video, and voice-applications, but also more demanding multimedia applications, such as gaming, video-conferencing, etc. In order to accomplish this, it is important that the traffic that is generated by these applications is properly served, in particular by sharing the available service capacity in a suitable manner among the various traffic classes.

In this monograph we analyze mathematical models for bandwidth sharing in such multi-service networks. It is important to distinguish between i) explicit scheduling in network nodes, and ii) bandwidth sharing as a consequence of the end-to-end rate control by end-users. For both cases, various bandwidth-sharing disciplines can be identified for either implementing or modeling bandwidth sharing.

Note that a communication network can be regarded as a system where customers arrive, possibly wait for their service, and leave after they have been served. Both the times at which customers arrive and the corresponding service requirements are stochastic in nature. Hence, it is natural to view a communication network as a queueing system. In this thesis we therefore apply queueing theory as a tool to analyze the performance of several bandwidth-sharing mechanisms.

This thesis consists of two parts, preceded by an introductory chapter. Part I is devoted to case i) mentioned above, whereas Part II considers case ii).

In Part I, consisting of Chapters 2-5, our goal is to study the performance of a mechanism that can implement differentiated sharing in a network node. In this part we assume that traffic can be modeled as a continuous fluid flow. We consider systems with Gaussian inputs, which provide a general and versatile class of fluid input processes, covering a broad range of correlation structures.

In Chapter 2 we first present the machinery that will be used in Part I. The use of this machinery is illustrated for a single queue with Brownian input (a special case of Gaussian input). We determine the joint distribution function of the workloads at two different times, which also allows us to calculate their correlation coefficient.

In the next chapter we analyze simple networks of Brownian queues, namely: a two-node parallel queue and a two-node tandem queue. For both systems, we derive the joint distribution function of the workloads of the first and second queue. We

also analyze a two-class priority queue, in which the low-priority class is only served if there is no backlog of high-priority traffic.

Chapter 4 considers a single node that serves two traffic classes, each having a different Gaussian input stream. We assume that capacity is allocated to the two classes according to the Generalized Processor Sharing (GPS) discipline. The GPS mechanism works as follows. Each class is assigned a weight, and this weight determines a guaranteed service rate for that class. In case a class does not fully use its minimum rate, the excess rate becomes available to the other class. Assigning all weight to a single class, implies that the other class can only be served if there is no traffic of this single class queued. Thus, priority queueing can be regarded as special case of GPS. We focus on the probability that the virtual delay of a particular class exceeds some threshold. In particular, we derive the delay asymptotics, and show that, depending on the GPS weights, three kinds of asymptotics appear.

In the last chapter of Part I we again study the system of Chapter 4. In this chapter, we focus on the problem of selecting GPS weights that maximize the traffic-carrying capacity. The results suggest that the weight-setting is not so crucial, and that simple priority strategies may suffice for practical purposes.

Part II, consisting of Chapters 6-8, considers bandwidth sharing as a consequence of the rate control by end-users. In that case the bandwidth shares are strongly affected by the protocol that governs the transfer of packets through the network. At large time scales, we consider the sequence of all packets from the beginning of a transfer until the end as a single flow. In particular, in Part II we deal with elastic flows, which are produced by the transfer of Web pages, e-mails, etc., and are characterized by a transmission rate that is continuously adapted over time, based on the level of congestion in the network. The above scenario can be modeled by assuming that bandwidth is shared according to an Alpha-Fair Sharing (AFS) policy, which covers a broad range of sharing policies. We assume that flows arrive according to a Poisson process, and have exponentially distributed service requirements.

In Chapter 6 we consider a general AFS network topology and focus on the probability that, conditional on the network population being in a given state a time zero, the network is in some other set of states after some predefined time. In particular, we assume that the underlying event is rare, so that the corresponding probability is small. We devise an Importance Sampling algorithm, i.e., an algorithm that can be used to simulate the system with new interarrival and service time distributions, in order to efficiently obtain an unbiased estimate for the probability of interest.

In the next chapter we analyze a linear network that operates under an AFS policy. In this system there is one class that requires service at all nodes simultaneously, whereas the other classes only require service at a single node. We derive approximations for the mean number of active users of each class, by assuming that one or two of the nodes are heavily loaded.

