Prioriteitswachtlijnen met gelimiteerde capaciteit

Priority Queues with Limited Capacity

Thomas Demoor

Promotoren: prof. dr. ir. H. Bruneel, prof. dr. ir. J. Walraevens
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2014 - 2015

UNIVERSITEIT
GENT

ISBN 978-90-8578-732-7
NUR 919
Wettelijk depot: D/2014/10.500/78

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Telecommunicatie en Informatieverwerking

Prioriteitswachtlijnen met gelimiteerde capaciteit

Priority Queues with Limited Capacity

Thomas Demoor

Proefschrift tot het bekomen van de graad van
Doctor in de Ingenieurswetenschappen
Academiejaar 2014-2015

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Informatietechnologie

Promotoren: Prof. dr. ir. Herwig Bruneel
Prof. dr. ir. Joris Walraevens

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Telecommunicatie en Informatieverwerking
Onderzoeksgroep SMACS

St-Pietersnieuwstraat 41, B-9000 Gent, België
Tel.: +32-9-264.34.12
Fax.: +32-9-264.42.95

Proefschrift tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen
Academiejaar 2014-2015

DANKWOORD

Uiteraard kan ik niet aan de plicht verzaken diegenen die een (onrechtstreekse) bijdrage aan dit doctoraat hebben geleverd te bedanken, uiteraard in de eerste plaats mijn promotor Herwig. Niet zozeer omdat hij mij ooit gevraagd heeft bij hem te komen doctoreren op prioriteitswachtlijnen, of omdat hij me daardoor jarenlang "tewerkgesteld" heeft, maar voor de manier waarop zijn onderzoeksgroep ingericht is. De wetenschappelijke vrijheid en het minimaliseren van administratieve overlast scheppen een perfect kader voor het bedrijven van "wetenschap om de wetenschap". Ik kon maandenlang ongestoord mijn hoofd breken over een onderzoeksprobleem zonder verantwoording af te moeten leggen voor die stilstand op tussentijdse evaluatievergaderingen of via "progress reports". Soms moest ik een probleem opbergen om later (of nooit meer) nog eens de strijd aan te gaan. Soms had ik geluk en schoot me plots de (achteraf gezien natuurlijk triviale) oplossing te binnen. Zo'n "eureka-moment" vond (wat de oude Grieken ook moge beweren) niet per se in bad plaats maar evengoed aan mijn bureau of tijdens discussies (met de collega's) op café . Dat we in een tijd leven waar sommigen dit "Don Quichot-achtige" proces proberen omvormen tot een industrieel fabricageproces heb ik bij de onderzoeksgroep SMACS amper gemerkt, waarvoor dank.

Van Joris, co-promotor, die ook mijn licenciaatsthesis (Master-thesis voor de jongelingen onder jullie) begeleid heeft, heb ik geleerd door te zetten tijdens het bovenvermelde "vast zitten" op een probleem. Blijven prutsen, vereenvoudigde gevallen bestuderen, . . . gewoon hard werken.

> On the mountains of truth you can never climb in vain: either you will reach a point higher up today, or you will be training your powers so that you will be able to climb higher tomorrow.
>
> — Friedrich Nietzsche

Behalve professor is Joris een "down-to-earth" prachtkerel met een brede interesse.

"Informele co-promotor", Dieter sprong in de bres toen Joris voor enkele maanden naar het buitenland trok toen ik nog maar 1 maand onderzoeker was. Hij liet me kennis maken met een hele rits aan methoden binnen het gebied van de toegepaste probabiliteit. Hoe meer werktuigen je hebt, hoe meer kans dat één ervan je de mogelijkheid geeft aan de oplossing te sleutelen.

> The surest way to corrupt a youth is to instruct him to hold in higher esteem those who think alike than those who think differently.
>
> — Friedrich Nietzsche

Hij leerde me hoe je op intuitie een goede benadering van de oplossing kunt bedenken, wat ongelooflijk nuttig is om te weten of je in de juiste richting zit. Verder was hij de ideale sparringpartner voor ideologische en filosofische discussies.

Ik kwam als vierde man in de bureau van de "anciens" terecht, waar iedereen zijn onderzoeksproblemen op het bord kwam uiteenzetten in de hoop geholpen te worden door een van de "wijzen" . Al zal het jullie niet verbazen dat ik na enkele weken ook daar het hoogste woord voerde, de aard van het beestje . . . Naast Joris en Dieter huisde daar ook een zekere Stijn, die vanwege zijn didactische kwaliteiten de persoon was aan wie je "domme" vragen stelde, maar ook auteur (artiest) van menig grafisch hoogstandje in MetaPost (waarvan enkele in deze thesis verwerkt zijn) en alom gekend als idealist. Al verloor hij die laatste stempel toen hij de stal verliet om professor te worden in het "industrieel management".

Verder sta ik erop de juryleden te bedanken voor het spenderen van hun overbevraagde tijd aan het evalueren van mijn proefschrift.

Bedankt aan de collega's voor de toffe werkomgeving: Koen de modelonderzoeker (correct?), Tom de mobiliteitsmaniak, Wouter de Walibi-wacko, Dieter de "pwat", de Roeselarenaars voor hun accent, de interne-dienstverlenings-strategen, de tennissers voor de geladen partijtjes, de middageters voor het verdragen van mijn "rants" richting beeldbuis, de mensen waarmee ik op congressen plezier mee gemaakt heb en alle andere SMACSers en TELINners.

De voetsoldaten van TELIN: Annette, Sylvia, Patrick, Davy, Philippe voor het vooral met daad bijstaan van deze hulpeloze onderzoeker bij allerhande administratieve en technische uitdagingen tijdens (de eindfase van) mijn doctoraat.

David en Sherolyn van het, nabij de vakgroep gelegen, Geuzenhuis, voor het aanbieden van een onderzoeks- en plezier-vriendelijke gelagzaal waar de vrijdagavondfile richting kust altijd maar langer werd.

Mijn vrienden Jeroen, Pieter, Frederik, Sofie, Pieter & Annelies, Siegfried & Shirley en diegenen die ik vergeten ben.

Mijn ouders voor de verkenningsmogelijkheden en het bijhorende vertrouwen en mijn broer David voor de fratsen tijdens die verkenningstochten. De rest van de (schoon)familie voor hun onvoorwaardelijke steun.

Mijn vrouw Céline voor de steun bij het bereiken van een volgende mijlpaal in ons leven en mijn dochter Anna, de enige mijlpaal die er toe doet.

De duur van de "schrijffase" van vele doctoraten (en ook het mijne) volgt een geometrische distributie. Elke maand dacht ik nog een maandje werk te hebben, ongeacht van hoelang ik eigenlijk al aan het schrijven was.

> When you screw up and nobody's saying anything to you anymore, that means they gave up. Your critics are the ones telling you they still love you.
>
> — Randy Pausch

Ik zou dus de collega's, vrienden, familieleden en vooral mijn vrouw Céline willen bedanken om te blijven vragen wanneer ik nu eindelijk klaar zou zijn. Bij deze . . .

Thomas Demoor — September 2014, De Haan

Table of Contents

I Two-class Priority Queue

II Partial Buffer Sharing

III Conclusions

IV Appendices

NEDERLANDSTALIGE SAMENVATTING –SUMMARY IN DUTCH–

Wachtlijnen zijn alomtegenwoordig. Iedereen heeft wel al eens in de rij gestaan in de supermarkt, het postkantoor, het pretpark of met de wagen in de file. Wachtlijnen zijn ook in fabricageprocessen, ketenbeheer en logistieke netwerken overvloedig aanwezig. Verder kan men ook aspecten van verschillende takken van de computerwetenschappen modelleren door middel van een wachtlijn. Bijvoorbeeld als in een telecommunicatienetwerk verschillende verbindingen tezelfdertijd datapakketten over dezelfde verbinding in het netwerk willen sturen.

Het bestuderen van wachtlijnsystemen, ook wel wachtlijntheorie genoemd, ligt op het kruispunt van toegepaste probabiliteit, door de stochastische (willekeurige) aard van het tijdstip waarop klanten aankomen en de tijd die nodig is om hen te bedienen, en operationeel onderzoek, door de veelvuldige praktische toepassingen. Een stochastisch model wordt gestuurd door willekeur (toevalsveranderlijken) en heeft tot doel het kwantificeren (door middel van probabiliteitsdistributies) van de willekeur van de uitkomsten (performantiematen), veroorzaakt door de willekeur van (een of meer van de) invoerveranderlijken.

Dit doctoraat bestudeert prioriteitswachtlijnen die bepaalde scenario's in telecommunicatienetwerken modelleren. De razendsnelle evolutie van telecommunicatienetwerken heeft tot een veelvoud van performantie-eisen voor verschillende soorten netwerkverkeer geleid. Uiteraard is het verzekeren van goede "Quality of Service" (QoS) voor alle types verkeer uitermate belangrijk. Door de telecommunicatie-context zullen we de klanten van het wachtlijnsysteem pakketten noemen.

In telecommunicatienetwerken kan je de verschillende soorten netwerkverkeer grosso modo in twee klassen (types) opsplitsen (er bestaan natuurlijk ook veel fijnere opsplitsingen). "Real-time" verkeer uit videostreaming en spraaktoepassingen, zoals een conversatie op Skype, vereist lage wachttijden maar kan een beperkte mate van pakketverlies verdragen. Dataverkeer daarentegen verdraagt helemaal geen pakketverlies maar kan wel wat tijdsvertraging aan. Globaal zegt men dat pakketten die hun vertraging willen minimaliseren tijdsprioriteit vereisen en dat plaatsprioriteit gevraagd wordt om de pakketverlieskans te minimaliseren.

In het eerste deel van dit proefschrift bestuderen we een prioriteitswachtlijnsysteem die de situatie uit de vorige paragraaf modelleert. Er zijn twee klassen pakketten die elk toekomen in een afzonderlijke wachtlijn. Beide wachtlijnen worden bediend door dezelfde bedieningseenheid maar klasse-1 pakketten krijgen absolute (tijds)prioriteit om de vertraging van deze hoge-prioriteitspakketten te minimaliseren. Bijgevolg kunnen lage-prioriteitspakketten (klasse-2) enkel bediend worden als er geen klasse-1 pakketten in het systeem aanwezig zijn. Het model heeft als bijzonderheid dat de capaciteit van de wachtlijn voor klasse-1 pakketten beperkt is tot N maar er oneindig veel klasse-2 pakketten in de wachtlijn aanwezig kunnen zijn. Daardoor noemen we dit model het N/∞ model. In de literatuur daarentegen veronderstelt men gewoonlijk dat beide wachtlijnen een ongelimiteerde capaciteit bezitten (het ∞/∞ model). Uiteraard bestuderen we de convergentie van het N/∞ model naar het ∞/∞ model als N groeit. De analyse van het N/∞ model gebeurt gelijktijdig in het probabiliteitsdomein voor klasse 1 en in het getransformeerd domein voor klasse 2 door het gebruik van een vector/matrix-representatie. In een eerste hoofdstuk veronderstellen we dat de bediening van een pakket altijd slechts een enkel slot duurt. In het volgende hoofdstuk kan de bedieningstijd een algemene distributie volgen. Dit deel wordt afgesloten door een hoofdstuk over de staartprobabiliteiten van de systeembezetting van de lage-prioriteitsklasse. De convergentie van het N/∞ model naar het ∞/∞ model voor N naar oneindig is vanzelfsprekend maar dit was niet af te leiden uit de analytische uitdrukkingen voor beide systemen. Onze analyse legt een cruciale relatie tussen de karakteristieke vergelijking van een recursierelatie in het eindige geval en de “kernel”, die de impliciet gedefinieerde functie veroorzaakt in het oneindige geval, bloot. Verder tonen we in verschillende praktische voorbeelden aan dat onder bepaalde omstandigheden (kleine capaciteit van de wachtlijn, hoge klasse-1 belasting, aankomsten met zware staarten) de resultaten van het N/∞ model merkbaar verschillen van deze die men bekomt door de capaciteit van de klasse-1 wachtlijnen door oneindig te benaderen.

In een tweede (korter) deel van het proefschrift bestuderen we een prioriteitswachtlijnsysteem met twee klassen die samen een enkele wachtlijn met eindige capaciteit delen volgens een “partial buffer sharing” (PBS) strategie. Hier spelen zowel tijds- als plaatsprioriteit een belangrijke rol. Een van de klassen krijgt tijdsprioriteit en heeft dus net als in het vorige deel absolute prioriteit bij het bedienen. Een van de klassen wordt plaatsprioriteit toegewezen. Als de wachtlijn minder pakketten bevat dan een zekere drempelwaarde worden alle pakketten toegelaten tot de wachtlijn maar wanneer deze waarde overschreden wordt laat het systeem pakketten zonder plaatsprioriteit niet meer tot de wachtlijn toe en worden ze verwijderd. Uiteraard kan geen enkel pakket de wachtlijn vervoegen als deze zijn maximale capaciteit bereikt heeft. Er zijn vier mogelijke combinaties van de twee prioriteitstypes maar we moeten er maar twee beschouwen omdat de andere twee symmetrisch zijn. In het ene scenario krijgt een klasse zowel tijds- als plaatsprioriteit en in de andere krijgt een klasse tijdsprioriteit en de andere plaatsprioriteit. Het laatste scenario past goed in the telecommunicatie-context die we ook in het eerste deel bespraken waar real-time verkeer tijdsprioriteit krijgt en dataverkeer plaatsprioriteit. Het ander scenario modelleert bvb. een “scalable video coding” (SVC) omgeving. SVC is een formaat voor het versturen van stromende video en gebruikt twee types pakketten: basislaagpakketten en augmentatielaagpakketten. De eerste zijn noodzakelijk om de video af te kunnen spelen aan lage kwaliteit terwijl de tweede enkel de kwaliteit verhogen en op zichzelf nutteloos zijn. In een dergelijke context is het dus verstandig om zowel tijds- als plaatsprioriteit aan basislaagpakketten te geven. We modelleren beide scenario’s op een geünificeerde manier en analyseren ze met matrix-analytische oplossingsmethoden. Men kan besluiten dat de QoS differentiatie die door deze modellen geleverd wordt groot is en dat het vinden van de juiste drempelwaarde voor de PBS strategie cruciaal is.

Summary in English

Queues are ubiquitous. Everyone has queued (waited in line) at a supermarket, post office, amusement park or has been stuck in traffic. Queues are also omnipresent in manufacturing plants, supply chains, logistics networks and all other kinds of processes. Queues also arise in several branches of computer science, e.g. in telecommunications when multiple connections concurrently want to send traffic over the same link of the network.

The study of queues, queueing theory, lies on the intersection of applied probability, due to the stochastic nature (randomness) associated with customer arrivals and the duration of service, and operations research, due to the myriad of real-life applications. A stochastic model is governed by randomness (random variables). Its purpose is to quantify (through probability distributions) the randomness of its outputs (performance measures) , which is caused by the randomness of one (or more) of its inputs.

This dissertation studies priority queues that model certain settings in telecommunications networks. The rapid development of modern telecommunication networks has resulted in a wide variety of performance demands for various types of traffic. Evidently, allowing all traffic to meet their Quality of Service (QoS) requirements is of paramount importance. Due to the telecommunications context, we refer to the customers of the studied queueing system as packets.

In telecommunications networks, a rather coarse, but very practical, classification distributes packets in two traffic classes. Real-time traffic, such as streaming video and voice, e.g. a Skype conversation, requires low delays but can endure a small amount of packet loss. On the other hand, data traffic, such as file transfer, benefits from low packet loss but has less stringent delay characteristics. In general, packets requiring minimal mean delay and delay jitter are said to request time priority whereas space priority is requested for minimizing packet loss.

In the first part of this dissertation, a two-class priority queueing system will be studied that models the setup described in the previous paragraph. There are two classes of packets, each arriving in a dedicated queue. Both queues are served by the same server but the server gives absolute (time) priority to class-1 packets in order to minimize the delay of these high-priority packets. Consequently, the (low-priority) class-2 packet waiting at the head of the class-2 queue can only enter the server if there are no class-1 packets in the system. The peculiarity of this model is that the class-1 queue capacity is limited to N, which is a finite positive integer, but the class-2 capacity is infinitely large. Therefore, we denote this the N/∞ model. In contrast, in the literature, the queue capacity is generally assumed to be infinite for both classes (the ∞/∞ model). Evidently, as N increases, the N/∞ priority queue is increasingly similar to a system where both queues are presumed to be of

infinite capacity, and we thus investigate this behavior. The analysis of the N/∞ model simultaneously takes place in the probability domain for class-1 and in the transform domain for class-2 through the use of a vector/matrix representation. In a first chapter, we assume that service of a packet always takes a single slot. In the subsequent chapter, we let the service times follow a general distribution. The part is concluded with a chapter on determining the tail behavior of the distribution of the low-priority system content. It is evident that, in the limit for N to ∞, the results for the N/∞ model must converge to those for the ∞/∞ model. However this was not clear from the formulas of both systems. Our analysis has uncovered a crucial relation between the characteristic polynomial of a recurrence relation in the finite case and the kernel, which causes the implicitly defined function, in the infinite case. Furthermore, through several numerical examples, we have showed that, under certain conditions (small queue capacity, relatively high class-1 load, power-law arrivals), the results for the N/∞ model are considerably different from the ones obtained if one assumes infinite class-1 queue capacity.

In the second (shorter) part of this dissertation, we study a two-class priority queueing system where both classes share a single queue with finite capacity according to a partial buffer sharing (PBS) policy. Here, both time and space priority play a crucial role. One of the classes receives absolute time priority. As in the previous part, these packets receive service before the packets of the other class. Additionally, one of the classes receives space priority. When the queue contains less packets than a (predetermined) threshold value, PBS accepts all packets but when the queue (also called buffer) level is over a predetermined threshold, packets with low space priority cannot enter the queue and are dropped by the system. Evidently, when the system is entirely full, all arriving packets are dropped. There are four possible combinations of the two priority types. However, we only need to consider two as the other two then follow directly by swapping the classes. The two scenarios are thus giving both time and space priority to one of the classes or giving time priority to one class and space priority to the other. In a general telecommunications context, as detailed in the previous part, one would of course give time priority to real-time packets and space priority to data packets. In contrast, in a scalable video coding (SVC) setting one would prefer the other scenario. SVC uses two types of packets: base layer and enhancement layer packets. The former are required to decode and playback the video, although at poor quality, whereas enhancement packets increase quality but are useless without base packets. Here, it thus makes sense to give both time and space priority to base packets. We present a unified way to model both scenarios and analyze them using well-known matrix analytic solution techniques. One can conclude that the range of QoS differentiation covered by these models is large and that determining an appropriate value for the threshold of the PBS policy is of paramount importance.

1

Introduction

1.1 The queue and you

A queue is formed when multiple “customers” concurrently require access to a “service”. Queueing is ubiquitous. Everyone has queued (waited in line) at a supermarket, post office, amusement park or has been stuck in traffic. Queues are also omnipresent in manufacturing plants, supply chains, logistics networks and all other kinds of processes. Queues also arise in several branches of computer science, e.g. in file storage, when multiple files concurrently need to be written to or read from a storage medium, such as a hard disk, or in telecommunications, when multiple connections concurrently want to send traffic over the same link of the network.

Note 1. *As the research presented in this dissertation was performed at the Department of Telecommunications and Information Processing, the studied models were chosen with telecommunications applications in mind. Therefore, we will often denote the customers of the queueing systems by packets. However, this is mere terminology. The mathematical modeling and analysis is completely independent of the application. Therefore, if a model is a sensible approximation of reality in the application at hand, the analysis method and the results presented in this dissertation can be applied in any practical setting.*

1.2 Queueing theory

The formal (mathematical) study of queues is over a hundred years old. In 1908, Agner Krarup Erlang, a Danish engineer who worked for the Copenhagen Telephone Exchange, published the first “queueing paper”. In this era, automatic switching of telephone calls was in its very early development. Before that, making a telephone call consisted of calling a telephone exchange and asking to the operator to connect you to your intended destination. Calls were manually switched by connecting both lines on a switching board. Erlang realized that overload at the telephone exchange was caused by the fact that telephone calls arrive randomly in time (according to a Poisson process) and can thus be clumped together and that call durations are highly variable. His subsequent papers studied the waiting time and the occurrence of lost calls due to all lines being busy. His work was of great practical value for dimensioning telephone switching boards but remained unknown to those outside the field of teletraffic theory until the 1930s, when the Russian mathematicians discovered this field, immediately noted its probabilistic (also called stochastic) nature and formalized the results by fitting them into the probability framework developed by Markov. However, the queueing theory domain did not become popular among the mathematical community until after the second world war. During the war, mathematicians had to tackle operations research problems (for military and logistics purposes) of stochastic nature and hence started to appreciate queueing theory. Telephone systems are less of a "hot" topic today but queueing theory has solidified its position at the intersection of applied probability and operations research because, as mentioned in the previous section, it has a myriad of applications. For a more extensive history of the field, see [1],which was written in celebration of "100 years of queueing" .

Note 2. *For those who are unaccustomed to reading scientific texts, remark that references to the bibliography, which can be found at the back of this dissertation, are indicated by their index number between brackets: e.g. [1].*

1.3 Stochastic modeling

Note 3. *One can fill an entire library with books on this topic so being exhaustive would not only be impossible but also detrimental to our goals. Here, we focus on the aspects of stochastic modeling relevant to the setting used in this dissertation, discrete-time Markov chains with countable state spaces. Furthermore, in order to grasp this text, some very basic notions of probability, see e.g. [2],are required.*

A stochastic model is governed by randomness (random variables). Its purpose is to quantify (through probability distributions) the randomness of its outputs (performance measures) , which is caused by the randomness of one (or more) of its inputs.

A discrete random variable x, taking values in $\mathbb{N}$, is generally described through its probability mass function (pmf) given by

$$x(n) = \Pr[x = n], \quad n \geq 0, \tag{1.1}$$

which gives the probability that x is equal to n, for all possible values n and is often called "the distribution of x".

1.3.1 Discrete-time Markov chains

Time is discretized by dividing it into fixed-length periods, called slots. Consider a sequence of random variables $\{x_k\}_{k=1}^{\infty}$, where k indicates the slot index. The value of x_k is called the state in slot k. Loosely speaking, this sequence "forms" a Markov chain if, in each slot, the next state only depends on the present state (what is the current state), not on the past states (how did we get in the state). Formally, the sequence $\{x_k\}_{k=1}^{\infty}$ is said to satisfy the Markov property if, for all k (and all $i_1, \ldots, i_k$),

$$\Pr[\, x_{k+1} = i_{k+1} \mid x_k = i_k, x_{k-1} = i_{k-1}, \ldots, x_1 = i_1] = \Pr[\, x_{k+1} = i_{k+1} \mid x_k = i_k] \,. \quad (1.2)$$

In this dissertation (and in general), one mostly encounters time-homogeneous Markov chains, where the transitions from one slot to the next are independent of the slot number k. A time-homogeneous Markov chain is completely characterized by a single transition matrix $\boldsymbol{P}$, where the element on the i-th row, j-th column is given by

$$p_{ij} = \Pr\left[\, x_{k+1} = j \;\middle|\; x_k = i \,\right] . \quad (1.3)$$

Note 4. *Throughout this dissertation, matrices and vectors are set in boldface, the former represented by capital letters and the latter by a lowercase letter.*

States of countable Markov chains are commonly classified based on three criteria. First, a state is called k-periodic if a Markov chain starting in the state only returns at times that are multiples of k. For $k = 1$, the state is aperiodic. Secondly, a state is called recurrent if the return time is finite with probability one, and transient if it is not. If one state of a Markov chain is periodic, then the Markov chain is called periodic. Likewise, if one state is recurrent, then the Markov chain is called recurrent. Finally, states i and j are said to communicate if there exists a positive probability of going from i to j and from j to i. The communication relation forms an equivalence relation, dividing the system into equivalence classes which in this context are known as recurrence classes. A Markov chain with a single recurrence class is called irreducible.

The "most interesting output" one can get from a Markov chain is its steady-state (or stationary) distribution given by (the row vector) $\boldsymbol{\pi}$. This is the distribution indicating the probability that the chain is in a state, once the influence of the initial distribution, the state in slot 1, has dissolved and letting the Markov chain transition to the next slot no longer changes this distribution. This is generally indicated by taking the limit for k, the slot number, to ∞. For an irreducible Markov chain with transition matrix $\boldsymbol{P}$, the stationary distribution $\boldsymbol{\pi}$ is the unique solution of

$$\begin{cases} \boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P} \\ 1 = \boldsymbol{\pi}\boldsymbol{e} \end{cases} . \quad (1.4)$$

Note 5. *Throughout this dissertation, **e** denotes a column vector of ones of appropriate size.*

1.3.2 FIFO queue with finite capacity

Let us make the reasoning in the previous section a bit more tangible by exploring an example. Consider a First-In-First-Out (FIFO), meaning that packets are served in order of arrival, queue with a single server that serves one packet per slot. This is the most basic type of queue. Let the number of arriving packets in slot k be given by the sequence of independent and identically distributed (i.i.d.) random variables $\{a_k\}_{k=1}^{\infty}$. Each a_k thus follows the same distribution, with pmf $a(n)$, independent of the rest of the sequence.

Furthermore, let the queue capacity, this is the maximum number of packets that can concurrently wait in the queue, be limited to N and assume that packets that arrive at a full system are dropped (they do not enter the system, they are discarded). The number of packets in the queueing system, called the system content, at the beginning of slot k is denoted by u_k.

Note 6. *The next subsection will explain the exact meaning of "at the beginning of a slot". In the meanwhile, the context should provide enough information to grasp the essentials.*

This sequence clearly is Markovian as u_{k+1}, the system content in slot $k+1$, is found by subtracting the packet served in slot k from u_k (if there was a packet in service, thus if $u_k > 0$), and then adding the number of arriving packets in slot k, given by a_k, independent of the system content in the slots before slot k.

Our goal is to quantify the steady-state distribution of the sequence $\{u_k\}_{k=1}^{\infty}$ through the pmf

$$u(n) = \lim_{k\to\infty} \Pr[u_k = n], \quad 0 \le n \le N, \tag{1.5}$$

which is stochastic due to the stochastic nature of the arrival process at the queue.

Note 7. *Notice the difference between an i.i.d. sequence, where all random variables have the same distribution, and a Markov sequence, where they "converge" to a steady-state distribution.*

In fact, using the terminology developed in the previous section, the transition matrix of this Markov chain is given by

$$\boldsymbol{P} = \begin{bmatrix} a(0) & a(1) & \cdots & a(N-1) & \sum_{n=N}^{\infty} a(n) \\ a(0) & a(1) & \cdots & a(N-1) & \sum_{n=N}^{\infty} a(n) \\ 0 & a(0) & \cdots & a(N-2) & \sum_{n=N-1}^{\infty} a(n) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & a(0) & \sum_{n=1}^{\infty} a(n) \end{bmatrix}, \tag{1.6}$$

which is irreducible and aperiodic for practical arrival processes, and the stationary distribution is given by

$$\boldsymbol{\pi} = \begin{bmatrix} u(0) & \ldots & u(N) \end{bmatrix}. \tag{1.7}$$

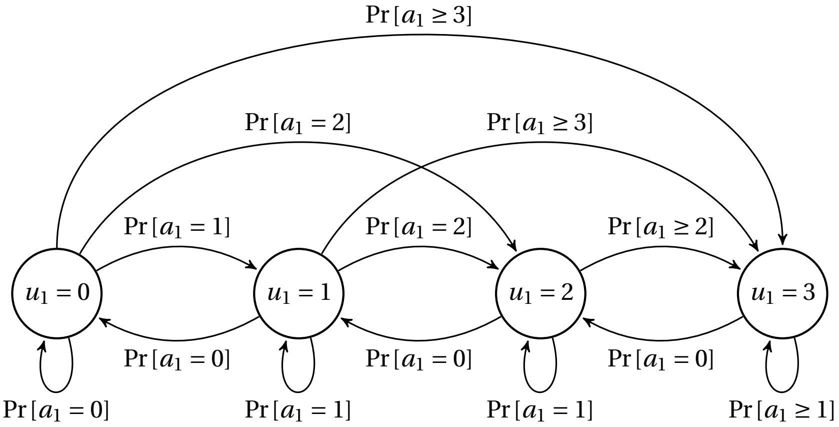

Figure 1.1: States and transitions for Markov chain of a FIFO queue with capacity $N = 3$.

The structure of $\boldsymbol{P}$ is perhaps clearer by investigating a visual representation of the Markov chain, which is depicted in figure 1.1 for $N = 3$. For the specific case where the queue capacity is limited to $N = 3$, we have

$$\boldsymbol{P} = \begin{bmatrix} a(0) & a(1) & a(2) & \sum_{n=3}^{\infty} a(n) \\ a(0) & a(1) & a(2) & \sum_{n=3}^{\infty} a(n) \\ 0 & a(0) & a(1) & \sum_{n=2}^{\infty} a(n) \\ 0 & 0 & a(0) & \sum_{n=1}^{\infty} a(n) \end{bmatrix}. \tag{1.8}$$

Now, if we specify the distribution of the arrival process, one can simply calculate these probabilities numerically. Let the arrival process follow a Poisson distribution with parameter $\lambda = 0.5$. This is the most common distribution in queueing theory, detailed in appendix A.1. When $\lambda = 0.5$, the probabilities of the number of arrivals in a slot are given by $a(0) = 0.6065306597$, $a(1) = 0.3032653298$, $a(2) = 0.0758163324$, $a(3) = 0.01263605541$, $a(4) = 0.001579506926$, … and, on average, one packet arrives per two slots. Then we have,

$$\boldsymbol{P} = \begin{bmatrix} 0.6065306597 & 0.3032653298 & 0.0758163324 & 0.0143876780 \\ 0.6065306597 & 0.3032653298 & 0.0758163324 & 0.0143876780 \\ 0 & 0.6065306597 & 0.3032653298 & 0.0902040104 \\ 0 & 0 & 0.6065306597 & 0.3934693403 \end{bmatrix}, \tag{1.9}$$

$$\boldsymbol{\pi} = \begin{bmatrix} 0.5077 \\ 0.3293 \\ 0.1244 \\ 0.038 \end{bmatrix}^T. \tag{1.10}$$

Note 8. *The transpose of a vector $\boldsymbol{a}$ (matrix $\boldsymbol{A}$) is denoted by $\boldsymbol{a}^T$ ($\boldsymbol{A}^T$).*

For Markov chains with a finite (and relatively small) state space like the one studied here, we can simply compute the evolution through time numerically from slot to slot, illustrating the convergence to a stationary distribution. For instance, let us start from an empty queue in slot 1. Thus, $u_1 = 0$ and the distribution over the states is given by the vector $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$. In the next slot, the distribution is then given by

$$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \boldsymbol{P} = \begin{bmatrix}0.6065\\0.3033\\0.0758\\0.0143\end{bmatrix}^T, \tag{1.11}$$

after 5 slots by

$$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \boldsymbol{P}^4 = \begin{bmatrix}0.5136\\0.3285\\0.1212\\0.0365\end{bmatrix}^T, \tag{1.12}$$

and in slot 50 by

$$\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \boldsymbol{P}^{49} = \begin{bmatrix}0.5077\\0.3293\\0.1244\\0.0383\end{bmatrix}^T. \tag{1.13}$$

One can thus "see" the convergence to $\boldsymbol{\pi}$.

Note 9. *Evidently, Markov chains are not solely used to model queueing processes. For instance, in part 2 of this dissertation, we consider a Markovian arrival process, where a Markov chain modulates the background state of the arrival process in order to introduce time-correlation.*

1.3.3 Order of events in a slot

In the example in the previous section, some (tacit) assumptions were made on the order of arrivals, departures and the moment of observation. Here following assumptions were made. Service of a packet starts and is completed at slot boundaries. Observation of the system (content) "at the beginning of a slot" happens immediately after the slot boundary, before any arrivals. Arrivals occur during the slot. Consequently, service of a packet arriving into an empty system cannot start in that slot but only at the next slot boundary.

These assumptions, which will also be used in part 1 of this dissertation, are not set in stone. In part 2 of this dissertation, a different convention is used. However, the different sets of assumptions almost always yield similar results [3] and are thus merely a technicality. The different possibilities are called early/late arrivals, delayed access,etc. More information can be found, a.o. in [4, vol. 3].

Note 10. *The fact that one needs to define some assumptions due to the discretization of time into slots is, in my opinion, one of the main reasons of the (perceived) messiness of discrete-time queueing models in comparison with continuous-time queueing models.*

1.3.4 Recursive formulation of Markov chains

There is a more intuitive and easier to construct description of a Markov chain describing a queueing process than by defining the transition matrix. Consider an auxiliary sequence of i.i.d. random variables $\{y_k\}_{k=1}^{\infty}$. The sequence $\{x_k\}_{k=1}^{\infty}$ is a Markov chain if there exists a function f such that, for $k > 1$,

$$x_{k+1} = f(x_k, y_k)\,. \tag{1.14}$$

This representation is common when describing a (reflected) random walk, a type of Markov chain of which queueing models are a prime example. For the FIFO queue with finite capacity studied in the previous subsection, the recursive representation is given by

$$u_{k+1} = \min(N, (u_k - 1)^+ + a_k)\,. \tag{1.15}$$

Note 11. *In this dissertation and in the queueing literature in general,* $(x)^+$ *denotes the maximum of* x *and* 0*. This operator exemplifies that a queueing process is a reflected random walk. As the system content cannot be negative, the process is reflected at 0.*

Notice how this representation intuitively follows how we think “about” a queue: what was already present minus what was served plus new arrivals.

1.3.5 Generating functions

The probability generating function (pgf) of a random variable x, is simply a transformation of the pmf, defined by

$$X(z) = \mathrm{E}\left[z^x\right] = \sum_{n=0}^{\infty} \Pr[x = n]\, z^n\,. \tag{1.16}$$

This can be seen as a function in complex variable z, which can be proven to be analytic for $|z| < 1$.

Note 12. *Due to the ubiquity of* z *as argument of a probability generating function, the pgf is often called the* z*-transform.*

Now, let us present some properties that make pgfs “easier to work with” compared to pmfs. Firstly, let a and b be independent random variables, with respective pgfs $A(z)$ and $B(z)$. The random variable $c = a + b$, representing their sum, has pgf $C(z)$ given by

$$C(z) = \mathrm{E}\left[z^{a+b}\right] = \mathrm{E}\left[z^a z^b\right] = A(z)B(z)\,. \tag{1.17}$$

The final transition is based on the independence of the random variables. The power of generating functions is apparent as the pgf is much easier to handle than the equivalent pmf

$$\Pr[c = n] = \sum_{i=0}^{n} \Pr[a = i] \Pr[b = n - i]\,. \tag{1.18}$$

Secondly, let $\{b_i\}_{i=1}^{\infty}$ be a sequence of i.i.d. random variables, each with pgf $B(z)$ and let a be a random variable independent of the b_i with pgf $A(z)$. The random variable $c = b_1 + b_2 + \cdots + b_a$ has a remarkably neat pgf $C(z)$ as

$$\begin{aligned} C(z) &= \mathrm{E}\left[z^{\sum_{i=1}^{a} b_i}\right] \\ &= \sum_{n=0}^{\infty} \mathrm{E}\left[z^{\sum_{i=1}^{a} b_i} \,\middle|\, a=n\right] \Pr[a=n] \\ &= \sum_{n=0}^{\infty} \mathrm{E}\left[z^{\sum_{i=1}^{n} b_i}\right] \Pr[a=n] \\ &= \sum_{n=0}^{\infty} B(z)^n \Pr[a=n] \\ &= A\big(B(z)\big)\,. \end{aligned} \tag{1.19}$$

The second transition holds due to the law of total expectation and the second to last one due to (1.17). Again, this expression in the transform domain is much neater than the equivalent expression in the probability domain.

Finally, the moments of a random variable are easily found from its pgf as the k-th derivative in z of $X(z)$ is the k-th factorial moment of x

$$\mathrm{E}[x(x-1)\ldots(x-k+1)] = X^{(k)}(1)\,. \tag{1.20}$$

Hence, one can also easily compute the regular moments of x, such as the mean (average value): $\mathrm{E}[x] = X'(1)$, the variance: $\mathrm{Var}[x] = X''(1) + X'(1) - (X'(1))^2$, etc. Consequently, if one has calculated the pgf of one of the "output" random variables of a queueing system (e.g. the system content), the moments of this random variable, which are key performance measures of the system, can be readily obtained.

Note 13. *We interchangeably use several notations for the derivative of a function. Let $f'(z)$ denote the first derivative of $f(z)$ in z and $f''(z)$ the second derivative. Higher order derivatives are indicated differently. The k-th derivative is given by $f^{(k)}(z)$ or equivalently by $\frac{\partial^k f(z)}{\partial z^k}$.*

However, using pgfs also has a downside. Computing the moments is easy, but obtaining the probabilities, called "inverting the pgf" (making the inverse transformation from the pgf to the pmf) is far from straightforward. In theory, one can simply see the pgf as a Taylor series around $z = 0$ and hence

$$x(n) = \frac{1}{n!} \frac{\partial^n X(z)}{\partial z^n}\bigg|_{z=0}, \quad n \geq 0\,, \tag{1.21}$$

but this becomes computationally infeasible as n becomes large. Unfortunately, the probabilities for large values of n, the so-called "tail of the distribution", are of great interest as these "rare-events" often correspond to worst-case behavior, disasters, etc. Appendix C details a method for inverting a pgf, which will be frequently used in chapter 4.

1.3.6 FIFO queue with infinite capacity

Let us again consider the FIFO queue but let us remove the restriction on the queue capacity. Then, the recursive representation of the system content is given by

$$u_{k+1} = (u_k - 1)^+ + a_k\,. \tag{1.22}$$

Notice that the stochastic recursion here is much simpler than (1.15). Consequently, most papers in the queueing literature assume the queue capacity to be infinite. In the case of a finite queue capacity N, one has an additional reflecting boundary (as the system content at the beginning of a slot cannot exceed N) similar to the one at the origin, and thus additional complexity.

However, the transition matrix $\boldsymbol{P}$ is now infinitely large and the method used previously is no longer feasible, but the generating functions, introduced in the previous subsection, are also ideal for tackling a recursion of the form (1.22). Recall that the arrivals are denoted by i.i.d. random variables $\{a_k\}_{k=1}^{\infty}$ with common pmf $a(n)$. Let the corresponding pgf be given by $A(z)$ and let the pgf of the system content in slot k be given by $U_k(z)$. Then, (1.22) yields

$$\begin{aligned}
U_{k+1}(z) &= \mathrm{E}\left[z^{u_{k+1}}\right]\\
&= \mathrm{E}\left[z^{(u_k-1)^+ + a_k}\right]\\
&= \mathrm{E}\left[z^{(u_k-1)^+ + a_k}\,1\{u_k=0\}\right] + \mathrm{E}\left[z^{(u_k-1)^+ + a_k}\,1\{u_k>0\}\right]\\
&= \mathrm{E}\left[z^{a_k}\,1\{u_k=0\}\right] + \mathrm{E}\left[z^{u_k-1+a_k}\,1\{u_k>0\}\right]\\
&= U_k(0)A(z) + \frac{1}{z}\big(U_k(z) - U_k(0)\big)A(z)\,.
\end{aligned} \tag{1.23}$$

Note 14. *The indicator function 1*$\{x=i\}$ *is* 1 *if* $x=i$ *and equals* 0 *otherwise.*

Here, the third transition uses the law of total expectation to split the expression into two terms corresponding to the two "states" of the server: either it is serving a packet (when $u_k > 0$) or idle (when the system is empty). Given the state of the server, expression (1.23) simplifies considerably. The fifth transition is based on the fact that a_k is independent of u_k.

Our goal is to obtain the pgf $U(z)$, which is the pgf of the steady-state distribution of the sequence $\{u_k\}_{k=1}^{\infty}$. This queueing system reaches a steady-state equilibrium if the mean number of arrivals, denoted by $\lambda = \mathrm{E}[a] = A'(1)$, does not exceed the service capacity per slot which equals 1, thus if $\lambda < 1$. By taking the limit on both sides of (1.23), one finds that

$$\lim_{k\to\infty} U_{k+1}(z) = \lim_{k\to\infty} U_k(0)A(z) + \frac{1}{z}(U_k(z) - U_k(0))A(z)\,, \tag{1.24}$$

and, as $U(z) = \lim_{k\to\infty} U_{k+1}(z) = \lim_{k\to\infty} U_k(z)$, hence

$$\begin{aligned}
U(z) &= U(0)A(z) + \frac{1}{z}\big(U(z) - U(0)\big)A(z)\\
&= U(0)\frac{(z-1)A(z)}{z - A(z)}\,.
\end{aligned} \tag{1.25}$$

In steady state, the probability that the system is empty equals $U(0) = 1 - \lambda$. This can be obtained through several different arguments. For instance, as $U(z)$ is a pgf, $U(1) = \sum_{i=0}^{\infty} \Pr[u = i] = 1$ and setting $z = 1$ in (1.25) and using L'Hôpital's rule yields

$$1 = U(0)\frac{1}{1-\lambda}. \tag{1.26}$$

Alternatively, the identity can also be obtained through the following classic steady-state argument that, roughly, states "what goes in must come out".

Theorem 1.1. *When a system is in steady state, the average number of packets accepted per slot equals the average number of packets served per slot.*

A packet is served by the system when the system content at the beginning of the slot is larger than 0. Consequently, invoking the theorem yields $\lambda = 1 - U(0)$. Consequently, the pgf of the steady-state system content can be expressed in terms of the arrival process as

$$U(z) = (1-\lambda)\frac{(z-1)A(z)}{z - A(z)}. \tag{1.27}$$

Thus, by specifying a specific arrival process one can numerically calculate $U(z)$, and all its moments, directly. For instance, for Poisson arrivals with parameter $\lambda = 0.5$ the pgf is given by $A(z) = e^{0.5(z-1)}$.

1.3.7 Other methods

The previous subsection demonstrated that a system with an infinitely large transition matrix $\boldsymbol{P}$ is readily solved through generating functions. However, a myriad of useful methods, each with their own strengths and weaknesses, have been developed for handling these kind of systems. Most notably, numerical methods , such as the matrix-geometric/matrix-analytic methods pioneered by Neuts [5, 6], have been very succesfull. We will use a numerical method in part 2 of this work. Furthermore, large deviations and software simulation are other notable approaches for tackling the kind of problems considered in this dissertation.

1.4 Telecommunication networks

The rapid development of modern telecommunication networks has resulted in a wide variety of performance demands for various types of traffic. Evidently, allowing all traffic to meet their Quality of Service (QoS) requirements is of paramount importance. One of the more popular attempts to supply improved QoS is Differentiated Services (DiffServ) [7],[8], a computer networking architecture in Internet Protocol (IP) networks that distributes packets in various traffic classes. It provides QoS differentiation by basing the order in which packets are transmitted on class-dependent priority rules. In DiffServ each packet is forwarded according to its Per-Hop Behavior (PHB). Obviously, implementation of DiffServ is particularly interesting in networks that struggle to provide acceptable QoS because bandwidth is limited and/or variable.

A rather coarse, but very practical, classification distributes packets in two traffic classes. Real-time traffic, such as streaming video and voice, e.g. a Skype conversation, requires low delays but can endure a small amount of packet loss. On the other hand, data traffic, such as file transfer, benefits from low packet loss but has less stringent delay characteristics. In general, packets requiring minimal mean delay and delay jitter are said to request time priority whereas space priority is requested for minimizing packet loss.

1.5 Two-class priority queue

In the first part of this dissertation, a two-class priority queueing system will be studied. There are two classes of packets, each arriving in a dedicated queue. Both queues are served by the same server but the server gives absolute (time) priority to class-1 packets. Consequently, the class-2 packet waiting at the head of the class-2 queue can only enter the server if there are no class-1 packets in the system.

This setup models a DiffServ implementation where real-time traffic (Expedited Forwarding PHB) has strict priority scheduling over data traffic (Default PHB). Although this scheduling algorithm is drastic, as data packets are only served if the system is void of real-time packets, it minimizes the delay of the real-time packets hence delivering maximum performance to real-time traffic. As real-time packets receive absolute priority, they can occupy the server (almost) permanently, denying data traffic of any service, if no admission control is performed. Therefore, the amount of real-time traffic allowed into the system should be regulated. Moreover, queueing a very large amount of real-time packets is useless anyway as they require small delays. These two observations emphasize the importance of limiting the queue capacity for real-time packets, evidently, without neglecting packet loss constraints. On the other hand, data packets require a very low amount of loss to achieve their QoS requirements. Therefore, the queue capacity for data packets should be as large as practically feasible. Hence, we can assume that the capacity for data packets is sufficiently large to be approximated by infinity but that the capacity for real-time packets should be modeled exactly. Here, space priority is only implicitly present, through the limited queue capacity for real-time packets, as each class has its dedicated queue.

We study this queueing system and compare it to the related system where the capacity for both classes is unbounded and identify the conditions under which they yield different results. Crucially, we have developed a model that is amenable to singularity analysiis allowing us to provide insight into the convergence between these two systems when letting the finite capacity restriction increase to infinity.

1.6 Partially shared buffer

In the second part of this dissertation, we study a two-class priority queueing system where both classes share a single queue with finite capacity according to a partial buffer sharing (PBS) policy. Here, both time and space priority play a crucial role. One of the classes receives absolute time priority. As in the previous section, these packets receive service before the packets of the other class. Additionally, one of the classes receives space priority. When the queue contains less packets than a (predetermined) threshold value, PBS accepts all packets but when the queue (also called buffer) level is over a predetermined threshold, packets with low space priority cannot enter the queue and are dropped by the system. Evidently, when the system is entirely full, all arriving packets are dropped.

There are four possible combinations of the two priority types. However, we only need to consider two as the two others then follow directly by swapping the classes. The two scenarios are thus giving both time and space priority to one of the classes or giving time priority to one class and space priority to the other. In a general telecommunications context, as detailed in the previous section, one would of course give time priority to real-time packets and space priority to data packets. In contrast, in a scalable video coding (SVC) setting one would prefer the other scenario. SVC uses two types of packets: base layer and enhancement layer packets. The former are required to decode and playback the video, although at poor quality, whereas enhancement packets increase quality but are useless without base packets. Here, it thus makes sense to give both time and space priority to base packets.