In the last chapter we consider a network in which, besides classes that use specific

routes, one class of users can split its traffic over several routes. We consider two different load balancing policies, and compare the performance of the network under these two policies.

Samenvatting

Moderne communicatienetwerken beogen gelijktijdig verschillende soorten applicaties te ondersteunen. Hierbij kan men denken aan standaard data-, video- en voice-applicaties, maar daarnaast ook aan multimedia applicaties zoals bijvoorbeeld gaming, video-conferencing, etc. Om dit te realiseren, is het belangrijk dat het verkeer dat door deze applicaties wordt geproduceerd op een juiste manier wordt afgehandeld, in het bijzonder door de beschikbare servercapaciteit op een geschikte wijze te verdelen over de verschillende verkeersklassen.

In dit proefschrift bestuderen we wiskundige modellen voor het verdelen van de servercapaciteit in zulke multi-service netwerken. Hierbij is het belangrijk om onderscheid te maken tussen i) het expliciet toewijzen van capaciteit in netwerk knooppunten, en ii) het delen van capaciteit als gevolg van de regulering van de transmissiesnelheid door eindgebruikers. Voor beide scenario's zijn al verschillende mechanismen voorgesteld om het delen van capaciteit te implementeren dan wel te modelleren.

Merk op dat een communicatienetwerk gezien kan worden als een systeem waar klanten aankomen, eventueel wachten op hun bediening, en vertrekken nadat ze zijn geholpen. Zowel de aankomstmomenten als de hoeveelheden werk die de klanten meebrengen hebben een stochastisch karakter. Dit impliceert dat een communicatienetwerk gezien kan worden als een wachtrijsysteem. In dit proefschrift passen we daarom wachtrijtheorie toe als hulpmiddel om de prestatie van verschillende mechanismen te analyseren.

Dit proefschrift omvat twee delen, voorafgegaan door een inleidend hoofdstuk van algemene aard. In deel I richten we ons op bovengenoemd scenario i), terwijl in deel II scenario ii) aan de orde komt.

In deel I, dat hoofdstuk 2 tot en met 5 omvat, is het doel om te onderzoeken hoe goed een bepaald mechanisme werkt dat service differentiatie kan bewerkstelligen in een netwerk knooppunt. In dit deel veronderstellen we dat het gegenereerde verkeer als een continue vloeistofstroom kan worden opgevat. We beschouwen systemen met Gaussische inputstromen; deze klasse van modellen omvat een breed scala aan correlatiestructuren.

In hoofdstuk 2 beschrijven we eerst de belangrijkste technieken die worden gebruikt in deel I van het proefschrift. Daarna passen we deze technieken ter illustratie toe op

een systeem dat bestaat uit een enkele server met een Brownse inputstroom. Brownse inputstromen zijn een speciaal geval van Gaussische inputstromen. We bepalen eerst de simultane verdeling van de bufferinhoud op twee verschillende tijdstippen. Aan de hand hiervan leiden we ook een exacte uitdrukking af voor de correlatie van de bufferinhoud op deze twee tijdstippen.

Het volgende hoofdstuk richt zich op eenvoudige netwerken met een Brownse inputstroom, zoals een parallel systeem en een tandem systeem, beide met twee servers. Voor beide systemen bepalen we de simultane verdeling van de inhoud van de eerste en tweede buffer op een bepaald tijdstip. We analyseren daarnaast ook een systeem dat bestaat uit een enkele server, maar dat nu twee verschillende soorten types verkeer bedient, die elk een verschillende Brownse inputstroom hebben. We veronderstellen hier dat de ene klasse prioriteit heeft boven de andere. Dat wil zeggen dat de klasse met de laagste prioriteit alleen kan worden bediend als er geen verkeer van de klasse met hogere prioriteit is.

Hoofdstuk 4 beschouwt een enkele server, die twee verschillende soorten verkeer bedient, elk met een verschillende Gaussische inputstroom. De capaciteit van de server wordt verdeeld over de twee klassen door middel van de Generalized Processor Sharing (GPS) discipline. Bij GPS wordt aan elke klasse een gewicht toegewezen. Dit gewicht bepaalt de fractie van de capaciteit van de server die gegarandeerd beschikbaar is voor de betreffende klasse. Wanneer een klasse zijn gegarandeerde capaciteit niet volledig gebruikt, dan komt het teveel aan capaciteit beschikbaar voor de andere klasse. Merk op dat het prioriteitsysteem een speciaal geval is van het GPS systeem (wanneer men al het gewicht aan één klasse toekent). We bepalen de asymptotiek van de kans dat de wachttijd van een bepaalde klasse een drempelwaarde overschrijdt, en laten zien dat drie verschillende regimes van asymptotiek bestaan.

In het laatste hoofdstuk van deel I wordt wederom het systeem van hoofdstuk 4 beschouwd. In dit hoofdstuk wordt onderzocht hoe men de GPS gewichten optimaal kan toewijzen aan de twee klassen, zodanig dat het systeem in staat is om zoveel mogelijk verkeer te ondersteunen. De verrassende conclusie is dat het niet zoveel uitmaakt hoe men de gewichten toekent, en dat men kan volstaan met het toekennen van het volledige gewicht aan één bepaalde klasse, wat overeenkomt met het bovengenoemde prioriteitsysteem.

Deel II, dat hoofdstuk 6 tot en met 8 omvat, beschouwt, in tegenstelling tot deel I, het delen van capaciteit als gevolg van de regulering van de transmissiesnelheid door eindgebruikers. In dat geval worden de toewijzingen sterk beïnvloed door een protocol dat het versturen van datapakketjes door het netwerk reguleert. Op grote tijdschaal kan men een stroom van kleine datapakketjes zien als een individuele klant, ook wel een flow genoemd. In het bijzonder richten we ons op elastische flows. Deze worden geproduceerd door het transport van webpagina's, e-mails, etc., en zijn gekarakteriseerd door een transmissiesnelheid die fluctueert over de tijd. Het bovenstaande scenario kan goed gemodelleerd worden door aan te nemen dat de capaciteit

wordt verdeeld volgens een Alpha-Fair Sharing (AFS) strategie, hetgeen een groot aantal verschillende vormen om capaciteit te delen omvat. In de verschillende modellen die worden behandeld in dit deel, nemen we aan dat flows aankomen volgens een Poissonproces, en dat ze exponentieel verdeelde bedieningstijden hebben.

In hoofdstuk 6 beschouwen we een netwerk van algemene topologie die verschillende soorten klanten bedient, waarin de capaciteit van de servers wordt verdeeld volgens een AFS strategie. We richten ons op de kans dat de netwerkpopulatie zich op een zeker tijdstip in een bepaalde toestand bevindt, gegeven dat de netwerkpopulatie zich op een eerder tijdstip in een bepaalde andere toestand bevindt. We veronderstellen dat dit tijdstip en deze toestanden zo zijn dat deze kans erg klein is. We leiden vervolgens een Importance Sampling algoritme af, d.w.z. een algoritme waarmee het systeem wordt gesimuleerd met andere verdelingen voor de tussenaankomsttijden en bedieningstijden van de verschillende klassen, waarmee snel een zuivere schatter met lage variantie voor de betreffende kans wordt verkregen.

Hoofdstuk 7 richt zich op een lineair netwerk, waarin de capaciteit van de servers wederom wordt verdeeld volgens een AFS strategie. In dit systeem is er één klasse die gelijktijdig door alle servers moet worden bediend, terwijl alle andere klassen maar bediend hoeven te worden door een enkele server. We leiden benaderingen af voor het gemiddelde aantal aanwezige klanten van elke klasse, door te veronderstellen dat één of twee van de servers in het netwerk zwaar belast zijn.

In het laatste hoofdstuk beschouwen we een netwerk, waarin één klasse zijn verkeer kan splitsen over verschillende routes, terwijl alle andere klassen specifieke routes gebruiken. We beschouwen twee methodes om dit te bewerkstelligen, en onderzoeken hoe goed het netwerk werkt voor deze twee methodes.

About the author

Pascal Lieshout was born in Amsterdam, The Netherlands, on March 5, 1981. He completed grammar school at the Gooi's Lyceum, Bussum, in June 1999. In September 2003 he received his master's degree in Econometrics (cum laude) from the University of Amsterdam (UvA). One year later, in October 2004, he also obtained his master's degree in Operations Research and Management (cum laude) from the same university. Subsequently, he became a Ph.D. student under the supervision of Michel Mandjes and Sem Borst at the Centrum Wiskunde & Informatica (CWI). Since October 2006 he has also been affiliated with the UvA.