We present a unified way to model and analyze both scenarios. One can conclude that this queueing system allows for a broad spectrum of QoS differentiation and that tuning the threshold parameter is of paramount importance.

1.7 Publications

1.7.1 Publications in international journals

- [9] J. Walraevens, T. Demoor, T. Maertens, and H. Bruneel. Stochastic queueing-theory approach to human dynamics. *Phys. Rev. E*, 85:021139, 2012

- [10] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. Partially shared buffers with full or mixed priority. *Journal of Industrial and Management Optimization*, 7(3):735–751, 2011
  ⇒ This paper corresponds to part 2 of this dissertation.

- [11] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Performance analysis of a priority queue : expedited forwarding PHB in DiffServ. *AEU-International Journal of Electronics and Communications*, 65(3):190–197, 2011

- [12] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Influence of real-time queue capacity on system contents in Diffserv's expedited forwarding per-hop-behavior. *Journal of Industrial and Management Optimization*, 6(3):587–602, 2010
  ⇒ This paper is related to chapter 3 of this dissertation.

1.7.2 Publications in international conferences

- [13] J. Walraevens, T. Demoor, D. Fiems, and H. Bruneel. Uncovering the evolution from finite to infinite high-priority capacity in a priority queue. In *2013 International Conference on Computing, Networking and Communications (IEEE ICNC), San Diego*, 2013
  ⇒ This paper is related to chapter 4 of this dissertation.

- [14] D. Fiems, S. Andreev, T. Demoor, H. Bruneel, Y. Koucheryavy, and K. De Turck. Analytic evaluation of power saving in cooperative communication. In *Conference on Future Internet Communications (CFIC), Coimbra, Portugal*, 2013

- [15] T. Demoor, S. Andreev, K. De Turck, H. Bruneel, and D. Fiems. On the effect of combining cooperative communication with sleep mode. In *9th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Courmayeur, Italy*, 2012

- [16] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. The impact of queue capacities on asymptotics in priority queues. In *International conference on Stochastic Modelling and Simulation, Chennai, India*, pages 29–29, 2011
  ⇒ This paper is related to chapter 4 of this dissertation.

- [17] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Tail behaviour of a finite-/infinite-capacity priority queue. In *3rd Madrid conference on Queueing Theory, Toledo, Spain*, pages 31–32, 2010
  ⇒ This paper is related to chapter 4 of this dissertation.

- [18] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. The preemptive repeat hybrid server interruption model. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010), Cardiff, Wales. Lecture Notes in Computer Science*, volume 6148, pages 59–71. Springer, Springer, 2010

- [19] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Time and space priority in a partially shared priority queue. In *5th International conference on Queueing Theory and Network Applications, Beijing, China*, pages 125–131. Association for Computing Machinery (ACM), 2010
  ⇒ This paper corresponds to part 2 of this dissertation.

- [20] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Mixed finite-/infinite-capacity priority queue with general class-1 service times. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2009), Madrid, Spain. Lecture Notes in Computer Science*, volume 5513, pages 264–278. Springer, 2009
  ⇒ This paper is related to chapter 3 of this dissertation and won the best paper award at the conference.

- [21] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Modelling queue sizes in an expedited forwarding DiffServ router with service differentiation. In *4th International conference on Queueing Theory and Network Applications, Singapore, Singapore*. Association for Computer Machinery (ACM), 2009
  ⇒ This paper is related to chapter 3 of this dissertation.

- [22] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Mixed finite-/infinite-capacity priority queue with interclass correlation. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA), Nicosia, Cyprus. Lecture Notes in Computer Science*, volume 5055, pages 61–74. Springer-Verlag, 2008
  ⇒ This paper is related to chapter 2 of this dissertation.

1.7.3 Publications in national conferences

- [23] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. Controlling delay and loss in a DiffServ router with expedited forwarding PHB. In *23rd National Conference of the Belgian Operations Research Society*, pages 98–98, 2009

- [24] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Performance analysis of a two-class priority queue with finite high-priority queue capacity. In *22nd National Conference of the Belgian Operations Research Society*, pages 54–56, 2008

Part I

Two-class Priority Queue

2

N/∞ Priority Queue - Single-slot Service Times

2.1 Introduction

In this chapter, we study a discrete-time queueing system with a single server fed by two queues, one per priority class, and with an absolute priority scheduling algorithm in order to minimize the delay of the high-priority packets. This means that low-priority packets (class 2) are only served if there are no high-priority packets (class 1) in the system. This is the most drastic scheduling method, minimizing class-1 delay at the cost of class-2 performance. To the best of our knowledge, such a system, called a priority queue, was first studied over half a century ago, more precisely in 1954, in [25]. Survey [26] gives an overview up to 1960 and one of the first books dedicated to priority queues [27] appeared in 1968.

Note 15. *One can identify a pioneering work by the number of cited references. The seminal paper [25] mentions two references: Feller's 1950 book on probability (which, in modern times, is just assumed to be the invisible reference [0] in any paper using probability) and a paper from Bell Labs to show that there are practical applications (telephone systems). That must have been magical times. Now please do not go and check the number of references at the end of this dissertation, dear reader. Please proceed by reading the next paragraph.*

In the more recent literature, various studies on priority queueing systems (in continuous and in discrete time) have been performed, e.g.[28, 29, 30, 31]. A complete overview up to 2003 can be found in the PhD dissertation [32]. In the last decade, the research in this domain has moved in several directions but we will not survey all of them immediately. The contributions focusing on general distributions for the service times are discussed in the next chapter of this dissertation. Furthermore, there is a huge interest in the cause of different types of tail probabilities, but, as this is the subject of chapter 4, we defer the discussion of the relevant literature until then. For now, let us focus on the queue capacity, this is the number of packets that can concurrently be waiting for service in the queue, which is the most distinctive feature of the model discussed in this chapter.

In practice, queueing systems have a limited queue capacity. In contrast, analytic studies of queueing systems generally assume infinite queue capacity as this facilitates the mathematical analysis of the system. Papers describing priority queues with finite capacity for both classes exist, but are far less numerous than their infinite capacity counterparts. The pioneering work in this area was performed in 1984 [33]. A more recent work [34] references most of the relevant papers in this area. Evidently, if the queue capacity is finite, one must detail what happens when the queue is completely full and an additional packet tries to enter the system. The packet can be dropped, it can push out another packet, etc. More information will be provided in part II of this dissertation, as this type of queueing systems are related to the one that is studied there.

Note 16. *The motivations for studying a system with limited class-1 queue capacity are not solely of intellectual but also of practical nature. Some systems have an obvious (physical) limit to the queue capacity. Also, enforcing a limited queue capacity provides a mechanism for mitigating the effect of the priority scheduling, thereby (partly) protecting class-2 customers from having to give priority to an endless stream of class-1 customers, as mentioned in the introductory chapter of this dissertation. On a more philosophical level one might even argue that infinite capacity queues do not exist, as the number of people standing in line is constrained by the global population, the number of emails in an inbox by computer storage, etc. However, infinite capacity models have proven to be very accurate approximations in many situations, but, as will become clear in this dissertation, this is not always the case.*

In this part of the dissertation, we will study a two-class discrete-time priority queueing model where the class-1 queue capacity is limited to N, which is a finite positive integer, but the class-2 capacity is infinitely large. We will refer to this system as the N/∞ priority queueing system. Furthermore, class-1 packets that find the class-1 queue to be full upon their arrival (as there are already N packets waiting for service) are dropped by the system. Note that, in contrast to the queueing systems mentioned in the previous paragraph, the state-space of the underlying Markov chain of an N/∞ priority queueing system is not finite.

Evidently, as N increases, the N/∞ priority queue is increasingly similar to a system where both queues are presumed to be of infinite capacity, which is studied in the paper [30] and the corresponding chapter in the PhD dissertation [32], which we will call the ∞/∞ priority queue. Investigating how (the performance measures of) the N/∞ system converge to those of the ∞/∞ system in the limit for $N \to \infty$ is the main goal of this part of the dissertation.

Note 17. *Therefore, we will often refer to the results of the corresponding ∞/∞ system studied in the dissertation [32], which can be downloaded from the author's website `http://telin.ugent.be/~jw/PhDthesis.pdf`.*

The N/∞ priority queue is considered in [35] as well, where packet loss ratio and tail behavior are analyzed using a matrix-analytic method. This method is perfectly suited for studying two-dimensional Markov processes if (at least) one of the dimensions is finite. However, this method is sometimes inaccurate for the ∞/∞ system and discovering how the limit for $N \to \infty$ works out is thus not completely solved. The methodology developed in the current chapter will provide extra insight in this limiting behavior, as will be shown in chapter 4. Furthermore, concurrently with the research presented here, the continuous-time equivalent of the N/∞ priority queue was studied in [36].

The remainder of this chapter is organized as follows: first, in section 2.2, the N/∞ priority queueing model is thoroughly described. Several subsequent sections each detail how to analyze a particular performance measure for the system. Next, section 2.7 demonstrates how one can calculate the moments of the performance measures. Then, the applicability of the analysis is exemplified through several numerical examples and finally the chapter is concluded with some closing remarks.

2.2 Model

We consider a discrete-time single-server priority queueing system with 2 classes, finite class-1 queue capacity N and an infinite class-2 queue. Class-1 packets are served with absolute priority over class-2 packets and within a class the queueing discipline is First-Come-First-Served (FCFS). First, in order to give some insights into the queuing model, let us consider a specific sample path of the system depicted in figure 2.1. On the left, the queueing system is depicted. One can discern the two queues and the server. The class-1 queue capacity is limited to $N = 3$. To the right, the evolution of the system content, influenced by arrivals and completed services, is depicted, aligned horizontally, during 20 slots. Class-1 (class-2) information is indicated in dark- (light-)grey and full (dotted) lines respectively. The queue content of both classes is shown on the positive vertical axis whereas the content of the server is visible on the negative one (the aggregation of queue content and server content is the system content). Arrows indicate service time durations (always a single slot). Time is divided into fixed-length slots corresponding to the transmission time of a packet and time progresses from left to right in the picture as can be deduced from the slot numbers. A packet can only enter the server at the beginning of a slot, even if it arrives in an empty system (slot 19). The Tail Drop queue management algorithm is used for the class-1 queue, hence the system accepts packets until the corresponding queue is entirely filled up and packets that arrive at a full queue are dropped by the system (slots 6 and 19).The system can contain up to $N+1$ class-1 packets simultaneously in a slot, N in the queue and 1

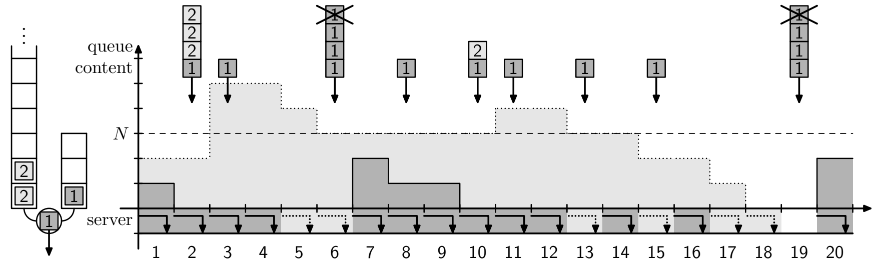

Figure 2.1: Evolution of the finite/infinite queueing system with $N = 3$ *over 20 slots. The content of both the high-priority queue (class-1 packets, dark grey) and low-priority queue (class-2 packets, light grey) is shown.*

in the server. Consequently, there are at most N class-1 packets in the system at the beginning of a slot. Also note that a class-1 packet thus resides in the system for at most N slots, which bounds its delay.

We assume that for both classes the numbers of arrivals in consecutive slots form a sequence of independent and identically distributed (i.i.d.) random variables. We define $a_{i,k}$ as the number of class-i ($i = 1,2$) packet arrivals during slot k. The arrivals of both classes are characterized by the joint probability mass function (pmf)

$$a(m,n) = \Pr\left[a_{1,k} = m, a_{2,k} = n\right] , \tag{2.1}$$

and joint probability generating function (pgf)

$$A(z_1, z_2) = \mathrm{E}\left[z_1^{a_{1,k}} z_2^{a_{2,k}}\right] . \tag{2.2}$$

Note that the arrival process allows correlation between both classes. Let the mean number of class-i arrivals per slot (class-i load) be

$$\lambda_i = \mathrm{E}\left[a_{i,k}\right] = \left.\frac{\partial A(z_1,z_2)}{\partial z_i}\right|_{z_1=1,z_2=1} , \quad (i = 1,2) . \tag{2.3}$$

The total (arrival) load equals $\lambda_T = \lambda_1 + \lambda_2$.

The bivariate pgf $A(z_1, z_2)$ is the standard representation of the arrival process for tackling two-dimensional queueing processes, such as the ∞/∞ system [32]. However, due to the limited class-1 queue capacity, the queueing process we study has an additional boundary requiring us to track class-1 more explicitly. Therefore, let us define the (partial) pgf of the class-2 arrivals in a slot with i (i or more) class-1 arrivals as $A_i(z)$ ($A_i^*(z)$), yielding

$$A_i(z) = \mathrm{E}\left[z^{a_{2,k}}\, 1\left\{a_{1,k} = i\right\}\right] , \qquad A_i^*(z) = \sum_{j=i}^{\infty} A_j(z) . \tag{2.4}$$

Note 18. *Recall that the indicator function 1*$\{x = i\}$ *is* 1 *if* $x = i$ *and equals* 0 *otherwise.*

In this dissertation, one will only encounter these pgfs for $i = 0, \ldots, N$ as the class-1 queue capacity is limited to N and hence at most N packets can be admitted into the queue during a slot.

It is immediately clear that the representations are related as

$$A(z_1, z_2) = \sum_{i=0}^{\infty} A_i(z_2) z_1^i \,. \tag{2.5}$$

Furthermore, note that, as $\Pr\left[a_{1,k} = i\right] = A_i(1)$, (2.3) is equivalent to

$$\lambda_1 = \sum_{i=1}^{\infty} i A_i(1) \,,\; \lambda_2 = \frac{d}{dz} A_0^*(z)\Big|_{z=1} = A_0^{*\prime}(1) \,. \tag{2.6}$$

2.3 System content

First, we study how the system content evolves from slot k to $k+1$. This is rather straightforward using pgfs as the process $\{(u_{1,k}, u_{2,k}), k \geq 1\}$ forms a Markov chain and thus a bivariate representation is sufficient. However, due to the limited class-1 capacity of N packets, rather than obtaining an expression for a bivariate pgf like in [32] a system of $N+1$ equations of partial pgfs is established. Next, the steady-state behaviour is investigated. Finally, as systems of equations are rather messy, a more intuitive matrix representation, which will be used throughout the entire dissertation, is presented.

2.3.1 Relating consecutive slots

Let the class-i system content at the beginning of slot k be denoted by $u_{i,k}$. As expected, we study the evolution of the queueing system in consecutive slots. Relating the system content at the beginning of slots k and $k+1$ yields

$$\begin{aligned} u_{1,k+1} &= \min(N, (u_{1,k}-1)^+ + a_{1,k}) \,, \\ u_{2,k+1} &= \begin{cases} (u_{2,k}-1)^+ + a_{2,k} & \text{if } u_{1,k} = 0 \,, \\ u_{2,k} + a_{2,k} & \text{if } u_{1,k} > 0 \,. \end{cases} \end{aligned} \tag{2.7}$$

Note 19. *Recall that* $(x)^+$ *denotes the maximum of* x *and* 0.

Clearly, as the class-1 capacity is bounded by N, the queue cannot always accommodate all arriving class-1 packets. Let the number of effective class-1 arrivals in slot k be denoted by $a_{1,k}^e$. This random variable is clearly influenced by the class-1 system content in slot k and can be characterized by

$$a_{1,k}^e = \min(a_{1,k}, N - (u_{1,k}-1)^+) \,. \tag{2.8}$$

Consequently, the evolution of the class-1 system content, as detailed in (2.7), can equivalently be expressed, in more traditional fashion, as

$$u_{1,k+1} = (u_{1,k}-1)^+ + a_{1,k}^e \,. \tag{2.9}$$

The (partial) pgf of the class-2 system content in slot k with class-1 system content equal to i is defined as

$$U_{i,k}(z) = \mathrm{E}\left[z^{u_{2,k}}\,1\left\{u_{1,k}=i\right\}\right]. \tag{2.10}$$

These pgfs can be related to eachother by the system equations (2.7). The pgf in slot $k+1$ is obtained as

$$\begin{aligned}
U_{i,k+1}(z) &= \mathrm{E}\left[z^{u_{2,k+1}}\,1\left\{u_{1,k+1}=i\right\}\right] \\
&= \mathrm{E}\left[z^{u_{2,k+1}}\,1\left\{u_{1,k+1}=i,\, u_{1,k}=0,\, u_{2,k}=0\right\}\right] \\
&\quad + \mathrm{E}\left[z^{u_{2,k+1}}\,1\left\{u_{1,k+1}=i,\, u_{1,k}=0,\, u_{2,k}>0\right\}\right] \\
&\quad + \mathrm{E}\left[z^{u_{2,k+1}}\,1\left\{u_{1,k+1}=i,\, u_{1,k}>0\right\}\right] \\
&= \mathrm{E}\left[z^{a_{2,k}}\,1\left\{\min(N,a_{1,k})=i,\, u_{1,k}=0,\, u_{2,k}=0\right\}\right] \\
&\quad + \mathrm{E}\left[z^{u_{2,k}-1+a_{2,k}}\,1\left\{\min(N,a_{1,k})=i,\, u_{1,k}=0,\, u_{2,k}>0\right\}\right] \\
&\quad + \mathrm{E}\left[z^{u_{2,k}+a_{2,k}}\,1\left\{\min(N,u_{1,k}-1+a_{1,k})=i,\, u_{1,k}>0\right\}\right].
\end{aligned} \tag{2.11}$$

Here, we first conditioned on the "state" of the server in slot k. There are three different cases: no service (system empty), class-2 packet in service (class-1 empty and class-2 non-empty), class-1 packet in service (class-1 non-empty). Then, in each of these cases, the relation between slots $k+1$ and k described in (2.7) becomes straightforward as the operator $(x)^+$ has vanished. However, due to the presence of the minimum operator, $U_{N,k+1}$, the pgf when the class-1 queue is entirely full, must be treated separately in order to proceed.

For $0 \le i < N$, we have

$$\begin{aligned}
U_{i,k+1}(z) &= \mathrm{E}\left[1\left\{u_{1,k}=0,\, u_{2,k}=0\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}=i\right\}\right] \\
&\quad + \frac{1}{z}\mathrm{E}\left[z^{u_{2,k}}\,1\left\{u_{1,k}=0,\, u_{2,k}>0\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}=i\right\}\right] \\
&\quad + \sum_{j=1}^{i+1}\mathrm{E}\left[z^{u_{2,k}}\,1\left\{u_{1,k}=j\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}=i+1-j\right\}\right] \\
&= U_{0,k}(0)A_i(z) + \frac{1}{z}\left(U_{0,k}(z)-U_{0,k}(0)\right)A_i(z) + \sum_{j=1}^{i+1}U_{j,k}(z)A_{i-j+1}(z).
\end{aligned} \tag{2.12}$$

As the arrivals during slot k are independent of the system content at the beginning of slot k, they can be separated into a different expectation operator. Also, note that, as at most one (class-1) packet can leave the system each slot, $u_{1,k+1}=i$ implies $u_{1,k} \le i+1$ which defines the range of the sum in the last term.

Analogously,

$$\begin{aligned} U_{N,k+1}(z) &= \mathrm{E}\left[1\left\{u_{1,k}=0,\, u_{2,k}=0\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}\geq N\right\}\right] \\ &\quad+\frac{1}{z}\mathrm{E}\left[z^{u_{2,k}}\,1\left\{u_{1,k}=0,\, u_{2,k}>0\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}\geq N\right\}\right] \\ &\quad+\sum_{j=1}^{N}\mathrm{E}\left[z^{u_{2,k}}\,1\left\{u_{1,k}=j\right\}\right]\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}\geq N+1-j\right\}\right] \\ &= U_{0,k}(0)A_N^*(z)+\frac{1}{z}\left(U_{0,k}(z)-U_{0,k}(0)\right)A_N^*(z) \\ &\quad+\sum_{j=1}^{N}U_{j,k}(z)A_{N+1-j}^*(z)\,. \end{aligned} \tag{2.13}$$

In this derivation, due to the limitation of the class-1 system content to N, the sum only runs to N and it is necessary to account for all arriving packets (also those that are dropped by the system) leading to the appearance of the pgfs $A_i^*(z)$.

Summarizing, the system content in slot $k+1$ is determined by

$$\begin{aligned} U_{i,k+1}(z) &= \frac{1}{z}U_{0,k}(z)A_i(z)+\frac{z-1}{z}U_{0,k}(0)A_i(z) \\ &\quad+\sum_{j=1}^{i+1}U_{j,k}(z)A_{i-j+1}(z), \quad i=0\ldots N-1, \\ U_{N,k+1}(z) &= \frac{1}{z}U_{0,k}(z)A_N^*(z)+\frac{z-1}{z}U_{0,k}(0)A_N^*(z) \\ &\quad+\sum_{j=1}^{N}U_{j,k}(z)A_{N-j+1}^*(z)\,. \end{aligned} \tag{2.14}$$

2.3.2 Steady state

Under the assumption that the system reaches steady state, on which we will elaborate later on, let us define

$$\begin{aligned} U_i(z) &= \lim_{k\to\infty}U_{i,k}(z)=\lim_{k\to\infty}U_{i,k+1}(z), \quad i=0\ldots N, \\ U(z_1,z_2) &= \lim_{k\to\infty}U_k(z_1,z_2)=\sum_{i=0}^{N}U_i(z_2)z_1^i\,. \end{aligned} \tag{2.15}$$

Furthermore, dropping the slot index k in the notation of a random variable indicates the corresponding random variable in a random slot in steady state, e.g. u_1 is the class-1 system content in a random slot in steady state. Accordingly, let us define u_2, a_1, a_2 and a_1^e. In steady-state, the system of equations (2.14) becomes

$$\begin{aligned} U_i(z) &= \frac{1}{z}U_0(z)A_i(z) + \frac{z-1}{z}U_0(0)A_i(z) \\ &\quad + \sum_{j=1}^{i+1} U_j(z)A_{i-j+1}(z), \quad i = 0 \ldots N-1, \\ U_N(z) &= \frac{1}{z}U_0(z)A_N^*(z) + \frac{z-1}{z}U_0(0)A_N^*(z) \\ &\quad + \sum_{j=1}^{N} U_j(z)A_{N-j+1}^*(z). \end{aligned} \tag{2.16}$$

2.3.3 Matrix representation

Systems of linear equations are more conveniently handled using matrices. In the representation used here, the elements of the vector/matrix express class-2 information through a partial pgf and the position of this element encodes the class-1 information. Let us define the row vector representing the system content in slot k by

$$\boldsymbol{u}_k(z) = \begin{bmatrix} U_{0,k}(z) & U_{1,k}(z) & \cdots & U_{N,k}(z) \end{bmatrix}. \tag{2.17}$$

Thus, $[\boldsymbol{u}_k(z)]_i$ is the partial pgf of the class-2 system content in slot k with class-1 system content equal to $i-1$.

Note 20. *Row- and column numbers of vectors (and matrices) are assumed to start at 1. Thus, e.g. the first element of the vector $\boldsymbol{u}_k(z)$ is $[\boldsymbol{u}_k(z)]_1 = U_{0,k}(z)$. Accordingly, the element on the i-th row, j-th column of a matrix $\boldsymbol{M}$ is given by $[\boldsymbol{M}]_{i,j}$.*

Accordingly, let us define the $(N+1)\times(N+1)$ "arrival matrix"

$$\boldsymbol{A}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z) \\ 0 & A_0(z) & \cdots & A_{N-2}(z) & A_{N-1}^*(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & A_0(z) & A_1^*(z) \\ 0 & \cdots & \cdots & 0 & A_0^*(z) \end{bmatrix}. \tag{2.18}$$

Adding arriving packets to the system content can thus be represented by the multiplication $\boldsymbol{u}_k(z)\boldsymbol{A}(z)$. Informally, for $1 \le i, j \le N+1$, given that the class-1 system content is $i-1$, $\boldsymbol{A}(1)_{ij}$ is the probability that $j-i$ class-1 packets are effectively allowed into the system and $\boldsymbol{A}(z)_{ij}$ is the corresponding partial pgf of the packets added to the class-2 queue.

Furthermore, for notational purposes, let us introduce the matrices

$$\boldsymbol{H}_0 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0}^T & \boldsymbol{O} \end{bmatrix}, \boldsymbol{H}_{>0} = \boldsymbol{I} - \boldsymbol{H}_0, \boldsymbol{D}_H = \begin{bmatrix} \mathbf{0} & 0 \\ \boldsymbol{I} & \mathbf{0}^T \end{bmatrix}. \tag{2.19}$$

Note 21. *Throughout this dissertation,* $\mathbf{0}$ *denotes a row vector of zeroes of appropriate size,* $\mathbf{x}^T$ *represents the transpose of vector* $\mathbf{x}$, $\boldsymbol{I}$ *and* $\boldsymbol{O}$ *respectively denote the identity matrix and the zero matrix of appropriate size.*

Right-multiplying by $\boldsymbol{H}_0$ and $\boldsymbol{H}_{>0}$ filters for "only the first column" and "all but the first column" respectively. Recall that whether or not a class-1 packet is in service is the defining factor for the evolution of the queuing system to the next slot. To that end, $\boldsymbol{u}_k(z)\boldsymbol{H}_0 = \begin{bmatrix} U_{0,k}(z) & 0 & \cdots & 0 \end{bmatrix}$ can be "informally" seen as "when $u_{1,k} = 0$" and $\boldsymbol{u}_k(z)\boldsymbol{H}_{>0} = \begin{bmatrix} 0 & U_{1,k}(z) & \cdots & U_{N,k}(z) \end{bmatrix}$ as "when $u_{1,k} > 0$" (the letter H was chosen as visual reminder that the matrix operations concern high-priority, i.e. class-1, and the subscripts were chosen for clarification purposes as well). Right multiplying by the matrix $\boldsymbol{D}_H$ shifts all elements to the left and thus $\boldsymbol{u}_k(z)\boldsymbol{D}_H = \begin{bmatrix} U_{1,k}(z) & \cdots & U_{N,k}(z) & 0 \end{bmatrix}$ "informally" can be stated to represent $u_{1,k} - 1$, i.e. a high-priority departure (again influencing the notation, $\boldsymbol{D}_H$).

In view of these definitions, (2.14) is identical to

$$\boldsymbol{u}_{k+1}(z) = \boldsymbol{u}_k(0)\boldsymbol{H}_0\boldsymbol{A}(z) + (\boldsymbol{u}_k(z) - \boldsymbol{u}_k(0))\,\boldsymbol{H}_0\frac{1}{z}\boldsymbol{A}(z) + \boldsymbol{u}_k(z)\boldsymbol{H}_{>0}\boldsymbol{D}_H\boldsymbol{A}(z)\,. \tag{2.20}$$

Again, the different terms can be seen to reflect the state of the server as the first term corresponds to no-service (system empty), the second to class-2 service (class-1 empty, class-2 not) and the final term to class-1 service. Rearranging leads to

$$\boldsymbol{u}_{k+1}(z) = \boldsymbol{u}_k(z)\left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(z) + \frac{z-1}{z}\boldsymbol{u}_k(0)\boldsymbol{H}_0\boldsymbol{A}(z)\,. \tag{2.21}$$

Note 22. *Although $\boldsymbol{H}_{>0}\boldsymbol{D}_H = \boldsymbol{D}_H$, I choose not to invoke this identity in order to increase the legibility of the expressions by "humans".*

Now, introducing the $(N+1)\times(N+1)$ matrix

$$\begin{aligned}\tilde{\boldsymbol{X}}(z) &= \left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(z)\\ &= \begin{bmatrix} \frac{1}{z}A_0(z) & \frac{1}{z}A_1(z) & \cdots & \frac{1}{z}A_{N-1}(z) & \frac{1}{z}A_N^*(z)\\ A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z)\\ 0 & A_0(z) & \cdots & A_{N-2}(z) & A_{N-1}^*(z)\\ \vdots & \ddots & \ddots & \vdots & \vdots\\ 0 & \cdots & 0 & A_0(z) & A_1^*(z)\end{bmatrix},\end{aligned} \tag{2.22}$$

allows us to write (2.21) as

$$\boldsymbol{u}_{k+1}(z) = \boldsymbol{u}_k(z)\tilde{\boldsymbol{X}}(z) + (z-1)\begin{bmatrix} U_{0,k}(0) & \boldsymbol{0}\end{bmatrix}\tilde{\boldsymbol{X}}(z)\,, \tag{2.23}$$

because

$$\begin{aligned}\begin{bmatrix} U_{0,k}(0) & \boldsymbol{0}\end{bmatrix}\tilde{\boldsymbol{X}}(z) &= \begin{bmatrix} U_{0,k}(0) & \boldsymbol{0}\end{bmatrix}\left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(z)\\ &= \boldsymbol{u}_k(0)\boldsymbol{H}_0\left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(z)\\ &= \boldsymbol{u}_k(0)\left(\frac{1}{z}\boldsymbol{H}_0\boldsymbol{H}_0 + \boldsymbol{H}_0\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(z)\\ &= \boldsymbol{u}_k(0)\left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{O}\right)\boldsymbol{A}(z)\,.\end{aligned} \tag{2.24}$$

However, the fractions in the first row of $\tilde{\boldsymbol{X}}(z)$ make manipulating this matrix rather tedious. As, in chapter 4, further investigation of this model is performed and such manipulations are frequently needed, an alternative but equivalent representation is introduced by defining

$$\boldsymbol{X}(z) = z\tilde{\boldsymbol{X}}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z) \\ zA_0(z) & zA_1(z) & \cdots & zA_{N-1}(z) & zA_N^*(z) \\ 0 & zA_0(z) & \cdots & zA_{N-2}(z) & zA_{N-1}^*(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & zA_0(z) & zA_1^*(z) \end{bmatrix}. \tag{2.25}$$

for which (2.23) becomes

$$\boldsymbol{u}_{k+1}(z) = \boldsymbol{u}_k(z)\frac{\boldsymbol{X}(z)}{z} + (z-1)\begin{bmatrix} U_{0,k}(0) & \boldsymbol{0} \end{bmatrix}\frac{\boldsymbol{X}(z)}{z}. \tag{2.26}$$

Consequently, the corresponding steady-state row vector

$$\boldsymbol{u}(z) = \lim_{k\to\infty} \boldsymbol{u}_k(z) = \begin{bmatrix} U_0(z) & U_1(z) & \cdots & U_N(z) \end{bmatrix}, \tag{2.27}$$

can be expressed by

$$\boldsymbol{u}(z) = \frac{1}{z}\boldsymbol{u}(z)\boldsymbol{X}(z) + \frac{z-1}{z}\begin{bmatrix} U_0(0) & \boldsymbol{0} \end{bmatrix}\boldsymbol{X}(z), \tag{2.28}$$

finally leading to

$$\boldsymbol{u}(z) = (z-1)\begin{bmatrix} U_0(0) & \boldsymbol{0} \end{bmatrix}\boldsymbol{X}(z)\big(z\boldsymbol{I} - \boldsymbol{X}(z)\big)^{-1}. \tag{2.29}$$

The unknown constant $U_0(0)$ is yet to be obtained. To that end, let us substitute $z = 1$ in (2.28) yielding $\boldsymbol{u}(1) = \boldsymbol{u}(1)\boldsymbol{X}(1)$ or, equivalently,

$$\boldsymbol{u}(1)\big(\boldsymbol{I} - \boldsymbol{X}(1)\big) = \boldsymbol{0}. \tag{2.30}$$

Note 23. *The identity $\boldsymbol{u}(1) = \boldsymbol{u}(1)\boldsymbol{X}(1)$ is not surprising at all. Considering class-1 in isolation is identical to studying a standard single-server FIFO queue with finite capacity, as presented in the introductory chapter of this dissertation, with steady-state vector $\boldsymbol{u}(1)$ and transition matrix $\boldsymbol{X}(1)$.*

However, $\boldsymbol{I} - \boldsymbol{X}(1)$ is not invertible, as $\boldsymbol{X}(1)$ is a right stochastic matrix. We have

$$\mathrm{Rank}\big(\boldsymbol{I} - \boldsymbol{X}(1)\big) = N. \tag{2.31}$$

We thus require one additional relation in order to determine the $N+1$ unknowns in the vector $\boldsymbol{u}(1)$. The normalisation property of probability distributions and recalling that $U_i(1) = \Pr[u_1 = i]$ lead to

$$\sum_{i=0}^{N} U_i(1) = 1 \Leftrightarrow \boldsymbol{u}(1)\boldsymbol{e} = 1. \tag{2.32}$$

By replacing one of the relations in (2.30) by this normalisation condition, the pmf of the class-1 system content is found to be

$$\boldsymbol{u}(1) = \begin{bmatrix} \boldsymbol{0} | 1 \end{bmatrix}\begin{bmatrix} \boldsymbol{I} - \boldsymbol{X}(1) | \boldsymbol{e} \end{bmatrix}^{-1}. \tag{2.33}$$

Note 24. *Recall that* $\boldsymbol{e}$ *denotes a column vector of ones of appropriate size. By* $[\boldsymbol{A}|\boldsymbol{b}]$ *we denote the matrix* $\boldsymbol{A}$ *with the last column replaced by the column vector* $\boldsymbol{b}$ *and by* $[\boldsymbol{a}|b]$ *the vector* $\boldsymbol{a}$ *with the last element replaced by* b*.*

We are now ready to determine the unknown constant $U_0(0)$, the probability that the system is empty, using theorem 1.1, as in the introductory chapter. In the current model, class-1 packets are not affected by class-2 packets, due to the priority scheduling, and hence class-1 can be studied separately. Consequently, we first apply this theorem on a queueing system where we neglect class-2 packets. The mean number of class-1 packets accepted by that system during a slot is denoted by λ_1^e. A class-1 packet is served by the system when the class-1 system content at the beginning of the slot is larger than 0. Consequently, invoking the theorem yields

$$\lambda_1^e = 1 - U_0(1)\,. \tag{2.34}$$

This quantity is known from (2.33) as $U_0(1)$ is simply the first element of $\boldsymbol{u}(1)$. Let λ_T^e be the total effective load, i.e. the mean number of packets accepted by the system, thus $\lambda_T^e = \lambda_1^e + \lambda_2$. Repeating the argument above for the complete system (containing packets of both classes) yields

$$\lambda_T^e = 1 - U_0(0)\,. \tag{2.35}$$

Combining (2.34) and (2.35) finally provides

$$U_0(0) = U_0(1) - \lambda_2\,. \tag{2.36}$$

The system content of both classes is thus completely expressed in terms of the arrival process (through $\boldsymbol{X}(z)$, λ_T^e) as (2.29) becomes

$$\boldsymbol{u}(z) = (1-\lambda_T^e)(z-1)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}\boldsymbol{X}(z)\big(z\boldsymbol{I}-\boldsymbol{X}(z)\big)^{-1}\,. \tag{2.37}$$

Note 25. *As the model studied here is of M/G/1-type, the stationary distribution could have been derived immediately using the famous Pollaczeck-Khinchine formula, see f.i. [37]. However, the analysis above was preferred to make this dissertation more self-containing and to use a similar solution method for both system content and delay.*

2.3.4 Stability

As the class-1 queue has finite capacity and excess packets are thus dropped the class-1 system is always stable. For the entire system to reach steady state it is required that the average number of class-2 packets that can be served exceeds the average number of class-2 arrivals, or that $\lambda_2 < 1 - \lambda_1^e$. Notice that requiring that $U_0(0) > 0$ is an equivalent stability constraint.

2.4 Queue content

The class-i queue content at the beginning of slot k, denoted by $q_{i,k}$, is defined as the number of class-i packets in the queue, thus in the system but not in the server. An explicit expression for the queue content will facilitate calculating the packet delay in the following sections. Evidently, the queue content is closely related to the system content as

$$\begin{aligned} q_{1,k} &= (u_{1,k}-1)^+\,, \\ q_{2,k} &= \begin{cases} (u_{2,k}-1)^+ & \text{if } u_{1,k}=0\,, \\ u_{2,k} & \text{if } u_{1,k}>0\,. \end{cases} \end{aligned} \tag{2.38}$$

The corresponding (partial) pgfs are defined by

$$Q_{n,k}(z) = \mathrm{E}\left[z^{q_{2,k}}\,1\{q_{1,k}=n\}\right], \quad n=0\dots N-1\,. \tag{2.39}$$

Recall that the queue can contain up to $N-1$ packets at the beginning of the slot. Thus (2.38) yields

$$\begin{aligned} Q_{0,k}(z) &= \mathrm{E}\left[z^{(u_{2,k}-1)^+}\,1\{u_{1,k}=0\}\right] + \mathrm{E}\left[z^{u_{2,k}}\,1\{u_{1,k}=1\}\right] \\ &= U_{0,k}(0) + \frac{1}{z}\left(U_{0,k}(z)-U_{0,k}(0)\right) + U_{1,k}(z)\,, \end{aligned} \tag{2.40}$$

and, for $n=1\dots N-1$,

$$Q_{n,k}(z) = \mathrm{E}\left[z^{u_{2,k}}\,1\{u_{1,k}=n+1\}\right] = U_{n+1,k}(z)\,. \tag{2.41}$$

Again, let us express these relations using matrices. Consider the vector

$$\boldsymbol{q}_k(z) = \begin{bmatrix} Q_{0,k}(z) & Q_{1,k}(z) & \cdots & Q_{N-1,k}(z) \end{bmatrix}. \tag{2.42}$$

Using this vector representation, combining (2.40) and (2.41) yields the steady-state (as before, this is indicated by dropping the slot index k from the notation) vector $\boldsymbol{q}(z)$ of the queue content of both classes, given by

$$\boldsymbol{q}(z) = \lim_{k\to\infty} \boldsymbol{q}_k(z) = \boldsymbol{u}(z)\left(\frac{1}{z}\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix} + \frac{z-1}{z}\begin{bmatrix} U_0(0) & \boldsymbol{0}\end{bmatrix}. \tag{2.43}$$

Note 26. *The* $(N+1)\times N$ *"resizing" matrix* $\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}$*, which removes the last element, resolves the difference in size between* $\boldsymbol{q}(z)$ *and* $\boldsymbol{u}(z)$*.*

2.5 Class-1 delay

The delay of a packet is defined as the number of slots during which it is present in the queueing system for the entire slot. This thus does not include the packet's arrival slot. The delay can be subdivided into two major parts: the time spent waiting in the queue and the service time.

2.5.1 Direct method

As stated earlier, class-1 packets are not affected by class-2 packets in the model under consideration and thus class 1 can be studied in isolation as a standard FIFO queueing system with finite capacity. Consider an infinitely long sample path of the system and arbitrarily tag a class-1 packet that is accepted into the system (the delay of dropped packets is undefined). As one choses a packet randomly (out of the infinitely large number of packets), with probability 1, the system is in steady state during its arrival slot. Let the delay of the tagged packet be denoted by d_1 and the arrival slot of the packet by k. The class-1 (packet) delay is given by

$$d_1 = q_{1,k} + \hat{a}_{1,k} + 1\,. \tag{2.44}$$

Here, $\hat{a}_{1,k}$ denotes the class-1 packets arriving in the same slot as, but before, the tagged packet.

Note 27. *For the delay, class 1 and 2 are studied separately, as choosing a class-1 packet at random and choosing a class-2 packet at random are completely unrelated. In contrast, when choosing a random slot, one can observe the system content of both classes in that random slot.*

The class-1 delay consists of at least one slot (the packet's service time), if the packet arrives in an empty queue, and at most N slots (service of the $N-1$ packets present upon the packet's arrival plus its own service), if the packet fills up the queue. Consequently, the pmf of the delay of an accepted class-1 packet in steady state is given by

$$\Pr\left[\, d_1 = n \mid \text{packet accepted}\right] = \frac{\Pr\left[d_1 = n, \text{packet accepted}\right]}{\Pr\left[\text{packet accepted}\right]}, \quad n = 1 \ldots N\,. \tag{2.45}$$

The denominator can be obtained easily, as, by a counting argument,the (long-run) probability that a packet is accepted equals

$$\Pr\left[\text{packet accepted}\right] = \lambda_1^e / \lambda_1\,. \tag{2.46}$$

In order to determine the numerator, it is crucial to note that the arrival slot of the tagged packet is not a random slot, e.g. it is impossible that no packets arrive (as the tagged one does). Randomly tagging a packet favours slots with a lot of arrivals. This classical renewal-theory inspection paradox is well-known, see a.o. [4, 38]. Let $\tilde{a}_{1,k}$ denote the number of arrivals in the arrival slot of the tagged packet ($\tilde{}$ indicates that slot k is not a random slot). Then, the corresponding pmf is given by

$$\Pr\left[\tilde{a}_{1,k} = n\right] = \frac{n \Pr\left[a_1 = n\right]}{\lambda_1}\,. \tag{2.47}$$

As the tagged packet is accepted into the system per definition, so are the packets arriving before it in the same slot ($\hat{a}_{1,k}$). However, packets arriving after the tagged packet (which are a subset of the packets in $\tilde{a}_{1,k}$) can potentially be dropped.

Now, we can proceed by noting that, for $n = 1 \dots N$,

$$\begin{aligned}
&\Pr\left[d_1 = n, \text{packet accepted}\right] \\
&= \Pr\left[q_{1,k} + \hat{a}_{1,k} + 1 = n, \text{packet accepted}\right] \\
&= \sum_{m=0}^{n-1} \Pr\left[q_{1,k} = m, \hat{a}_{1,k} = n - m - 1\right] \\
&= \sum_{m=0}^{n-1} \sum_{i=n-m}^{\infty} \Pr\left[\tilde{a}_{1,k} = i\right] \Pr\left[\, q_{1,k} = m, \hat{a}_{1,k} = n - m - 1 \mid \tilde{a}_{1,k} = i\right] \\
&= \sum_{m=0}^{n-1} \sum_{i=n-m}^{\infty} \frac{i \Pr\left[a_1 = i\right]}{\lambda_1} \Pr\left[\, q_{1,k} = m \mid \tilde{a}_{1,k} = i\right] \\
&\qquad\qquad \Pr\left[\, \hat{a}_{1,k} = n - m - 1 \mid q_{1,k} = m, \tilde{a}_{1,k} = i\right] \\
&= \frac{1}{\lambda_1} \sum_{m=0}^{n-1} \Pr\left[q_{1,k} = m\right] \sum_{i=n-m}^{\infty} i \Pr\left[a_1 = i\right] \frac{1}{i} \\
&= \frac{1}{\lambda_1} \sum_{m=0}^{n-1} \Pr\left[q_1 = m\right] A^*_{n-m}(1)\,.
\end{aligned} \tag{2.48}$$

The second transition considers all possible values for the queue content, which can maximally run up to $n - 1$ as the tagged packet's service slot also has to be incorporated in the delay. Furthermore, for the values of n and m considered here, $q_{1,k} = m$ and $\hat{a}_{1,k} = n - m - 1$ guarantee that the tagged packet is accepted. The third transition is obtained by considering all possible numbers of total arrivals (accepted or not) in the arrival slot of the tagged packet. Next, we invoke (2.47) and also use the definition of conditional probability. The fitfh transition holds as the queue content at the beginning of a slot ($q_{1,k}$) is independent of the total number of arrivals in that slot ($\tilde{a}_{1,k}$) and as choosing the tagged packet so that there are $n - m - 1$ packets arriving before it equals choosing a packet uniformly out of the i arriving packets. Finally, as we are in steady-state, the slot index k can be dropped because $q_{1,k}$ and q_1 are statistically indistinguishable.

Then, plugging (2.48) and (2.46) into (2.45) leads to

$$\Pr\left[\, d_1 = n \mid \text{packet accepted}\right] = \frac{1}{\lambda_1^e} \sum_{m=0}^{n-1} \Pr\left[q_1 = m\right] A^*_{n-m}(1), \quad n = 1 \dots N. \tag{2.49}$$

As before, we will revert to a matrix representation and consider system behaviour in steady state. The arguments made in (2.48) lead to the matrix representation $\frac{1}{\lambda_1} \hat{\boldsymbol{A}}_1$ for $\hat{a}_1$, with $\hat{\boldsymbol{A}}_1$ the $N \times N$ matrix

$$\hat{\boldsymbol{A}}_1 = \begin{bmatrix} A^*_1(1) & A^*_2(1) & \cdots & A^*_N(1) \\ 0 & A^*_1(1) & \cdots & A^*_{N-1}(1) \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & A^*_1(1) \end{bmatrix}. \tag{2.50}$$

Then, from (2.49), the (probability distribution) vector of the steady-state class-1 delay is found by

$$\begin{aligned}\boldsymbol{d}_1 &= \lim_{k\to\infty}\left[\Pr\left[d_1 = n\right]\right]_{n=1\ldots N}\\ &= \frac{1}{\lambda_1^e}\boldsymbol{q}(1)\hat{\boldsymbol{A}}_1\\ &= \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\left(\boldsymbol{H}_0 + \boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\hat{\boldsymbol{A}}_1\,.\end{aligned} \tag{2.51}$$

Note 28. *The paradox that by considering a randomly selected packet one does not consider a random slot is one of these "counter-intuitive" but evident results that make probability utterly fascinating to me (and I hope us, dear reader) but is often considered "witchcraft" by the general public.*

2.5.2 Distributional form of Little's Law

There is an alternative method for computing the delay of a random (accepted) class-1 packet using (an extension of) probably the most famous theorem in queueing theory.

Theorem 2.1 (Little's Law)**.** *The average number of customers in a system in steady state is equal to the average effective arrival rate multiplied by the average time a customer spends in the system.*

In the current model, this theorem thus states that $\mathrm{E}\left[u_1\right] = \lambda_1^e \mathrm{E}\left[d_1\right]$.

For simple queueing systems, such as the finite FIFO queue one considers when studying class-1 in isolation in the current queueing model, an even stronger result holds as not only the expectations but the entire distributions adhere to such a law. This distributional form of Little's Law is thoroughly detailed in [39, 40] and states that

$$\Pr\left[d_1 = n\right] = \frac{\Pr\left[u_1 = n\right]}{\lambda_1^e}, \quad n = 1\ldots N, \tag{2.52}$$

or, equivalently,

$$\boldsymbol{d}_1 = \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\boldsymbol{D}_H\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}. \tag{2.53}$$

Consequently, the right-hand sides of equations (2.51) and (2.53) should be proven to be identical. To that end, consider the $(N+1)\times(N+1)$ matrix, which is similarly structured as $\boldsymbol{A}(z)$,

$$\boldsymbol{A}^*(z) = \begin{bmatrix} A_0^*(z) & A_1^*(z) & \cdots & A_{N-1}^*(z) & A_N^*(z)\\ 0 & A_0^*(z) & \cdots & A_{N-2}^*(z) & A_{N-1}^*(z)\\ \vdots & \ddots & \ddots & \vdots & \vdots\\ \vdots & & \ddots & A_0^*(z) & A_1^*(z)\\ 0 & \cdots & \cdots & 0 & A_0^*(z)\end{bmatrix}, \tag{2.54}$$

and observe that then, (2.51) yields

$$\begin{aligned}
\boldsymbol{d}_1 &= \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\hat{\boldsymbol{A}}_1\\
&= \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\left(\boldsymbol{A}^*(1)-\boldsymbol{A}(1)\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \frac{1}{\lambda_1^e}\left(\boldsymbol{u}(1)\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}^*(1)-\boldsymbol{u}(1)\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}(1)\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \frac{1}{\lambda_1^e}\left(\boldsymbol{u}(1)\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}^*(1)-\boldsymbol{u}(1)\boldsymbol{X}(1)\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\left(\left(\boldsymbol{H}_0+\boldsymbol{H}_{>0}\boldsymbol{D}_H\right)\boldsymbol{A}^*(1)-\boldsymbol{I}\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}\\
&= \frac{1}{\lambda_1^e}\boldsymbol{u}(1)\left(\boldsymbol{D}_H+\boldsymbol{F}\right)\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{0}\end{bmatrix}.
\end{aligned} \tag{2.55}$$

Here, the fifth transition holds through (2.30) and the final one uses $A_0^*(1)=1$. Furthermore, the matrix $\boldsymbol{F}$ is given by

$$\boldsymbol{F}=\begin{bmatrix}
0 & A_1^*(1) & A_2^*(1) & A_3^*(1) & \cdots & A_{N-1}^*(z) & A_N^*(z)\\
0 & -A_0(1) & A_2^*(1) & A_3^*(1) & \cdots & A_{N-1}^*(z) & A_N^*(z)\\
\vdots & \ddots & -A_0(1) & A_2^*(1) & \cdots & A_{N-2}^*(z) & A_{N-1}^*(z)\\
\vdots & & \ddots & -A_0(1) & \ddots & \vdots & \vdots\\
\vdots & & & \ddots & \ddots & A_2^*(1) & A_3^*(1)\\
\vdots & & & & \ddots & -A_0(1) & A_2^*(1)\\
0 & \cdots & \cdots & \cdots & \cdots & 0 & -A_0(1)
\end{bmatrix}. \tag{2.56}$$

In the remainder of this subsection it is proven that $\boldsymbol{u}(1)\boldsymbol{F}=\boldsymbol{0}$ which asserts that (2.55) equals (2.53) and thus (2.51) equals (2.53). First, the general proof is detailed and then a practical example for $N=3$ is provided for clarification purposes.

Recall that class-1 in isolation is simply a finite Markov chain with $N+1$ states, as studied in the introductory chapter. The stationary distribution is found by requiring that the rates to and from a state need to be equal. Moreover, a similar result holds for "partitions" of the state space, resulting in "local balance" equations, first discovered in [41].

Theorem 2.2. *Consider a Markov chain with state space Ω. Select some subset of states Ω_{out}, and let $\Omega_{in}=\Omega\setminus\Omega_{out}$ denote its complement. In steady state, the rate from Ω_{in} to Ω_{out} equals the rate from Ω_{out} to Ω_{in}.*

Let p_j, $j=0\ldots N$, be the partition of the state space $\Omega=\{0,\ldots,N\}$ into the set of states $\Omega_{out}=\{j,\ldots,N\}$ and the set of states $\Omega_{in}=\Omega\setminus\Omega_{out}$. If partitioning happens according to partition p_{j-1}, $[\boldsymbol{F}]_{i,j}$ contains the sum of all transitions from state $u_1=i-1\in\Omega_{in}$ to any state in Ω_{out} minus the sum of all transitions to state $u_1=i-1\in\Omega_{in}$ from any state in Ω_{out} (recall that matrix indices start at 1). Consequently,

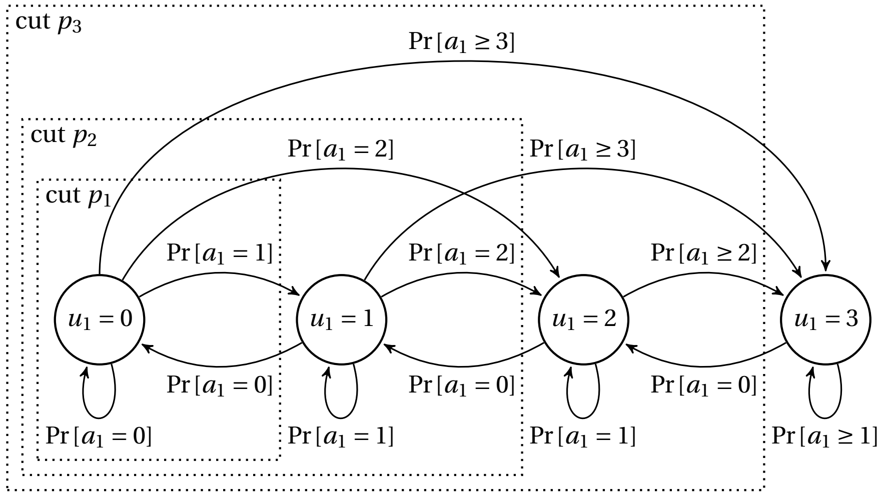

Figure 2.2: Markov states and transitions of the model for $N = 3$

$([\boldsymbol{u}(1)\boldsymbol{F})]_j$ characterizes the rates between Ω_{in} and Ω_{out} when these are partitioned according to p_j and theorem 2.2 thus asserts that $\boldsymbol{u}(1)\boldsymbol{F} = \boldsymbol{0}$, which completes our proof.

Let us consider the case $N = 3$, detailed in figure 2.2. For each partition, we have visibly indicated the "cut" that encapsulates the states in set Ω_{in}. Informally, one could say that the theorem states that the rates passing "through" the cut from inner to outer should cancel out those from outer to inner. Partition p_0 is trivial as then Ω_{out} encompasses all states and there are thus no transitions between Ω_{out} and $\Omega_{in} = \emptyset$. For partition p_1, the transitions $\Pr[a_1 = 1], \Pr[a_1 = 2], \Pr[a_1 \geq 3]$ go from Ω_{in} to Ω_{out} and the transition $\Pr[a_1 = 0]$ from Ω_{out} to Ω_{in}. The corresponding rates are found by incorporating the probability that the system is in the state from which the transition originates, leading to $\Pr[u_1 = 0]\Pr[a_1 \geq 1]$ from Ω_{in} to Ω_{out} and $\Pr[u_1 = 1]\Pr[a_1 = 0]$ from Ω_{out} to Ω_{in}. As the rates should be equal, we have

$$\Pr[u_1 = 0]\Pr[a_1 \geq 1] = \Pr[u_1 = 1]\Pr[a_1 = 0]\,, \tag{2.57}$$

or $[(\boldsymbol{u}(1)\boldsymbol{F})]_2 = 0$. For partition p_2 this argument leads to

$$\Pr[u_1 = 0]\Pr[a_1 \geq 2] + \Pr[u_1 = 1]\Pr[a_1 \geq 2] = \Pr[u_1 = 2]\Pr[a_1 = 0]\,, \tag{2.58}$$

which amounts to $[\boldsymbol{u}(1)\boldsymbol{F}]_3 = 0$, and for partition p_3 to

$$\begin{split}\Pr[u_1 = 0]\Pr[a_1 \geq 3] + \Pr[u_1 = 1]\Pr[a_1 \geq 3] + \Pr[u_1 = 2]\Pr[a_1 \geq 2]\\ = \Pr[u_1 = 3]\Pr[a_1 = 0]\,,\end{split} \tag{2.59}$$

which equals $[\boldsymbol{u}(1)\boldsymbol{F}]_4 = 0$.

2.6 Class-2 delay

Class-2 packets have to give priority to class-1 packets causing them to reside in the system for a (potentially) longer period of time as not only class-1 packets present in the system upon arrival of a class-2 packet, but also class-1 packets arriving while the class-2 packet waits in the queue are to be served before the class-2 packet. Consider an arbitrarily tagged class-2 packet. Let the delay of the packet be denoted by d_2 and the arrival slot of the packet by k. The class-2 (packet) delay is given by

$$d_2 = r_{1,k+1} + \sum_{i=1}^{q_{2,k}+\hat{a}_{2,k}} t_{2,k_i} + 1\,. \tag{2.60}$$

From the point of view of class 2, the system resembles a queueing system with server vacations as the server becomes unavailable for class 2 (takes a vacation) when class-1 packets are served. As the tagged class-2 packet arrives in slot k, its delay starts in slot $k+1$. The first part of the delay is the remaining class-1 busy period in slot $k+1$, $r_{1,k+1}$, as class-1 packets have priority over the tagged packet. It consists of the (single-slot) service times of $u_{1,k+1}$, the class-1 packets in the system at the beginning of slot $k+1$ and, as even future class-1 arrivals have priority over class-2 packets, the service times of the class-1 packets arriving during these service times, and the service times of the class-1 packets arriving during those service times, and etc. In short, it is the time until the system is void of class-1 packets for the first time after slot k. Then, the server becomes available for class-2 packets. Two "groups" of class-2 packets have to be served before the tagged packet. The first group is formed by the class-2 queue content in slot k, given by $q_{2,k}$. It consists of the class-2 packets in the system at the beginning of slot k, except the class-2 packet in service, if any, as that packet leaves the system by slot $k+1$ and thus does not contribute to the delay. The second group, represented by $\hat{a}_{2,k}$, consists of the class-2 packets arriving in the same slot as, but before, the tagged packet. Each class-2 packet to be served before the tagged packet contributes a (single-slot) service time to the tagged packet's delay. However, if class-1 packets arrive during this service slot, an entire class-1 busy period is added to the delay before the next class-2 packet can be served. This period, the service slot possibly extended with a class-1 busy period, is called the extended service completion time of a class-2 packet and is denoted by t_{2,k_i}. Here, k_i indicates the slot number of the service of the ith class-2 packet served after slot k. Finally, the tagged packet's own service slot completes its delay.

In order to clarify the components of the class-2 delay, a concrete example is portrayed in figure 2.3. Let the tagged packet be the second class-2 packet arriving in slot 2 (thus $k=2$). Its delay starts at the beginning of slot 3. As the system contains class-1 packets ($u_{1,3}=1$), the first contribution to the tagged packet's delay is the remaining class-1 busy period, $r_{1,3}=2$, consisting of the service times of the class-1 packet in the system and the class-1 packet arriving in slot 3. Next, we can start serving class-2 packets. Note that $q_{2,3}=2$ as there were two packets already waiting in the queue at the beginning of slot 2 and that the tagged packet

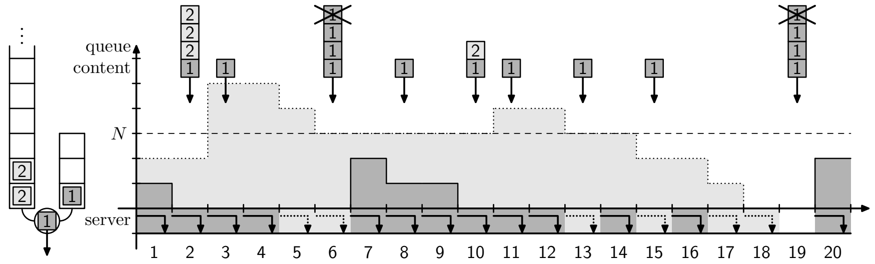

Figure 2.3: Sample path of the queueing system with N = 3 *over 20 slots. The content of both the high-priority queue (class-1 packets, dark grey) and low-priority queue (class-2 packets, light grey) is shown.*

was the second class-2 packet in its slot thus $\hat{a}_{2,2} = 1$. The first of these packets is served in slot 5 and no class-1 packets arrive in slot 5 so its extended service completion time equals its (single slot) service time ($t_{2,5} = 1$) and the next class-2 packet can be served in the next slot. However, that packet's extended service completion time, $t_{2,6}$, equals its service time (slot 6) and the entire class-1 busy period (slots 7 to 13) as four class-1 packets arrive at the system in slot 6 (of which three are accepted). Analogously, the final class-2 packet to be served before the tagged packet, the packet in $\hat{a}_{2,2}$, has an extended service time, $t_{2,13}$, consisting of its service time (slot 13) and a (in this case short) class-1 busy period (slot 14). Finally, the tagged class-2 packet is served in slot 15. Note that the class-1 packet arriving in slot 15 does not contribute to the delay. Summarizing, the delay is given by $d_2 = 2+1+7+2+1 = 13$ from slot 3 to slot 16.

Note 29. *Notice that "from slot k to slot l" denotes a time period that includes slot k and excludes slot l.*

In the remainder of this section, analytical expressions are obtained for the random variables involved in (2.60).

2.6.1 Remaining class-1 busy period

A (class-1) busy period is a period of consecutive slots with strictly-positive (class-1) system content or, equivalently, the number of slots between two consecutive instants where the system does not contain any (class-1) packets. Consequently, the remaining class-1 busy period in slot k, $r_{1,k}$, can be seen as the number of slots until the next slot with class-1 system content equal to 0. Evidently, it depends on the system content in slot k as a larger queue generally leads to a longer remaining busy period. Consequently, let us define the conditional pgf of the remaining class-1 busy period at the beginning of slot k, if the class-1 system content at the beginning of that slot equals n. We have

$$R_{1,k}(z|n) = \mathrm{E}\left[\, z^{r_{1,k}} \;\middle|\; u_{1,k} = n \right], \quad n = 0 \ldots N. \tag{2.61}$$

Note 30. *Remark that, here, the pgf tracks the number of slots instead of the number of packets.*

Recall that class-1 packets are unaffected by class-2 packets as they have priority. Evidently, $R_{1,k}(z|0) = 1$ as the class-1 busy period ends when the class-1 queue is empty. For $n = 1 \dots N$, relating slots k and $k+1$ yields

$$\begin{aligned}
R_{1,k}(z|n) &= \mathrm{E}\left[z^{r_{1,k}} \mid u_{1,k} = n \right] \\
&= \mathrm{E}\left[z^{1+r_{1,k+1}} \mid u_{1,k} = n \right] \\
&= z \sum_{m=0}^{\infty} \mathrm{E}\left[z^{r_{1,k+1}}\, 1\left\{a_{1,k} = m\right\} \mid u_{1,k} = n \right] \\
&= z \sum_{m=0}^{\infty} \mathrm{E}\left[z^{r_{1,k+1}} \mid u_{1,k} = n, a_{1,k} = m \right] \mathrm{E}\left[1\left\{a_{1,k} = m\right\} \right] \\
&= z \left(\sum_{m=0}^{N-n} \mathrm{E}\left[z^{r_{1,k+1}} \mid u_{1,k+1} = n-1+m \right] \Pr\left[a_{1,k} = m\right] \right. \\
&\quad \left. + \mathrm{E}\left[z^{r_{1,k+1}} \mid u_{1,k+1} = N \right] \Pr\left[a_{1,k} \geq N-n+1\right] \right) \\
&= z \left(\sum_{m=0}^{N-n} R_{1,k+1}(z|n-1+m) A_m(1) + R_{1,k+1}(z|N) A^*_{N-n+1}(1) \right).
\end{aligned} \tag{2.62}$$

Evidently, the remaining busy period in slot k is one slot longer than in slot $k+1$ (if $u_{1,k} = n > 0$). Next, we differentiate between all possible arrival patterns in slot k through the law of total probability. Then, observe that, for the remaining busy period in slot $k+1$, the pair of random variables $(u_{1,k}, a_{1,k})$ contains the same information as $u_{1,k+1}$. Finally, note the separate treatment of a completely full queue.

Again, let us introduce a more convenient vector notation. Consider following (column!) vector

$$\boldsymbol{r}_{1,k}(z) = \left[R_{1,k}(z|0) \cdots R_{1,k}(z|N) \right]^T. \tag{2.63}$$

Then, (2.62) leads to

$$\boldsymbol{r}_{1,k}(z) = \begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}^T + z\boldsymbol{H}_{>0}\boldsymbol{X}(1)\boldsymbol{r}_{1,k+1}(z). \tag{2.64}$$

Again, notice the intuitiveness of the matrix representation. The first term covers the case $u_{1,k} = 0$ and thus $R_{1,k}(z|0) = 1$, which marks the end of the busy period, whereas the second term starts with $\boldsymbol{H}_{>0}$, selecting for $u_1 > 0$, thus for a continuation of the busy period, which consists of the single elapsed slot k (z), during which the evolution of the class-1 queue content is governed by $\boldsymbol{X}(1)$ and of the remaining busy period at the beginning of the next slot $(\boldsymbol{r}_{1,k+1}(z))$.

Note 31. *Remark that, as matrix-multiplication is not commutative, the order of the vectors/matrices has to be consistent with the evolution of time in order to correctly encode the evolution of the class-1 queue. Therefore, the expression above is of the form $\boldsymbol{r}_{1,k}(z) \sim \boldsymbol{X}(1)\boldsymbol{r}_{1,k+1}(z)$ as we express slot k in terms of slot $k+1$ whereas, in section 2.3, the system content was determined by expressing slot $k+1$ in terms of slot k yielding $\boldsymbol{u}_{k+1}(z) \sim \boldsymbol{u}_k(z)\boldsymbol{X}(z)$.*

Consequently, as $\boldsymbol{H}_{>0}\boldsymbol{X}(1) = \boldsymbol{D}_H\boldsymbol{A}(1)$ the steady-state vector of conditional pgfs of the remaining class-1 busy period given the system content is given by

$$\boldsymbol{r}_1(z) = \lim_{k\to\infty} \boldsymbol{r}_{1,k}(z) = \left(\boldsymbol{I} - z\boldsymbol{D}_H\boldsymbol{A}(1)\right)^{-1} \begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}^T . \tag{2.65}$$

2.6.2 Class-1 busy period

A class-1 busy period starts in slot k if, in the previous slot, the class-1 system content is zero and the number of class-1 arrivals is greater than zero. Let us denote the length of this period by $b_{1,k}$. We have

$$\begin{aligned} B_{1,k}(z) &= \mathrm{E}\left[\, z^{b_{1,k}} \;\middle|\; u_{1,k-1} = 0\,, a_{1,k-1} > 0\right] \\ &= \sum_{n=1}^{\infty} \mathrm{E}\left[\, z^{b_{1,k}}\, 1\{a_{1,k-1} = n\} \;\middle|\; u_{1,k-1} = 0\,, a_{1,k-1} > 0\right] \\ &= \sum_{n=1}^{N-1} \mathrm{E}\left[\, z^{r_{1,k}} \mid u_{1,k} = n\right] \Pr\left[\, a_{1,k-1} = n \mid a_{1,k-1} > 0\right] \\ &\quad + \mathrm{E}\left[\, z^{r_{1,k}} \mid u_{1,k} = N\right] \Pr\left[\, a_{1,k-1} \geq N \mid a_{1,k-1} > 0\right] \\ &= \sum_{n=1}^{N-1} R_{1,k}(z|n) \frac{A_n(1)}{1 - A_0(1)} + R_{1,k}(z|N) \frac{A_N^*(1)}{1 - A_0(1)} . \end{aligned} \tag{2.66}$$

Notice that, once the conditions for starting a busy period are fulfilled, the busy period evidently equals the remaining busy period. Hence, in steady-state, the class-1 busy period is given by

$$\begin{aligned} B_1(z) = \lim_{k\to\infty} B_{1,k}(z) &= \frac{1}{1 - A_0(1)} \left(\sum_{n=1}^{N-1} R_1(z|n) A_n(1) + R_1(z|N) A_N^*(1) \right) \\ &= \frac{1}{1 - \begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix} \boldsymbol{A}(1)\boldsymbol{H}_0\boldsymbol{e}} \begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix} \boldsymbol{A}(1)\boldsymbol{H}_{>0}\boldsymbol{r}_1(z) . \end{aligned} \tag{2.67}$$

2.6.3 Extended service completion time

The extended service completion time of a class-2 packet starts when the packet starts service and lasts until the next slot wherein a class-2 packet can be served. Let $t_{2,k}$ denote the extended service completion time of a class-2 packet starting service in slot k. We have

$$t_{2,k} = \begin{cases} 1 & \text{if } a_{1,k} = 0\,, \\ 1 + b_{1,k+1} & \text{if } a_{1,k} > 0\,. \end{cases} \tag{2.68}$$

If no class-1 packets arrive during the service-slot of the packet, the server can handle another class-2 packet in the next slot. If there are class-1 arrivals, we have to wait for a class-1 busy period after the service-slot until the service of another class-2 packet can start. The corresponding pgf is given by

$$\begin{aligned} T_{2,k}(z) &= \mathrm{E}\left[z^{t_{2,k}} \right] \\ &= \Pr\left[a_{1,k} = 0\right] z + \Pr\left[a_{1,k} > 0\right] z B_{1,k+1}(z) . \end{aligned} \tag{2.69}$$

We can thus express t_2, the extended service completion time of a class-2 packet in steady state, through its pgf

$$T_2(z) = \lim_{k\to\infty} T_{2,k}(z) = A_0(1)z + \big(1 - A_0(1)\big)zB_1(z)\,. \tag{2.70}$$

Note 32. *The notion of extended service completion times stems from queueing systems with server vacations [42]. Such a queueing system typically has one class of customers and a server that can be unavailable. It is used to model real-life phenomena such as repair in a production line, temporary shut-down of transmission for energy savings purposes in wireless communications, etc. An extended service completion time encapsulates both the service time and the vacation time. In order to asses the performance of this system, an alternative (standard) queuing system (without vacations!) is considered with service time set to the extended service completion time in the original queueing model. For many performance metrics these two models behave indistinguishably and the formulas for the standard model are evidently readily available. This methodology cannot be applied to the model studied here, as it generally requires the arrival process to be independent of the vacation process. In the current model, the server vacations perceived by class-2 packets are caused by class-1 packets and the arrivals of the two types of packets can be correlated. However, the concept of extended service completion times remains very useful to us.*

2.6.4 Arrivals in the same slot as the tagged packet to be served before it

The following analysis is similar to that in subsection 2.5.1 but the total number of class-1 arrivals ($a_{1,k}$) needs to be tracked simultaneously with the number of class-2 arrivals before the tagged packet ($\hat{a}_{2,k}$) as all arriving class-1 packets need to be served before the tagged packet due to the priority scheduling. Furthermore, as the arrivals of both classes are correlated it is clear that $a_{1,k}$ and $\hat{a}_{2,k}$ are correlated as well. Recall that slot k, the arrival slot of the tagged class-2 packet, is not a random slot, as it is more likely to have more class-2 arrivals, but that the system is in steady state. Let $\tilde{a}_{2,k}$ denote the number of arriving class-2 packets in this slot k. Analogous to (2.47), we have

$$\Pr\big[a_{1,k} = m, \tilde{a}_{2,k} = n\big] = \frac{n\Pr[a_1 = m, a_2 = n]}{\lambda_2}\,. \tag{2.71}$$

Now, the (joint) probability of the class-1 arrivals and the class-2 arrivals before the tagged packet can be obtained as

$$\begin{aligned}\Pr\big[a_{1,k} = m, \hat{a}_{2,k} = n\big] &= \sum_{i=n+1}^{\infty} \Pr\big[\,\hat{a}_{2,k} = n \,\big|\, \tilde{a}_{2,k} = i\big]\Pr\big[a_{1,k} = m, \tilde{a}_{2,k} = i\big]\\ &= \sum_{i=n+1}^{\infty} \frac{\Pr[a_1 = m, a_2 = i]}{\lambda_2}\,.\end{aligned} \tag{2.72}$$

Again, the selection of the tagged class-2 packet such that there are n class-2 packets arriving before it, boils down to choosing uniformly from the i arriving packets and, as the system is in steady state, the slot index k could be dropped because the respective random variables are statistically indistinguishable. We define the corresponding (partial) pgfs by

$$\begin{aligned}\hat{A}_i(z) &= \sum_{n=0}^{\infty} \Pr\left[a_{1,k}=i,\, \hat{a}_{2,k}=n\right] z^n = \frac{A_i(z)-A_i(1)}{\lambda_2(z-1)}\,,\\ \hat{A}_i^*(z) &= \sum_{l=i}^{\infty} \hat{A}_l(z) = \frac{A_i^*(z)-A_i^*(1)}{\lambda_2(z-1)}\,.\end{aligned} \tag{2.73}$$

Then, the corresponding matrix, characterizing the (class-1 and class-2) arrivals in the same slot as the tagged class-2 packet, that are to be served before it, is given by

$$\hat{\boldsymbol{A}}_2(z) = \frac{\boldsymbol{A}(z)-\boldsymbol{A}(1)}{\lambda_2(z-1)}\,. \tag{2.74}$$

2.6.5 Class-2 delay

Finally, we have adequate tools for characterizing the class-2 delay. From (2.60), $D_2(z)$, the pgf of the delay of a (randomly tagged) class-2 packet, with arrival slot k, is obtained as

$$\begin{aligned}D_2(z) &= \mathrm{E}\left[z^{d_2}\right]\\ &= \sum_{n=0}^{N} \mathrm{E}\left[z^{d_2}\, 1\{u_{1,k+1}=n\}\right]\\ &= \sum_{n=0}^{N} \mathrm{E}\left[z^{r_{1,k+1}+\sum_{i=1}^{q_{2,k}+\hat{a}_{2,k}} t_{2,k_i}+1}\, 1\{u_{1,k+1}=n\}\right]\\ &= z\sum_{n=0}^{N} \mathrm{E}\left[z^{r_{1,k+1}} \mid u_{1,k+1}=n\right] \mathrm{E}\left[z^{\sum_{i=1}^{q_{2,k}+\hat{a}_{2,k}} t_{2,k_i}}\, 1\{u_{1,k+1}=n\}\right]\\ &= z\sum_{n=0}^{N} R_{1,k+1}(z|n) \mathrm{E}\left[z^{\sum_{i=1}^{q_{2,k}+\hat{a}_{2,k}} t_{2,k_i}}\, 1\left\{q_{1,k}+a^e_{1,k}=n\right\}\right].\end{aligned} \tag{2.75}$$

The last transition holds as $q_{1,k}=(u_{1,k}-1)^+$ and thus $u_{1,k+1}=q_{1,k}+a^e_{1,k}$. Now, observe that the information concerning slot $k+1$ and slot k has been separated into two distinct expectations. Then, as we are in steady state, the slot indices (k and $k+1$) can be dropped in each part. Furthermore, the successive extended service completion times are i.i.d., with pgf $T_2(z)$ given by (2.70). Recalling (1.19), one can then proceed by

$$\begin{aligned}
D_2(z) &= z \sum_{n=0}^{N} R_1(z|n) \mathrm{E}\left[T_2(z)^{q_2+\hat{a}_2}\, 1\{q_1 + a_1^e = n\}\right] \\
&= z \left(\sum_{n=0}^{N-1} R_1(z|n) \sum_{m=0}^{n} \mathrm{E}\left[T_2(z)^{q_2}\, 1\{q_1 = m\}\right] \mathrm{E}\left[T_2(z)^{\hat{a}_2}\, 1\{a_1 = n-m\}\right] \right. \\
&\quad \left. + R_1(z|N) \sum_{m=0}^{N-1} \mathrm{E}\left[T_2(z)^{q_2}\, 1\{q_1 = m\}\right] \mathrm{E}\left[T_2(z)^{\hat{a}_2}\, 1\{a_1 \geq N-m\}\right] \right) \\
&= z \sum_{m=0}^{N-1} \mathrm{E}\left[T_2(z)^{q_2}\, 1\{q_1 = m\}\right] \left(\sum_{n=m}^{N-1} \mathrm{E}\left[T_2(z)^{\hat{a}_2}\, 1\{a_1 = n-m\}\right] R_1(z|n) \right. \\
&\quad \left. + \mathrm{E}\left[T_2(z)^{\hat{a}_2}\, 1\{a_1 \geq N-m\}\right] R_1(z|N) \right) \\
&= \begin{bmatrix} \boldsymbol{q}\big(T_2(z)\big) & 0 \end{bmatrix} \hat{\boldsymbol{A}}_2\big(T_2(z)\big) \boldsymbol{r}_1(z) z .
\end{aligned} \tag{2.76}$$

Note that, in the second transition, one has to take into account that q_1 cannot exceed $N-1$. In the next transition, we interchange both sums.

Note 33. *Again, notice that the order of the matrices mimics the evolution of time.*

2.7 Calculating performance measures

From the expressions obtained in the previous sections, several performance measures can be derived. Typically, for class-1 this is straightforward. For instance, as the entire distribution of the class-1 system content is provided by the finitely sized vector $\boldsymbol{u}(1)$, computing moments of u_1 is trivial. Furthermore, from the class-1 system content we easily obtain the class-1 packet loss ratio plr_1. This is the fraction of class-1 packets that arrive at the system but are dropped. We have

$$plr_1 = \frac{\lambda_1 - \lambda_1^e}{\lambda_1} = 1 - \frac{1 - U_0(1)}{\lambda_1} . \tag{2.77}$$

Deriving performance measures is less straightforward for class 2. The moment-generating property of pgfs enables determination of the moments of the class-2 system content. First, in order to find the mean, $\mathrm{E}[u_2] = \boldsymbol{u}'(1)\boldsymbol{e}$, let us compute the first derivative of (2.37) yielding

$$\boldsymbol{u}'(z)\big(z\boldsymbol{I} - \boldsymbol{X}(z)\big) + \boldsymbol{u}(z)\big(\boldsymbol{I} - \boldsymbol{X}'(z)\big) = (1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}\big(\boldsymbol{X}(z) + (z-1)\boldsymbol{X}'(z)\big). \tag{2.78}$$

Then, setting $z = 1$ implies

$$\boldsymbol{u}'(1)\big(\boldsymbol{I} - \boldsymbol{X}(1)\big) + \boldsymbol{u}(1)\big(\boldsymbol{I} - \boldsymbol{X}'(1)\big) = (1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}\boldsymbol{X}(1). \tag{2.79}$$

However, again, as $\boldsymbol{I} - \boldsymbol{X}(1)$ is not invertible, an additional relation is required. To that end, let us take the second derivative of (2.37) producing

$$\begin{aligned}
&\boldsymbol{u}''(z)\big(z\boldsymbol{I} - \boldsymbol{X}(z)\big) + 2\boldsymbol{u}'(z)\big(\boldsymbol{I} - \boldsymbol{X}'(z)\big) - \boldsymbol{u}(z)\boldsymbol{X}''(z) \\
&\qquad = (1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}\big(2\boldsymbol{X}'(z) + (z-1)\boldsymbol{X}''(z)\big).
\end{aligned} \tag{2.80}$$

Now, putting $z=1$ yields

$$\boldsymbol{u}''(1)\big(\boldsymbol{I}-\boldsymbol{X}(1)\big)+\boldsymbol{u}'(1)2\big(\boldsymbol{I}-\boldsymbol{X}'(1)\big)-\boldsymbol{u}(1)\boldsymbol{X}''(1)=(1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}2\boldsymbol{X}'(1)\,. \tag{2.81}$$

Multiplying both sides of the equation with $\boldsymbol{e}$ makes the term in $\boldsymbol{u}''(z)$ vanish as $(\boldsymbol{I}-\boldsymbol{X}(1))\boldsymbol{e}=\boldsymbol{0}$. This is quite convenient as it completes our quest for an additional relation for $\boldsymbol{u}'(1)$. We have

$$\boldsymbol{u}'(1)2\big(\boldsymbol{I}-\boldsymbol{X}'(1)\big)\boldsymbol{e}-\boldsymbol{u}(1)\boldsymbol{X}''(1)\boldsymbol{e}=(1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}2\boldsymbol{X}'(1)\boldsymbol{e}\,. \tag{2.82}$$

Finally, combining (2.79) and (2.82), while recalling the $[\boldsymbol{A}|\boldsymbol{b}]$ notation defined in note 24, leads to

$$\begin{aligned}\mathrm{E}[u_2]&=\boldsymbol{u}'(1)\boldsymbol{e}\\&=\Big(\boldsymbol{u}(1)\big[\boldsymbol{X}'(1)-\boldsymbol{I}\big|\boldsymbol{X}''(1)\boldsymbol{e}\big]+(1-\lambda_T^e)\begin{bmatrix}1 & \boldsymbol{0}\end{bmatrix}\big[\boldsymbol{X}(1)\big|2\boldsymbol{X}'(1)\boldsymbol{e}\big]\Big)\\&\qquad\Big[\boldsymbol{I}-\boldsymbol{X}(1)\Big|2\big(\boldsymbol{I}-\boldsymbol{X}'(1)\big)\boldsymbol{e}\Big]^{-1}\boldsymbol{e}\,.\end{aligned} \tag{2.83}$$

Note that the variance of the arrival process appears in this equation (through $X''(1)$), which is consistent with the ∞/∞ model [32, p. 25].

Completely analogously, higher-order moments are obtained. The variance, $\mathrm{Var}[u_2]=\boldsymbol{u}''(1)\boldsymbol{e}+\mathrm{E}[u_2]-\mathrm{E}[u_2]^2$, requires the computation of $\boldsymbol{u}''(1)$, which is found from (2.81). Again, as $\boldsymbol{I}-\boldsymbol{X}(1)$ is not invertible, calculating the third derivative of (2.37), setting $z=1$ and multiplying by $\boldsymbol{e}$ yields the required additional relation. This procedure can be repeated ad infinitum to compute all moments. It allows for very efficient computations as it expresses the moments in terms of derivatives of $\boldsymbol{X}(z)$ evaluated in z=1. Thus, we can directly express the moments of the performance measures in terms of real numbers, costly symbolic inversion is not needed, and computations are thus very efficient.

2.8 Numerical examples

In the numerical examples, we will primarily focus on topics that identify the impact of the limited class-1 queue capacity. This is achieved by comparing the results for the N/∞ priority queue to those for the ∞/∞ priority queue. Other interesting topics on two-class priority queues, e.g. the impact of correlation between the two arrival classes, are extensively detailed throughout the literature. In the remainder of this dissertation, the values for the performance measures for the ∞/∞ priority queue were calculated using the expressions in [32].

Note 34. *Let us introduce the shorthand notation "finite case" and "infinite case" for the N/∞ and the ∞/∞ priority queue respectively.*

2.8.1 Output-queueing switch

In order to illustrate how the type of research performed in this dissertation might be applied, let us consider a relatively practical example. Therefore, let us first study

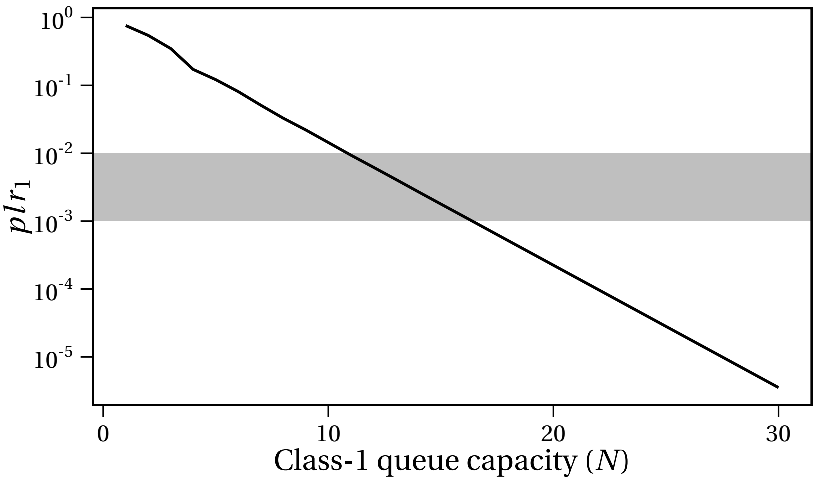

Figure 2.4: Class-1 Packet loss ratio vs. class-1 queue capacity.

one of the outputs of an 8 × 8 output-queueing switch of a computer network, detailed in appendix A.4. Assume the batches to have size $b = 4$ and arrive with probability $v_1 = v_2 = 0.1$ yielding $\lambda_1 = \lambda_2 = 0.4$. On average the system thus receives the same amount of packets of each class.

First, let us study the performance measures, using these parameters, versus the class-1 queue capacity. Figure 2.4 depicts the class-1 packet loss ratio versus the class-1 queue capacity N. Obviously the packet loss decreases with increasing N. The region between 10^{-2} and 10^{-3}, which we have marked in grey, is particularly interesting as, in a practical computer networking setting, most real-time applications tolerate this amount of packet loss. In this region, we will show that system content and packet delay, which are evidently exact for the finite case, are not accurately approximated by the infinite case, hence identifying a practical use for the queueing model developed in this chapter. A packet loss ratio over 10^{-2} causes the Quality of Service (QoS) delivered to real-time applications to be unacceptable and is hence impractical in a DiffServ setting. Systems with a very small packet loss ratio ($plr_1 << 10^{-3}$) are accurately modelled by the infinite case.

In figure 2.5, the mean and standard deviation of the delay of both classes are plotted versus the class-1 queue capacity N. We clearly see the effect of priority scheduling as the low mean and standard deviation for the class-1 delay demonstrate the performance boost, at the cost of the class-2 performance measures. The values increase for increasing N, as the number of dropped class-1 packets decreases. For larger N, the values clearly approach the values corresponding with the infinite case, which are represented by the horizontal dotted lines. This validates that, for N going to infinity, the finite case converges to the infinite case, as the number of dropped class-1 packets tends to zero. In the region with 10^{-2} <

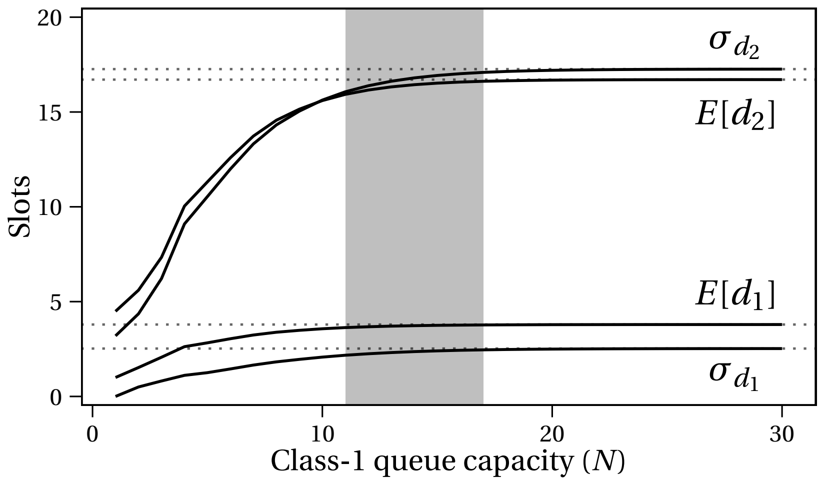

Figure 2.5: Moments of class-1/2 delay vs. class-1 queue capacity.

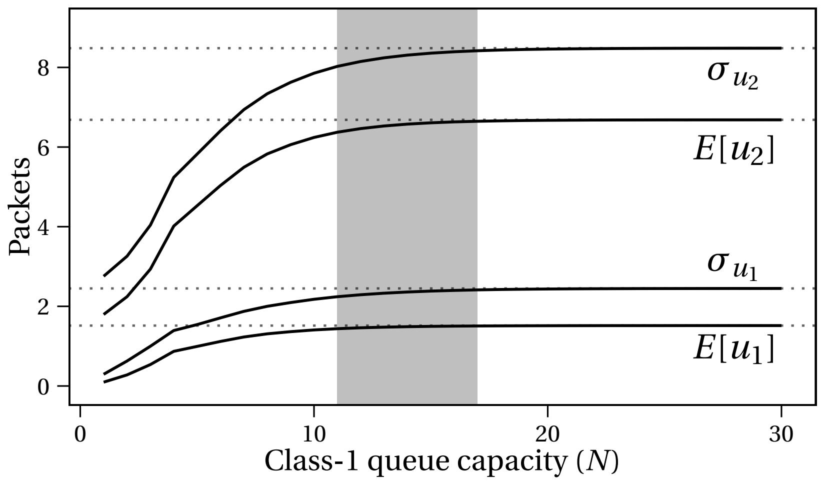

Figure 2.6: Moments of class-1/2 system content vs. class-1 queue capacity.

$plr_1 < 10^{-3}$, determined in figure 2.4 and again marked in grey here, the infinite system considerably overestimates the mean and standard deviation of the delay of both classes. For instance, at $N = 11$, the mean and standard deviation of the class-2 system content are overestimated by 5-7%. Figure 2.6 depicts the moments of the system content in the same scenario. When comparing these figures, one can “literally“ see Little’s Law at work, as the respective means have similar graphs.

Note 35. *Notice that for the standard deviations the graphs are not similar, even for class-1 where the distributional form of Little’s Law holds.*

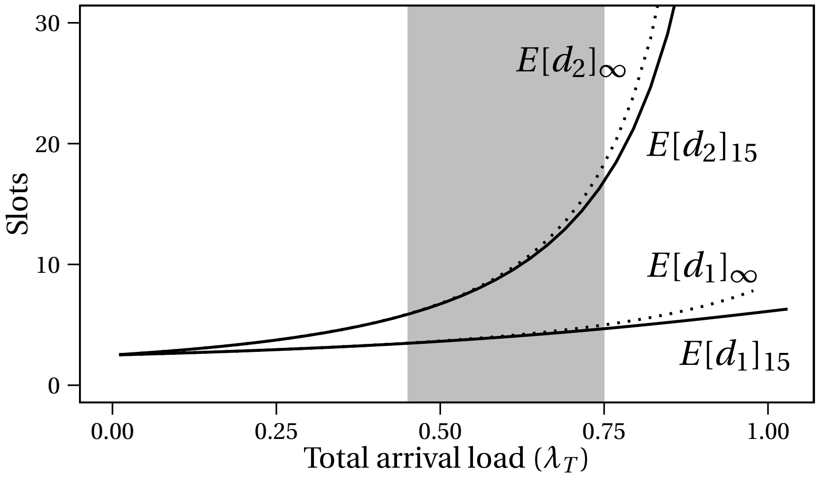

Figure 2.7: Mean class-1/2 delay vs. total load, with $\alpha = 0.75$, *for class-1 queue capacity 15 and* ∞.

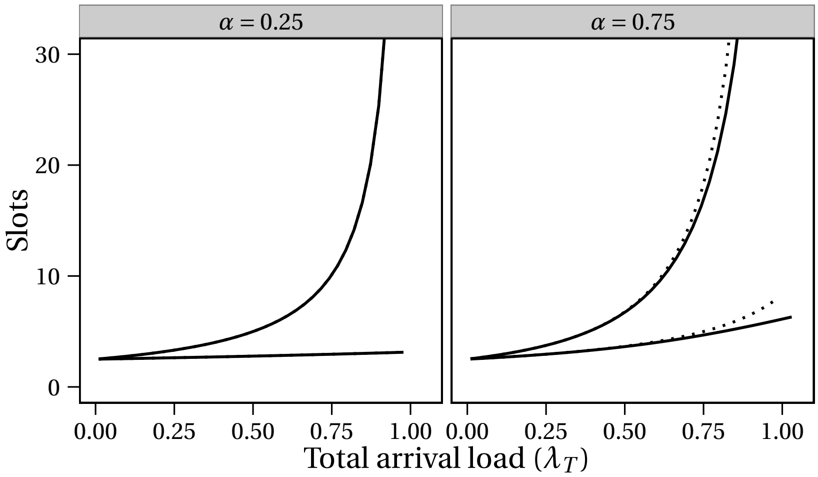

Figure 2.8: Comparison of figure 2.7 with the same scenario but with $\alpha = 0.25$.

Next, define the "arrival mix" α to be the fraction of class-1 packets out of all arriving packets, thus

$$\alpha = \lambda_1/\lambda_T\,. \tag{2.84}$$

For figure 2.7, let us fix $N = 15$ and set $\alpha = 0.75$, meaning that 3 out of 4 arriving packets are of class 1. Thus, we vary the total arrival load λ_T by varying ν_1, ν_2, while keeping $\nu_1 = 3\nu_2$ and plot the mean system content of both classes. Furthermore, we have again marked the region where $10^{-3} < plr_1 < 10^{-2}$, in grey. The effect of the priority scheduling is apparent, class-2 packets reside in the system for a much longer time than class-1 packets. Especially note the long class-2 delays as the load increases. For $\lambda_T = 0.75$, where the packet loss ratio plr_1 approaches the 1% boundary, the error introduced by using the infinite model over our model amounts to 7% and 10% overestimation for class-1 and class-2 respectively. Also note that the total load can exceed 1 in the finite case (and not in the infinite case) as excess class-1 packets are dropped. Here, it can run up to approximately 1.03. This is caused by the fact that the finite system is stable as long as $\lambda_2 < 1 - \lambda_1^e$.

Note 36. *Notice that, even for very low load, the delays are considerably longer than a single slot due to the batch size* $b = 4$.

Now, through figure 2.8, the influence of α is investigated by comparing figure 2.7 (on the right) to the scenario where all parameters are the same except now $\alpha = 0.25$ (on the left). Evidently, for $\alpha = 0.25$, as there are less class-1 packets in the system (and more class-2 packets), class-1 packet loss is reduced. It is even so low ($plr_1 << 10^{-3}$) that there is hardly any difference between the finite and the infinite cases. Furthermore, increasing α, thus increasing the number of class-1 packets and decreasing the number of class-2 packets), increases the delays for both(!) class-1 and class-2 packets. For the former, this is due to the queueing effect, for the latter due to the priority mechanism. Finally, note that the region where the system is stable is smaller for $\alpha = 0.25$, λ_T can hardly exceed 1.

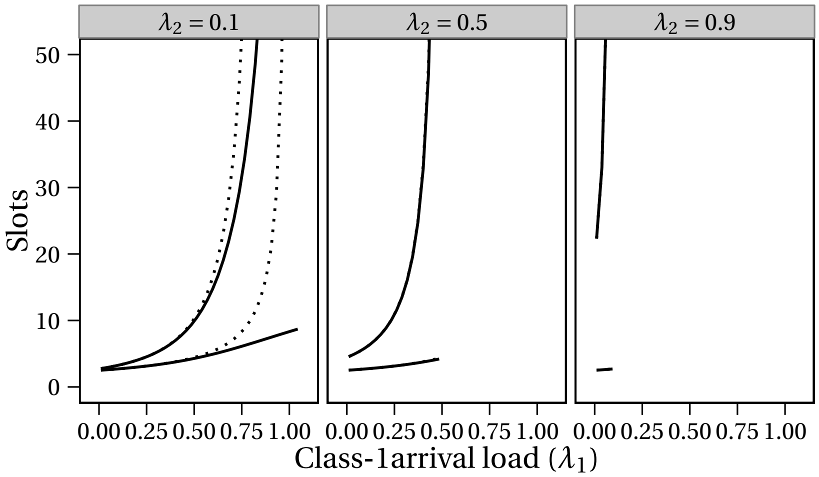

Figure 2.9: Mean class-1/2 delay vs. class-1 load for class-1 queue capacity 15 and ∞.

Figure 2.9 exhibits the region of stability for different arrival loads. It plots the class-1/2 mean delay versus the class-1 load for, from left to right, class-2 load equal to 0.1, 0.5, 0.9 respectively. The dotted lines, representing the infinite case, evidently stop at $\lambda_T = 1$ thus at $\lambda_1 = 0.9, 0.5, 0.1$ respectively. For $\lambda_2 = 0.1$, the finite case is stable up to $\lambda_1 = 1.05$, meaning the system can support a total load of 115%

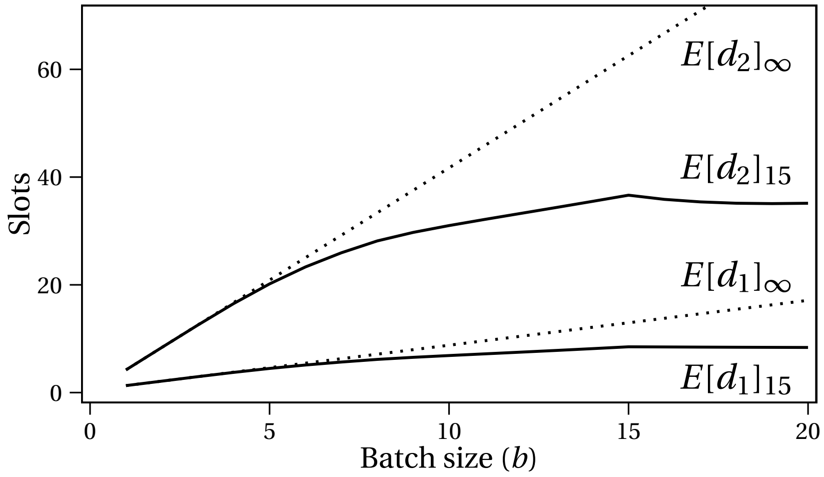

Figure 2.10: Mean class-1/2 delay vs. batch size for class-1 queue capacity 15 and ∞*.*

($\lambda_T = 1.15$). For $\lambda_2 = 0.5$, the system can barely support λ_T over 100% ($\lambda_1 = 0.51$) and for $\lambda_2 = 0.9$ the excess supported load is negligible. Evidently, this difference is caused by the difference in packet loss and, as packet loss causes λ_1^e to be smaller than λ_1, this directly follows from the stability condition $\lambda_2 < 1 - \lambda_1^e$. Furthermore, one again sees that there only is a noticeable difference between the finite and infinite case if the fraction of class-1 packets (α) is large enough.

Finally, let us study the effect of the variance of the arrival process. Assume class-1 queue capacity $N = 15$ and $\lambda_1 = \lambda_2 = 0.4$. We vary the batch size b while adjusting the v_i accordingly in order to keep the arrival load λ_i constant. For increasing b the system thus receives the same amount of packets but the variance of the number of arrivals increases, as packets are more clumped together. In figure 2.10, we depict the mean delays of both classes, as well as the mean delays of the infinite case (as dotted lines), versus the batch size b. We clearly see that the delay increases and that the infinite case leads to inaccurate results when the variance in the arrival process increases. The decrease of the mean delays for $b \geq 15$ can be attributed to a high loss rate since for $b \geq 15$ the batch size exceeds the class-1 queue capacity.

2.8.2 Poisson and power-law arrivals

Now, we verify if the results from the previous subsection, where the arrival process was driven by the switch, are also applicable for other arrival processes. Therefore, let us investigate the queueing system studied in this chapter when arrivals occur according to processes that are traditionally used in the queueing literature. The Poisson distribution is the most frequently used distribution for describing the arrival process of a queuing system. In contrast, in real-life, power-law arrival pro-

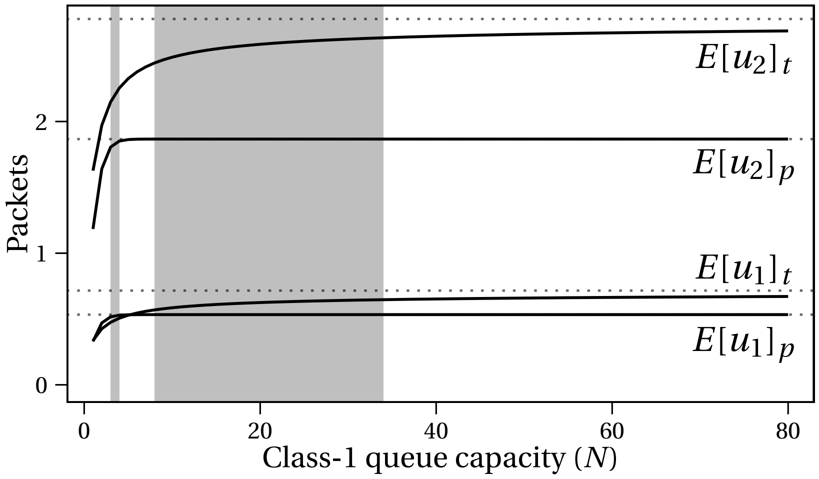

Figure 2.11: Moments of class-1/2 system content vs. class-1 queue capacity.

cesses are very common (see appendix A.3). Therefore, let us compare the impact of using these arrival processes. Let us assume that the arrivals of class 1 and class 2 are mutually independent. Then, the expressions describing the bivariate arrival process ($A_i(z)$) for the Poisson and for the power-law arrival process are detailed in appendix A.1.1 and A.3.1 respectively. In the figures in this subsection, we will use the subscripts p and t to indicate Poisson and power-law (heavy-tailed) arrivals respectively.

Figure 2.11 depicts the mean system content for both classes vs. the class-1 queue capacity, where, for the Poisson arrivals, we let $\lambda_1 = \lambda_2 = 0.4$ and, for the power-law arrivals, we let $\gamma_1 = \gamma_2 = 3.5$, hence the mean and variance exist but the higher moments are infinitely large, and in order to also have $\lambda_1 = \lambda_2 = 0.4$, we set $\beta_1 = \beta_2 = 0.3359655818$. The mean numbers of arrivals are thus the same as in figure 2.6 for the switch, allowing for a comparison of both figures (only for the means as the standard deviation is infinite for the power-law arrivals). Clearly, notice that Poisson arrivals yield a smaller mean system content for both classes than power-law arrivals, which in turn is smaller than those for the switch arrival process. This is caused by the increasing amounts of variances of these processes (in the switch case, the batch arrivals cause high variance). One can see that this is also true for the infinite case, where the mean system content is completely determined by the mean and variance of the arrival process [32, p. 25]. Again, regions with moderate packet loss have a grey background. As demonstrated in figure 2.12, the grey area for $N = 3-4$ marks the region where $10^{-3} < plr_1 < 10^{-2}$ for Poisson arrivals, whereas the other grey area for $N = 8-34$ corresponds to power-law arrivals. In figure 2.11, also notice that convergence to the infinite case is almost immediate with Poisson arrivals but, more importantly, that the convergence is very slow with

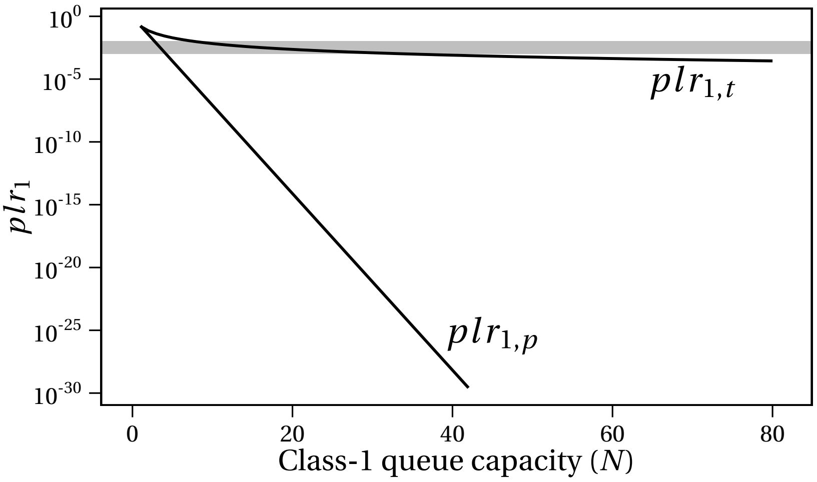

Figure 2.12: Class-1 packet loss ratio system content vs. class-1 queue capacity.

power-law arrivals. For $N = 30$, the overestimation by the infinite case amounts to 5% for class 1 and 11% for class 2 and it remains considerably large at 3% and 6% respectively for $N = 80$. This is caused by the heavy tail of the class-1 arrival distribution, which is carried over to the class-1 packet loss ratio, which is apparent in 2.12. Even for packet loss ratios smaller than 10^{-4}, the difference between the infinite and the finite case is considerably large as most of the class-1 packets arriving in case of a "tail event" are dropped by the system, even when N is large. Therefore, in the finite case, not only mean and variance of the arrival process, but the entire arrival process and also the class-1 queue capacity N have a major influence on system performance.

Finally, in order to further investigate the impact of power-law arrivals, let us reconsider the situation studied in figure 2.7, but with power-law arrivals, again with $\gamma_1 = \gamma_2 = 3.5$. In 2.13, we vary the total arrival load λ_T by varying β_1 and β_2, while keeping $\beta_1 = 3\beta_2$ and plot the mean system content of both classes. It looks similar to figure 2.7 but, for all values of λ_T, it holds that $10^{-3} < plr_1 < 10^{-2}$ and the entire background is thus grey. Also, note that the stability region is smaller for this arrival process. The divergence for higher loads is clear but the difference is smaller than with the switch arrival process, as the arrival process has a smaller variance. However, here, there is already a difference between the finite and the infinite case for moderate values of λ_T, caused by the heavy-tailednesss of the arrival process. Due to the ubiquity of power-law processes in practical applications, the results of this subsection are of significant importance.

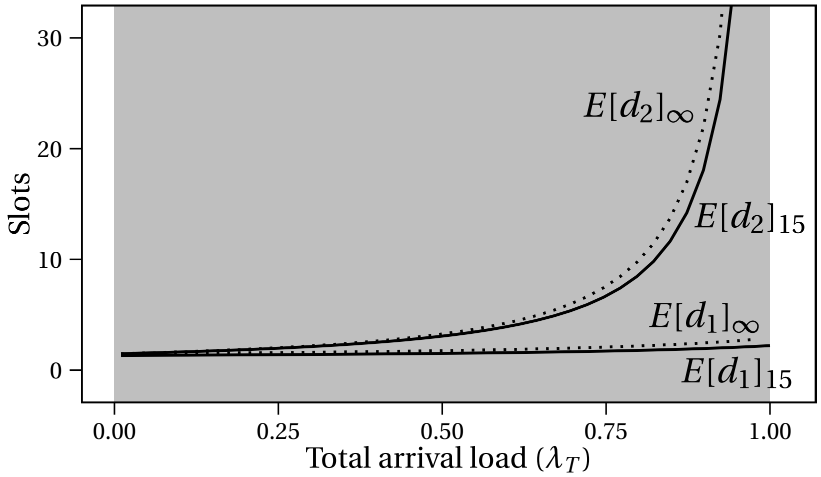

Figure 2.13: Mean class-1/2 delay vs. total load, with $\alpha = 0.75$*, for class-1 queue capacity 15 and* ∞*.*

2.9 Concluding remarks

This chapter exhibits the methodology for studying a two-class priority queue with finite class-1 queue capacity. The analysis simultaneously takes place in the probability domain for class-1 and in the transform domain for class2 through the use of a vector/matrix representation where the elements are partial generating functions tracking class-2 while the position of an element encodes class-1 information. This enables the determination of various performance measures of such a queueing system from the probability mass functions of the high-priority (class 1) and the probability generating functions of the low-priority (class 2) system content and delay, one derives through the analysis. Under certain conditions (small queue capacity, relatively high class-1 load, power-law arrivals), our results are considerably different from the ones obtained if one assumes infinite class-1 queue capacity, as is standard in the literature, hence justifying our queueing model.

3

N/∞ Non-preemptive Priority Queue - General Service Times

3.1 Introduction

In many real-life queueing systems, not all customers require the same (fixed) service time. Evidently, the model with single-slot service times studied in previous chapter cannot (accurately) describe such systems. Therefore, this chapter extends that model by treating the service times as random variables that follow any general distribution. Moreover, in priority systems, customers can have different service requirements from class to class and thus the service times follow a (potentially different) general distribution for each class.

Here, we will study a non-preemptive priority queue, which means that class-2 service is modeled to be uninterruptible. This means that a class-1 packet arriving at the system while a class-2 packet is in service, thus when there are no class-1 packets in the system, has to wait (potentially for multiple slots) until the ongoing class-2 service is finished, before it can enter the server. Consequently, the class-1 system cannot be studied separately as a FIFO queue, as detailed in the previous chapter, because class-1 performance depends on the class-2 system content and service time distribution.

Numerous studies of non-preemptive and preemptive (which has different variants depending on whether an interrupted service can be resumed, has to be be restarted completely or a mixture of these) priority queues with infinite capacity have been performed with all kinds of arrival processes. Recall that the first chapter on priority queues was written in 1954 [25]. Furthermore, Takagi's book [4, vol. 3] treats several of these models and the PhD dissertation [32] surveys the literature, such as [43, 31], extensively. To the best of our knowledge, priority queues with general service distributions and finite queue capacities were first studied in [33, 44]. More work in this area is found, a.o. in [45] and in [46], with Markovian arrival processes.

This chapter is structured in the same manner as the previous one. First, the N/∞ priority queueing model is detailed. Next, the analysis of the system content and delay of both classes is performed. Finally, the applicability of the formulas is demonstrated through some numerical examples and the chapter is concluded by some summarizing remarks.

3.2 Model

This chapter studies a discrete-time single-server two-class non-preemptive priority queueing system where class-1 (real-time) packets receive absolute priority over class-2 (data) packets. Packets are handled in a FIFO manner within a class. We limit the capacity of the class-1 queue to N packets such that real-time packets that arrive at a full queue are dropped by the system. The system can hence contain up to $N+1$ class-1 packets simultaneously, N in the queue and 1 in the server. In contrast, the class-2 queue has infinite capacity. Time is divided into fixed-length slots and a packet can only enter the server at slot boundaries, even if arriving in an empty system.

Let s_j $(j = 1,2)$ denote a generic random service time of a class-j packet . These independent variables have corresponding mean values μ_j $(j = 1,2)$ and pgfs

$$S_j(z) = \sum_{n=1}^{\infty} \Pr\left[s_j = n\right] z^n\,, \tag{3.1}$$

where the sum starts at 1 as service times are assumed to take at least one slot ($\Pr\left[s_j = 0\right] = 0$), which is standard in the queueing literature. As in the previous chapter, observing the system at the beginning of a slot happens after the departure (if any) at the slot boundary but before arrivals in the subsequent slot.

Note 37. *In the queueing literature, μ generally denotes the service rate so that the mean service time is $1/\mu$. However, we have chosen our notation to match that of [32], facilitating the comparison of equations.*

We assume that, for both classes, the numbers of arrivals in consecutive slots form a sequence of independent and identically distributed (i.i.d.) random variables. We define $a_{i,k}$ as the number of class-i $(i = 1,2)$ packet arrivals during slot k. The arrivals of both classes are characterized by the joint probability mass function (pmf)

$$a(m,n) = \Pr\left[a_{1,k} = m, a_{2,k} = n\right] \tag{3.2}$$

which allows us to take into account dependence between both classes. The corresponding joint probability generating function (pgf) is denoted by

$$A(z_1,z_2)=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}a(i,j)z_1^iz_2^j\,. \tag{3.3}$$

The partial pgf of the number of class-2 arrivals in a slot with i and i or more class-1 arrivals are respectively denoted by $A_i(z)$ and $A_i^*(z)$. We establish

$$A_i(z)=\mathrm{E}\left[z^{a_{2,k}}\,1\left\{a_{1,k}=i\right\}\right]=\sum_{j=0}^{\infty}a(i,j)z^j\,,\ A_i^*(z)=\sum_{j=i}^{\infty}A_j(z)\,. \tag{3.4}$$

The mean number of class-1 and class-2 arrivals per slot are respectively expressed as

$$\lambda_1=\sum_{i=1}^{\infty}iA_i(1)\,,\ \lambda_2=\frac{d}{dz}A_0^*(z)\Big|_{z=1}=A_0^{*\prime}(1)\,. \tag{3.5}$$

Hence, the loads per class are $\rho_1=\lambda_1\mu_1$ and $\rho_2=\lambda_2\mu_2$. Furthermore, the total mean number of arrivals and the total load are respectively denoted by $\lambda_T=\lambda_1+\lambda_2$ and $\rho_T=\rho_1+\rho_2$.

3.3 System content

Again, let the class-i ($i=1,2$) system content at the beginning of slot k be denoted by $u_{i,k}$. For this model, one can no longer directly relate the system content of both classes in consecutive slots as the process $\{(u_{1,k},u_{2,k}),k\geq 1\}$ no longer forms a Markov chain because the service process, which evidently has an effect on the system, spans multiple slots. One popular approach to tackling this issue introduces supplementary random variables, $x_{i,k}$, denoting the remaining number of slots required for the service of the class-i packet in service in slot k (if any). Then, the four-dimensional process $\{(u_{1,k},u_{2,k},x_{1,k},x_{2,k}),k\geq 1\}$ is a Markov chain.

Another popular approach, the path by which we will travel, considers a process that tracks the system content of both classes only at specific slots rather than at all slots. These specific slots, slots where a packet starts service or the system is empty, are called start-slots. Let $n_{i,l}$ denote the class-i system content at the beginning of start-slot l. The start-slots are chosen in such a way that the process $\{(n_{1,l},n_{2,l}),l\geq 1\}$ forms a Markov chain allowing for an analysis along the lines of the one in the previous chapter. These slots are also called embedded points and the process the “embedded” Markov chain. Once the properties of the embedded process are known, one can easily study the process $\{(u_{1,k},u_{2,k}),k\geq 1\}$.

Note 38. *I prefer the embedded Markov chain approach over the supplementary variable method as it divides the problem in several sub-problems which are tackled separately and, as added bonus, often yield intermediate results with a clear probabilistic interpretation.*

This section proceeds as follows. First, the operation of the system is detailed through the study of a specific sample path and the relation between the system content of both classes at the beginning of start-slots is established. The next subsection addresses the characterization of arrivals during a class-i service. This enables determination of the system content at start-slots in subsection 3.3.3. Next, the system content at the beginning of random slots is derived from those at start-slots but this requires several intermediate results which make up the preceding subsections.

3.3.1 Relating consecutive start-slots

A start-slot is a slot where the server is available for starting service of a packet, thus if a service actually starts or the system is empty. In figure 3.1, the evolution of the system is exemplified for a specific sample path in order to clarify the concept of start-slots and give some insights into the system studied in this chapter. On the left, the queueing system is depicted for $N = 3$. To its right, the evolution of the system content, influenced by arrivals and completed services, is depicted, aligned horizontally, during 37 slots. Class-1 (class-2) information is indicated in dark- (light-)grey and full (dotted) lines respectively. Time evolves on the horizontal axis and the queue content of both classes is shown on the positive vertical axis whereas the state of the server is visible on the negative one. For the server, arrows indicate service time durations and start-slots are highlighted with a '•'. The following events are particularly interesting. In slot 10, class-1 packets arrive in a system void of class-1 packets. Although class-1 packets have priority, class-1 service can only start after the class-2 service in progress is completed as the priority is non-preemptive. Therefore, class-1 performance is dependent on class-2 traffic, in contrast with the model studied in the previous chapter. Slot 12 exemplifies that the class-1 queue can only hold N packets, while slot 26 demonstrates that packets cannot enter the server upon arrival, but only at slot boundaries. Moreover, slot 26 clarifies that the class-1 system content is limited to N at start-slots, whereas, at the non-start-slots 28, 29 and 32 it can amount to $N+1$ if the queue is full and a class-1 packet is in service. Remark that a slot where the system is empty at the beginning of the slot is a start-slot as well (slots 25 and 26). Furthermore, notice the dotted line in slot 32 visualizing the class-2 queue content "behind" the class-1 content (as the former is smaller).

Evidently, the time period between the beginning of two consecutive start-slots consists of s_1, s_2 or a single slot(s), depending on the type of packet in the server (if any). Therefore, study of the evolution of the system during a service time is of paramount importance to the analysis. Let $e_{i,j,k}$ represent the number of class-i arrivals during a class-j service that starts in slot k. We have

$$e_{i,j,k} = \sum_{m=0}^{s_j-1} a_{i,k+m} \,. \tag{3.6}$$

Recall that $n_{i,l}$ denotes the class-i system content at the beginning of start-slot l. The process $\{(n_{1,l}, n_{2,l}), l \geq 1\}$ forms a Markov chain. Assume that start-slot l

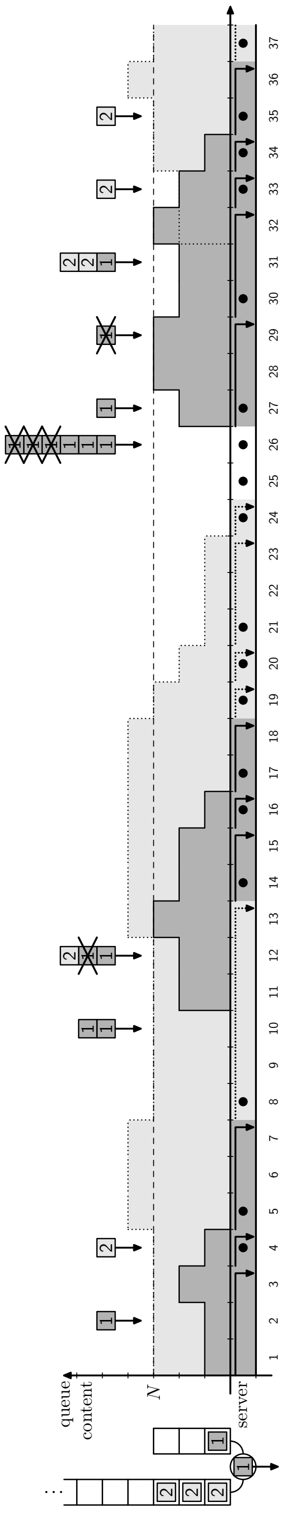

Figure 3.1: Evolution of the finite/infinite non-preemptive queueing system with $N = 3$ over 37 slots.

corresponds with slot k. Relating start-slots l and $l+1$ establishes the set of system equations

$$
\begin{aligned}
n_{1,l+1} &= \begin{cases} \min(N, a_{1,k}) & \text{if } n_{1,l}=0, n_{2,l}=0\,, \\ \min(N, e_{1,2,k}) & \text{if } n_{1,l}=0, n_{2,l}>0\,, \\ \min(N, n_{1,l}-1+e_{1,1,k}) & \text{if } n_{1,l}>0\,, \end{cases} \\
n_{2,l+1} &= \begin{cases} a_{2,k} & \text{if } n_{1,l}=0, n_{2,l}=0\,, \\ n_{2,l}-1+e_{2,2,k} & \text{if } n_{1,l}=0, n_{2,l}>0\,, \\ n_{2,l}+e_{2,1,k} & \text{if } n_{1,l}>0\,, \end{cases}
\end{aligned}
\tag{3.7}
$$

The system equations can be explained as follows: if the system is empty, start-slot $l+1$ is slot $k+1$, thus only the arrivals during slot k contribute to the system content. If $n_{1,l}=0, n_{2,l}>0$, a class-2 packet starts service at the beginning of start-slot l and it leaves the system immediately before start-slot $l+1$. For each class, admitted arrivals during this class-2 service contribute to the system content at the beginning of start-slot $l+1$. On the other hand, if $n_{1,l}>0$, a class-1 packet starts service at the beginning of start-slot l and it leaves the system immediately before start-slot $l+1$. For each class, admitted arrivals during this class-1 service contribute to the system content at the beginning of start-slot $l+1$. Note that the class-1 system content at the beginning of start-slots cannot exceed N, the class-1 queue capacity.

3.3.2 Arrivals during a service

Evidently, we try to stick to the analysis method that was successful in the previous chapter. In order to proceed from (3.7), we need expressions characterizing the $e_{i,j,k}$. Without loss of generality, one can drop the index k as the $e_{i,j,k}$ are i.i.d. (for different k) as the $a_{i,k}$ are i.i.d. and independent of the s_j. Consequently, the corresponding partial pgfs of the number of class-2 arrivals during a class-j service, during which i ($0 \le i \le N$) and i or more class-1 packets arrive, respectively denoted by $E_{i,j}(z)$ and $E^*_{i,j}(z)$, are given by

$$
E_{i,j}(z) = \mathrm{E}\left[z^{e_{2,j,k}}\, 1\left\{e_{1,j,k}=i\right\}\right]\,, E^*_{i,j}(z) = \sum_{m=i}^{\infty} E_{m,j}(z)\,. \tag{3.8}
$$

Mimicking the matrix-based approach of the previous chapter, leads to

$$
\boldsymbol{E}_j(z) = \begin{bmatrix} E_{0,j}(z) & E_{1,j}(z) & \cdots & E_{N-1,j}(z) & E^*_{N,j}(z) \\ 0 & E_{0,j}(z) & \cdots & E_{N-2,j}(z) & E^*_{N-1,j}(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & E_{0,j}(z) & \vdots \\ 0 & \cdots & \cdots & 0 & E^*_{0,j}(z) \end{bmatrix}. \tag{3.9}
$$

Unfortunately, obtaining these partial pgfs is, in general, a tedious task. We have

$$
E_{i,j}(z) = \frac{1}{i!}\frac{d^i}{dx^i} S_j\big(A(x,z)\big)\Big|_{x=0}\,, E^*_{i,j}(z) = S_j\big(A(1,z)\big) - \sum_{k=0}^{i-1} E_{k,j}(z)\,. \tag{3.10}
$$

However, from a computational point of view, this is infeasible for general $S_j(z)$ and $A(z_1, z_2)$.

Note 39. *A very common approach to tackling these kind of problems would be to invert this two-dimensional transform using the Fourier-series method, which is fast but approximate, rendering it useless for our purposes of finding an exact closed-form solution.*

Fortunately, the following alternative method is interesting, especially from a numerical point of view as a lot of the computational effort is reused in the further analysis of the system. Recall, the matrix of arrivals in a slot, given by

$$\boldsymbol{A}(z) = \begin{bmatrix} A_0(z) & A_1(z) & \cdots & A_{N-1}(z) & A_N^*(z) \\ 0 & A_0(z) & \cdots & A_{N-2}(z) & A_{N-1}^*(z) \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & A_0(z) & \vdots \\ 0 & \cdots & \cdots & 0 & A_0^*(z) \end{bmatrix}. \tag{3.11}$$

Intuitively, the matrix $\boldsymbol{E}_j(z)$, governing the arrivals during a service, should be the multiplication of s_j, the number of slots in a service time, times the matrix $\boldsymbol{A}(z)$, which governs the arrivals in a slot. It turns out that the fact that s_j is a random variable does not pose any issues and the preceding argument is completely correct yielding

$$\boldsymbol{E}_j(z) = S_j\big(\boldsymbol{A}(z)\big), \; j = 1, 2. \tag{3.12}$$

Evaluating a function in a matrix is perfectly feasible by grace of the spectral decomposition theorem (see appendix B), if $S_j(z)$ satisfies the requirements detailed in (B.2).

This theorem provides us with a very powerful tool from a computational point of view. Instead of having to evaluate the matrix power series $\sum_{n=0}^{\infty} \Pr\left[s_j = n\right] \boldsymbol{A}(z)^n$, we only need to evaluate the function $S_j(z)$ and its derivatives for scalar arguments and compute a finite number of matrix multiplications. The downside is that the eigenvalues of $\boldsymbol{A}(z)$ have to be calculated, as well as the matrices $\mathbf{G}_j$. In general, this can prove to be quite difficult but in our case the downsides are virtually non-existent as the eigenvalues and spectral projectors are surprisingly easy to obtain.

Computing the eigenvalues is straightforward because of the special eigenstructure of $\boldsymbol{A}(z)$. As this matrix has a triangular form, the eigenvalues simply are its diagonal elements. There are two distinct eigenvalues: $\xi_1 = A_0^*(z)$, with index 1, and $\xi_2 = A_0(z)$, with index N.

Note 40. *Note that we obtain an (exact) analytic expression for the eigenvalues whereas finding eigenvalues usually requires some numerical calculations*.

The corresponding spectral projectors are easily shown to be given by

$$\boldsymbol{G}_1 = \begin{bmatrix} \mathbf{0}^T & \cdots & \mathbf{0}^T & \boldsymbol{e} \end{bmatrix}, \; \boldsymbol{G}_2 = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{e} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{3.13}$$

Remarkably, they are independent of z.

Note 41. *Recall that $\boldsymbol{I}$ denotes the identity matrix of appropriate size, $\boldsymbol{x}^T$ is the transpose of vector $\boldsymbol{x}$ and $\boldsymbol{e}$ and $\boldsymbol{0}$ indicate the column vectors of appropriate size with all elements equal to 1 and 0 respectively.*

Finally, applying the spectral decomposition theorem (B.1) yields

$$\boldsymbol{E}_j(z) = S_j\big(\boldsymbol{A}(z)\big) = S_j\big(A_0^*(z)\big)\boldsymbol{G}_1 + \sum_{k=0}^{N-1} \frac{S_j^{(k)}\big(A_0(z)\big)}{k!}\big(\boldsymbol{A}(z) - A_0(z)\boldsymbol{I}\big)^k \boldsymbol{G}_2 . \tag{3.14}$$

Note 42. *Recall that the notation $f^{(k)}(z)$ denotes the k-th derivative of $f(z)$.*

Notice that $\boldsymbol{E}_1(z)$ and $\boldsymbol{E}_2(z)$ share all factors except (the derivatives of) the functions $S_1(z)$ and $S_2(z)$. Especially note that the (computationally expensive) powers of $\big(\boldsymbol{A}(z) - A_0(z)\boldsymbol{I}\big)$ are shared.

3.3.3 System content at the beginning of start-slots

Let us denote the partial pgf of the class-2 system content at the beginning of start-slot l that has class-1 system content equal to i by

$$N_{i,l}(z) = \mathrm{E}\left[z^{n_{2,l}}\, 1\{n_{1,l} = i\}\right] . \tag{3.15}$$

The corresponding row vector of size $N+1$ of the system content at the lth start-slot is given by

$$\boldsymbol{n}_l(z) = \left[N_{i,l}(z)\right]_{i=0..N} . \tag{3.16}$$

Again, define the $(N+1)\times(N+1)$ matrices

$$\boldsymbol{H}_0 = \begin{bmatrix} 1 & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{O} \end{bmatrix}, \boldsymbol{H}_{>0} = \boldsymbol{I} - \boldsymbol{H}_0, \boldsymbol{D}_H = \begin{bmatrix} \boldsymbol{0} & 0 \\ \boldsymbol{I} & \boldsymbol{0}^T \end{bmatrix} . \tag{3.17}$$

By conditioning on the state of the server at start-slot l, a relation between $\boldsymbol{n}_l(z)$ and $\boldsymbol{n}_{l+1}(z)$ is derived from the system equations (3.7). We have

$$\boldsymbol{n}_{l+1}(z) = \boldsymbol{n}_l(0)\boldsymbol{H}_0\boldsymbol{A}(z) + \big(\boldsymbol{n}_l(z) - \boldsymbol{n}_l(0)\big)\boldsymbol{H}_0\frac{1}{z}\boldsymbol{E}_2(z) + \boldsymbol{n}_l(z)\boldsymbol{H}_{>0}\boldsymbol{D}_H\boldsymbol{E}_1(z) . \tag{3.18}$$

This can be explained as follows. The first term corresponds with an empty server. Therefore, $n_{2,l} = 0, n_{1,l} = 0$ and start slot $l+1$ is the next slot thus we take into account the arrivals in a single slot (start-slot l). The second term represents the evolution of the system when a class-2 service starts at start-slot l. This yields that $n_{2,l} > 0, n_{1,l} = 0$, that by start-slot $l+1$ the class-2 packet in service will have left the system and that we need to consider arrivals during a class-2 service. The final term corresponds with a class-1 packet starting service at start-slot l. Then, $n_{1,l} > 0$ and the class-1 packet in service will have left the system by start-slot $l+1$ and packets arriving during this class-1 service need to be accounted for.

Assume that the system has reached steady state and define following steady-state values

$$\boldsymbol{n}(z) = \lim_{l\to\infty} \boldsymbol{n}_l(z) = \lim_{l\to\infty} \boldsymbol{n}_{l+1}(z) = \left[N_i(z)\right]_{i=0..N} . \tag{3.19}$$

Taking the limit of (3.18) for $l \to \infty$ induces

$$\boldsymbol{n}(z)\big(z\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(z) - z\boldsymbol{D}_H\boldsymbol{E}_1(z)\big) = \boldsymbol{n}(0)\boldsymbol{H}_0\big(z\boldsymbol{A}(z) - \boldsymbol{E}_2(z)\big)\,. \tag{3.20}$$

Note 43. *From now on, we opt for brevity over clarity in order to reduce formula sizes. Consequently, we have invoked the identity $\boldsymbol{H}_{>0}\boldsymbol{D}_H = \boldsymbol{D}_H$ in the formula above.*

As $\boldsymbol{n}(0)\boldsymbol{H}_0 = \begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}$, (3.20) becomes

$$\boldsymbol{n}(z)\big(z\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(z) - z\boldsymbol{D}_H\boldsymbol{E}_1(z)\big) = \begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}\big(z\boldsymbol{A}(z) - \boldsymbol{E}_2(z)\big)\,. \tag{3.21}$$

The constant $N_0(0)$ is the only unknown. It is found in two steps.

First, evaluation of (3.21) in $z = 1$ produces

$$\boldsymbol{n}(1)\big(\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(1) - \boldsymbol{D}_H\boldsymbol{E}_1(1)\big) = \begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}\big(\boldsymbol{A}(1) - \boldsymbol{E}_2(1)\big)\,. \tag{3.22}$$

As the matrices $\boldsymbol{E}_j(1),\ j = 1,2$ are right-stochastic by construction, each row of matrix $[\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(1) - \boldsymbol{D}_H\boldsymbol{E}_1(1)]$ sums to 0 and it hence has rank N and is not invertible. We thus require an additional relation in order to obtain the vector $\boldsymbol{n}(1)$. The normalization condition provides $\boldsymbol{n}(1)\boldsymbol{e} = 1$. Combining this with (3.22) yields

$$\boldsymbol{n}(1) = \Big[\begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}\big(\boldsymbol{A}(1) - \boldsymbol{E}_2(1)\big)\Big|1\Big]\Big[\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(1) - \boldsymbol{D}_H\boldsymbol{E}_1(1)\Big|\boldsymbol{e}\Big]^{-1}\,. \tag{3.23}$$

Second, differentiation of (3.21) with respect to z yields

$$\begin{aligned}&\boldsymbol{n}(z)\big(\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2'(z) - \boldsymbol{D}_H\boldsymbol{E}_1(z) - z\boldsymbol{D}_H\boldsymbol{E}_1'(z)\big)\\ &+ \boldsymbol{n}'(z)\big(z\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(z) - z\boldsymbol{D}_H\boldsymbol{E}_1(z)\big) = \begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}\big(\boldsymbol{A}(z) + z\boldsymbol{A}'(z) - \boldsymbol{E}_2'(z)\big)\,.\end{aligned} \tag{3.24}$$

Observe that $\boldsymbol{E}_j(1)$ ($j = 1,2$) and $\boldsymbol{A}(1)$ are right-stochastic matrices by construction. Therefore,

$$\big(\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(1) - \boldsymbol{D}_H\boldsymbol{E}_1(1)\big)\boldsymbol{e} = \boldsymbol{0}\,, \begin{bmatrix} 1 & \boldsymbol{0}\end{bmatrix}\boldsymbol{A}(1)\boldsymbol{e} = 1\,. \tag{3.25}$$

Keeping these identities in mind, evaluation of (3.24) in $z = 1$ and multiplication of both sides of the resulting equation by $\boldsymbol{e}$ yields

$$N_0(0) = \frac{\boldsymbol{n}(1)\big(\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2'(1) - \boldsymbol{D}_H\boldsymbol{E}_1(1) - \boldsymbol{D}_H\boldsymbol{E}_1'(1)\big)\boldsymbol{e}}{1 + \begin{bmatrix} 1 & \boldsymbol{0}\end{bmatrix}\big(\boldsymbol{A}'(1) - \boldsymbol{E}_2'(1)\big)\boldsymbol{e}}\,. \tag{3.26}$$

Substitution of (3.23) for $\boldsymbol{n}(1)$ and solving the resulting expression for $N_0(0)$ finally provides this probability.

Now that $N_0(0)$ has been obtained, (3.23) provides $\boldsymbol{n}(1)$, the probability mass function (pmf) of the class-1 system content at the beginning of a start-slot in steady state and as (3.21) leads to

$$\boldsymbol{n}(z) = \begin{bmatrix} N_0(0) & \boldsymbol{0}\end{bmatrix}\big(z\boldsymbol{A}(z) - \boldsymbol{E}_2(z)\big)\big(z\boldsymbol{I} - \boldsymbol{H}_0\boldsymbol{E}_2(z) - z\boldsymbol{D}_H\boldsymbol{E}_1(z)\big)^{-1}\,, \tag{3.27}$$

the pgf of the class-2 system content at the beginning of a start-slot in steady state is found as $\boldsymbol{n}(z)\boldsymbol{e}$.

Note 44. *As in the previous chapter, the stationary distribution could have been derived immediately using the Pollaczeck-Khinchine formula.*

3.3.4 Moving from start-slots to random slots

Consider a randomly chosen slot, say k. We will construct the system content in slot k starting from the preceding start-slot l as the system content at start-slots is known from the previous subsection. Recall that, if slot k happens to be a start-slot, the preceding start-slot l is slot k itself. The key observation to make is that the state of the server (the type of the packet in service, if any) in slot k equals the state of the server in the preceding start-slot l, as in an empty system these slots coincide and in a non-empty system a class-j ($j = 1,2$) packet enters the server in start-slot l and remains there until the following start-slot. Therefore, if we condition on the state of the server in slot k, the state of the server in start-slot l is known and one only has to focus on the evolution of the queue content from start-slot l to slot k. To that end, one needs to account for the class-i queue content at (the beginning of) start-slot l, denoted by $m_{i,l}$, and for the packets arriving between start-slot l (inclusive) and slot k (exclusive). If a class-j ($j = 1,2$) packet is in service during a random slot, the time period between the beginning of that slot and the beginning of the preceding start-slot is called the elapsed service time s_j^- of that packet. Furthermore, let $e_{i,j,k}^-$ represent the number of class-i arrivals during the elapsed class-j service time up to slot k. Parallel to (3.6), we then have

$$e_{i,j,k}^- = \sum_{m=1}^{s_j^-} a_{i,k-m} \,. \tag{3.28}$$

Formally, the arguments above can be stated as

$$\begin{aligned} u_{1,k} &= \begin{cases} 0 & \text{if no service}\,, \\ m_{1,l} + e_{1,1,k}^- + 1 & \text{if class-1 service}\,, \\ m_{1,l} + e_{1,2,k}^- & \text{if class-2 service}\,, \end{cases} \\ u_{2,k} &= \begin{cases} 0 & \text{if no service}\,, \\ m_{2,l} + e_{2,1,k}^- & \text{if class-1 service}\,, \\ m_{2,l} + e_{2,2,k}^- + 1 & \text{if class-2 service}\,. \end{cases} \end{aligned} \tag{3.29}$$

Evidently, if there is no service then no packets are present at the beginning of slot k (start-slot l). Otherwise, we simply add the queue contents at start slot l, the packets arriving during the elapsed service time and the packet in service (to the appropriate class). In the following subsections we characterize these random variables.

3.3.5 Queue contents at start-slots conditioned on the server

Obtaining the queue content in function of the system content was performed in the previous chapter. Consequently, completely analogous to (2.38), we have

$$\begin{aligned} m_{1,l} &= (n_{1,l} - 1)^+\,, \\ m_{2,l} &= \begin{cases} (n_{2,l} - 1)^+ & \text{if } n_{1,l} = 0\,, \\ n_{2,l} & \text{if } n_{1,l} > 0\,. \end{cases} \end{aligned} \tag{3.30}$$

Recall that, here, as we study the start-slot l preceding random slot k, the state of the server is identical in these slots and thus $u_{i,k}$ and $m_{i,l}$ are correlated. Consequently, we are interested in the queue content conditioned on the state of the server. Furthermore, let the $1 \times N+1$ vector $\boldsymbol{m}_l\left(z \mid j\right)$, $(j = 0, 1, 2)$ denote the queue content at the beginning of a start-slot l when the server is empty $(j = 0)$, serving a class-1 packet $(j = 1)$ or a class-2 packet $(j = 2)$ respectively. Then, by conditioning on the state of the server, (3.30) leads to

$$\begin{aligned}
\boldsymbol{m}_l(z\,|0) &= \left[\mathrm{E}\left[\, z^{m_{2,l}}\, 1\left\{m_{1,l} = i\right\} \mid \text{no service}\right]\right]_{i=0..N} \\
&= \left[\mathrm{E}\left[\;1\left\{0 = i\right\} \mid n_{1,l} = 0,\, n_{2,l} = 0\right]\right]_{i=0..N} \\
&= \begin{bmatrix}1 & \mathbf{0}\end{bmatrix}, \\
\boldsymbol{m}_l(z\,|1) &= \left[\mathrm{E}\left[\, z^{m_{2,l}}\, 1\left\{m_{1,l} = i\right\} \mid \text{class-1 service}\right]\right]_{i=0..N} \\
&= \left[\mathrm{E}\left[\, z^{n_{2,l}}\, 1\left\{n_{1,l} - 1 = i\right\} \mid n_{1,l} > 0\right]\right]_{i=0..N} \\
&= \frac{1}{1 - N_{0,l}(1)} \boldsymbol{n}_l(z) \boldsymbol{H}_{>0} \boldsymbol{D}_H, \\
\boldsymbol{m}_l(z\,|2) &= \left[\mathrm{E}\left[\, z^{m_{2,l}}\, 1\left\{m_{1,l} = i\right\} \mid \text{class-2 service}\right]\right]_{i=0..N} \\
&= \left[\mathrm{E}\left[\, z^{n_{2,l}-1}\, 1\left\{0 = i\right\} \mid n_{1,l} = 0,\, n_{2,l} > 0\right]\right]_{i=0..N} \\
&= \frac{1}{N_{0,l}(1) - N_{0,l}(0)} \left(\boldsymbol{n}_l(z) - \boldsymbol{n}_l(0)\right) \boldsymbol{H}_0 \frac{1}{z}.
\end{aligned} \tag{3.31}$$

Notice that knowing the state of the server implies knowledge about the possible values for $n_{1,l}$ and $n_{2,l}$. Furthermore, as before, $\boldsymbol{D}_H$ is used to “subtract” a class-1 packet, in this case the packet moving from the queue to the server. For class-2, this happens through the factor z^{-1}. The corresponding steady-state expressions $\boldsymbol{m}\left(z \mid j\right)$, $(j = 0, 1, 2)$, the steady-state queue content given the state of the server, are trivially obtained by dropping the index l, yielding

$$\begin{aligned}
\boldsymbol{m}(z\,|0) &= \begin{bmatrix}1 & \mathbf{0}\end{bmatrix}, \\
\boldsymbol{m}(z\,|1) &= \frac{1}{1 - N_0(1)} \boldsymbol{n}(z) \boldsymbol{D}_H, \\
\boldsymbol{m}(z\,|2) &= \frac{1}{z\big(N_0(1) - N_0(0)\big)} \big(\boldsymbol{n}(z) - \boldsymbol{n}(0)\big) \boldsymbol{H}_0.
\end{aligned} \tag{3.32}$$

3.3.6 Arrivals during an elapsed service time

If a class-$j (j = 1, 2)$ packet is in service in slot k, the elapsed service time s_j^- runs from the preceding start-slot l up to slot k. The goal of this subsection is obtaining an expression (jointly) quantifying $e^-_{1,j,k}$ and $e^-_{2,j,k}$. First, recall that slot k was chosen randomly. However, the total service time of the packet in service, $\tilde{s}_j$ is not statistically indistinguishable from s_j due to the renewal theory inspection paradox. Similar to arrivals before a randomly tagged packet, a random slot during a service is more likely to belong to a long service time. Consequently, analogous to (2.47), we have

$$\Pr\left[\tilde{s}_j = n\right] = \frac{n \Pr\left[s_j = n\right]}{\mu_1}, \tag{3.33}$$

and thus

$$\Pr\left[s_j^- = n\right] = \sum_{i=n+1}^{\infty} \frac{\Pr\left[s_j = i\right]}{\mu_j}, \tag{3.34}$$

Hence, the pgf of the elapsed service time of the class-j packet in service in a random slot is given by

$$S_j^-(z) = \frac{S_j(z) - 1}{\mu_j (z-1)}. \tag{3.35}$$

Note 45. *The link with equation (2.73) is evidently due to the conceptual similarity of "arrivals before a tagged packet" and "slots before a tagged slot".*

Finding an expression for the arrivals during the elapsed service time follows the template of subsection 3.3.2 where the arrivals during a service were computed using the spectral decomposition theorem, but here $S_j^-(z)$ evidently is used instead of $S_j(z)$. Hence, the number of arrivals during an elapsed class-j service time are characterized by the matrix

$$\begin{aligned} \boldsymbol{E}_j^-(z) &= S_j^-\left(\boldsymbol{A}(z)\right) \\ &= S_j^-\left(A_0^*(z)\right)\boldsymbol{G}_1 + \sum_{k=0}^{N-1} \frac{S_j^{-(k)}\left(A_0(z)\right)}{k!}\left(\boldsymbol{A}(z) - A_0(z)\boldsymbol{I}\right)^k \boldsymbol{G}_2 . \end{aligned} \tag{3.36}$$

Note 46. *Almost all factors were previously obtained in (3.14), which is very interesting from a computational point of view.*

3.3.7 System content at the beginning of random slots

Let the vector $\boldsymbol{u}_k(z)$, of size $N+2$ denote the system content (of both classes) at the beginning of slot k, given by

$$\boldsymbol{u}_k(z) = \left[\mathrm{E}\left[z^{u_{2,k}}\, 1\{u_{1,k} = i\}\right]\right]_{i=0..N+1} . \tag{3.37}$$

Note that $0 \le u_{1,k} \le N+1$ as the class-1 queue can hold up to N packets and the server can hold a single packet.

Conditioning on the state of the server in slot k and invoking (3.29) easily yields

$$\begin{aligned} \boldsymbol{u}_k(z) = &\left[\mathrm{E}\left[z^{u_{2,k}}\, 1\{u_{1,k} = i, \text{ no service in slot } k\}\right]\right]_{i=0..N+1} \\ &+ \left[\mathrm{E}\left[z^{u_{2,k}}\, 1\{u_{1,k} = i, \text{ class-1 service in slot } k\}\right]\right]_{i=0..N+1} \\ &+ \left[\mathrm{E}\left[z^{u_{2,k}}\, 1\{u_{1,k} = i, \text{ class-2 service in slot } k\}\right]\right]_{i=0..N+1} \\ = &\Pr[\text{no service in slot } k]\, \boldsymbol{m}_l(z\,|\,0) \\ &+ \Pr[\text{class-1 service in slot } k]\, \boldsymbol{m}_l(z\,|\,1)\, \boldsymbol{E}_1^-(z)\left[\boldsymbol{0} \quad \boldsymbol{I}\right] \\ &+ \Pr[\text{class-2 service in slot } k]\, \boldsymbol{m}_l(z\,|\,2)\, \boldsymbol{E}_2^-(z) z \left[\boldsymbol{I} \quad \boldsymbol{0}\right] . \end{aligned} \tag{3.38}$$

As mentioned at the beginning of subsection 3.3.4, the crucial step lies in acknowledging that knowing the state of the server in slot k implies the same server state in the preceding start-slot l. Also, note that tracking the packet in service is accounted for by $\begin{bmatrix}\mathbf{0}^T & \boldsymbol{I}\end{bmatrix}$ for a class-1 packet and $z\begin{bmatrix}\boldsymbol{I} & \mathbf{0}^T\end{bmatrix}$ for a class-2 packet. This also transforms the vector dimension to incorporate the server capacity (of one).

Now, in order to take the limit of the equation above to move to steady state, one must first determine the steady-state probability of the server being in one of the three possible states (idle, serving class-1/2). To that end , recall that if the server is idle during a slot, that slot is a start-slot. Therefore, let us first determine the probability that a slot is a start-slot. On average, the time period between the beginning of start-slots l and $l+1$ consists of a single slot if the system is empty ($\Pr\left[n_{1,l} = n_{2,l} = 0\right]$), of μ_2 slots if a class-2 packet is served ($\Pr\left[n_{1,l} = 0, n_{2,l} > 0\right]$) or of μ_1 slots if a class-1 packet is served ($\Pr\left[n_{1,l} > 0\right]$). Therefore, γ, the steady-state probability that a random slot is a start-slot, is defined as

$$\begin{aligned}\gamma &= \lim_{k\to\infty} \Pr[\text{slot k is a start-slot}] \\ &= \frac{1}{N_0(0) + \big(N_0(1) - N_0(0)\big)\mu_2 + \big(1 - N_0(1)\big)\mu_1}\,.\end{aligned} \tag{3.39}$$

Then, the probability that the system is empty at the beginning of a random slot in is given by

$$\begin{aligned}U_0(0) &= \lim_{k\to\infty} \Pr\left[u_{1,k} = u_{2,k} = 0\right] \\ &= \lim_{k,l\to\infty} \Pr\left[n_{1,l} = n_{2,l} = 0, \text{slot k is a start-slot}\right] \\ &= \gamma N_0(0)\,.\end{aligned} \tag{3.40}$$

In steady state, the system is in stochastic equilibrium. Therefore, on average, the amount of packets effectively accepted by the system equals the amount of packets served by the system. This yields that the effective total load is found as $\rho_T^e = 1 - U_0(0)$. The effective class-1 load and mean number of effective class-1 arrivals are therefore expressed as $\rho_1^e = \rho_T^e - \rho_2$ and $\lambda_1^e = \rho_1^e/\mu_1$ respectively. Consequently, the steady-state probabilities of the state of the server are given by

$$\begin{aligned}\Pr[\text{no service}] &= U_0(0) = 1 - \rho_T^e\,, \\ \Pr[\text{class-1 service}] &= \big(1 - U_0(0)\big)\frac{\rho_1^e}{\rho_T^e} = \rho_1^e\,, \\ \Pr[\text{class-2 service}] &= \big(1 - U_0(0)\big)\frac{\rho_2}{\rho_T^e} = \rho_2\,.\end{aligned} \tag{3.41}$$

Hence, taking the limit of (3.38) produces

$$\boldsymbol{u}(z) = (1-\rho_T^e)\boldsymbol{m}(z\,|\,0) + \rho_1^e\boldsymbol{m}(z\,|\,1)\,\boldsymbol{E}_1^-(z)\begin{bmatrix}\mathbf{0} & \boldsymbol{I}\end{bmatrix} + \rho_2\boldsymbol{m}(z\,|\,2)\,\boldsymbol{E}_2^-(z)z\begin{bmatrix}\boldsymbol{I} & \mathbf{0}\end{bmatrix}, \tag{3.42}$$

and substitution of (3.32) finally results in

$$\begin{aligned}\boldsymbol{u}(z) = (1 - \rho_T^e)\begin{bmatrix}1 & \mathbf{0}\end{bmatrix} &+ \frac{\rho_1^e}{1 - N_0(1)}\boldsymbol{n}(z)\boldsymbol{D}_H\boldsymbol{E}_1^-(z)\begin{bmatrix}\mathbf{0}^T & \boldsymbol{I}\end{bmatrix} \\ &+ \frac{\rho_2}{N_0(1) - N_0(0)}\big(\boldsymbol{n}(z) - \boldsymbol{n}(0)\big)\boldsymbol{H}_0\boldsymbol{E}_2^-(z)\begin{bmatrix}\boldsymbol{I} & \mathbf{0}^T\end{bmatrix}.\end{aligned} \tag{3.43}$$

Note 47. *Note the conciseness of this expression achieved by breaking up the problem in different parts allowed by the embedded Markov chain approach. However, the relation is not expressed "immediately" in terms of the arrival and service process but through intermediate concepts (e.g. system contents at start-slots, arrivals during the elapsed part of service). We feel this is justifiable because these concepts have a clear (probabilistic) meaning and because, the "immediate" expression would be devoid of any practical use as it is clear that the gargantuan expression one would end up with when substituting the expressions (3.36),(3.39) and subsequently replacing all occurrences of* $\boldsymbol{n}(z)$, $\boldsymbol{n}(0)$, $N_0(0)$, $N_0(1)$, $\ldots$ *by the corresponding expressions found in subsection 3.3.3 into the formula above would provide little insight.*

3.4 Class-1 delay

The main assumption used for calculating the class-1 delay in the previous chapter no longer holds. General service times and non-preemptive service imply that a class-1 packet arriving in a system void of other class-1 packets cannot start service in the next slot if the system is currently performing a class-2 service, occupying the server until completion of that task (thus until the next start-slot). For example, recall the sample path depicted in figure 3.1. In slot 10, the first class-1 packet that arrives only enters the server in slot 14. Consequently, one can no longer study class-1 in isolation in order to obtain the class-1 delay and the distributional form of Little's Law does not hold forcing us to a direct approach. However, the intermediate concepts derived in the previous subsection will be of great use. Furthermore, unlike in the previous chapter, calculating the class-1 delay in the probability domain is cumbersome as the service times can be unboundedly long and thus an upper bound for the class-1 delay does no longer exist. Consequently, we revert to the use of pgfs.

Again, randomly tag a class-1 packet and let its delay be denoted by d_1, the arrival slot of the packet by k and the preceding start slot by l. Recall that, if a packet is in service during slot k, that packet's service time, denoted by $\tilde{s}_j$, is different from s_j due to the renewal theory paradox. Furthermore, similar to the elapsed service time s_j^- , let us define the remaining service time s_j^+, the number of slots between slot k (exclusive) and the next start slot $(l+1)$. Consequently, upon arrival of the tagged packet in slot k , the ongoing service consists of an elapsed service time, slot k itself and a remaining service time, or equivalently $\tilde{s}_j = s_j^- + 1 + s_j^+$.

Again, the state of the server plays a crucial role. If the system is empty at the beginning of slot k, only the service times of the class-1 packets arriving in slot k before the tagged packet and the service time of the tagged packet itself contribute to the delay. If the tagged packet arrives during a class-1 service, these periods are also part of the delay, but they are preceeded by the remaining service time of the ongoing class-1 service, the service times of the class-1 packets waiting in the queue at the start of the ongoing service and the service times of the class-1 packets that have arrived during the elapsed part of the ongoing service. If the tagged packet arrives during a class-2 service, the situation is similar but the ongoing service is of class 2 and thus it is impossible that there were class-1 packets in the system at the beginning of the service. Formally, this is expressed by

$$d_1 = \begin{cases} \sum_{i=1}^{\hat{a}_{1,k}+1} s_{1,k_i} & \text{if no service in slot } k, \\ s_1^+ + \sum_{i=1}^{m_{1,l}+e^-_{1,1,k}+\hat{a}_{1,k}+1} s_{1,k_i} & \text{if class-1 service in slot } k, \\ s_2^+ + \sum_{i=1}^{e^-_{1,2,k}+\hat{a}_{1,k}+1} s_{1,k_i} & \text{if class-2 service in slot } k. \end{cases} \tag{3.44}$$

Here, s_{1,k_i} represents the service time of the i-th class-1 packet starting service after slot k. Furthermore, recall that $\hat{a}_{1,k}$ denotes the class-1 packets arriving in the same slot as but before the tagged packet and that $m_{1,l}$ consists of the class-1 packets in the queue at start-slot l. This formula is completely analogous, barring some difference in notation, to [32].

Recall from the previous chapter, that one must take into account that it only makes sense to consider the delay of an accepted packet. The (long-run) probability that a class-1 packet is accepted is given by

$$\Pr\left[\text{packet accepted}\right] = \lambda_1^e / \lambda_1 . \tag{3.45}$$

The goal of this section is computing the pgf of the class-1 delay $D_1(z)$, given by

$$D_1(z) = \lim_{k,l\to\infty} \mathrm{E}\left[z^{d_1} \,\middle|\, \text{packet accepted}\right] . \tag{3.46}$$

Proceeding from (3.44) seems rather complicated at first but remark that, unlike $\tilde{s}_j$, the service times s_{1,k_i} are all statistically indistinguishable from s_1. This allows following key insight: the delay of the tagged packet equals the time between slot k (exclusive) and start-slot $l+1$ plus a service time s_1 for each class-1 packet whose complete service time contributes to the delay (thus not the potential packet already in service during slot k). Thus, let us construct a row vector which, in the transform domain, tracks the part of the delay up to start-slot $l+1$ and, through the position in the vector, tracks the number of class-1 packets that will contribute to the delay of the tagged packet through their service times. Then, simply multiplying this row vector with a column vector with i-th element equal to $S_1(z)^i$ yields $D_1(z)$.

Almost all of the "tools" required for this construction have already been developed in the previous sections and the previous chapter. The only new tool we need is the description of the three "parts" of an ongoing class-j service time upon arrival of the tagged packet. As slot k can be each slot of the ongoing service with equal probability [4, vol 3, p. 31], the corresponding joint pgf is given by

$$\begin{aligned}
\tilde{S}_j(z_1,z_2,z_3) &= \lim_{k\to\infty} \mathrm{E}\left[z_1^{s_j^-} z_2 \, z_3^{s_j^+} \right] \\
&= \sum_{n=0}^{\infty} \Pr\left[\tilde{s}_j = n+1\right] \frac{1}{n+1} \sum_{i=0}^{n} z_1^i z_2 z_3^{n-i} \\
&= \sum_{n=0}^{\infty} \frac{1}{\mu_j} \Pr\left[s_j = n+1\right] z_3^n \frac{\left(\frac{z_1}{z_3}\right)^{n+1} - 1}{\left(\frac{z_1}{z_3}\right) - 1} z_2 \\
&= \frac{S_j(z_1) - S_j(z_3)}{\mu_j(z_1 - z_3)} z_2 \,.
\end{aligned} \tag{3.47}$$

Recall, from (2.50), that the class-1 arrivals before a tagged class-1 packet are given by $\hat{\boldsymbol{A}}_1$. Now, conditioning on the state of the server, we have

$$\begin{aligned}
D_1(z) = \frac{\lambda_1}{\lambda_1^e} \Big[& \Pr[\text{no service}] \begin{bmatrix} 1 & \boldsymbol{0} \end{bmatrix} \hat{\boldsymbol{A}}_1 \\
& + \Pr[\text{class-1 service}]\, \boldsymbol{m}(1|1) \tilde{S}_1\left(\boldsymbol{A}(1), \hat{\boldsymbol{A}}_1, z\right) \\
& + \Pr[\text{class-2 service}] \begin{bmatrix} 1 & \boldsymbol{0} \end{bmatrix} \tilde{S}_2\left(\boldsymbol{A}(1), \hat{\boldsymbol{A}}_1, z\right) \Big] \boldsymbol{s}_1(z)\,,
\end{aligned} \tag{3.48}$$

with $\boldsymbol{s}_1(z)$ the column vector

$$\boldsymbol{s}_1(z) = \begin{bmatrix} S_1(z) & \cdots & S_1(z)^N & 0 \end{bmatrix}^T , \tag{3.49}$$

where the last element is zero as it is impossible for a packet to enter the system when the class-1 queue is full (recall that we can only tag an accepted packet). However, evaluating $\tilde{S}_j(z_1,z_2,z_3)$ in multiple matrix arguments is not straightforward as there is no multivariate version of the spectral decomposition theorem. Fortunately, one can easily sidestep the problem for the arguments needed here. The specific form of the expression developed in (3.47) and the fact that matrices and scalars commute, yields

$$\tilde{S}_j\left(\boldsymbol{A}(1), \hat{\boldsymbol{A}}_1, z\right) = \tilde{S}_j\left(\boldsymbol{A}(1), 1, z\right) \hat{\boldsymbol{A}}_1 \,. \tag{3.50}$$

Then, one can see this as a function in one (matrix) argument by assuming the second and third arguments to be constants, enabling the use of spectral decomposition. This leads to

$$\begin{aligned}
\tilde{S}_j\left(\boldsymbol{A}(1), 1, z\right) = \tilde{S}_j\left(A_0^*(1), 1, z\right) \boldsymbol{G}_1 & \\
+ \sum_{k=0}^{N-1} \frac{1}{k!} \left. \frac{\partial^k \tilde{S}_j(x,1,z)}{\partial x^k} \right|_{x=A_0(1)} & \left(\boldsymbol{A}(1) - A_0(1)\boldsymbol{I}\right)^k \boldsymbol{G}_2 \,.
\end{aligned} \tag{3.51}$$

Note 48. *Recall that the i-th order partial derivative in x of a function f is indicated by $\frac{\partial^i f}{\partial x^i}$.*

Finally, by invoking (3.32) and (3.41), (3.48) becomes

$$\begin{aligned} D_1(z) = \frac{\lambda_1}{\lambda_1^e}\Big(&(1-\rho_T^e)\,[1 \quad \mathbf{0}] + \frac{\rho_1^e}{1-N_0(1)}\boldsymbol{n}(1)\boldsymbol{D}_H\tilde{S}_1\big(\boldsymbol{A}(1),1,z\big) \\ &+\rho_2\,[1 \quad \mathbf{0}]\,\tilde{S}_2\big(\boldsymbol{A}(1),1,z\big)\Big)\hat{\boldsymbol{A}}_1\boldsymbol{s}_1(z)\,. \end{aligned} \tag{3.52}$$

3.5 Class-2 delay

The delay of a class-2 packet is (potentially) longer as the packet not only has to wait for the service of the packets arriving before it to complete, it also has to give priority to all class-1 packets accepted into the system before the class-2 packet starts service. As before, we randomly tag a class-2 packet. Assume the arrival slot of the packet to be slot k and the preceding start-slot to be start-slot l. The delay of the tagged packet, denoted by d_2, is given by

$$d_2 = \begin{cases} r_{1,l+1} + \sum\limits_{i=1}^{\hat{a}_{2,k}} t_{2,k_i} + \bar{s}_2 & \text{if no service in slot } k\,, \\ s_1^+ + r_{1,l+1} + \sum\limits_{i=1}^{m_{2,l}+e^-_{2,1,k}+\hat{a}_{2,k}} t_{2,k_i} + \bar{s}_2 & \text{if class-1 service in slot } k\,, \\ s_2^+ + r_{1,l+1} + \sum\limits_{i=1}^{m_{2,l}+e^-_{2,2,k}+\hat{a}_{2,k}} t_{2,k_i} + \bar{s}_2 & \text{if class-2 service in slot } k\,. \end{cases} \tag{3.53}$$

Let us first elaborate on this formula and the involved random variables. Once again, the state of the server in slot k is crucial. If the system is empty at the beginning of slot k, the tagged packet's delay starts with the remaining class-1 busy period in start-slot $l+1$, denoted by $r_{1,l+1}$, which is caused by (potential) class-1 arrivals in slot k. Next, for each class-2 packet arriving in slot k before the tagged packet ($\hat{a}_{2,k}$), an extended service completion time has to be accounted for, as class-1 packets arriving during their service have priority. Let t_{2,k_i} represent the extended service completion time of the i-th class-2 packet starting service after slot k. Finally, the service time of the tagged packet itself, denoted by $\bar{s}_2$, concludes the delay. If the tagged packet arrives during a class-1 service, the same periods contribute to the delay but one of course also has to include the remaining service time of the ongoing class-1 service and take into account the class-2 packets already waiting in the queue at the start of the ongoing service and the ones arriving during the elapsed part of the ongoing service. Furthermore, the remaining class-1 busy period in start-slot $l+1$ is induced by the class-1 packets in the queue in start-slot l plus the ones arriving during the entire ongoing service. If the tagged packet arrives during a class-2 service, the situation is similar but the ongoing service is of class 2 and thus it is impossible that there were class-1 packets in the system at the beginning of the service so the remaining class-1 busy period in start-slot $l+1$ is induced only by the class-1 packets arriving during the ongoing service.

Next, as in the previous chapter, expressions for the different parts contributing to the class-2 delay are first developed in separate subsections. Obtaining these expressions only requires some small variations of the methods used in the previous chapter.

3.5.1 Remaining class-1 busy period

As busy periods start and end at start slots, it seems natural to study the evolution of the remaining busy period between consecutive start-slots. Let us define the conditional pgf of the remaining class-1 busy period at the beginning of a start-slot l, with class-1 system content equal to n, by

$$R_{1,l}(z|n) = \mathrm{E}\left[z^{r_{1,l}} \mid n_{1,l} = n \right], \quad n = 0 \ldots N, \tag{3.54}$$

and the corresponding column vector by

$$\boldsymbol{r}_{1,l}(z) = \left[R_{1,l}(z|0) \cdots R_{1,l}(z|N) \right]^T . \tag{3.55}$$

Then, one can follow the exact same reasoning used in the previous chapter, but now accounting for the service time between two consecutive start-slots during the remaining busy period. The busy period ends if $n_{1,l} = 0$. Otherwise, we track the number of slots until the next start-slot, obviously a class-1 service, while at the same time accounting for the changes in system content, consisting of the arrivals during the service and the departure of the served packet. This leads to

$$\boldsymbol{r}_{1,l}(z) = \left[1 \quad \boldsymbol{0} \right]^T + \boldsymbol{D}_H S_1\big(\boldsymbol{A}(1)z\big)\boldsymbol{r}_{1,l+1}(z) . \tag{3.56}$$

Consequently, in steady state, the vector of conditional pgfs of the remaining class-1 busy period in a start-slot, given the system content, is expressed as

$$\boldsymbol{r}_1(z) = \lim_{l\to\infty} \boldsymbol{r}_{1,l}(z) = \Big(\boldsymbol{I} - \boldsymbol{D}_H S_1\big(\boldsymbol{A}(1)z\big) \Big)^{-1} \left[1 \quad \boldsymbol{0} \right]^T . \tag{3.57}$$

Note 49. *Note that $S_1\big(\boldsymbol{A}(1)z\big) \neq \boldsymbol{E}_1(z)$. As mentioned above, the similarity with equation (2.65) in the previous chapter is natural.*

3.5.2 Extended service completion time

The general service times make calculating the extended service completion time of a class-2 packet quite intricate. Recall that the extended service completion time of a class-2 packet starts when the packet starts service and lasts until the next slot wherein a class-2 packet can be served. Let $t_{2,k}$ denote the extended service completion time of a class-2 packet starting service in slot k (start-slot l). We have

$$t_{2,k} = \begin{cases} s_2 & \text{if } e_{1,2,k} = 0 , \\ s_2 + r_{1,l+1} & \text{if } e_{1,2,k} > 0 . \end{cases} \tag{3.58}$$

However, as one can expect that during longer class-2 service times more class-1 packets arrive, $e_{1,2,k}$ and s_2 are correlated and furthermore, this also causes a longer subsequent class-1 busy period. The extended service completion time starts in start slot l with the start of a class-2 service, thus in a system void of class-1 packets. The class-1 packets arriving during this service form the class-1 system content in start-slot $l+1$, hence triggering a subsequent remaining class-1 busy period. Consequently, we have

$$T_{2,k}(z) = \mathrm{E}\left[z^{t_{2,k}}\right] = \begin{bmatrix}1 & \mathbf{0}\end{bmatrix} S_2\big(\boldsymbol{A}(1)z\big)\big(\boldsymbol{H}_0\boldsymbol{e} + \boldsymbol{H}_{>0}\boldsymbol{r}_{1,l+1}(z)\big)\,. \tag{3.59}$$

Evidently, the corresponding pgf in steady state is given by

$$T_2(z) = \begin{bmatrix}1 & \mathbf{0}\end{bmatrix} S_2\big(\boldsymbol{A}(1)z\big)\boldsymbol{r}_1(z)\,. \tag{3.60}$$

3.5.3 Class-2 delay

Now, we are ready to compute the pgf of the class-2 delay, given by

$$D_2(z) = \lim_{k,l\to\infty} \mathrm{E}\left[z^{d_2}\right]. \tag{3.61}$$

In order to proceed from (3.53), one can follow a strategy similar to the one used for computing the class-1 delay in the previous section. Again we construct a row vector that tracks the number of class-1 packets in the system at start-slot $l+1$ through the position in the vector. Multiplying with $\boldsymbol{r}_1(z)$ then accounts for the remaining class-1 busy period. The other parts of the delay, associated with the class-2 packets to be served before the tagged packet, are tracked in the transform domain as the elements of the row vector. Also, note that the extended service completion times t_{2,k_i} are each statistically indistinguishable from t_2 and the service time of the tagged packets $\tilde{s}_2$ is also not distinguishable from s_2. Furthermore, recall that $\hat{\boldsymbol{A}}_2(z)$ is given by (2.74). These arguments lead to

$$\begin{aligned} D_2(z) = \Bigg[& (1-\rho_T^e)\begin{bmatrix}1 & \mathbf{0}\end{bmatrix}\hat{\boldsymbol{A}}_2\big(T_2(z)\big) \\ & + \rho_1^e \boldsymbol{m}\big(T_2(z)|1\big)\tilde{S}_1\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right) \\ & + \rho_2 \boldsymbol{m}\big(T_2(z)|2\big)\tilde{S}_2\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right)\Bigg]\boldsymbol{r}_1(z)S_2(z)\,. \end{aligned} \tag{3.62}$$

Here, evaluating $\tilde{S}_j\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right)$, the class-$j$ ongoing service time ($j=1,2$), is not straightforward. Recall that there is no multivariate version of the spectral decomposition theorem and the “trick” used in the previous section is not applicable here as all three arguments contain matrices. However, if we specify the function $\tilde{S}(z_1,z_2,z_3)$ by its power series expansion, as in (3.47), we can apply the spectral decomposition theorem on the arguments separately. Power series expansion produces

$$\begin{aligned} & \tilde{S}_j\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right) \\ & = \mathrm{E}\left[\boldsymbol{A}\big(T_2(z)\big)^{s_j^-}\hat{\boldsymbol{A}}_2\big(T_2(z)\big)\left(\boldsymbol{A}(1)z\right)^{s_j^+}\right] \\ & = \sum_{n=0}^{\infty}\frac{1}{n+1}\Pr\left[\tilde{s}_j = n+1\right]\sum_{i=0}^{n}\boldsymbol{A}\big(T_2(z)\big)^i\hat{\boldsymbol{A}}_2\big(T_2(z)\big)\big(\boldsymbol{A}(1)z\big)^{n-i}\,. \end{aligned} \tag{3.63}$$

Now, one can perform spectral decomposition (B.1) on the function $x \to x^i$ and on the function $x \to x^{n-i}$ separately. Both decompositions share the same spectral projectors $\mathbf{G}_1$ and $\mathbf{G}_2$. The eigenvalues and their index are respectively denoted by

$$\begin{aligned}&\xi_1(z) = A_0^*\big(T_2(z)\big) \text{ with } k_1 = 1\,,\ \xi_2(z) = A_0\big(T_2(z)\big) \text{ with } k_2 = N\,,\\ &\xi_1'(z) = A_0^*(1)z \text{ with } k_1' = 1\,,\ \xi_2'(z) = A_0(1)z \text{ with } k_2' = N\,.\end{aligned} \tag{3.64}$$

Note that the eigenvalues are functions in z. Then, the straightforward but tedious task of substituting the two spectral decompositions into (3.63), expanding all terms of the resulting equation and finally reconstructing the generating function from the power series in each term yields

$$\begin{aligned}&\tilde{S}_j\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right)\\ &= \sum_{n=1}^{2}\sum_{m=0}^{k_n-1}\sum_{n'=1}^{2}\sum_{m'=0}^{k'_{n'}-1}\frac{1}{m!}\frac{1}{m'!}\frac{\partial^{m+m'}}{\partial x^m y^{m'}}\tilde{S}(x,1,y)\bigg|_{\substack{x=\xi_n(z)\\ y=\xi_{n'}(z)}}\\ &\qquad\qquad \Big(\boldsymbol{A}\big(T_2(z)\big) - \xi_n(z)\mathbf{I}\Big)^m \mathbf{G}_n\hat{\boldsymbol{A}}_2\big(T_2(z)\big)\Big(\boldsymbol{A}(1)z - \xi'_{n'}(z)\mathbf{I}\Big)^{m'}\mathbf{G}_{n'}\,.\end{aligned} \tag{3.65}$$

Finally, substituting (3.32) in (3.62) yields

$$\begin{aligned}D_2(z) = \Bigg[&(1-\rho_T^e)\begin{bmatrix}1 & \mathbf{0}\end{bmatrix}\hat{\boldsymbol{A}}_2\big(T_2(z)\big)\\ &+ \frac{\rho_1^e}{1-N_0(1)}\boldsymbol{n}\big(T_2(z)\big)\boldsymbol{D}_H\tilde{S}_1\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right)\\ &+ \frac{\rho_2}{T_2(z)\big(N_0(1)-N_0(0)\big)}\Big(\boldsymbol{n}\big(T_2(z)\big) - \boldsymbol{n}(0)\Big)\boldsymbol{H}_0\\ &\qquad\times\tilde{S}_2\left(\boldsymbol{A}\big(T_2(z)\big), \hat{\boldsymbol{A}}_2\big(T_2(z)\big), \boldsymbol{A}(1)z\right)\Bigg]\boldsymbol{r}_1(z)S_2(z)\,.\end{aligned} \tag{3.66}$$

3.6 Numerical Examples

Let us validate the analysis performed in this chapter through some numerical examples. In this section, the main goals are comparing the results of the N/∞ queue to those of the ∞/∞ priority queue and assessing the impact of general service times. Again, let us focus on the cases were the models yield different results as the default behavior of priority queues is well-documented [32].

Again, let us consider the N/∞ queueing system at one of the outputs of an output-queueing switch, detailed in appendix A.4. Let us study a 4 × 4 output-queueing switch. Let the parameters of the arrival process be batch size $b = 5$ and probability that a batch arrives and is of class 1/2 by $v_1 = v_2 = 0.04$. Consequently, the arrival loads for each class are the same ($\lambda_1 = \lambda_2 = 0.2$). However, let the pgfs of the service times be given by

$$S_1(z) = 1/4z^2 + 1/2z^3 + 1/4z^4\ , S_2(z) = 1/2z + 1/2z^2\,. \tag{3.67}$$

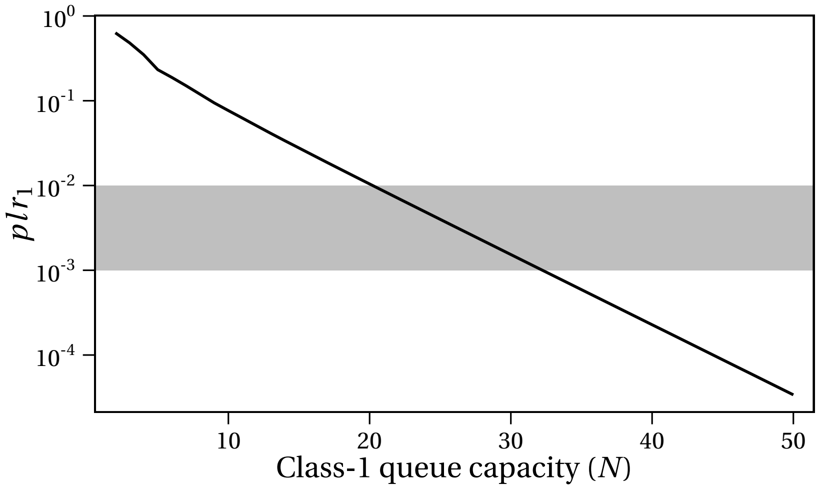

Figure 3.2: Class-1 Packet loss ratio vs. class-1 queue capacity.

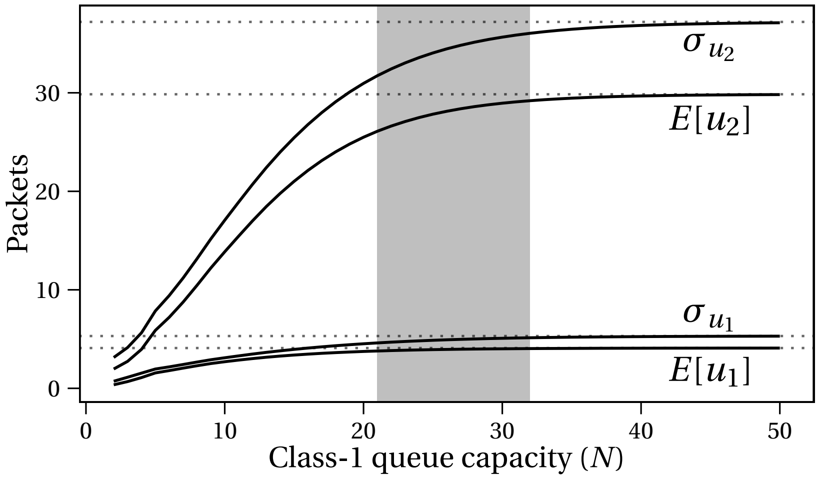

Figure 3.3: Moments of class-1/2 system content vs. class-1 queue capacity.

Consequently, class-1 service times are, on average, longer than those for class-2 packets as $\mu_1 = 3$ and $\mu_2 = 1.5$. Note that, consequently, the arrival load is given by $\rho_T = 0.9$.

Figure 3.2 depicts the packet loss ratio versus the class-1 queue capacity N. Obviously the packet loss decreases with increasing N. Again, the region between 10^{-2} and 10^{-3}, which we have marked in grey, is particularly interesting as most real-time applications tolerate this amount of packet loss. Systems with a very small packet loss ratio ($<< 10^{-3}$) are accurately modeled by the infinite system. The mean and the standard deviation of the system content at the beginning of random slots

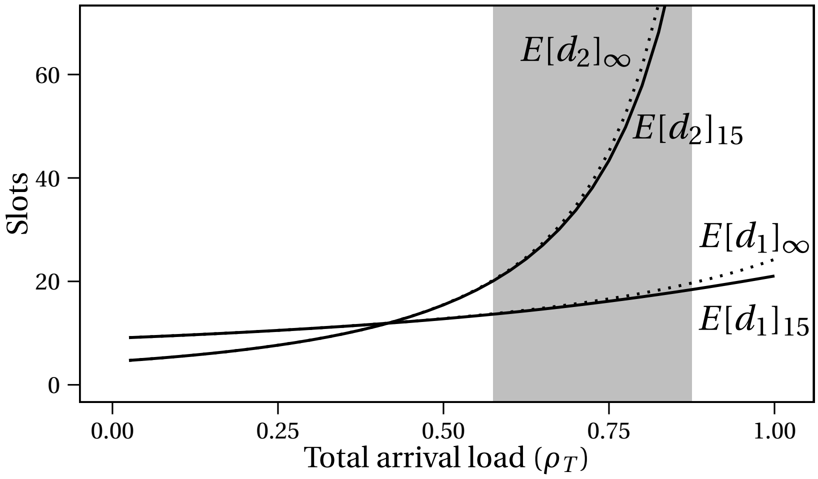

Figure 3.4: Mean class-1/2 delay vs. total load for class-1 queue capacity 20 and ∞.

are plotted for both classes versus the class-1 queue capacity N in figure 3.3. The effect of the priority scheduling is apparent as the class-2 system content exceeds the class-1 system content, despite that, on average, the system receives the same amount of packets of each class and that class-1 packets generally have longer service times. The values increase for increasing N and clearly converge to the values corresponding with the infinite system, represented by the horizontal dotted lines. This validates that, for N going to infinity, the N/∞ system considered in this chapter tends to the infinite system, as the number of dropped class-1 packets tends to zero. However, in our region of interest (again marked in grey) the infinite system considerably overestimates the mean value and standard deviation of the system content of both classes. For instance, at $N = 21$, the smallest value for N where $plr_1 < 10^{-2}$, the mean and standard deviation of the class-2 system content are overestimated by 14% and 17% respectively.

Next, for figure 3.4 we fix $N = 20$ and vary $v_1 = v_2$ to consequently vary the total arrival load ρ_T between 0% and 100% while plotting the mean system content of both classes. Instead of showing the packet loss on a separate figure, we have again marked the region of interest, where $10^{-3} < plr_1 < 10^{-2}$, in grey. Again the effect of the priority scheduling is apparent. Especially note the class-2 starvation as the curves for the N/∞ system and the infinite system seem to be close together, due to the steep slope, but for $\rho_T \cong 0.875$, where plr_1 approaches the 1% boundary, the error introduced by using the infinite model over our model amounts to 8 and 13% for class-1 and class-2 respectively.

To conclude this subsection, let us focus on the impact of the mean class-2 service time. For the arrival and service processes, consider the parameters

$$N = 25,\ v_1 = 0.06,\ S_1(z) = z^2\ ,\ v_2 = 0.06/i,\ S_2(z) = z^i\ . \tag{3.68}$$

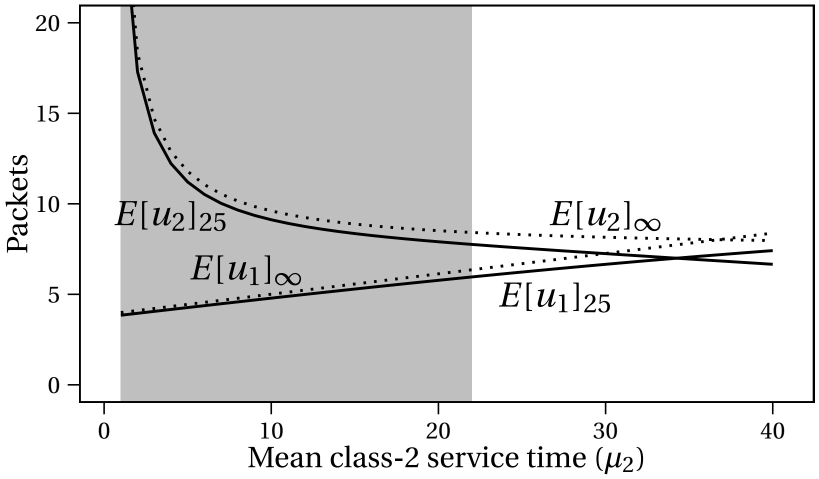

Figure 3.5: Mean system content versus mean class-2 service time for class-1 queue capacity 25 and ∞.

Note that $\mu_2 = i$. By increasing i, the average class-2 service time increases, while, at the same time, we let the average number of class-2 arrivals decrease, thereby keeping the arrival load ρ_T constant at 0.9. Figure 3.5 depicts the average system content at the beginning of random slots of both classes for class-2 service times from 1 to 40. Evidently, the decrease of $E[u_2]$ is caused by the longer service time as λ_2 decreases and less packets arrive. Furthermore, this figure exemplifies the effect of class-2 traffic on class-1 traffic, as increasing values of μ_2 yield an increased probability that a class-1 packet arrives during the service time of a class-2 packet in a system void of class-1 packets and has to wait until the end of this service before it can enter the server. Again, it is clear that the infinite capacity approximation is inaccurate under certain conditions.

3.7 Concluding remarks

A two-class non-preemptive priority queue with finite capacity for high-priority packets has been studied. Analytical formulas for system content and delay of both traffic classes were determined making extensive use of the spectral decomposition theorem to cope with the difficulties that arise when considering general service times for both classes. Several numerical examples indicate the impact of small but practically feasible amounts of real-time packet loss on system performance, which is considerably different from what was predicted by existing models.

4

N/∞ Priority Queue - Tail Behavior

4.1 Introduction

In this chapter, we investigate the tail behavior of the class-2 system content in the priority-queueing model with single-slot service times studied in chapter 2 and compare it to the ∞/∞ model. The latter assumes that both queues can hold an infinite amount of packets. This assumption is retracted for the high-priority class in the N/∞ model, where only N high-priority packets can be stored simultaneously. Evidently, this has an impact on (the tail of) the class-2 system content.

It has been observed that modeling the high-priority queue capacity as finite or infinite leads to different low-priority tail behavior, which is closely related to packet loss, but it remains unclear how this shift in behavior arises. Studying the asymptotic behaviour of these type of queues started with [47]. Abate and Whitt [48] were the first to prove exactly that tails in an infinitely-sized priority queue are not necessarily exponential, even if the distributions of inter-arrival and service times are exponentially decaying. They heavily rely on singularity analysis of the Laplace transform of the low-priority waiting time in the complex plane and characterize three types of tails of the pmf of the delay of low-priority customers in a two-class $M/G/1$ priority queue, namely (i) $\sim \alpha t^{-3/2} e^{-\eta t}$, (ii) $\sim \alpha e^{-\eta t}$ and (iii) $\sim \alpha t^{-1/2} e^{-\eta t}$, with α and η constants depending on the arrival and service-time distributions. Depending on the parameters of the arrival and service processes, one of these three types of tail behavior appears.

Type (i) is encountered when the 'priority effect' dominates (large low-priority waiting time due to blocking by high-priority customers). In this case, the tail of the low-priority waiting time is related to the tail of the busy period of the high-priority queue, which has a Laplace transform expressed by an implicitly defined function.

This implicit function, a solution of the so-called kernel equation, is the origin of the non-exponentiality of the tail, as it has a branch cut in the complex plane, rather than simple poles (the latter lead to exponentially decaying tails). This type is observed when the load of the low-priority class is low relative to the high-priority load. On the other hand, exponential tails (ii) are encountered when the 'queueing effect' dominates (large low-priority waiting time due to many low-priority customers blocking each other) and is observed when the load of the low-priority class is relatively high. Type (iii) is the boundary between the other cases (both of the aforementioned effects are equally strong). Furthermore, using large deviations principles, these results were verified [49].

In contrast, when the capacity for the high-priority customers is limited, tail probabilities of the low-priority system content are *always* exponentially decaying (i.e., for all possible values of the involved parameters). Here, all singularities of the transform are (simple) poles, leading to purely exponential tails. This is also apparent using matrix-analytic techniques [35, 36], where the terms "levels" and "phases" are used for the two dimensions of the queueing system. The high-priority capacity corresponds to the number of phases but treating an infinite amount of phases remains an open problem. Furthermore, it has been shown [50, 51, 52] that the standard practices to transform the state-space, such as truncation, can lead to erroneous results concerning tail behavior. Recent research in matrix-analytic techniques has therefore focused on trying to cope with an infinite number of phases. Primary attention has been paid to obtaining the boundary condition for exponentiality, i.e. finding conditions under which the tails are exponential, for several subclasses of random walks [53, 54]. Consequently, the methods from literature can either handle infinite or finite capacity, but the evolution/limit from finite to infinite capacity is still not fully discovered (although for the QBD sub-case some recent results give some hope [55, 56, 57]).

Note 50. *Recall that the N/∞ and the ∞/∞ priority queue are respectively called the finite and infinite case/system.*

This chapter is structured as follows. First, we summarize the results for the ∞/∞ model. Next, we investigate the location of all singularities for some practical examples and we numerically compute the pmf by calculating all poles and the respective residues. Then, under the restriction of maximum two high-priority arrivals in a slot, a crucial relation between the characteristic polynomial of this recurrence relation in the finite case and the kernel in the infinite case is established. Part of this reasoning is then repeated under the relaxed restriction of maximum S high-priority arrivals in a slot.

Note 51. *This chapter is rather strange. The content ranges from numerical exploration to a more rigorous mathematical style with theorems and lemmas with corresponding proofs and a lot of the results only hold under certain assumptions. This might be untraditional but I think it is an original way to simultaneously indicate the boundaries of the performed research and highlight in what directions the most interesting future work is to be found.*

4.2 Summary: ∞/∞ priority queue

In this chapter, the comparisons between the N/∞ and the ∞/∞ model are so frequent that we briefly summarize the relevant results from the latter here. Evidently, the reader should consult [32] for a more detailed view.

The joint pgf of the steady-state system content is given by

$$U(z_1,z_2) = \mathrm{E}\left[z_1^{u_1} z_2^{u_2}\right] = (1-\lambda_T)\frac{A(z_1,z_2)(z_2-1)(Y(z_2)-z_1)}{(z_2-Y(z_2))(A(z_1,z_2)-z_1)}\,, \tag{4.1}$$

where the pgf $Y(z)$ is implicitly defined as it is the unique (as will be proven later) root in x of the kernel $F(x,z)$ when $|x|<1$, $|z|<1$, with

$$F(x,z) = A(x,z) - x\,. \tag{4.2}$$

The total system content is given by

$$U_T(z) = U(z,z) = (1-\lambda_T)\frac{A_T(z)(z-1)}{z-A_T(z)}\,, \tag{4.3}$$

and the class-2 system content by

$$U_2(z) = U(1,z) = (1-\lambda_T)\frac{A_2(z)(z-1)(Y(z)-1)}{(z-Y(z))(A_2(z)-1)}\,. \tag{4.4}$$

We obtain an approximation for the tail probabilities by studying the dominant singularity of these expressions.

Note 52. *In the entire chapter (unless mentioned otherwise), we assume that the pgfs of the arrival processes are meromorphic (pgfs and their derivatives go to infinity for z equal to their radii of convergence or for $z\to\infty$), which is correct for "standard" arrival distributions. Even for most pgfs that do not fulfill this assumptions, the reasoning can be adjusted and the results still hold but this process is quite messy, as these arrival processes introduce extra singularities. Here, we want to focus on the singularities emerging from the priority scheduling mechanism, which makes singularities transferred from the arrival process undesirable.*

4.2.1 Tail of total system content

We prove that the dominant singularity of $U_T(z)$ is a pole with multiplicity 1 and is a zero of $z-A_T(z)$. From Pringsheim's theorem [58, p. 242], we know that the dominant singularity lies on the positive real axis. We first look at the zeros of $f(z)=z-A_T(z)$. Its smallest zero on the positive real axis is $z=1$. Since $f'(1)=1-\lambda_T>0$, this is a zero with multiplicity 1. This is however not a pole of $U_T(z)$ since pgfs remain finite at $z=1$. Starting from $z=1$, we look for the next zero of $f(z)$ by increasing z. It is seen that $f(z)>0$ at first (since $f(1)=0$ and $f'(1)>0$). However since $A_T'(z)$ is a strictly increasing function, $f'(z)=1-A_T'(z)$ is a strictly decreasing function. Therefore $f(z)$ reaches a certain maximum and then decreases. For a certain z_T,

$f(z)$ equals zero (again) and $f'(z_T) < 0$. Therefore z_T is a zero with multiplicity 1 of $z - A_T(z)$. Since z_T is inside the region of convergence of $A_T(z)$, z_T is smaller than the (possible) dominant singularity of $A_T(z)$ and is thus the dominant singularity of $U_T(z)$.

4.2.2 Investigating $Y(z)$

The tail behavior of the system content of class-2 packets is a bit more involved, since it is not a priori clear what the dominant singularity of $U_2(z)$ is due to the occurrence of the implicitly defined function $Y(z)$ in (4.4). First we investigate $Y(z)$ on the (positive) real axis. The first derivative of $Y(z)$ is given by

$$Y'(z) = \frac{A^{(2)}(Y(z),z)}{1 - A^{(1)}(Y(z),z)}. \tag{4.5}$$

Consequently, $Y(z)$ has a singularity, denoted by z_B, where the denominator of $Y'(z)$ becomes 0, i.e., $A^{(1)}(Y(z_B),z_B) = 1$. One can prove that $Y(z_B)$ is finite and that $Y(z)$ is a pgf and thus can be written as a power series with non-negative coefficients:

$$Y(z) = \sum_{n=0}^{\infty} y(n) z^n, \tag{4.6}$$

thus with $y(n)$ a pmf. If we find an explicit function approximating $Y(z)$ in the neighborhood of z_B, we can use Flajolet's singularity analysis, detailed in appendix C to obtain the tail probabilities $y(n)$. In [59] it is shown (in the more general context of a set of functional equations) that, in the neighborhood of z_B, $Y(z)$ is approximately given by

$$Y(z) \sim Y(z_B) - K_Y (z_B - z)^{1/2}, \tag{4.7}$$

with

$$K_Y = \sqrt{\frac{2A^{(2)}(Y(z_B),z_B)}{A^{(11)}(Y(z_B),z_B)}}. \tag{4.8}$$

Using theorem (C.1) on expression (4.7), we get

$$y(n) = -\frac{K_Y \sqrt{z_B}}{\Gamma(-1/2)} n^{-3/2} z_B^{-n}. \tag{4.9}$$

4.2.3 Tail of class-2 system content

A first singularity of $U_2(z)$ is z_B as taking the first derivative yields

$$U_2'(z) = \frac{(1-\lambda_T)\left\{\begin{array}{l} (z-1)(1-Y(z))(z-Y(z))A_2'(z) \\ +A_2(z)(1-Y(z))^2(1-A_2(z)) \\ -A_2(z)(z-1)^2(1-A_2(z))Y'(z) \end{array}\right\}}{(z-Y(z))^2(1-A_2(z))^2}, \tag{4.10}$$

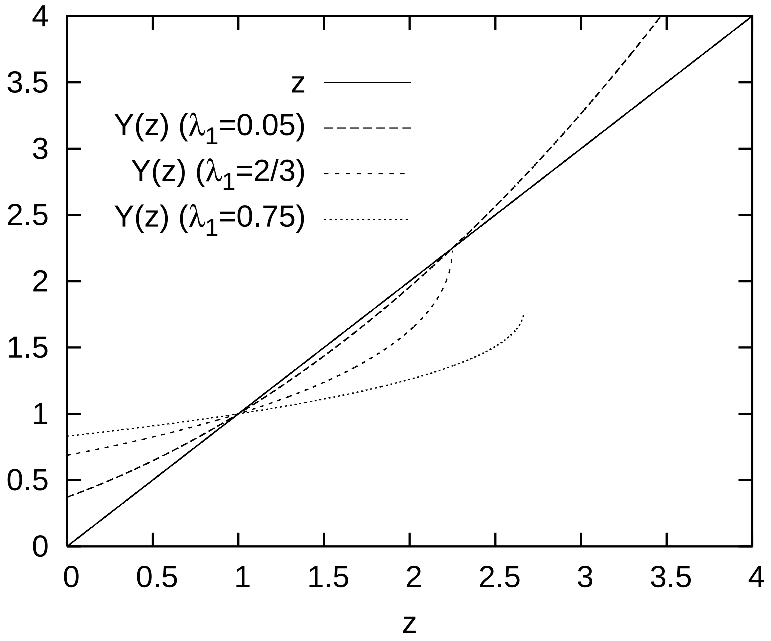

Figure 4.1: Types of behavior of $Y(z)$

and this clearly goes to infinity as $Y'(z) \to \infty$, or, as $z \to z_B$.

A second possible singularity z_L of $U_2(z)$ on the real axis is given by the positive zero of the denominator $z - Y(z)$, and it is easily proved to be equal to z_T, if z_L exists. Figure 4.1 gives three typical types of behavior of $Y(z)$. In the case indicated with a long-dashed line, $Y(z)$ intersects z twice (for $z = 1$ and for $z = z_L$), before reaching the branch point. When the line is short-dashed, $Y(z)$ intersects z once in $z = 1$ and equals z in its branch point. In case of the dotted line, $Y(z)$ intersects z once in $z = 1$ and reaches its branch point before intersecting z a second time. In this case, z_L does not exist, or alternatively, $z - Y(z) \neq 0$ for real $z > 1$ (and for z for which $Y(z)$ exists), as z_T does not lie on Y(z) but on the other branch ($Y(z) = A(Y(z), z)$ has multiple solutions).

The tail behavior of the system content of class-2 packets is thus characterized by z_L or z_B, depending on which is the dominant (i.e., smallest) singularity, which is dependent of the used arrival process and its parameters, on which we will elaborate at the end of this subsection. Furthermore, z_L equals z_T when it is dominant (or equivalently, when it exists). Three situations may thus occur, namely when $z_L = z_T < z_B$, z_L does not exist, and $z_L = z_T = z_B$. We study the (approximate) behavior of $U_2(z)$ in the neighborhood of its dominant singularity (thus for all three cases). In the first case, the single pole z_T is dominant and thus

$$U_2(z) \sim \frac{K_2^{(1)}}{z_T - z}, \tag{4.11}$$

for $z \to z_T$. $K_2^{(1)}$ can be calculated by substituting expression (4.4) in the previous expression and setting $z = z_T$ yielding

$$K_2^{(1)} = \frac{(1-\lambda_T) A_2(z_T)\,(z_T-1)^2}{(A_2(z_T)-1)\,(Y'(z_T)-1)}. \tag{4.12}$$

In the second case, i.e., when z_L does not exist, the branch point z_B is dominant. We study the behavior of $U_2(z)$ in the neighborhood of z_B. Using expression (4.7) in (4.4), we find

$$U_2(z) \sim (1-\lambda_T)\frac{A_2(z)(z-1)\left(Y(z_B)-K_Y\,(z_B-z)^{1/2}-1\right)}{\left(z-Y(z_B)+K_Y\,(z_B-z)^{1/2}\right)(A_2(z)-1)} \tag{4.13}$$

$$\sim (1-\lambda_T)\frac{\left\{\begin{array}{l} A_2(z)(z-1)\left(Y(z_B)-K_Y\,(z_B-z)^{1/2}-1\right) \\ \times\left(z-Y(z_B)-K_Y\,(z_B-z)^{1/2}\right)\end{array}\right\}}{\left((z-Y(z_B))^2-K_Y^2\,(z_B-z)\right)(A_2(z)-1)}. \tag{4.14}$$

This expression leads to

$$U_2(z) \sim U_2(z_B) - K_2^{(3)}\,(z_B-z)^{1/2}, \tag{4.15}$$

in the neighborhood of z_B — note that we have used the notation $K_2^{(3)}$ instead of (the expected) $K_2^{(2)}$ because we will switch the last two types of tail behavior in the end formulas — with

$$K_2^{(3)} = \frac{(1-\lambda_T)K_Y A_2(z_B)\,(z_B-1)^2}{(A_2(z_B)-1)\,(z_B-Y(z_B))^2}. \tag{4.16}$$

In the third case, z_T and z_B coincide. Again, we will study the behavior of $U_2(z)$ in the neighborhood of this dominant singularity. The approximation of $U_2(z)$ in the neighborhood of z_B is again found by substituting expression (4.7) in expression (4.4):

$$U_2(z) \sim (1-\lambda_T)\frac{A_2(z)(z-1)\left(z_B-K_Y\,(z_B-z)^{1/2}-1\right)}{\left(z-z_B+K_Y\,(z_B-z)^{1/2}\right)(A_2(z)-1)} \tag{4.17}$$

$$\sim (1-\lambda_T)\frac{A_2(z)(z-1)\left(z_B-K_Y\,(z_B-z)^{1/2}-1\right)}{(z_B-z)^{1/2}\left((z_B-z)^{1/2}+K_Y\right)(A_2(z)-1)}, \tag{4.18}$$

where we have used that $Y(z_B) = z_B$. This leads to the following form of $U_2(z)$ in the neighborhood of its dominant singularity:

$$U_2(z) = \frac{K_2^{(2)}}{(z_B-z)^{1/2}}, \tag{4.19}$$

with

$$K_2^{(2)} = \frac{(1-\lambda_T) A_2(z_B)\,(z_B-1)^2}{K_Y\,(A_2(z_B)-1)}. \tag{4.20}$$

Summarizing, $U_2(z)$ can be approximated in the neighborhood of its dominant singularity by

$$U_2(z) \sim \begin{cases} \dfrac{K_2^{(1)}}{z_T - z} & \text{if } z_L = z_T < z_B \\ \dfrac{K_2^{(2)}}{(z_B - z)^{1/2}} & \text{if } z_L = z_T = z_B \\ U_2(z_B) - K_2^{(3)}\,(z_B - z)^{1/2} & \text{if } z_L \text{ does not exist,} \end{cases} \tag{4.21}$$

where the constants $K_2^{(i)}$ $(i = 1, 2, 3)$ are given by expressions (4.12), (4.20) and (4.16) respectively (note that we switched the second and third case). Using theorem (C.1), we find the tail probabilities for the three possible cases:

$$u_2(n) = \text{Prob}[u_2 = n] \sim \begin{cases} \dfrac{(1-\lambda_T)A_2(z_T)\,(z_T - 1)^2\,z_T^{-n-1}}{(A_2(z_T) - 1)\,(Y'(z_T) - 1)} \\ \dfrac{(1-\lambda_T)A_2(z_B)\,(z_B - 1)^2\,n^{-1/2} z_B^{-n}}{K_Y\sqrt{z_B\pi}\,(A_2(z_B) - 1)} \\ \dfrac{(1-\lambda_T)K_Y A_2(z_B)\,(z_B - 1)^2\,n^{-3/2} z_B^{-n}}{2\sqrt{\pi/z_B}\,(A_2(z_B) - 1)\,(z_B - Y(z_B))^2}, \end{cases} \tag{4.22}$$

for large enough n, if $z_L = z_T < z_B$, if $z_L = z_T = z_B$ and if z_L does not exist, respectively. The first expression constitutes a typical *geometric* (or *exponential*) tail behavior, while the third expression is a typical *non-geometric* tail behavior. The second expression exhibits a behavior in between the two other cases, and we will thus call this tail behavior of *transition* type. As mentioned before, the type of tail behavior is driven by the arrival process (and thus also the class-1/2 load). For the switch arrival process with $S = 16$ and $b = 1$, figure 4.2 marks the regions where different tail behavior is observed, in terms of the relative amount of high priority load versus low-priority load.

4.3 Numerically determining tail behavior in the N/∞ priority queue

4.3.1 Numerically computing the poles

First, in order to get some (visual) understanding of how the singularities (and hence the tail behavior) for the N/∞ model and the ∞/∞ model are related, let us calculate them numerically and plot them for a number of different scenarios.

Recall from (2.29) that

$$\boldsymbol{u}(z) = \Big((z-1)U_0(0)\,[1 \quad 0 \quad \cdots \quad 0]\,\boldsymbol{X}(z)\Big)\big(z\boldsymbol{I} - \boldsymbol{X}(z)\big)^{-1}. \tag{4.23}$$

Let us slightly change the notation by adding the subscript N (and substituting $U_0(0)$ by $p_{0,N}$) to explicitly indicate the size of the class-1 queue capacity. The pgf

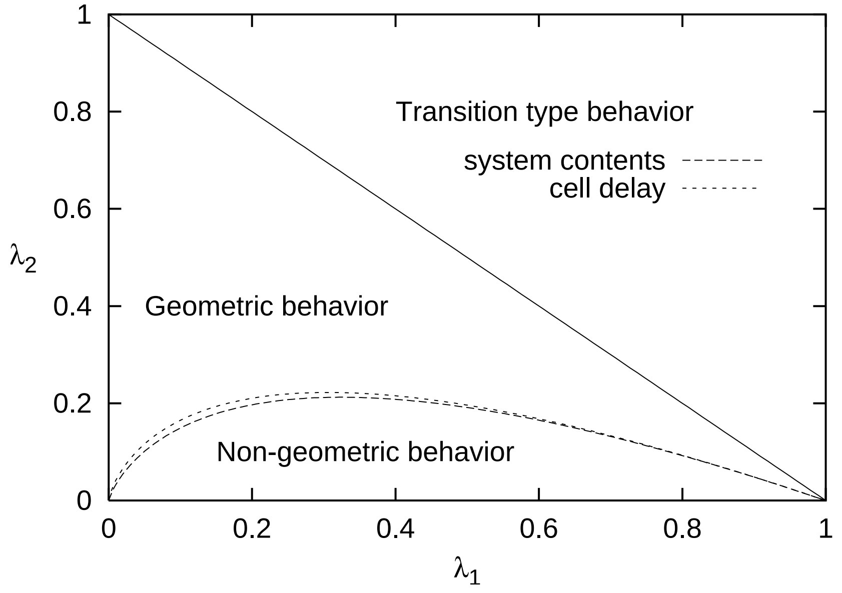

Figure 4.2: Tail behavior in the different regions of the parameter space (λ_1, λ_2).

of the class-2 system content in N/∞ model is denoted by $U_N(z)$

$$U_N(z) = \boldsymbol{u}(z)\boldsymbol{e} = \Big((z-1)p_{0,N}\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}\boldsymbol{X}_N(z)\Big)\big(z\boldsymbol{I} - \boldsymbol{X}_N(z)\big)^{-1}\boldsymbol{e}. \tag{4.24}$$

As a shorthand, define $\boldsymbol{P}(z) = z\boldsymbol{I} - \boldsymbol{X}(z)$. Invoking the definition of the matrix inverse, equation (4.24) can also be written as

$$U_N(z) = \Big((z-1)p_{0,N}\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}\boldsymbol{X}_N(z)\Big)\frac{\text{adj}(\boldsymbol{P}_N(z))}{\det(\boldsymbol{P}_N(z))}\boldsymbol{e}. \tag{4.25}$$

Note 53. *Let adj($\boldsymbol{A}$) denote the adjugate matrix of a matrix $\boldsymbol{A}$ and det($\boldsymbol{A}$) the determinant of $\boldsymbol{A}$.*

As the pgfs of the arrival processes are meromorphic and calculating the adjugate of a matrix does not introduce singularities, the singularities of $U_N(z)$ are the zeroes of $\det(\boldsymbol{P}_N(z))$.

Note 54. *These zeroes can be numerically found using any root-finding algorithm. Kravanja's method [60, 61] does not require an initial value and is well-suited for the kind of searching that needs to be done here.*

Let us verify that all the singularities of the N/∞ model are singular poles and compare the location of these poles to the singularities of the ∞/∞ model for some practical examples. Here, we use a different symbol to depict each type of singularity. We have

∞/∞ pole: ○
∞/∞ branch point: □
N/∞ pole: •

First, let us again study an output-queueing switch as described in appendix A.4.

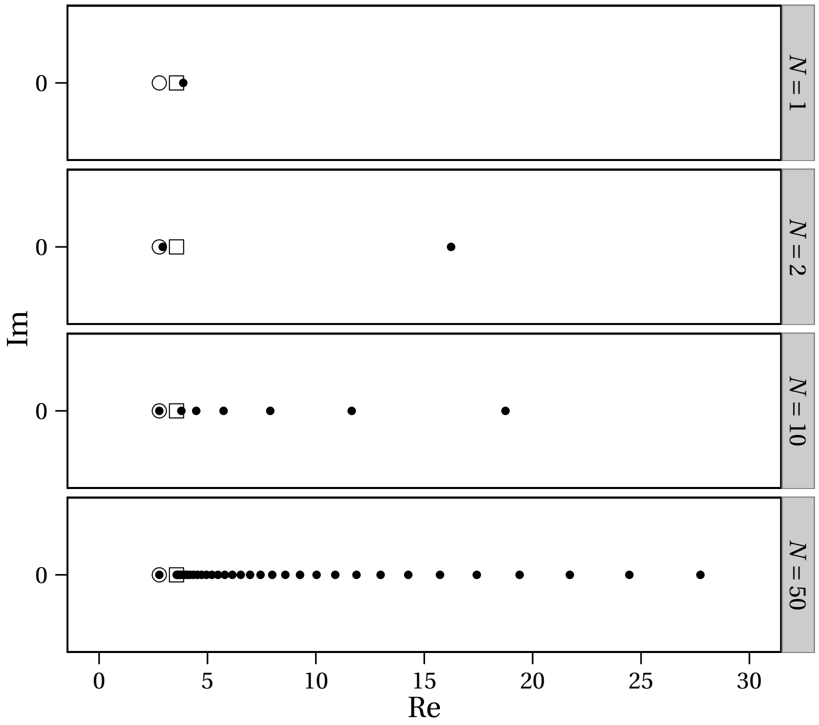

Figure 4.3: Location of poles for 2×2 *switch: geometric.*

Consider a 2×2 switch with $b = 1$ and $\lambda_1 = 0.40$, $\lambda_2 = 0.35$. Figure 4.3, depicts the location of the poles in the complex plane, with the real part shown on the horizontal axis and the imaginary part on the vertical axis, for different values of the class-1 queue capacity N for the finite case. Furthermore, the singularities for the infinite case are also shown, which for the parameter settings considered here, correspond to geometric tail behavior. For increasing N, the number of poles increases and their modulus decreases. The pole with smallest modulus clearly converges to the value of the (dominant) pole in the infinite case and all other poles converge to the branch point of the infinite case and it is apparent that in the limit a branch cut will be formed. Note that we focus on the range close to z_L and z_B and thus poles with very large modulus are not visible on the figure.

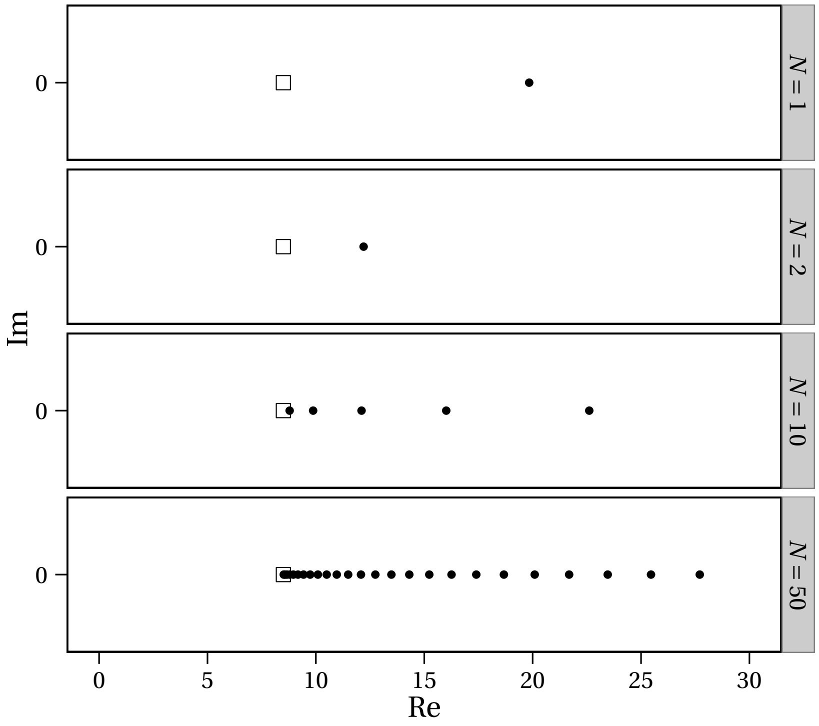

Figure 4.4: Location of poles for 2 × 2 *switch: non-geometric.*

Next, in figure 4.4, consider the same arrival process but with $\lambda_1 = 0.40$, $\lambda_2 = 0.12$, which corresponds to non-geometric tail behavior, as z_L does not exist and the branch point z_B is dominant. In the finite case, we see that all poles now seem to converge to the branch point, and, in contrast to in the previous figure, none of the poles have a modulus smaller than z_B.

Now, let us check if the entire (structure of the) arrival process influences the location of the poles. Let us study a 4 × 4 switch in figure 4.5, where we chose $b = 1$ and $\lambda_1 = 0.40$, $\lambda_2 = 0.47$. As this corresponds to geometric tail behavior in the infinite case there is again a single pole that converges to the dominant pole of the infinite case. However, there are now 3 branch points and poles accumulate at each of them.

Finally, let us also study the bivariate independent geometric arrival process detailed in A.2.1 for $p_1 = 1/7$, $p_2 = 1/3$. Consider figure 4.6. Again, this corresponds to the geometric tail behavior and thus one pole converges to the dominant pole in the infinite case and the others line up at the branch point. However, for this arrival process there are fewer poles Also note that the arrival process has a pole around 7.7 that is transferred to the class-2 system content.

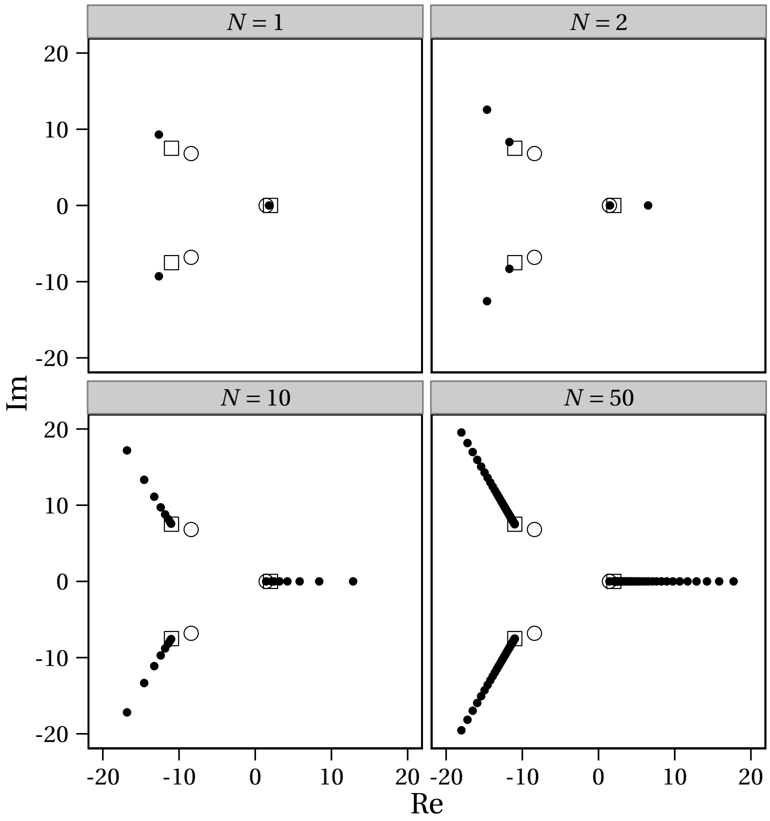

Figure 4.5: Location of poles for 4 × 4 *switch: geometric.*

4.3.2 Numerically computing the tail behavior

Once the singularities have been located (numerically), the tail of u_{2_N} is obtained by investigating the behavior of its pgf around its poles by computing the residue in a pole. One is not restricted to the dominant-pole approximation but can also use all poles. In figure 4.7, we plot the tail of $u_{2_{20}}$ using only the dominant pole (squares), all poles (circles) and the tail of u_{2_∞} (triangles) for the 6 × 6 switch arrival process with $b = 1$, $\lambda_1 = 0.4$, $\lambda_2 = 0.05$. The single (dominant) pole approximation performs badly (difference $\sim 10^2$) as the poles lie close together. When all poles are taken into account, the results lie very close to the ones obtained in the infinite case.

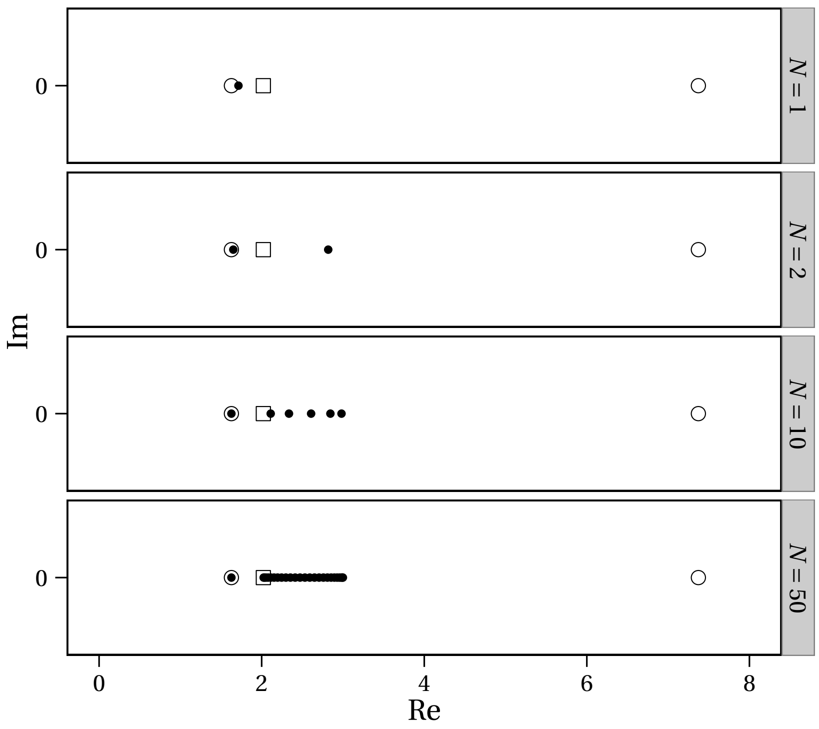

Figure 4.6: Location of poles for bivariate independent geometric arrival process : geometric.

4.4 Explicit expression for $U_N(z)$ for arrival processes with maximum 2 class-1 arrivals per slot

We restrict the maximum number of class-1 arrivals per slot to two. Thus, $\forall z$: $A_i(z) = A_i^*(z) = 0$, $i > 2$. And, to exclude the degenerate case where there is no queueing for class-1, let $A_2(1) > 0$. We focus on the steady-state class-2 system content in the finite and infinite case.

For the ∞/∞ model, recall expression (4.4) for the class-2 system content. As with the N/∞ model, let us adopt a change of notation replacing the subscript 2 by ∞ in order to avoid confusion with the N/∞ model. We then have

$$U_\infty(z) = p_{0,\infty} \frac{A_0^*(z)(z-1)(Y_1(z)-1)}{(z-Y_1(z))(A_0^*(z)-1)}, \tag{4.26}$$

with $p_{0,\infty} = 1 - \lambda_1 - \lambda_2$. As $Y_1(z)$ is the unique root of the kernel with $|x| < 1$ when $|z| < 1$, the kernel plays a crucial role. The kernel is given by

$$F(x,z) = \sum_{i=0}^{\infty} A_i(z)x^i - x \tag{4.27}$$

$$= A_2(z)x^2 + (A_1(z)-1)x + A_0(z). \tag{4.28}$$

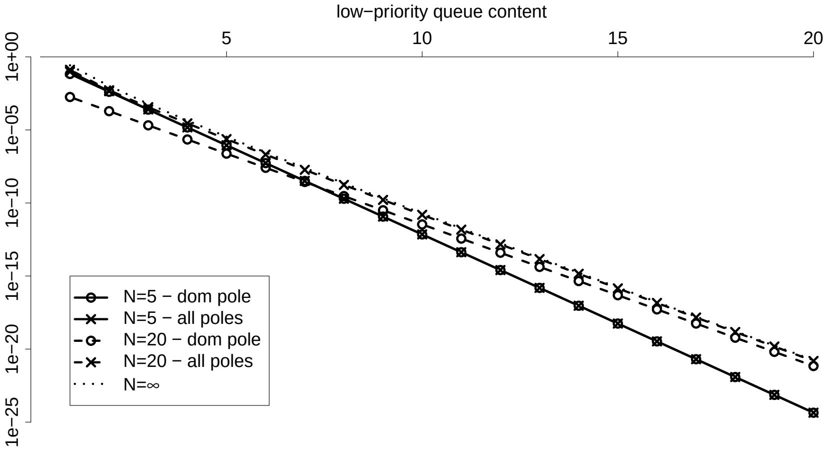

Figure 4.7: Low-priority system content for $N = \infty$ and $N = 20$ (using dominant pole approximation and all poles.

As the kernel turns out to be quadratic in x due to the restriction on the class-1 arrivals, it has two roots given by

$$Y_1(z) = \frac{1 - A_1(z) - \sqrt{(1 - A_1(z))^2 - 4A_0(z)A_2(z)}}{2A_2(z)} \tag{4.29}$$

and

$$Y_2(z) = \frac{1 - A_1(z) + \sqrt{(1 - A_1(z))^2 - 4A_0(z)A_2(z)}}{2A_2(z)} \tag{4.30}$$

The square-root in the expression of $Y_1(z)$ causes the non-exponential tail probabilities in $U_\infty(z)$, as it gives rise to branch cuts and branch points (points where the expression under the square root equals 0). This paper will unveil that $Y_1(z)$ also appears in the expression for $U_N(z)$, but that its square-root is in fact canceled by the square-root of $Y_2(z)$.

4.4.1 Expressing $U_N(z)$in terms of $Y_i(z)$

Note 55. *Here, we require that $\lambda_1 + \lambda_2 < 1$, which guarantees that the corresponding ∞/∞ model is stable.*

In this subsection, an explicit expression for $U_N(z)$ is established. In the process, we uncover a crucial relation between the characteristic polynomial of a recurrence relation for the determinant in the finite case and the kernel in the infinite case. Furthermore, the expression for $U_N(z)$ also correctly converges to $U_\infty(z)$ when taking the limit for N. First, some manipulations on the matrices in (4.25) will be performed.

Note 56. *In the remainder, we start the count of rows and columns of matrices and vectors at* 0.

Lemma 4.1. *The function $U_N(z)$ can be written as*

$$U_N(z) = (z-1)p_{0,N}\frac{T_N(z)}{z^N D_N(z)}, N \geq 2, \tag{4.31}$$

with $T_N(z) = \left(\sum_{i=0}^{2} A_i(z) \sum_{j=0}^{N} adj(\boldsymbol{P}_N(z))_{ij}\right)$ and $D_N(z)$ the determinant of

$$\boldsymbol{Q}_N(z) = \begin{bmatrix} z-A_0(z) & -A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2^*(z) \\ 0 & & \cdots & 0 & -A_0(z) & 1-A_1^*(z) \end{bmatrix}. \tag{4.32}$$

Proof. Multiplication of vector $\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}$ and matrix $\boldsymbol{X}_N(z)$ in (4.25) results in the vector $\begin{bmatrix}A_0(z) & A_1(z) & A_2(z) & 0 & \cdots & 0\end{bmatrix}$. Multiplication of the adjugate matrix of $\boldsymbol{P}_N(z)$ with $\boldsymbol{e}$ leads to the column vector $[\sum_{j=0}^{N} \text{adj}(\boldsymbol{P}_N(z))_{ij}]$. Furthermore, all elements of $\boldsymbol{P}_N(z)$ but the ones in the first row have a factor z. Therefore, it is easily seen that $\det(\boldsymbol{P}_N(z)) = z^N D_N(z)$. □

Let us commence by calculating $D_N(z)$. To that end, a linear homogeneous recurrence relation for $\{D_N(z)\}_{N=0}^{\infty}$ is constructed, which turns out to be crucial. This recurrence relation is then solved by means of generating functions.

Theorem 4.1. *The determinant $D_N(z)$ is a solution of the recurrence relation*

$$D_N(z) = (1-A_1(z))D_{N-1}(z) - A_0(z)A_2(z)D_{N-2}(z), \quad N \geq 2 \tag{4.33}$$

with seed functions

$$D_0(z) = z-1, \tag{4.34}$$

$$D_1(z) = z(1-A_1^*(z)) - A_0(z). \tag{4.35}$$

Proof. We first subtract the second row of $\boldsymbol{Q}_N(z)$ from its first row, which does not affect its determinant $D_N(z)$.

$$\begin{bmatrix} z & -1 & 0 & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & & -A_0(z) & 1-A_1(z) & -A_2^*(z) \\ 0 & & \cdots & & 0 & -A_0(z) & 1-A_1^*(z) \end{bmatrix}. \tag{4.36}$$

Laplace expansion along the last row (and then last column) of this matrix leads to

$$D_N(z) = (1-A_1^*(z))E_{N-1}(z) - A_0(z)A_2^*(z)E_{N-2}(z), \quad N \geq 2, \tag{4.37}$$

with $E_N(z)$ the determinant of the $(N+1)\times(N+1)$ matrix

$$\begin{bmatrix} z & -1 & 0 & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & & -A_0(z) & 1-A_1(z) & -A_2(z) \\ 0 & & \cdots & & 0 & -A_0(z) & 1-A_1(z) \end{bmatrix}. \tag{4.38}$$

Notice that the matrices giving rise to $E_N(z)$ and $D_N(z)$ only differ in the last column. Again performing Laplace expansion of (4.38) along the last row (and then last column), it is clear that $E_N(z)$ fulfills a recurrence relation:

$$E_N(z) = (1-A_1(z))E_{N-1}(z) - A_0(z)A_2(z)E_{N-2}(z), \quad N \geq 2, \tag{4.39}$$

with seed functions

$$E_1(z) = (1-A_1(z))E_0(z) - A_0(z), \tag{4.40}$$

$$E_0(z) = z. \tag{4.41}$$

Eliminating $E_{N-2}(z)$ from expressions (4.37) and (4.39) leads to

$$D_N(z) = E_N(z) - A_2(z)E_{N-1}(z), N \geq 1. \tag{4.42}$$

Note that expression (4.35) can also be obtained from this expression.

Since $D_N(z)$ is a linear combination of $E_N(z)$ and $E_{N-1}(z)$ for $N \geq 1$, $D_N(z)$ fulfils the same recurrence equation as $E_N(z)$, i.e.,

$$D_N(z) = (1-A_1(z))D_{N-1}(z) - A_0(z)A_2(z)D_{N-2}(z), \quad N \geq 3. \tag{4.43}$$

In order for this recurrence relation to be valid for $N = 2$, $D_0(z)$ should be chosen as in (4.34), which completes the proof. □

Note 57. *Note that $D_0(z)$ and $D_1(z)$ are chosen such that the recurrence relation is valid for all $N \geq 2$. Therefore, $D_0(z)$ has no real meaning as determinant.*

Note 58. *Theorem 4.1 states that $\{D_N(z)\}_{N=0}^{\infty}$ is a linear homogeneous recurrence relation of order 2. The order is due to the maximum number of class-1 arrivals in a slot.*

Solving the linear homogeneous recurrence relation in theorem 4.1 can easily be achieved, for instance by means of generating functions.

Lemma 4.2. *The generating function $D(x,z)$ of $\{D_N(z)\}_{N=0}^{\infty}$, defined as*

$$D(x,z) = \sum_{N=0}^{\infty} D_N(z)x^N, \tag{4.44}$$

is given by

$$D(x,z) = A_0(z)\frac{(1-A_0^*(z)-(z-1)A_2(z))x+(z-1)}{F(A_0(z)x,z)}, \tag{4.45}$$

with $F(x,z)$ the kernel in the infinite case (expression (4.28)).

Proof. Multiplying all terms in (4.33) by x^N and summing over all valid N leads to an expression for $D(x,z)$ as a function of $D_0(z)$ and $D_1(z)$. Inserting these seed functions leads to (4.45). □

Note 59. *The denominator of the generating function is directly related to the characteristic polynomial of the underlying recurrence relation and the roots of this polynomial lead to geometric terms in the final expression of $D_N(z)$. Both surprisingly and crucially, the characteristic polynomial is related to the kernel $F(x,z)$ in the infinite case (see expression (4.45)), and the root $Y_1(z)$ of the kernel will thus appear in the final expression of $D_N(z)$ in the finite case. This turns out to be the crucial link between the finite and the infinite cases.*

Theorem 4.2. *For $N \geq 0$, $D_N(z)$ is given by*

$$D_N(z) = \frac{(1-A_0^*(z))}{A_2(z)(Y_2(z)-Y_1(z))}\left[\frac{z-Y_1(z)}{1-Y_1(z)}\left(\frac{A_0(z)}{Y_1(z)}\right)^N - \frac{z-Y_2(z)}{1-Y_2(z)}\left(\frac{A_0(z)}{Y_2(z)}\right)^N\right]. \quad (4.46)$$

Here $Y_1(z)$ and $Y_2(z)$ are as defined in (4.29) and (4.30).

Proof. The function $D_N(z)$ is calculated by writing expression (4.45) as power series in x. This can be done by partial fraction expansion of the rational expression. As the denominator equals $F(xA_0(z),z)$, its roots in x are $Y_1(z)/A_0(z)$ and $Y_2(z)/A_0(z)$. The partial fraction expansion then equals

$$\begin{aligned} D(x,z) = & \frac{(z-Y_1(z))(A_2(z)Y_1(z)-A_0(z))Y_2(z)}{(Y_1(z)-Y_2(z))A_0(z)\left(1-x\dfrac{A_0(z)}{Y_1(z)}\right)} \\ & + \frac{(z-Y_2(z))(A_2(z)Y_2(z)-A_0(z))Y_1(z)}{(Y_2(z)-Y_1(z))A_0(z)\left(1-x\dfrac{A_0(z)}{Y_2(z)}\right)}. \end{aligned} \quad (4.47)$$

Writing both terms on the right as geometric series (cf. (4.44)), identifying the coefficients of x^N on both sides, and using that $Y_1(z)$ and $Y_2(z)$ are roots of kernel (4.28) leads to (4.46). □

Next, we concentrate on calculating $T_N(z)$, the numerator of $U_N(z)$ (4.31). In a long and tedious process, let us write the different terms in this numerator as functions of certain determinants, much like $D_N(z)$. For all these determinants, suitable linear recurrence relations will be constructed and solved, finally leading to an expression for $T_N(z)$. Recall that, here, $\boldsymbol{P}_N(z)$ is given by

$$\begin{bmatrix} z-A_0(z) & -A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ -zA_0(z) & z-zA_1(z) & -zA_2(z) & \cdots & 0 & 0 \\ 0 & -zA_0(z) & z-zA_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -zA_0(z) & z-zA_1(z) & -zA_2^*(z) \\ 0 & & \cdots & 0 & -zA_0(z) & z-zA_1^*(z) \end{bmatrix}. \quad (4.48)$$

Lemma 4.3. *The terms* $\sum_{j=0}^{N}[adj(\boldsymbol{P}_N(z))]_{ij}$ *(*$N \geq 2$, $i = 0,1,2$*) can be written as*

$$\sum_{j=0}^{N}[adj(\boldsymbol{P}_N(z))]_{0j} = z^{N-1}\Bigg[zF_{N-1}(z) - H_{N-1}(z) + \sum_{j=2}^{N}(-1)^j K_{j-1}(z)F_{N-j-1}(z)\Bigg], \tag{4.49}$$

$$\sum_{j=0}^{N}[adj(\boldsymbol{P}_N(z))]_{1j} = z^{N-1}\Bigg[zA_0(z)F_{N-2}(z) + (z - A_0(z))F_{N-2}(z) - \sum_{j=2}^{N}(-1)^j L_{j-1}(z)F_{N-j-1}(z)\Bigg], \tag{4.50}$$

$$\sum_{j=0}^{N}[adj(\boldsymbol{P}_N(z))]_{2j} = z^{N-1}\Bigg[zA_0(z)^2F_{N-3}(z) + (z - A_0(z))A_0(z)F_{N-3}(z) + \sum_{j=2}^{N}(-1)^j M_{j-1}(z)F_{N-j-1}(z)\Bigg]. \tag{4.51}$$

Here, $F_N(z)$, $H_N(z)$, $K_N(z)$, $L_N(z)$ *and* $M_N(z)$ *are the respective determinants of the following* $(N+1)\times(N+1)$ *matrices (*$N \geq 1$*)*

$$\begin{bmatrix} 1-A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2^*(z) \\ 0 & & \cdots & 0 & -A_0(z) & 1-A_1^*(z) \end{bmatrix}, \tag{4.52}$$

$$\begin{bmatrix} -A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2^*(z) \\ 0 & & \cdots & 0 & -A_0(z) & 1-A_1^*(z) \end{bmatrix}, \tag{4.53}$$

$$\begin{bmatrix} -A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ 1-A_1(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & 1-A_1(z) & -A_2(z) & 0 \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2(z) \end{bmatrix}, \tag{4.54}$$

$$\begin{bmatrix} z-A_0(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ -A_0(z) & -A_2(z) & 0 & \cdots & 0 & 0 \\ 0 & 1-A_1(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & -A_0(z) & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & 1-A_1(z) & -A_2(z) & 0 \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2(z) \end{bmatrix}, \tag{4.55}$$

and

$$\begin{bmatrix} z-A_0(z) & -A_1(z) & 0 & \cdots & 0 & 0 \\ -A_0(z) & 1-A_1(z) & 0 & \cdots & 0 & 0 \\ 0 & -A_0(z) & -A_2(z) & \cdots & 0 & 0 \\ 0 & 0 & 1-A_1(z) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & 1-A_1(z) & -A_2(z) & 0 \\ 0 & & \cdots & -A_0(z) & 1-A_1(z) & -A_2(z) \end{bmatrix}, \tag{4.56}$$

and $F_0(z) = 1 - A_1^*(z)$ *and* $F_{-1}(z) = 1$.

Proof. Since the element on the i-th row and j-th column of $\operatorname{adj}(\boldsymbol{P}_N(z))$ is given by $(-1)^{i+j}$ multiplied by the determinant of the $N \times N$ matrix that origins from deleting row j and column i of matrix $\boldsymbol{P}_N(z)$, it is easily seen that

$$[\operatorname{adj}(\boldsymbol{P}_N(z))]_{ij} = \begin{cases} z^N [\operatorname{adj}(\boldsymbol{Q}_N(z))]_{ij} & \text{if } j = 0 \\ z^{N-1} [\operatorname{adj}(\boldsymbol{Q}_N(z))]_{ij} & \text{if } j > 0. \end{cases} \tag{4.57}$$

First,

$$[\operatorname{adj}(\boldsymbol{Q}_N(z))]_{00} = F_{N-1}(z),\ N \geq 2, \tag{4.58}$$

with $F_N(z)$ the determinant of the $(N+1) \times (N+1)$ matrix in (4.52), $N \geq 1$. Similarly, we have

$$[\operatorname{adj}(\boldsymbol{Q}_N(z))]_{01} = -H_{N-1}(z),\ N \geq 2, \tag{4.59}$$

with $H_N(z)$ the determinant of the $(N+1) \times (N+1)$ matrix in (4.53), $N \geq 1$.

Deleting the j-th row ($2 \leq j \leq N$) and the zero-th column of $\boldsymbol{Q}_N(z)$ leads to a matrix of the form

$$\begin{bmatrix} \boldsymbol{B}_{j-1}(z) & \boldsymbol{0} \\ \boldsymbol{C}(z) & \boldsymbol{G}_{N-j-1}(z) \end{bmatrix}, \tag{4.60}$$

where $\boldsymbol{B}_N(z)$ and $\boldsymbol{G}_N(z)$ are the $(N+1) \times (N+1)$ matrices of (4.54) and (4.52), $\boldsymbol{0}$ is a zero matrix of appropriate size and $\boldsymbol{C}(z)$ is a matrix that is of no importance for the remainder. Indeed, the determinant of matrix (4.60) is the product of the determinants of $\boldsymbol{B}_{j-1}(z)$ and $\boldsymbol{G}_{N-j-1}(z)$, and thus,

$$\sum_{j=2}^{N} [\operatorname{adj}(\boldsymbol{Q}_N(z))]_{0j} = \sum_{j=2}^{N} (-1)^j K_{j-1}(z) F_{N-j-1}(z). \tag{4.61}$$

Note that $F_0(z)$ and $F_{-1}(z)$ have to be defined as in the lemma to make this expression valid. Expressions (4.57), (4.58), (4.59) and (4.61) lead to (4.49). The other two sums (expressions (4.50) and (4.51)) can be obtained in the same way; these calculations are therefore omitted here.

□

Lemma 4.4. *The functions $F_N(z)$ and $H_N(z)$ are respectively given by*

$$\begin{aligned} F_N(z) &= \frac{(A_1^*(z)+A_2(z)Y_1(z)-1)Y_2(z)}{Y_1(z)-Y_2(z)}\left(\frac{A_0(z)}{Y_1(z)}\right)^N \\ &+ \frac{(A_1^*(z)+A_2(z)Y_2(z)-1)Y_1(z)}{Y_2(z)-Y_1(z)}\left(\frac{A_0(z)}{Y_2(z)}\right)^N, \ N\geq 0, \qquad (4.62)\\ H_N(z) &= \frac{(A_0(z)A_1^*(z)-(1-A_0(z))A_2(z)Y_1(z))Y_2(z)}{A_0(z)(Y_1(z)-Y_2(z))}\left(\frac{A_0(z)}{Y_1(z)}\right)^N \\ &+ \frac{(A_0(z)A_1^*(z)-(1-A_0(z))A_2(z)Y_2(z))Y_2(z)}{A_0(z)(Y_1(z)-Y_2(z))}\left(\frac{A_0(z)}{Y_2(z)}\right)^N, \ N\geq 1. \qquad (4.63)\end{aligned}$$

Furthermore, $K_N(z)$, $L_N(z)$ and $M_N(z)$ are given by

$$K_N(z) = -(-A_2(z))^N, \ N\geq 1, \qquad (4.64)$$

$$L_N(z) = z(-A_2(z))^N, \ N\geq 1, \qquad (4.65)$$

$$M_N(z) = (z-A_0(z)-zA_1(z))(-A_2(z))^{N-1}, \ N\geq 1. \qquad (4.66)$$

Proof. $F_N(z)$ and $H_N(z)$ fulfil the same second-order linear recurrence relation as $D_N(z)$, see theorem 4.1, albeit with different seed functions:

$$F_0(z) = 1-A_1^*(z), \qquad (4.67)$$

$$F_1(z) = (1-A_1(z))(1-A_1^*(z))-A_0(z)A_2(z), \qquad (4.68)$$

$$H_0(z) = -A_1^*(z), \qquad (4.69)$$

$$H_1(z) = -A_1(z)(1-A_1^*(z))-A_0(z)A_2(z). \qquad (4.70)$$

Therefore, deriving (4.62)-(4.63) is achieved by an analogous method to the one used to obtain (4.46) from theorem 4.1 for $D_N(z)$, i.e. establishing generating functions for $\{F_N(z)\}_{N=0}^{\infty}$ and $\{H_N(z)\}_{N=0}^{\infty}$, using partial fraction expansion and writing its terms in power series.

The matrices (4.54), (4.55) and (4.56) are all of the form

$$\begin{bmatrix} \boldsymbol{B}_1(z) & \boldsymbol{0} \\ \boldsymbol{C}(z) & \boldsymbol{G}_{N-2}(z)\end{bmatrix}, \qquad (4.71)$$

with $\boldsymbol{B}_1(z)$ a 2×2 matrix and $\boldsymbol{G}_{N-2}(z)$ an $(N-1)\times(N-1)$ diagonal matrix with the diagonal elements equal to $-A_2(z)$. Therefore, the determinants equal the determinant of $\boldsymbol{B}_1(z)$ multiplied by $(-A_2(z))^{N-1}$ which leads to expressions (4.64)-(4.66). □

Lemma 4.5. *The numerator of expression (4.31) is given by*

$$T_N(z) = \frac{(z-1)p_{0,N}A_0^*(z)z^N\left[\left(\frac{A_0(z)}{Y_1(z)}\right)^N - \left(\frac{A_0(z)}{Y_2(z)}\right)^N\right]}{A_2(z)(Y_2(z)-Y_1(z))}. \tag{4.72}$$

Theorem 4.3. *The pgf $U_N(z)$ of the class-2 system content is given by*

$$U_N(z) = \left(\frac{1-\lambda_1}{1-(A_2(1)/A_0(1))^N} - \lambda_2\right)\frac{(z-1)A_0^*(z)}{(1-A_0^*(z))} \\ \frac{\left(Y_2(z)^N - Y_1(z)^N\right)}{\left(\frac{z-Y_1(z)}{1-Y_1(z)}Y_2(z)^N - \frac{z-Y_2(z)}{1-Y_2(z)}Y_1(z)^N\right)}. \tag{4.73}$$

Proof. The numerator of $U_N(z)$ is given by$T_N(z)$, while the denominator is given by $z^N D_N(z)$; $D_N(z)$ is given by equation (4.46). Finally, $p_{0,N}$ is calculated by using the normalization condition $U_N(1) = 1$, leading to

$$p_{0,N} = \frac{1-\lambda_1}{1-(A_2(1)/A_0(1))^N} - \lambda_2. \tag{4.74}$$

□

Corollary 4.1. *The correct limiting behavior from the finite to the infinite case is established as* $\lim_{N\to\infty} U_N(z) = U_\infty(z)$.

Proof. For z inside the complex unit circle, it is easily proved that $|Y_1(z)| < 1 < |Y_2(z)|$ (through e.g. Rouché's theorem and the implicit function theorem). Therefore, when taking the limit of (4.73) for $N \to \infty$, the terms in $Y_2(z)^N$ dominate those in $Y_1(z)^N$, both in numerator and denominator. Furthermore, for a stable system, $\lambda_1 < 1$, which results in $A_2(1) < A_0(1)$. Therefore $\lim_{N\to\infty}(A_2(1)/A_0(1))^N = 0$. These two observations immediately lead to (4.26). □

Corollary 4.2. *If the $A_i(z)$ ($i = 0,1,2$) are meromorphic, $U_N(z)$ is meromorphic and thus cannot have branch points.*

Proof. If the $A_i(z)$ ($i = 0,1,2$) are meromorphic so are the seed values of all recurrence relations used in this chapter. As the recurrence relations consist of basic operations and the meromorphic functions form a field with respect to the usual pointwise operations followed by redefinition at the removable singularities, evidently $D_N(z)$, $F_N(z)$, $H_N(z)$, $K_N(z)$, $L_N(z)$, $M_N(z)$ and finally $U_N(z)$ are all meromorphic. □

Note 60. *This corollary asserts that $U_N(z)$ (expression (4.73)) cannot contain branch points and thus the square root in $Y_1(z)$ is canceled by the square root in $Y_2(z)$ for* *each finite N. This is highly comparable to the expression of the N-th Fibonacci number,*

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^N - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^N, \tag{4.75}$$

that 'seems' to contain $\sqrt{5}$ while the Fibonacci numbers are obviously integer as the square-roots in both terms cancel out.

4.4.2 Mean class-2 system content

Although the main point of this section is obtaining an "explicit" expression for $U_N(z)$, achieved in (4.73), and studying the limit behavior to $U_\infty(z)$, this explicit expression can also be used for other purposes. By taking the derivative of (4.73) and evaluating in $z = 1$, the average of u_{2_N}, the class-2 system content in a priority queue with class-1 capacity N is

$$\begin{aligned} \mathrm{E}\left[u_{2_N}\right] = \lambda_2 + \frac{1}{Y_2(1)^N(1-\lambda_T)+\lambda_2)}&\left[Y_2(1)^N\lambda_{12} + \frac{Y_2(1)^N\lambda_{11}\lambda_2}{2(1-\lambda_1)}\right.\\ &\left.+\frac{(Y_2(1)^N-1)\lambda_{22}}{2} + \frac{\lambda_2}{Y_2(1)-1} + \frac{[(1-\lambda_1)Y_2'(1)-\lambda_2 Y_2(1)]Y_2(1)^{N-1}N}{(Y_2(1)^N-1)}\right]. \end{aligned} \tag{4.76}$$

Furthermore, this allows us to study how u_{2_N} approaches u_{2_∞}, the class-2 system content in a priority queue with infinite capacity. For arbitrarily large N , we have

$$\mathrm{E}\left[u_{2_N}\right] \sim \mathrm{E}\left[u_{2_\infty}\right] + \frac{[(1-\lambda_1)Y_2'(1)-\lambda_2 Y_2(1)]N}{(1-\lambda_T)Y_2(1)^{N+1}}. \tag{4.77}$$

Evidently, $\mathrm{E}\left[u_{2_N}\right] < \mathrm{E}\left[u_{2_\infty}\right]$ as $Y_2(1) > 1$, $Y_2'(1) < 0$. Also, as $Y_2(1) > 1$, the linear evolution in N in the numerator is dampened by the appearance of $Y_2(1)^N$ in the denominator so the geometric convergence rate is $1/Y_2(1)$.

4.4.3 Location of singularities of $U_N(z)$

We have established an explicit expression for $U_N(z)$, the pgf of the class-2 system content when the high-priority capacity equals N customers. We showed that the roots of the kernel in the infinite case appear in $U_N(z)$ and proved the limit for N going to ∞. In this section, singularity analysis is performed in the finite case, i.e., we take a closer look at the location of the singularities of $U_N(z)$ and investigate how the singularities accumulate into a branch cut for N going to infinity.

Lemma 4.6. *If $A_i(z)$ ($i = 0,1,2$) is meromorphic, a pole of $U_N(z)$ is a root of $D_N(z)$ or a pole of $A_0(z)$, $A_1(z)$ or $A_2(z)$.*

Proof. In the numerator and denominator of (4.31) only summations and multiplications of terms and factors in $A_i(z)$ occur, from which the lemma follows. □

Corollary 4.3. *If $A_i(z)$ ($i = 0, 1, 2$) is meromorphic, $U_N(z)$ is meromorphic as well and has only poles as singularities.*

Note 61. *By restricting the input pgfs (the $A_i(z)$) to meromorphic functions, we avoid having to deal with branch cuts introduced by these input pgfs, which would clutter observing the formation of the branch cut in the infinite case from poles in the finite case. In the remainder, we therefore use the more specific term 'pole' instead of 'singularity' in the finite case.*

Lemma 4.7. *The roots of $T_N(z)$ inlude* 0 *and* 1.

Proof. From (4.72), this is quite straightforward. Clearly, 0 and 1 are roots, with resp. multiplicities N and 1. Also, $A_0^*(z)$ is increasing by definition. Furthermore, when $Y_1(z) = Y_2(z)$, both the numerator and denominator of (4.72) are zero. Invoking L'Hôpital's rule asserts that this does not lead to a root. □

Corollary 4.4. *All roots of $D_N(z)$ except* 0 *or* 1 *are poles of $U_N(z)$.*

It is known that the dominant pole of a pgf lies on the positive real axis in the interval $]1, \infty]$. We first have following lemma.

Lemma 4.8. *The functions $D_N(z)$ ($N \geq 0$) have a root in* 1.

Proof. Since $A_0^*(1) = 1$, it is clear that $D_0(1) = D_1(1) = 0$. From recurrence relation (4.33) then follows that $D_N(1) = 0$ for all $N \geq 0$. □

However, since 1 is also a root of $T_N(z)$ this root is canceled once. The fact that 1 is not a pole of $U_N(z)$ is also immediate from following lemma.

Lemma 4.9. *The first derivatives of the functions $D_N(z)$ in* 1 *are ordered in the following way:*

$$0 < \ldots < D'_{N+1}(1) < D'_N(1) < D'_{N-1}(1) < \ldots < D'_1(1) < D'_0(1) = 1\,. \tag{4.78}$$

Proof. First, it is evident that $D'_0(1) = 1$. Furthermore, reverting to generating functions, (4.33) leads to

$$\sum_{N=2}^{\infty} D'_N(1)x^N = \sum_{N=2}^{\infty} (1 - A_1(1))D'_{N-1}(1)x^N - A_0(1)A_2(1) \sum_{N=2}^{\infty} D'_{N-2}(1)x^N\,. \tag{4.79}$$

Subsequent partial fraction expansion yields

$$D'_N(1) = A_0(1)^N + \frac{\lambda_2}{A_0(1) - A_2(1)}(A_2(1)^N - A_0(1)^N)\,. \tag{4.80}$$

Hence, as $A_0(1) - A_2(1) = 1 - \lambda_1$, we have

$$D'_N(1) = \frac{1 - \lambda_T}{1 - \lambda_1} A_0(1)^N + \frac{\lambda_2}{1 - \lambda_1} A_2(1)^N\,, \tag{4.81}$$

from which the lemma is evident. □

Next, it is proven that $D_N(z)$ has at least one root in $]1, R_A[$, with R_A the minimum of the radii of convergence of the $A_i(z)$. We call the smallest of such roots $z_{N,1}$.

Note 62. *In the remainder, frequent use of the intermediate value theorem for a continuous function is made. This theorem states that if f is a real-valued continuous function on the interval $[a,b]$, and u is a number between $f(a)$ and $f(b)$, then there is a $c \in]a,b[$ such that $f(c) = u$.*

Lemma 4.10. *If $A_i(z)$ ($i = 0,1,2$) is meromorphic, $D_N(z)$ has at least one root in $]1, R_A[$ for all $N \geq 1$. If the smallest of such roots is denoted by $z_{N,1}$, it holds that $z_{N,1} < z_{N-1,1}$, $N \geq 2$.*

Proof. We prove this by induction on N, starting with $N = 1$ and $N = 2$. Firstly, since $A_i(z)$ and $A_i^*(z)$ are partial generating functions that are meromorphic and since at least one of the $A_i(z)$ goes to infinity for $z \to R_A$, it is easily seen from (4.35) that $D_1(R_A) = -\infty$. Combining lemmata 4.8 and 4.9 yields $D_1(1+\epsilon) > 0$, for $\epsilon > 0$ and small enough. Hence, due to the intermediate value theorem, $D_1(z)$ must have a root $z_{1,1} \in]1, R_A[$, and (4.33) yields (for $N = 2$)

$$D_2(z_{1,1}) = -A_0(z_{1,1})A_2(z_{1,1})D_0(z_{1,1}). \tag{4.82}$$

As $A_0(z_{1,1}), A_2(z_{1,1}), D_0(z_{1,1}) > 0$, evidently $D_2(z_{1,1}) < 0$. Again, lemmata 4.8 and 4.9 and the intermediate value theorem lead to the existence of at least one zero of $D_2(z)$ in $]1, R_A[$ and $z_{2,1} < z_{1,1}$.

Similarly, say $z_{N-1,1}$ exists for an $N \geq 2$, then

$$D_N(z_{N-1,1}) = -A_0(z_{N-1,1})A_2(z_{N-1,1})D_{N-2}(z_{N-1,1}). \tag{4.83}$$

Thus, $D_N(z_{N-1,1})$ and $D_{N-2}(z_{N-1,1})$ have opposite signs, yielding $D_N(z_{N-1,1}) < 0$, as assuming that the lemma is fulfilled for $N-1$ produces $D_{N-2}(z_{N-1,1}) > 0$. As above, lemmata 4.8 and 4.9 and the intermediate value theorem then lead to the existence of $z_{N,1}$ satisfying $z_{N,1} < z_{N-1,1}$, completing the proof. □

Corollary 4.5. *If $A_i(z)$ ($i = 0,1,2$) is meromorphic, the dominant pole of $U_N(z)$ equals $z_{N,1}$, the smallest root of $D_N(z)$ in $]1,\infty[$.*

Note 63. *A direct consequence of this lemma and consequent corollary is that the tail of the probability mass function of the low-priority system content decays slower for larger N. Intuitively, this was to be expected, as larger N means that more high-priority customers are admitted into the system, which negatively influences the low-priority system content.*

4.5 Explicit expression for $D_N(z)$ for arrival processes with maximum S class-1 arrivals per slot

In this section, we will relax the restriction on the arrival process and prove that a similar explicit expression can be found for $D_N(z)$. Here, we limit the class of arrival processes by asserting that $a(m,n) = 0$ for $m > S$, hence also $A_i(z) = 0$ for $m > S$. Hence, the number of class-1 arrivals in a slot is at most S. Due to this restriction, the kernel turns out to be a polynomial of degree S in x, given by

$$F(x,z) = \sum_{i=2}^{S} A_i(z)x^i + (A_1(z) - 1)x + A_0(z). \tag{4.84}$$

Evidently, $F(x,z)$ then has S roots in x, denoted by $Y_i(z)$, $i = 1 \ldots S$. The $Y_i(z)$ contain radicals causing non-exponential tail probabilities and $Y_1(z)$ thus causes such behavior in $U_\infty(z)$.

Lemma 4.11. *$F(x,z)$ has a single unique root for $|x| < 1$, $|z| < 1$, denoted $Y_1(z)$.*

Proof. Choose a complex number z with $|z| < 1$. Rouché's Theorem states that the functions $x \to F(x,z)$ and $x \to x$ have the same amount of roots within the complex unit circle (for x), as these functions are both analytic in the complex unit circle. Evidently $x = 0$ is the only root of the latter causing the former to also have only a single root, say $Y_1(z)$. Invoking the implicity function therorema completes the proof. □

This section first establishes a recurrence relation for $D_N(z)$. Then, we postulate an expression for $D_N(z)$ and prove that it adheres to the recurrence.

4.5.1 Recurrence relation for $D_N(z)$

As in the previous section, but with the assumption of maximum S class-1 arrivals, the determinant $D_N(z) = \det(z\boldsymbol{I} - \boldsymbol{X}(z))/z^N$, is given by

$$\begin{vmatrix} z - A_0(z) & -A_1(z) & -A_2(z) & \cdots & -A_{N-1}(z) & -A^*_N(z) \\ -A_0(z) & 1 - A_1(z) & -A_2(z) & \cdots & -A_{N-1}(z) & -A^*_N(z) \\ 0 & -A_0(z) & 1 - A_1(z) & \cdots & -A_{N-2}(z) & -A^*_{N-1}(z) \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -A_0(z) & 1 - A_1(z) & -A^*_2(z) \\ 0 & & \cdots & 0 & -A_0(z) & 1 - A^*_1(z) \end{vmatrix}. \tag{4.85}$$

Again, in parallel with the previous section, we construct a linear homogeneous recurrence relation for $\{D_N(z)\}_{N=0}^{\infty}$.

Proposition 4.1. *Let*

$$r_k(z) = (1 - A_1(z))D_{k-1}(z) - \sum_{i=2}^{min(k,S)} A_0(z)^{i-1} A_i(z) D_{k-i}(z). \tag{4.86}$$

Then, $D_N(z)$ is a solution of the recurrence relation $D_N(z) = r_N(z)$ for $N > S$, with S seed values:

$$\begin{aligned} D_N(z) &= r_N(z) - A_0(z)^{N-1} A^*_{N+1}(z)(z-1)\,, 1 < N \leq S\,, \\ D_1(z) &= z(1 - A^*_1(z)) - A_0(z)\,, \\ D_0(z) &= z - A^*_0(z)\,. \end{aligned}$$

Proof. Laplace expansion of the determinant (4.85) along the last row (recursively) yields

$$D_N(z) = (1 - A_1^*(z))E_{N-1}(z) - \sum_{i=2}^{S} A_0(z)^{i-1} A_i^*(z) E_{N-i}(z)\,, N > S\,, \tag{4.87}$$

$$\begin{aligned} D_N(z) = (1 - A_1^*(z))E_{N-1}(z) - \sum_{i=2}^{N} A_0(z)^{i-1} A_i^*(z) E_{N-i}(z) \\ - A_0(z)^N A_N^*(z)\,, 1 \le N \le S\,, \end{aligned} \tag{4.88}$$

with $E_N(z)$ the determinant of the $(N+1)\times(N+1)$ matrix

$$\begin{bmatrix} z - A_0(z) & -A_1(z) & -A_2(z) & \cdots & A_{N-1}(z) & A_N(z) \\ -A_0(z) & 1 - A_1(z) & -A_2(z) & \cdots & A_{N-1}(z) & A_N(z) \\ 0 & -A_0(z) & 1 - A_1(z) & \cdots & A_{N-2}(z) & A_{N-1}(z) \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & & \cdots & -A_0(z) & 1 - A_1(z) & -A_2(z) \\ 0 & & \cdots & 0 & -A_0(z) & 1 - A_1(z) \end{bmatrix}. \tag{4.89}$$

Notice that the matrices giving rise to $E_N(z)$ and $D_N(z)$ only differ in the last column and that this column contains zero entries for $N > S$ as $A_i(z) = 0, i > S$. Furthermore, it is clear that $E_N(z)$ fulfils following recurrence relation

$$E_N(z) = (1 - A_1(z))E_{N-1}(z) - \sum_{i=2}^{S} A_0(z)^{i-1} A_i(z) E_{N-i}(z)\,, N > S\,, \tag{4.90}$$

$$E_N(z) = (1 - A_1(z))E_{N-1}(z) - \sum_{i=2}^{N} A_0(z)^{i-1} A_i(z) E_{N-i}(z) \tag{4.91}$$

$$- A_0(z)^N A_N(z)\,, 1 \le N \le S\,, \tag{4.92}$$

$$E_0(z) = z - A_0(z)\,. \tag{4.93}$$

Combining expressions (4.87) and (4.90) leads to

$$D_N(z) = E_N(z) - \sum_{i=1}^{N} A_0(z)^{i-1} A_{i+1}^*(z) E_{N-i}(z)\,, N > S\,. \tag{4.94}$$

Since $D_N(z)$ is a linear combination of $E_0(z), \ldots, E_N(z)$ it fulfils the same recurrence equation as $E_N(z)$, i.e.,

$$D_N(z) = (1 - A_1(z))D_{N-1}(z) - \sum_{i=2}^{S} A_0(z)^{i-1} A_i(z) D_{N-i}(z)\,, N > S\,. \tag{4.95}$$

□

Proposition 4.1 states that $\{D_N(z)\}_{N=0}^{\infty}$ is a linear homogeneous recurrence relation of order S. The order is due to the maximum number of class-1 arrivals in a slot.

Lemma 4.12. *The characteristic equation of this recurrence relation is given by*

$$G(x,z) = x^S - (1 - A_1(z))x^{S-1} + \sum_{i=2}^{S} A_0(z)^{i-1} A_i(z) x^{S-i}\,, \tag{4.96}$$

and has S zeros: $A_0(z)/Y_i(z)\,, i = 1..S$.

Proof. Trivial by Proposition 4.1, recalling that $A_i(z) = 0$ for $i > S$ and by

$$G(x,z) = \frac{x^S}{A_0(z)} F(A_0(z)/x, z)\,. \tag{4.97}$$

□

4.5.2 An expression for $D_N(z)$

Let us define the following shorthand notations. Let

$$\begin{aligned} g_j(z) &= \big(z - Y_j(z)\big) \prod_{\substack{i=1\\ i\neq j}}^{S} \frac{1 - Y_i(z)}{Y_j(z) - Y_i(z)} \\ W_j(z) &= A_0(z)/Y_j(z) \end{aligned}\,. \tag{4.98}$$

Following lemmata are mere technicalities but necessary for proving the subsequent theorem.

Lemma 4.13. *For* $1 < N \le S$:

$$\sum_{j=1}^{S} g_j(z) \sum_{i=N}^{S} A_0(z)^{i-1} A_i(z) W_j(z)^{N-i} = A_0(z)^{N-1} A_N^*(z)(z-1) \tag{4.99}$$

Proof.

$$\begin{aligned} &\sum_{j=1}^{S} g_j(z) \sum_{i=N}^{S} A_0(z)^{i-1} A_i(z) W_j(z)^{N-i} \\ &= A_0(z)^{N-1} \sum_{i=N}^{S} A_i(z) \sum_{j=1}^{S} g_j(z) A_0(z)^{i-N} W_j(z)^{N-i} \\ &= A_0(z)^{N-1} \sum_{i=N}^{S} A_i(z) \sum_{j=1}^{S} g_j(z) Y_j(z)^{i-N} \\ &= A_0(z)^{N-1} \sum_{i=N}^{S} A_i(z) \sum_{j=1}^{S} \big(z - Y_j(z)\big) Y_j(z)^{i-N} \prod_{\substack{i=1\\ i\neq j}}^{S} \frac{1 - Y_i(z)}{Y_j(z) - Y_i(z)} \\ &= A_0(z)^{N-1} \sum_{i=N}^{S} A_i(z)(z-1)\,. \end{aligned}$$

The final transition is achieved by noting that the Lagrange interpolation polynomial $L(x)$ of the polynomial function $f(x) = (z - x)x^{i-N}$, with interpolation points $Y_j(z)$, $j = 1 \dots S$, is exact, if the degree of $f(x)$ is smaller than the number of data points, thus for $i - N + 1 < S$. Then, replace $f(1)$ by $L(1)$. Note that the constraint on the degree is fulfilled for all $1 < N \le i \le S$. □

Lemma 4.14.

$$A_S(z) = (-1)^S A_0(z) \prod_{i=1}^{S} \frac{1}{Y_i(z)}, \tag{4.100}$$

$$A_N(z) = (-1)^N A_0(z) \sum_{\Omega \in p_n(S)} \prod_{i \in \Omega} \frac{1}{Y_i(z)}, 1 < N \leq S, \tag{4.101}$$

$$A_1(z) = 1 - A_0(z) \sum_{i=1}^{S} \frac{1}{Y_i(z)}, \tag{4.102}$$

with $p_n(m)$ the collection of sets formed by all possible ways of choosing n distinct elements out of the set $\{1, \ldots, m\}$.

Proof. Proof is trivial by equating the powers of x on both sides of

$$G(1/x, z) = \prod_{i=1}^{S} (1 - xW_i(z)) \tag{4.103}$$

□

Theorem 4.4. *$D_N(z)$ can be expressed explicitly in terms of the $Y_i(z)$ by*

$$\begin{aligned} D_N(z) &= \sum_{j=1}^{S} \big(z - Y_j(z)\big) \prod_{\substack{i=1 \\ i \neq j}}^{S} \frac{1 - Y_i(z)}{Y_j(z) - Y_i(z)} \Big(\frac{A_0(z)}{Y_j(z)}\Big)^N, \quad N > 0, \\ D_0(z) &= z - A_0^*(z), \end{aligned} \tag{4.104}$$

Proof. Proof by induction on N. i) $N > S$:
Substituting (4.104) into Proposition 4.1 leads to

$$\begin{aligned} D_N(z) &= (1 - A_1(z)) \sum_{j=1}^{S} g_j(z) W_j(z)^{N-1} - \sum_{i=2}^{S} A_0(z)^{i-1} A_i(z) \sum_{j=1}^{S} g_j(z) W_j(z)^{N-i} \\ &= \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} \Big(-G(W_j(z), z) + W_j(z)^S\Big) \\ &= \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} W_j(z)^S. \end{aligned}$$

The final transition, based upon Lemma 4.12, asserts that $G(W_j(z), z) = 0$.
ii) $1 < N \leq S$: First, note that, by using $D_0(z) = z - A_0^*(z)$, for these values of N:

$$\begin{aligned} &r_N(z) + A_0(z)^{N-1}\big(A_N(z)(1 - A_0^*(z)) + A_{N+1}^*(z)(1 - z)\big) \\ &= r_{N-1}(z) - A_0(z)^{N-1} A_N^*(z)(z - 1). \end{aligned}$$

Consequently, substituting (4.104) into Proposition 4.1 yields

$$\begin{aligned}
D_N(z) &= (1 - A_1(z)) \sum_{j=1}^{S} g_j(z) W_j(z)^{N-1} - \sum_{i=2}^{N-1} A_0(z)^{i-1} A_i(z) \sum_{j=1}^{S} g_j(z) W_j(z)^{N-i} \\
&\quad - A_0(z)^{N-1} A_N^*(z)(z-1) \\
&= \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} \Big((1 - A_1(z)) W_j(z)^{S-1} - \sum_{i=2}^{N-1} A_0(z)^{i-1} A_i(z) W_j(z)^{S-i} \Big) \\
&\quad - A_0(z)^{N-1} A_N^*(z)(z-1) \\
&= \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} \Big((1 - A_1(z)) W_j(z)^{S-1} - \sum_{i=2}^{N-1} A_0(z)^{i-1} A_i(z) W_j(z)^{S-i} \Big) \\
&\quad - \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} \sum_{i=N}^{S} A_0(z)^{i-1} A_i(z) W_j(z)^{S-i} \\
&= \sum_{j=1}^{S} g_j(z) W_j(z)^{N-S} W_j(z)^{S}.
\end{aligned}$$

The last two transitions are based upon Lemmata 4.13 and 4.12 respectively.
iii) $N = 1$: Straightforward but tedious through substitutions from lemma 4.14. □

Note 64. *Notice that all S roots $Y_i(z)$ appear in (4.104).*

As verification, note that (4.104) corresponds to (4.46) for $S = 2$, as, after tedious substitutions from lemma 4.14, (4.104) can be found to be equal to

$$D_N(z) = -\frac{1 - A_0^*(z)}{A_S(z)} \sum_{j=1}^{S} \frac{z - Y_j(z)}{1 - Y_j(z)} \prod_{\substack{i=1 \\ i \neq j}}^{S} \frac{1}{Y_j(z) - Y_i(z)} \left(\frac{A_0(z)}{Y_j(z)} \right)^N, \quad N > 0. \tag{4.105}$$

Note 65. *Porting over the reasoning used for $S = 2$ to general S, which was manageable for $D_N(Z)$, seems even more tedious for $T_N(z)$. However, a more appealing approach in proving that the kernel also drives the N/∞ model, lies in using formal power series, for which we have a conjecture and are working on a proof.*

4.6 Concluding remarks

We have (numerically) compared the location of the singularities in the ∞/∞ model and the N/∞ model, which indicates that there is "convergence" between both models as N increases, which was to be expected. Furthermore, the tail behavior was also numerically computed in a practical example. However, the main result of this chapter is the discovery of the crucial relation between the characteristic polynomial of the recurrence relation in the finite case and the kernel in the infinite case. We have restricted the arrival process so that no more than 2 class-1 packets arrival per slot. We identify that, in the finite case, all roots of the kernel influence system behavior but *cancel* out each others branch cuts. In the limit to the infinite case, we show that the root inside the unit circle *dominates* the other roots, and the branch cut of this root is no longer canceled. We have relaxed this to maximum S arrivals per slot and proven a similar but partial result. However, these restrictions are probably unnecessary, which is an interesting line of future work. Furthermore, although we have provided some more insight, additional information on the location of the poles and their behavior (the probable convergence to infinite case as the class-1 queue capacity N increases) remains a “holy grail”.

Part II

Partial Buffer Sharing

5
PARTIAL BUFFER SHARING

5.1 Introduction

In this part of the dissertation, we study a single-server queueing model with two traffic classes that share a single finitely-sized buffer. The system can differentiate the service it delivers by giving a class time priority and/or space priority. The class receiving time priority has absolute transmission priority over the other class. Furthermore, space priority is provided by adopting the partial buffer sharing (PBS) acceptance policy. As there are two classes, say class 1 and class 2, there are four possible combinations of the two priority types. However, we only need to consider two as the two others then follow directly by swapping class 1 and 2 around. Therefore, we can choose class 1 to have time priority. The model wherein this class also receives space priority is called Full Priority (FP), whereas the term Mixed Priority (MP) is used when class 2 receives space priority. Arrivals are modeled as a two-class discrete batch Markovian arrival process (2-DBMAP) [62]. A discrete background Markov chain drives this arrival process as the number of arrivals in a slot depend on the state (transitions) of this Markov process. Consequently, one can take the burstiness of network traffic as well as correlation between both classes into account. The two queueing models are studied in a unified manner and solved using matrix analytic methods.

Priority queueing systems providing time priority have been studied extensively in the first part of this dissertation. In these models, each class has its dedicated queue whereas in the current chapter both classes share a single queue hence introducing here space priority. PBS is easily implemented [63] and has been widely studied in, [64, 65, 66]. However, these models do not include time priority as packets of all classes are served in a First-In-First-Out manner. The current contribution encapsulates [19] and extends [67]. The former paper studies networks in the same setting as the first part of this dissertation where packets are categorized into two classes: real-time packets (multimedia, gaming,...) requiring time priority and data packets (file transfer, email,...) requiring space priority and consequently the MP-model is perfect for providing QoS. In contrast, [67] studies the FP-model in order to provide QoS in scalable video coding (SVC) (see e.g. [68]), which uses two types of packets: base layer and enhancement layer packets. The former are required to decode and playback the video, although at poor quality, whereas enhancement packets only increase quality. Here, the FP-model is clearly appropriate. However, it is assumed that, at a slot boundary, all base-layer packets arrive before enhancement packets. The current contribution does not make any assumptions on the order of arrivals and provides unified formulas for both models.

The remainder of this chapter is organized as follows. The queueing model is described in the next section and is subsequently analyzed in section 5.3. The next section determines how to obtain several performance measures and is followed by a section elaborating on intra-slot space priority. Section 5.6 illustrates the obtained results by means of some numerical examples. Finally, this chapter is concluded in section 5.7.

5.2 Model

We consider a discrete-time single-server priority queueing system. Time is divided into fixed-length intervals (slots) and arrivals and departures are synchronized with respect to slot boundaries. There are two classes of packets, say class 1 and 2. Transmission times of packets of both classes are assumed to be fixed and equal to the slot length. Each slot, a packet enters the server for transmission if any packets are present in the queue. In the remainder, we thus distinguish between the queue and the server. The queue capacity of the system under investigation is finite as the queue can only store up to N packets simultaneously.

Actually, we study two queueing models. In both models, time priority is granted to class 1 in the form of absolute transmission priority over class 2. In the first model, class 1 gets time and space priority over class 2 and it is called the FP-model. In contrast, class 2 receives space priority in the MP-model.

Space priority is provided by adopting the PBS buffer acceptance policy with threshold T ($0 \le T \le N$). A packet of the class with space priority can enter the buffer containing less than N packets upon arrival of the packet whereas a packet of the other class is only allowed into a buffer containing no more than T packets upon arrival of the packet. Thereby, packet loss is minimized for the prioritized class. Here, the packets that are present “upon arrival” of a certain packet include the packets that arrived at the same slot boundary but that entered the queue before this packet.

Both the finite queue capacity and the PBS mechanism give rise to packet loss and the order in which packets arrive at a slot boundary determines which of these packets are lost. In the literature, this peculiarity is avoided by assuming that all packets of a certain class arrive concurrently. However, this may not hold in practice. Furthermore, when rearrangement is possible, it often can be exploited to improve performance. We consider a more formal arrival process making no assumptions on the order of arrivals at a slot boundary by using a string representation leading to the notion of intra-slot space priority (ISP), which is discussed in section 5.5.

Note 66. *In the previous part of this dissertation, each class had a dedicated queue making the order of packet arrivals within a slot irrelevant.*

The arrival sequence at a slot boundary is embodied by a vector $\boldsymbol{x}$ with i-th element $x_i \in \{1,2\}$ denoting the class of the ith packet. The total number of arrivals obviously equals the total number of elements of $\boldsymbol{x}$, given by $\dim(\boldsymbol{x})$. For instance, a class-2 arrival followed by a class-1 arrival and another class-2 arrival is depicted by the vector $\boldsymbol{x} = [2\ 1\ 2]$ whereas a slot with no arrivals corresponds to $\boldsymbol{x} = [\,]$. For each $n \in \mathbb{N}$, there are 2^n vectors representing a possible arrival sequence. Let the set of all vectors representing an arrival sequence be denoted by Ω. The arrival process is then specified by defining appropriate probability measures on Ω.

Note 67. *The length of such sequences is, in general, unbounded. Furthermore, considering only a finite number of elements of a sequence is not sufficient as an infinite amount of packets without space priority can be dropped due to the threshold while the following packets with space priority are accepted. However, assigning probability to such sequences is straightforward as one can easily construct an equivalent finite representation where each element tracks the number of consecutive arrivals of the same class (similar to "run length encoding"). Due to the buffer finiteness, at most $N+1$ "transitions" between consecutive arrivals of the same class need to be considered to determine which packets enter the buffer.*

Let the vector $\boldsymbol{a}_k$ represent the arrival sequence at the kth slot boundary. In this chapter, class-1 and class-2 arrivals are modelled by means of a 2-class discrete-time batch Markovian arrival process (2-DBMAP). As we need to keep track of the entire sequence of arrivals, the definition of this process is more general than the standard one [69]. In the current contribution, a 2-DBMAP is completely characterized by the $Q \times Q$ matrices $\boldsymbol{A}(\boldsymbol{x})$ governing the transitions from slot to slot of the underlying discrete-time Markov chain when arrivals occur according to the sequence $\boldsymbol{x} \in \Omega$. Here, Q denotes the size of the state space of the underlying chain. We have

$$\boldsymbol{A}(\boldsymbol{x}) = \Big[\Pr[\boldsymbol{a}_k = \boldsymbol{x}, s_{k+1} = j | s_k = i]\Big]_{i,j=1,\ldots,Q}, \tag{5.1}$$

with s_k the state of the underlying Markov chain during slot k. As $\boldsymbol{A}(\cdot)$ is to be a proper (probability) measure, for any set $\Phi \subseteq \Omega$ we have

$$\boldsymbol{A}(\Phi) = \sum_{\boldsymbol{x}\in\Phi} \boldsymbol{A}(\boldsymbol{x})\,. \tag{5.2}$$

The number of class-m ($m = 1,2$) packets amongst the first n packets in an arrival sequence $\boldsymbol{x}$ is given by

$$c_m^n(\boldsymbol{x}) = \sum_{i=1}^{min(n,dim(\boldsymbol{x}))} 1\{x_i = m\}, \quad . \tag{5.3}$$

Note 68. *Recall that* $1\{.\}$ *is the indicator function, evaluating to* 1 *if its argument is true and to* 0 *if it is false. Also, the size (dimension) of a vector* $\boldsymbol{a}$ *is denoted by dim*$(\boldsymbol{a})$.

Obviously, the dimension of x is an upper bound on $c_m^n(\boldsymbol{x})$. The total number of arrivals at the kth boundary, $a_{T,k}$, equals $\dim(\boldsymbol{a}_k)$. Also note that the number of class-i packets arriving at the kth boundary, $a_{i,k}$, is easily found to be equal to $c_i^\infty(\boldsymbol{a}_k)$.

For further use, let λ_i denote the mean number of packets of class i ($i = 1,2$) that arrive at a slot boundary and be defined as

$$\lambda_1 = \sum_{\boldsymbol{x}\in\Omega} c_1^\infty(\boldsymbol{x})\boldsymbol{\psi}\boldsymbol{A}(\boldsymbol{x})\boldsymbol{e}\,, \quad \lambda_2 = \sum_{\boldsymbol{x}\in\Omega} c_2^\infty(\boldsymbol{x})\boldsymbol{\psi}\boldsymbol{A}(\boldsymbol{x})\boldsymbol{e}\,. \tag{5.4}$$

Here $\boldsymbol{e}$ is a column vector of ones and $\boldsymbol{\psi}$ is the steady-state probability row vector of the underlying Markov chain, i.e., it is the unique non-negative solution of

$$\boldsymbol{\psi} = \boldsymbol{\psi}\boldsymbol{A}(\Omega)\,, \quad \boldsymbol{\psi}\boldsymbol{e} = 1\,. \tag{5.5}$$

Furthermore, let $\lambda = \lambda_1 + \lambda_2$ denote the total load.

Due to the possible simultaneity of arrivals of both classes and departures at slot boundaries, one needs to specify the order in which these arrivals and departures are processed at a boundary. We here assume that the departure, if any, occurs before any arrivals. In the remainder, observation of the queue “at slot boundaries" means after possible departures but before arrivals.

5.3 System analysis

We first relate the total number of packets and the number of class-2 packets in the queue at consecutive slot boundaries. These relations contain the notion of effective arrivals and these are subsequently derived in the second subsection. Finally, a set of balance equations can be established and solved numerically.

5.3.1 System equations

Consider slot boundary k and let u_k and v_k denote the total queue content and the class-2 queue content — i.e., the total number of packets and the number of class-2 packets in the queue — at this slot boundary. Possibly, some arriving packets are not accepted into the queue giving rise to packet loss. Therefore, let $\tilde{a}_{1,k}$ and $\tilde{a}_{2,k}$ denote the number of class-1 and class-2 packets arriving at the kth slot boundary that the system accommodates, called effective arrivals.

The system equations relate the total queue content and the class-2 queue content at consecutive slot boundaries. As a packet leaves the queue at the $(k+1)$th boundary if there are any packets present, the total queue content evolves according to

$$u_{k+1} = (u_k + \tilde{a}_{1,k} + \tilde{a}_{2,k} - 1)^+ . \tag{5.6}$$

Note 69. *Recall that* $(\cdot)^+$ *is the usual shorthand notation for* $\max(\cdot, 0)$.

The evolution of the class-2 queue content is more intricate. If a class-1 packet enters the server at the $(k+1)$st slot boundary, this is if $u_k - v_k + \tilde{a}_{1,k} > 0$, class-2 packets obviously have no access to the server yielding

$$v_{k+1} = v_k + \tilde{a}_{2,k} . \tag{5.7}$$

On the other hand, if there are no class-1 packets present, this is if $u_k - v_k + \tilde{a}_{1,k} = 0$, a class-2 packet enters the server, if any is present. This produces

$$v_{k+1} = (v_k + \tilde{a}_{2,k} - 1)^+ . \tag{5.8}$$

Note 70. *Here, we observe the queue at slot boundaries, whereas in the first part of this dissertation, the queue was observed at the beginning of a slot. This is clearly visible in the system equations as here, the arrivals are also contained in the* $(\cdot)^+$ *operator. In the next section, the queue content at (the beginning of) a random slot will be computed from the one at slot boundaries. Furthermore, the terms queue content and the system content are equal in the current context. These discrepancies between both parts of the dissertation was chosen to keep each part in sync with the relevant literature in that area.*

5.3.2 Effective arrivals

Before constructing the balance equations from the system equations, we introduce some auxiliary functions which will allow us to describe both models in unified formulas. The number of effective arrivals when the queue content equals n and packets arrive according to the vector $\boldsymbol{x}$ are given by

$$\begin{aligned} \tilde{a}_1^n(\boldsymbol{x}) &= \begin{cases} \min(c_1^\infty(\boldsymbol{x}), N-n-\tilde{a}_2^n(\boldsymbol{x})), & \text{Full Priority} \\ c_1^{T-n}(\boldsymbol{x}), & \text{Mixed Priority} \end{cases} \\ \tilde{a}_2^n(\boldsymbol{x}) &= \begin{cases} c_2^{T-n}(\boldsymbol{x}), & \text{Full Priority} \\ \min(c_2^\infty(\boldsymbol{x}), N-n-\tilde{a}_1^n(\boldsymbol{x})), & \text{Mixed Priority} \end{cases} \end{aligned} . \tag{5.9}$$

Consequently, $\tilde{a}_{i,k} = \tilde{a}_i^{u_k}(\boldsymbol{a}_k)$. Note that the queue accommodates arriving packets of the class receiving space priority until there are N packets in the queue and packets of the other class until there are T packets in the queue. Especially note that the number of effective arrivals of the class without space priority is obtained first as it appears in the expression for that of the other class. This stems from the fact that, obviously, the threshold is reached before the entire buffer is full (or concurrently if $T = N$).

The maximum number of effective class-i arrivals in a slot given the queue content equals n is denoted by $\tilde{a}_i^{\max}$(n) yielding

$$\begin{aligned} \tilde{a}_1^{\max}(n) &= \begin{cases} N-n\,, & \text{Full Priority} \\ (T-n)^+\,, & \text{Mixed Priority} \end{cases} \\ \tilde{a}_2^{\max}(n) &= \begin{cases} (T-n)^+\,, & \text{Full Priority} \\ N-n\,, & \text{Mixed Priority} \end{cases} \end{aligned} . \tag{5.10}$$

Notice that, evidently, these functions exactly oppose each other for both models. Next, the class-2 queue content given that the total queue content equals n ranges from $v_{\min}(n)$ to $v^{\max}(n)$ with

$$\begin{aligned} v_{\min}(n) &= \begin{cases} 0\,, & \text{Full Priority} \\ (n+1-T)^+\,, & \text{Mixed Priority} \end{cases} \\ v^{\max}(n) &= \begin{cases} \min(T,n)\,, & \text{Full Priority} \\ n\,, & \text{Mixed Priority} \end{cases} \end{aligned} . \tag{5.11}$$

Especially note that, in the FP-model, there are at most T class-2 packets present in the buffer, whereas in the MP-model, there must be class-2 packets present if the total content exceeds $T-1$ as the buffer can only contain up to $T-1$ class-1 packets immediately following a departure. Furthermore, in the MP-model, $T = 0$ is an exceptional case as then $v_{\min}(n)$ should equal n and not $n+1$. Here, the system behaves as a FIFO queue with a single class of (class-2) packets as all class-1 packets are dropped.

Let $\tilde{\boldsymbol{A}}_u(m,n)$ denote the matrix governing the transitions of the underlying Markov chain at a slot boundary when there are m effective class-1 arrivals and n effective class-2 arrivals, given that there are u packets in the queue at that slot boundary. That is

$$\begin{aligned} \tilde{\boldsymbol{A}}_u(m,n) &= \Big[\Pr[\tilde{a}_{1,k} = m, \tilde{a}_{2,k} = n, s_{k+1} = j | s_k = i, u_k = u]\Big]_{i,j=1,\dots,Q} \\ &= \sum_{x\in\Omega} \boldsymbol{A}(\boldsymbol{x}) 1\{m = \tilde{a}_1^{u_k}(\boldsymbol{x}),\ n = \tilde{a}_2^{u_k}(\boldsymbol{x})\}\,, \end{aligned} \tag{5.12}$$

for $u = 0,\dots,N-1$, $m \geq 0$ and $n \geq 0$.

5.3.3 Balance equations

Clearly, the triple (u_k, v_k, s_k) describes the state of the queueing system at the kth slot boundary in the Markovian sense. Therefore, let $\boldsymbol{\pi}_k(m,n)$ denote the row vector whose ith entry is the probability to have $n-m$ class-1 and m class-2 packets in the queue at the kth slot boundary while the arrival process is in state i, i.e.,

$$\boldsymbol{\pi}_k(m,n) = \Big[\Pr[v_k = m, u_k = n, s_k = i]\Big]_{i=1,\dots,Q}\,, \tag{5.13}$$

for $n = 0,\dots,N-1$ and $m = v_{\min}(n),\dots,v^{\max}(n)$. In view of (5.6), (5.7) and (5.8), relating slots k and $k+1$ and conditioning on the state of the server yields

$$\begin{aligned}\boldsymbol{\pi}_{k+1}(m,n) = &\sum_{j=0}^{\min(n+1,N-1)} \sum_{i=v_{\min}(j)}^{v^{\max}(j)} \boldsymbol{\pi}_k(i,j)\tilde{\boldsymbol{A}}_j(n-m+i+1-j,m-i)\\ &+1\{n=m\}\sum_{j=0}^{\min(n+1,N-1)} \boldsymbol{\pi}_k(j,j)\tilde{\boldsymbol{A}}_j(0,n-j+1)\\ &+1\{n=m=0\}\boldsymbol{\pi}_k(0,0)\tilde{\boldsymbol{A}}_0(0,0)\,,\end{aligned} \tag{5.14}$$

for $n = 0,\dots,N-1$ and $m = v_{\min}(n),\dots,v^{\max}(n)$.

Grouping the vectors $\boldsymbol{\pi}_k(m,n)$ by total total queue content defines the row vectors

$$\boldsymbol{\pi}_k(n) = [\boldsymbol{\pi}_k(v_{\min}(n),n),\dots,\boldsymbol{\pi}_k(v^{\max}(n),n)]\,, \tag{5.15}$$

for $n = 0,\dots,N-1$. The set of equations (5.14) then has block matrix representation

$$\boldsymbol{\pi}_{k+1}(n) = \sum_{j=0}^{\min(n+1,N-1)} \boldsymbol{\pi}_k(j)\boldsymbol{C}(j,n)\,, \tag{5.16}$$

where the block elements (of size $Q \times Q$) of $\boldsymbol{C}(j,n)$ are given by

$$\begin{aligned}c_{i+1,m+1}(j,n) = &\tilde{\boldsymbol{A}}_j(n+1-j-m+i,m-i)\\ &+1\{m=n,\, i=j\}\tilde{\boldsymbol{A}}_j(0,n-j+1)\\ &+1\{n=j=0\}\tilde{\boldsymbol{A}}_0(0,0)\,,\end{aligned} \tag{5.17}$$

for $i = v_{\min}(j),\dots,v^{\max}(j)$ and $m = v_{\min}(n),\dots,v^{\max}(n)$. Note that $c_{i+1,m+1}(j,n)$ corresponds to the evolution of $\boldsymbol{\pi}_k(i,j)$ to $\boldsymbol{\pi}_{k+1}(m,n)$.

Under mild assumptions, the Markov chain under consideration has only one ergodic class. Consequently, there exists a unique stationary distribution (a non-negative normalized vector), denoted by the (block) vector $\boldsymbol{\pi} = [\boldsymbol{\pi}(0),\dots,\boldsymbol{\pi}(N-1)]$, with $\boldsymbol{\pi}(n) = [\boldsymbol{\pi}(v_{\min}(n),n),\dots,\boldsymbol{\pi}(v^{\max}(n),n)]$, satisfying the balance equations

$$\boldsymbol{\pi}(n) = \sum_{j=0}^{\min(n+1,N-1)} \boldsymbol{\pi}(j)\boldsymbol{C}(j,n)\,, \tag{5.18}$$

for $n = 0,\dots,N-1$. Consequently, the transition matrix of the priority queueing system under consideration has an upper-Hessenberg block-structure with varying block sizes which is efficiently solved by means of a linear level reduction algorithm [70, 62, 71]. In the block matrix, the level (block-row number) indicates the total queue content while the phase (size of a block element) indicates the class-2 queue content and the state of the arrival process. In general, the number of phases equals $(v^{\max}(n) - v_{\min}(n)) \times Q$ at level n. Consequently, for $n \le T$, the number of phases equals $(n+1) \times Q$ as, out of n packets in total, from 0 up to n packets can be of class 2. For levels $n > T$, the block size remains constant at $(T+1) \times Q$ and $T \times Q$ for the FP- and MP-model respectively as the class-2 queue content can vary from 0 to T and from $n+1-T$ to n respectively. Figure 5.1 demonstrates the block structure of the FP-model for a small example ($N = 6$, $T = 3$).

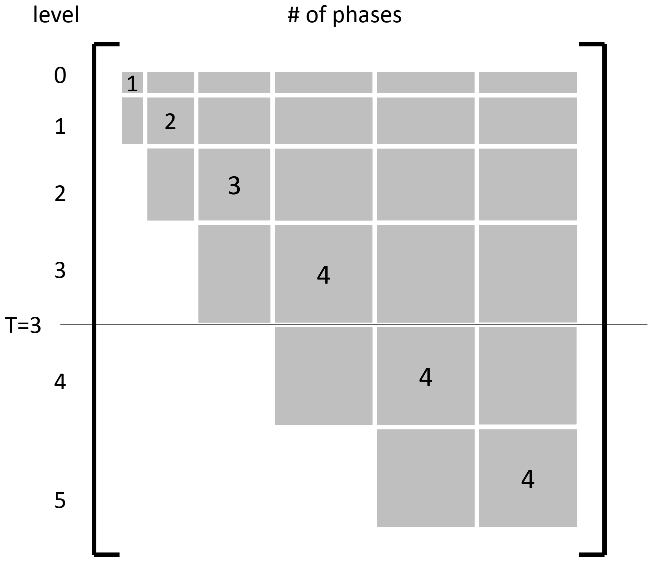

Figure 5.1: Transition matrix block structure for $N = 6$, $T = 3$.

5.4 Performance analysis

Once $\boldsymbol{\pi}(n)$ has been obtained, various performance measures can be derived. This section describes how to calculate supported load, packet loss, queue content at a random slot boundary and system content at (the beginning of a random slot) and mean packet waiting time.

The supported class-i load $\tilde{\lambda}_i$ is defined as the average number of class-i packets arriving at a slot boundary that are accommodated by the queue. They are determined by

$$\tilde{\lambda}_1 = \sum_{i=0}^{N-1} \sum_{m=0}^{\tilde{a}_1^{\max}} \sum_{n=0}^{\tilde{a}_2^{\max}} m\boldsymbol{\pi}(i)\boldsymbol{e}\tilde{\boldsymbol{A}}_i(m,n)\boldsymbol{e}\,, \quad \tilde{\lambda}_2 = \sum_{i=0}^{N-1} \sum_{m=0}^{\tilde{a}_1^{\max}} \sum_{n=0}^{\tilde{a}_2^{\max}} n\boldsymbol{\pi}(i)\boldsymbol{e}\tilde{\boldsymbol{A}}_i(m,n)\boldsymbol{e}. \tag{5.19}$$

Note that $\boldsymbol{\pi}(i)\boldsymbol{e}$ is a row vector of size Q with ith element denoting the probability that the queue contains i packets in total and that the underlying chain of the arrival process is in state j, $(1 \le j \le Q)$. Furthermore, the total supported load is given by $\tilde{\lambda} = \tilde{\lambda}_1 + \tilde{\lambda}_2$.

Alternatively, the supported load can also be retrieved by observing the departure process. As the system is stationary, the total supported load has to equal the probability that a packet leaves the queue at a random slot boundary. As a packet departs at each slot boundary except when the queue is empty, this produces

$$\tilde{\lambda} = 1 - \boldsymbol{\pi}(0,0)\tilde{\boldsymbol{A}}_0(0,0)\boldsymbol{e}. \tag{5.20}$$

Similarly, the class-1 supported load equals the probability of a class-1 departure at a random slot boundary. A class-1 packet leaves the queue if there are class-1 packets present in the queue. Thus

$$\tilde{\lambda}_1 = 1 - \sum_{m=0}^{v^{\max}(N-1)} \sum_{n=0}^{\tilde{a}_2^{\max}(m)} \boldsymbol{\pi}(m,m)\tilde{\boldsymbol{A}}_0(0,n)\boldsymbol{e}. \tag{5.21}$$

Note that the appearance of $v^{\max}(N-1)$ indicates that, at a slot boundary, the system can contain up to T and up to $N-1$ class-2 packets for the FP- and MP-model respectively. Also, the class-2 supported load is easily determined as $\tilde{\lambda}_2 = \tilde{\lambda} - \tilde{\lambda}_1$.

The packet loss ratio is the fraction of packets that cannot be accommodated by the queue. In view of the definitions of supported load and packet loss ratio, one easily derives the packet loss ratio of class-1 packets (plr_1), of class-2 packets (plr_2) and of all packets (plr) to be

$$\text{plr}_1 = 1 - \frac{\tilde{\lambda}_1}{\lambda_1}, \quad \text{plr}_2 = 1 - \frac{\tilde{\lambda}_2}{\lambda_2}, \quad \text{plr} = 1 - \frac{\tilde{\lambda}}{\lambda}. \tag{5.22}$$

Let u_1 and u_2 denote the class-1 and class-2 queue content at a random slot boundary. Since $\boldsymbol{\pi}(m,n)$ is the joint distribution of the queue content of both classes, all moments (mean, variance, etc.) of the random variables u_1 and u_2 are easily obtained. For instance, the i-th moment of the class-j queue content at random slot boundaries $\overline{u}_j^{(i)}$ is given by

$$\overline{u}_1^{(i)} = \sum_{n=0}^{N-1} \sum_{m=v_{\min}(n)}^{v^{\max}(n)} (n-m)^i \boldsymbol{\pi}(m,n)\boldsymbol{e}, \quad \overline{u}_2^{(i)} = \sum_{n=0}^{N-1} \sum_{m=v_{\min}(n)}^{v^{\max}(n)} m^i \boldsymbol{\pi}(m,n)\boldsymbol{e}. \tag{5.23}$$

The mean total queue content is given by $\overline{u}^{(1)} = \overline{u}_1^{(1)} + \overline{u}_2^{(1)}$. Similar expressions can be established for joint moments. For instance, the covariance between the queue content of both classes at a random slot boundary is given by

$$\text{Cov}(u_1,u_2) = \sum_{n=0}^{N-1} \sum_{m=v_{\min}(n)}^{v^{\max}(n)} (n-m)m\boldsymbol{\pi}(m,n)\boldsymbol{e} - \overline{u}_1^{(1)}\overline{u}_2^{(1)}. \tag{5.24}$$

Note 71. *Here, the i-th moment of a random variable a is denoted by $\overline{a}^{(i)}$.*

When the system is observed at (the beginning of) a random slot (or equivalently at random points in time), this is after all departures and arrivals occurred at the preceding slot boundary, let $\theta(m,n)$ denote the probability that it contains $n-m$ class-1 and m class-2 packets. These packets either were already present at the preceding slot boundary or have arrived at that slot boundary. Consequently, the queue content at random slots is easily obtained from the one at random slot boundaries yielding

$$\theta(m,n) = \sum_{i=0}^{m} \sum_{j=0}^{n} \boldsymbol{\pi}(i,j)\tilde{\boldsymbol{A}}_j(n-m-j+i, m-i)\boldsymbol{e}, \tag{5.25}$$

for $n = 0,\ldots,N$ and for $m = v_{\min}(n),\ldots,v^{\max}(n)$. Notice that the queue can now contain up to N packets as we no longer observe the system immediately following a departure. Again the i-th moment of the class-j queue content at random points in time $\overline{y}_i$ is given by

$$\overline{y}_1^{(i)} = \sum_{n=0}^{N} \sum_{m=v_{\min}(n)}^{v^{\max}(n)} (n-m)^i \theta(m,n)\boldsymbol{e}, \quad \overline{y}_2^{(i)} = \sum_{n=0}^{N} \sum_{m=v_{\min}(n)}^{v^{\max}(n)} m^i \theta(m,n)\boldsymbol{e}. \tag{5.26}$$

Alternatively, $y_i^{(1)}$ can also be obtained by noting that there are $\tilde{\lambda}_i$ class-i arrivals at a slot boundary on average yielding

$$\overline{y}_i^{(1)} = \overline{u}_i^{(1)} + \tilde{\lambda}_i. \tag{5.27}$$

Consequently, calculating $\theta(m,n)$ is superfluous when one is only interested in the mean values $y_i^{(1)}$.

Packet waiting time is defined as the number of slots a packet spends in the queueing system. Applying Little's law, the mean class-i $(i = 1,2)$ waiting time is found as

$$\overline{w}_i^{(1)} = \frac{1}{\tilde{\lambda}_i}\overline{y}_i^{(1)} = \frac{1}{\tilde{\lambda}_i}\overline{u}_i^{(1)} + 1. \tag{5.28}$$

Notice that here Little's result does not relate the mean waiting time to the mean queue content at random slot boundaries but to the mean queue content at the beginning of random slots. This is caused by the chosen order of arrival, observation and departure epochs in our queueing model as illustrated in [3].

5.5 Intra-slot space priority

The order in which packets arrive at a slot boundary can be seen as means of providing intra-slot space priority (ISP), as it partially determines which of these packets, if any, are dropped. Obviously, ISP will have a larger effect when a large number of packets arrive at a slot boundary. The literature generally assumes that all class-1/class-2 packets arrive before packets of the other class (class-1/2 ISP). In some applications, reordering the arrivals at a slot boundary is feasible. This can consequently be exploited to improve performance. For instance, as the FP-model provides time- and space priority to class-1 packets, it is beneficial to use class-1 ISP as well. In contrast, the MP-model gives space priority to class-2 packets so it seems natural to give these packets ISP as well. Furthermore, in a lot of real-life applications rearranging is infeasible and packets often arrive in a completely random order (no ISP).

Theoretically, ISP is achieved by only allowing certain forms of arrival sequences $\boldsymbol{x} \in \Omega$ to correspond with non-zero entries in the matrix $\boldsymbol{A}(\boldsymbol{x})$. Let us call the set of vectors of this form Ψ, making this formally equal to $\boldsymbol{A}(\Psi) = \boldsymbol{A}(\Omega)$. When Ψ contains a reasonably small number of vectors, determining $A_u(m,n)$ is straightforward by combining (5.3), (5.9) and (5.12). However, this becomes increasingly tedious as Ψ contains more elements. This can be avoided by giving up some generality on the order of arrivals. Here, information about the order of arrivals is removed from the arrival process but assumed to be generally known. This enables writing the arrival process as a standard 2-DBMAP [69] given by

$$\boldsymbol{A}(m,n) = \Big[\Pr[a_{1,k} = m, a_{2,k} = n, s_{k+1} = j | s_k = i]\Big]_{i,j=1..Q}, \tag{5.29}$$

that only keeps track of the number of arrivals of each class at a slot boundary. Several cases where the order of arrivals can be assumed to be generally known were mentioned above: class-1/2 ISP and no ISP. The remainder of this section will elaborate on this matter.

5.5.1 Class-1 intra-slot space priority.

Here, all class-1 packets are assumed to arrive before class-2 packets. Consequently, the set Ψ fulfilling $\boldsymbol{A}(\Psi) = \boldsymbol{A}(\Omega)$ is the set of all arrival sequences $\boldsymbol{x}$ of the form

$$\boldsymbol{x} = [\underbrace{1\dots1}_{m}\underbrace{2\dots2}_{n}], \tag{5.30}$$

for $m, n \geq 0$, representing a slot boundary with m class-1 and n class-2 arrivals. If class-1 ISP is assumed, the only information held by such a vector are the values of m and n. Consequently, (5.12) simplifies to

$$\tilde{\boldsymbol{A}}_u(m,n) = \sum_{i=m}^{\infty}\sum_{j=n}^{\infty} \boldsymbol{A}(i,j)1\{m = \min(i, \tilde{a}_1^{\max}(u)), n = \min(j, \tilde{a}_2^{\max}(u+m))\}. \tag{5.31}$$

5.5.2 Class-2 intra-slot space priority.

In this case, each non-zero probability arrival sequence $x \in \Psi$ is of the form

$$\boldsymbol{x} = [\underbrace{2\dots2}_{m}\underbrace{1\dots1}_{n}]. \tag{5.32}$$

Again, $\boldsymbol{A}(\Psi) = \boldsymbol{A}(\Omega)$ and only m and n need to be accounted for and thus (5.29) holds again . Here, (5.12) simplifies to

$$\tilde{\boldsymbol{A}}_u(m,n) = \sum_{i=m}^{\infty}\sum_{j=n}^{\infty} A(i,j)1\{m = \min(i, \tilde{a}_1^{\max}(u+n)), n = \min(j, \tilde{a}_2^{\max}(u))\}. \tag{5.33}$$

5.5.3 No intra-slot space priority.

This situation is more intricate. When i class-1 and j class-2 packets arrive, these $i + j$ packets are assumed to have a completely random order. We have

Full Priority:

$$\begin{aligned}\tilde{\boldsymbol{A}}_u(m,n) = \sum_{i=m}^{\infty}\sum_{j=n}^{\infty} \boldsymbol{A}(i,j)\Bigg(&1\{i+j < (T-u)^+)\}1\{m = i, n = j\}\\ &+ 1\{i+j \geq (T-u)^+)\}\frac{\binom{(T-u)^+}{n}\binom{(i+j-(T-u)^+)^+}{j-n}}{\binom{i+j}{i}}1\{m = \min(i, N-u-n)\}\Bigg)\end{aligned} \tag{5.34}$$

Mixed Priority:

$$\begin{aligned}\tilde{\boldsymbol{A}}_u(m,n) = \sum_{i=m}^{\infty}\sum_{j=n}^{\infty} \boldsymbol{A}(i,j)\Bigg(& 1\{i+j<(T-u)^+\}1\{m=i,n=j\} \\ & + 1\{i+j \geq (T-u)^+\}\frac{\binom{(T-u)^+}{m}\binom{(i+j-(T-u)^+)^+}{i-m}}{\binom{i+j}{i}} 1\{n=\min(j,N-u-m)\}\Bigg) \end{aligned} \qquad (5.35)$$

with $\binom{n}{k} = n!/(k!(n-k)!)$ denoting the binomial coefficient. In this case, a unified formula for both models (FP and MP) cannot be established as the class receiving space priority governs this equation. This can be seen as follows: when i class-1 and j class-2 packets arrive at a slot boundary, choosing i (out of $i+j$) positions for class-1 completely determines the arrival vector. The queue can accommodate $(T-u)^+$ packets until the threshold is reached and packets of the class without priority are no longer accepted. Consequently, in order to accept $m\,(n)$ of these packets, they have to be among the first $(T-u)^+$ arriving packets. The remaining $i-m\,(j-n)$ non-prioritized packets are lost, but all possible combinations among these vectors evidently have to be taken into consideration as well. Once the number of unprioritized effective arrivals is known, it is straightforward that prioritized packets are accepted as long as the queue is not entirely full.

5.6 Numerical examples

In this section, we investigate the impact of time priority, PBS and ISP on the performance measures of both classes in both the FP- and MP-model. Obviously, the impact of ISP increases as multiple packets arrive at the same slot boundary while it has no impact when only a single packet arrives. Therefore, a bursty arrival process where multiple packets arrive at the same slot boundary is considered in this section. Furthermore, ISP only has effect in slots where the threshold is crossed. If the threshold is not reached, all arriving packets are accepted, whereas, if the queue content already exceeds the threshold, only packets with space priority may enter the queue. Consequently, one would expect ISP to have a minor impact but in the following we demonstrate that ISP can considerably influence system performance. Furthermore, time priority and PBS have a large impact as expected.

We now study the queueing system described in this chapter when packets arrive according to the arrival process generated by M on/off sources as described in appendix A.5. The legends use following 3 character notation. The first character denotes the model: F for FP and M for MP, the second denotes the ISP: 1 and 2 for class-1 and class-2 ISP respectively and r for random (no ISP). This is followed by a hyphen and the class number (1 or 2). For instance, when the load is depicted for Fr-2 it denotes the class-2 load for the FP-model with no ISP. Each figure has two graphs. The left one depicts the results for the FP-model and the right one for the MP-model. Obviously, the results for no ISP will always lie between the values for class-1 and class-2-ISP. Furthermore, in order to make the graphs clearer, curves

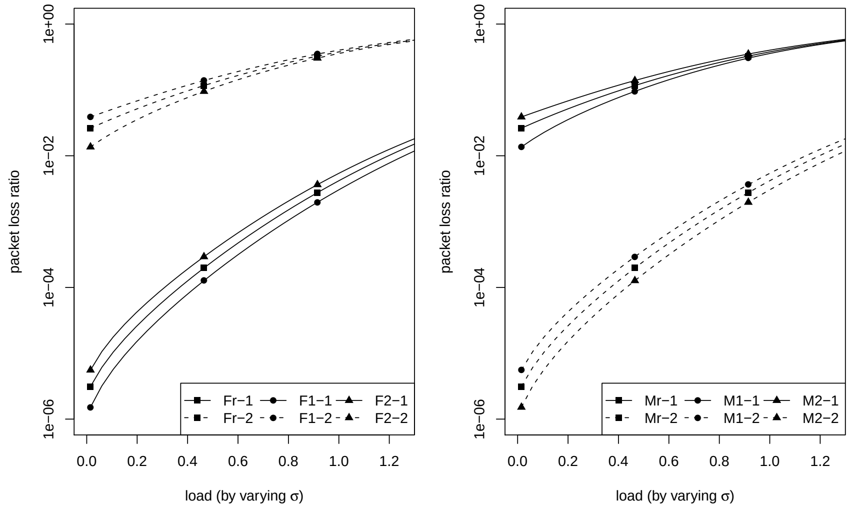

Figure 5.2: Loss vs. load with 3 ISP types for the FP-model (on the left) and the MP-model (on the right).

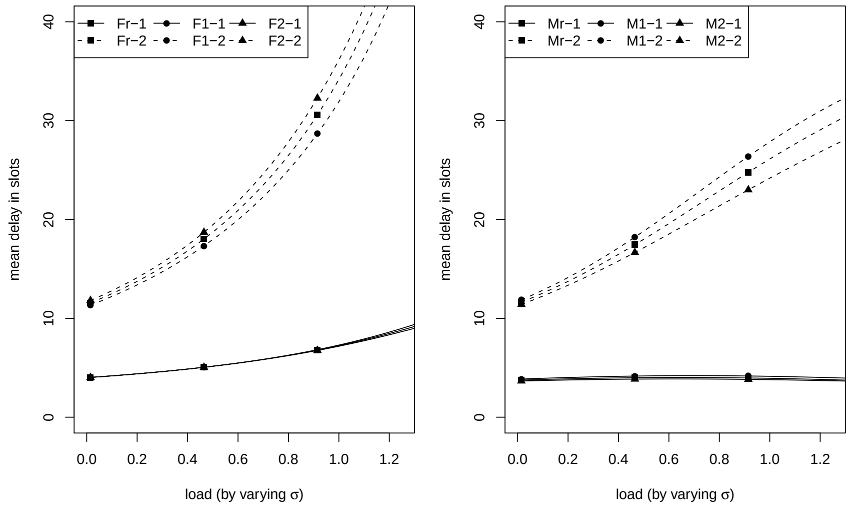

Figure 5.3: Delay vs. threshold with 3 ISP types for the FP-model (on the left) and the MP-model (on the right).

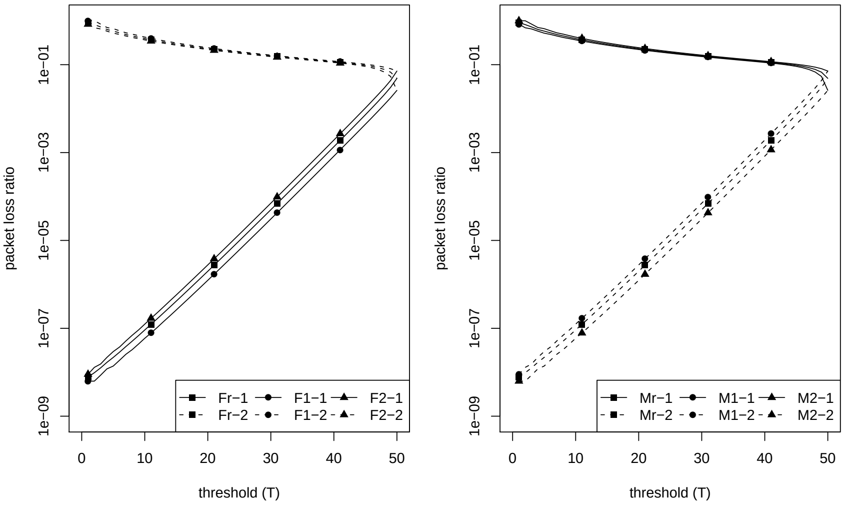

Figure 5.4: Packet loss ratio vs. threshold with 3 ISP types for the FP-model (on the left) and the MP-model (on the right).

are full for class-1 and dashed for class-2 and each type of ISP has a symbol: a circle for class-1 ISP, a triangle for class-2 ISP and a square for no ISP.

First, consider a buffer that can hold $N = 50$ packets and has threshold $T = 25$. Packets are generated by $M = 2$ sources with $K = 1.5$ and when a source is on it generates 4 packets of each class. The fraction of time a source is on σ is varied causing the load λ to vary from 0 to 1.3 (note that the system is finite and thus always stable). We investigate the impact hereof on the packet loss ratio in figure 5.2 and on the mean delay in figure 5.3.

We first study the packet loss ratio. Obviously, it increases when the load increases. The QoS differentiation provided by the model is immediately apparent. The loss is much lower for the class receiving space priority (class-1 on the left and class-2 on the right). Furthermore, the effect of ISP is easily observed as loss is up to three times higher (for light loads) between the different ISP types. For the class without space priority, all packets are discarded once the threshold T is exceeded and thus the ISP only plays a role in the slots where T is crossed. As the load increases the queue content surmounts the threshold more frequently and the packet loss becomes less dependent on the type of ISP and the three lines converge. Also note that, as time priority does not influence packet loss ratio, but only the order in which packets are served, the results are symmetric for the FP- and MP-model (swapping classes and ISP).

The mean delay of class 1 is lower than that of class 2 for both models as time priority is always provided to class 1. ISP affects mean class-2 delay considerably (5-20% difference) whereas class-1 packets are hardly influenced. This can be seen by noting that class-1 packets are not affected by other packets arriving while they

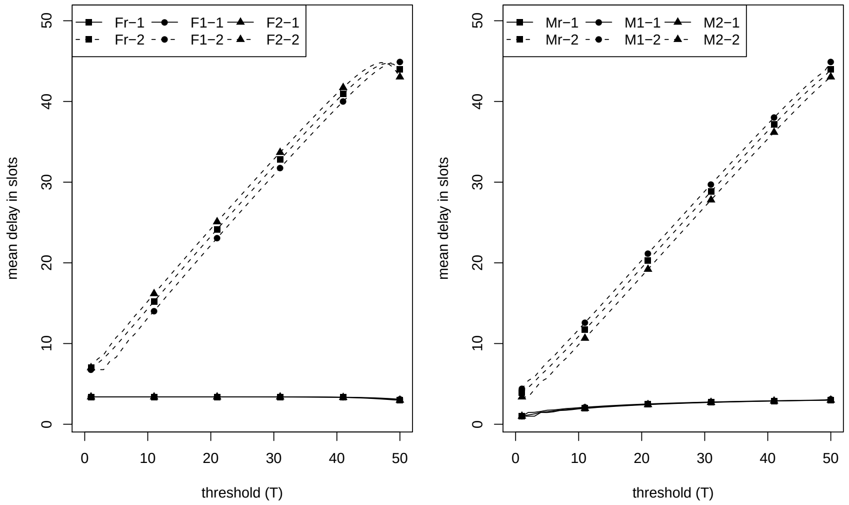

Figure 5.5: Delay vs. threshold with 3 ISP types for the FP-model (on the left) and the MP-model (on the right).

wait in the queue whereas class-2 packets have to give priority to any arriving class-1 packets and are consequently more reactive to packet drops and thus also to different ISP. For the FP-model, the mean class-1 delay increases with the load as more and more class-1 packets are allowed into the system. The mean class-2 delay increases as more and more packets enter the system. Note that the ISP resulting in the highest packet loss ratio also yields the lowest delay as more and more packets are dropped. In contrast, for the MP-model, the mean class-1 delay first increases slightly when the load increases and then starts decreasing as more and more class-1 packets are dropped as they do not have space priority and consequently the packets that do get accepted have shorter delay. Furthermore, both the mean class-1 and class-2 delay are lower than for the FP-model because, opposed to that model, the MP-model drops more class-1 packets than class-2 packets and the former have an impact on the delay of both classes whereas the latter only have an impact on the delay of other class-2 packets. This also explains why the ISP resulting in the highest class-2 packet loss ratio also yields the highest class-2 delay for the MP-model.

Next, we will investigate the effect of the threshold (T) as it controls how the available space (N) is distributed between both classes. Consider, $N = 50$, $M = 2$, $\sigma = 0.12$, $K = 1.5$ and $b_1 = b_2 = 2$ yielding a load $\lambda = 0.96$. We let T vary from 0 to N and again depict the packet loss ratio (figure 5.4) and the mean delay (figure 5.5).

For $T = 0$, the system behaves as a system with only one traffic class (those with space priority). The differentiation in packet loss ratio between both classes decreases as T increases as more and more packets are allowed into the system (packets of both classes can utilize the spaces up to T). For $T = N$ there is no space priority and thus no difference between both classes. Furthermore, as explained for figure 5.2, ISP has only a limited effect on class-1 packet loss for high load (recall that $\lambda = 0.96$) whereas its impact on class-2 is bigger. Obviously, the ISP equivalent to the class receiving space priority corresponds to the smallest amount of packet loss. Again, it is apparent that both models are symmetric concerning packet loss.

The mean class-1 delay is hardly affected by varying the threshold for the FP-model and for larger N it even decreases slightly as the system even starts to drop space prioritized (class-1) packets resulting in a shorter delay for packets of this class that are accepted. This also explains the decrease in class-2 delay when T approaches N. Furthermore, class-2 delay increases as the threshold increases as more and more class-2 packets are allowed into the system causing a longer delay for other packets of this class (recall that they do not affect the delay of class-1 packets). For the MP-model, when the threshold increases, more class-1 packets are allowed into the system at the cost of class-2 packets. But, as stated before, class-1 packets affect the mean delay of both classes which thus get longer as T increases. Concerning ISP, similar arguments as above lead to the same conclusions. It is clear that choosing the threshold T appropriately (with respect to the required QoS) is of paramount importance

5.7 Concluding remarks

This chapter studies a finite-sized discrete-time two-class priority queue where packets arrive according to a two-class discrete batch Markovian arrival process (2-DBMAP). Time and space priority are incorporated in the queueing model to provide different types of service to each class. One of both classes receives absolute time priority in order to minimize its delay. Space priority is implemented by the partial buffer sharing acceptance policy and can be provided to the class receiving time priority or to the other class. This choice gives rise to two different queueing models (Full and Mixed Priority) and this chapter analysed both these models in a unified manner. Furthermore, the buffer finiteness and the use of space priority make it interesting to consider a general order of arrivals at a slot boundary. This paper introduces a string representation for sequences of arriving packets. This naturally gives rise to intra-slot space priority (ISP) governing space priority between the packets arriving at a slot boundary. Performance of these queueing systems is then determined using matrix-analytic techniques. One can conclude that the range of service differentiation covered by these models is large and that ISP has a major impact for certain parameter settings and can thus not be neglected for bursty arrival processes. Determining an appropriate value for the threshold (space priority) is of paramount importance as it not only affects packet loss but also the queue content (and thus delay/time priority performance) of packets of both classes, especially for Mixed Priority.

Part III

Conclusions

6
CONCLUSIONS

In the first part of this dissertation, a two-class priority queueing system was studied modelling a node in a telecommunications network with two traffic classes. Real-time traffic, such as streaming video and voice, e.g. a Skype conversation, requires low delays but can endure a small amount of packet loss. On the other hand, data traffic, such as file transfer, benefits from low packet loss but has less stringent delay characteristics. Packets of each class arrive in a dedicated queue. Both queues are served by the same server but the server gives absolute (time) priority to class-1 packets in order to minimize the delay of these high-priority packets. Consequently, the (low-priority) class-2 packet waiting at the head of the class-2 queue can only enter the server if there are no class-1 packets in the system. The peculiarity of this model, denote this the N/∞ model, is that the class-1 queue capacity is limited to N, which is a finite positive integer, but the class-2 capacity is infinitely large. In contrast, in the literature, the queue capacity is generally assumed to be infinite for both classes (the ∞/∞ model). Evidently, as N increases, the N/∞ priority queue is increasingly similar to a system where both queues are presumed to be of infinite capacity, and we thus investigate this behavior. The analysis of the N/∞ model simultaneously took place in the probability domain for class-1 and in the transform domain for class-2 through the use of a vector/matrix representation. In a first chapter, we assumed that service of a packet always takes a single slot. In the subsequent chapter, we let the service times follow a general distribution. The part was concluded with a chapter on determining the tail behavior of the distribution of the low-priority system content for the model with single-slot service times studied earlier. It is evident that, in the limit for N to ∞, the results for N/∞ model must converge to those for ∞/∞ model. However this was not clear from the formulas for both systems. Our analysis has uncovered a crucial relation between the characteristic polynomial of a recurrence relation in the finite case and the kernel, which causes the implicitly defined function, in the infinite case. Furthermore, through several numerical examples, we have showed that, under certain conditions (small queue capacity, relatively high class-1 load, power-law arrivals), the result for the N/∞ model are considerably different from the ones obtained if one assumes infinite class-1 queue capacity.

In the second (shorter) part of this dissertation, we studied a two-class priority queueing system where both classes share a single queue with finite capacity according to a partial buffer sharing (PBS) policy. Here, both time and space priority played a crucial role. One of the classes receives absolute time priority. As in the previous section, these packets receive service before the packets of the other class. Additionally, one of the classes receives space priority. When the queue contains less packets than a (predetermined) threshold value, PBS accepts all packets but when the queue (also called buffer) level is over a predetermined threshold, packets with low space priority cannot enter the queue and are dropped by the system. Evidently, when the system is entirely full, all arriving packets are dropped. There are four possible combinations of the two priority types. However, we only needed to consider two as the two others then follow directly by swapping the classes. The two scenarios are thus giving both time and space priority to one of classes or giving time priority to one class and space priority to the other. In a general telecommunications context, as detailed in the previous paragraph, one would of course give time priority to real-time packets and space priority to data packets. In contrast, in a scalable video coding (SVC) setting one would prefer the other scenario. SVC uses two types of packets: base layer and enhancement layer packets. The former are required to decode and playback the video, although at poor quality, whereas enhancement packets increase quality but are useless without base packets. Here, it thus makes sense to give both time and space priority to base packets. We presented a unified way to model both scenarios and analyzed them using well-known matrix analytic solution techniques. One can conclude that the range of service differentiation covered by these models is large and that determining an appropriate value for the threshold of the PBS policy is of paramount importance.

Part IV

Appendices

A

APPENDIX: STOCHASTIC PROCESSES

This appendix describes the stochastic processes used in the numerical examples sections throughout this dissertation.

A.1 Poisson distribution

A discrete random variable x is said to have a Poisson distribution with parameter $\lambda > 0$, if the probability mass function (pmf) of x is given by

$$\Pr[x = n] = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n \geq 0. \tag{A.1}$$

Then, the corresponding probability generating function (pgf) reads

$$X(z) = \mathrm{E}\left[z^x\right] = e^{\lambda(z-1)}. \tag{A.2}$$

The mean and variance of a random variable with Poisson distribution happen to be identical, as

$$\mathrm{E}[x] = \mathrm{Var}[x] = \lambda. \tag{A.3}$$

A.1.1 Bivariate independent Poisson arrival process

Now, let us define a process appropriate for modelling the arrivals at a a two-class priority queue, the system studied in the first part of this dissertation. Let class-i ($i = 1,2$) arrivals occur according to a Poisson arrival process with parameter λ_i, independent of eachother. Using the notation developed there, we have

$$A_i(z) = \frac{(\lambda_1)^i e^{-\lambda_1}}{i!} e^{\lambda_2(z-1)}, \quad i \geq 0. \tag{A.4}$$

A.2 Geometric distribution

A discrete random variable x is said to have a geometric distribution with parameter $0 < p < 1$, if

$$\Pr[x = n] = (1-p)^n p, \quad n \geq 0\,, \tag{A.5}$$

and a shifted geometric distribution with parameter $0 < p < 1$, if

$$\Pr[x = n] = (1-p)^{n-1} p, \quad n \geq 1\,. \tag{A.6}$$

The latter corresponds to the number of trials until the first success in a Bernoulli experiment with success probability p, whereas the former denotes the number of failures before the first succes. Note that the support is different in each case as x does not take the value 0 in the shifted geometric case. The corresponding pgfs are given by

$$X(z) = \frac{p}{1-(1-p)z}\,, \tag{A.7}$$

and

$$X(z) = \frac{pz}{1-(1-p)z}\,. \tag{A.8}$$

A.2.1 Bivariate independent geometric arrival process

Evidently, we use the (regular) geometric distribution here in order to allow slots without arrivals. When class-i ($i = 1,2$) arrivals occur according to a geometric arrival process with parameter p_i, independent of eachother, we mean that

$$A_i(z) = \frac{(1-p_1)^i p_1 p_2}{1-(1-p_2)z}, \quad i \geq 0\,. \tag{A.9}$$

A.3 Power-law distribution

In the distributions described above, the probabilities decay rapidly as n increases. In practice, one very often encounters so called power-law distributions where the probabilities of “large” events remain significant. Such behaviour has been observed in city population, earthquake size, word frequency in texts (Zipf’s law), internet traffic, etc. Power-law distributions are also called heavy-tailed distributions, regularly varying distributions and Pareto distributions.

A power-law distribution with parameters β and γ is defined by $\Pr[x = 0] = 1-\beta$ and

$$\Pr[x = n] = \beta \frac{n^{-\gamma}}{\mathrm{Li}_\gamma(1)}, \quad n > 0\,, \tag{A.10}$$

with

$$\mathrm{Li}_\gamma(z) = \sum_{i=1}^{\infty} i^{-\gamma} z^i\,, \tag{A.11}$$

the so-called polylogarithm.

Therefore, the corresponding generating function

$$X(z) = 1 - \beta + \beta \frac{\mathrm{Li}_\gamma(z)}{\mathrm{Li}_\gamma(1)}\,. \tag{A.12}$$

has a branchpoint in $z = 1$.

Only the moments up to $\gamma - 1$ are finite, f.i. for $3 < \gamma < 4$ mean and variance are finite and skewness, kurtosis and all higher-order moments are infinite. If finite, the average is given by

$$\lambda = \beta \frac{\mathrm{Li}_{\gamma-1}(1)}{\mathrm{Li}_\gamma(1)}\,. \tag{A.13}$$

A.3.1 Bivariate independent power-law arrival process

Again let us define a process appropriate for modelling the arrivals at a a two-class priority queue. For each class, assume independent arrivals occurring according to a power-law process. Thus, by specifying the parameters (β_1, γ_1) and (β_2, γ_2) one completely describes the arrival process. In our notation, this yields

$$\begin{aligned} A_0(z) &= (1-\beta_1)\left(1 - \beta_2 + \beta_2 \frac{\mathrm{Li}_{\gamma_2}(z)}{\mathrm{Li}_{\gamma_2}(1)}\right), \\ A_i(z) &= \beta_1 \frac{i^{-\gamma_1}}{\mathrm{Li}_{\gamma_1}(1)}\left(1 - \beta_2 + \beta_2 \frac{\mathrm{Li}_{\gamma_2}(z)}{\mathrm{Li}_{\gamma_2}(1)}\right), \quad i > 0\,. \end{aligned} \tag{A.14}$$

A.4 Switch arrival process

Consider an output-queueing switch with S inlets and S outlets and two types of traffic. Such a switch is common in computer networks providing Differentiated Services (DiffServ) [7, 8]. In such networks, a switch is used to route incoming network traffic to several different destinations. This kind of switch is represented in figure A.1 for $S = 4$, where the queueing system we study is marked in gray. The switch operates as follows. Each inlet has the same input characteristics and traffic from the inlets is assumed to be destined for one of the outlets in a uniform manner. In front of each outlet a queueing system is in place, as up to S packets could try to access the same output in a (time-)slot, in order to buffer packets. The queuing systems in front of each outlet are statistically identical and one thus only needs to study one of them. Now, let us formally describe the arrival process at the studied queueing system.

On each inlet of the switch a batch arrives according to a Bernoulli process with parameter ν_T. A batch contains b (fixed) packets of class 1 with probability ν_1/ν_T or

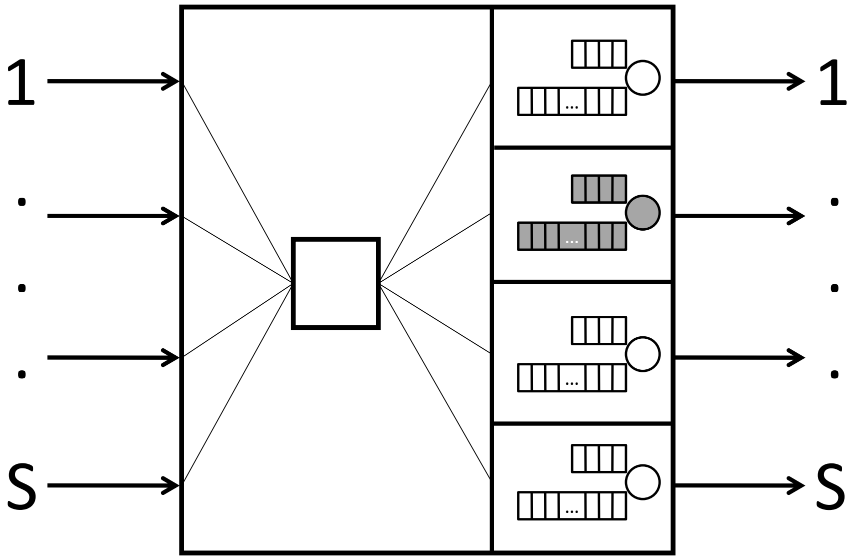

Figure A.1: Representation of a 4 × 4 *output queuing switch. The studied two-class queueing system is indicated in gray.*

b packets of class 2 with probability v_2/v_T (with $v_1+v_2 = v_T$). Incoming packets are routed uniformly to the outlets where they arrive at a two-class priority queueing system. The arrival process at the queueing system can consequently be described by the bivariate pmf

$$a(bn, bm) = \frac{S!\left(\frac{v_1}{S}\right)^n\left(\frac{v_2}{S}\right)^m\left(1-\frac{v_T}{S}\right)^{S-n-m}}{n!m!(S-n-m)!}, \tag{A.15}$$

for n and m integers with $n+m \le S$ and by $a(p,q) = 0$, for all other values of p and q. Because of the finite support (no more than Sb packets can arrive in a slot), constructing the corresponding functions $A_i(z)$ is straightforward. Furthermore, the mean number of class-i arrivals is given by $\lambda_i = b * v_i$.

Obviously the number of arrivals of class-1 and class-2 are negatively correlated as there can be no more than $Sb - i$ class-2 arrivals in a slot with i class-1 arrivals. For increasing values of S, the correlation increases and the numbers of arrivals of both types become uncorrelated for S going to infinity.

We have chosen this arrival process to, again, facilitate comparison with [32], where it is frequently used, but with $b = 1$. The addition of batch arrivals yields an easy way to increase variance in the arrival process.

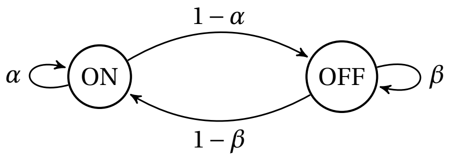

Figure A.2: Source transition diagram.

A.5 Multiple on-off sources

This arrival process is a discrete batch Markovian arrival process (DBMAP) [62], which is a very versatile class of arrival processes. Packets are generated by M on/off sources. Given that a source is on (off) at a slot boundary, it remains on (off) at the following slot boundary with probability α (β). This is demonstrated in figure A.2.

Consequently, consecutive on-periods (off-periods) constitute a series of geometrically distributed random variables with mean $1/(1-\alpha)$ ($1/(1-\beta)$). When a source is on at a slot boundary, it generates b_1 class-1 packets and b_2 class-2 packets. A source does not generate packets when it is off at a slot boundary. The aggregated DBMAP of these sources is easily established. The arrival process at the buffer is completely characterized by the quintuple $(M, b_1, b_2, \alpha, \beta)$. However, it is equivalent and often more convenient to use the quintuple (M, b_1, b_2, σ, K), where

$$\sigma = \frac{1-\beta}{2-\alpha-\beta}, \quad K = \frac{1}{2-\alpha-\beta}. \tag{A.16}$$

The parameter σ denotes the fraction of time a source is on and K is a measure for the absolute lengths of the on- and off-periods. The parameter K takes values between $\max(\sigma, 1-\sigma)$ and ∞. For $K < 1$, $K = 1$ and $K > 1$ the arrivals in consecutive slots are negatively correlated, not correlated and positively correlated respectively. Furthermore, the class-i arrival load is given by

$$\lambda_i = M\sigma b_i. \tag{A.17}$$

B

Appendix: Spectral Decomposition

Consider a square $m \times m$ matrix $\mathbf{A}$ and a scalar function f. The spectral decomposition theorem allows us to express the image of $\mathbf{A}$ under f by evaluating f (and its derivatives) in the eigenvalues of $\mathbf{A}$, see e.g. [72].

In the context of this dissertation, the function f is typically a power series $f(z) = \sum_{n=0}^{\infty} f_n z^n$ and the matrix $\mathbf{A}$ is non-diagonalisable. Such a matrix $\mathbf{A}$ cannot be reduced to a completely diagonal form by a similarity transform. However, any square matrix can be reduced to a form that is almost diagonal, called the Jordan normal form $\mathbf{J}$. Based on this reduction, it is possible to prove that the matrix $f(\mathbf{A})$ can be uniquely defined as

$$f(\mathbf{A}) = \sum_{j=1}^{s} \sum_{i=0}^{k_j-1} \frac{1}{i!} f^{(i)}(\xi_j)\,(\mathbf{A} - \xi_j \mathbf{I})^i\,\mathbf{G}_j\,, \tag{B.1}$$

see formula (7.9.9) in [72]. In this expression, $\{\xi_1, \ldots, \xi_s\}$ $(s \leq m)$ are the s distinct eigenvalues of $\mathbf{A}$, k_j denotes the index of eigenvalue ξ_j and $\boldsymbol{G}_j$ the spectral projector.

Note 72. *Recall that $f^{(i)}$ is the ith derivative of f. Here, we use ξ to denote an eigenvalue whereas most linear algebra texts use λ. However, throughout the queueing literature, λ is used to denote the mean number of arrivals in a slot. As this dissertation is primarily about queueing theory and linear algebra is merely a tool that helps us accomplish our goals, we have given "priority" to the queueing sense of λ.*

Obviously, it is required that the function f and its derivatives exist in the eigenvalues, i.e.

$$\xi_j \in \operatorname{dom} f^{(i)}\,, \qquad j = 1, \ldots, s, i = 0, \ldots, k_j - 1\,. \tag{B.2}$$

The matrices $\mathbf{G}_j$ are called the constituents or spectral projectors of $\mathbf{A}$ belonging to the eigenvalue ξ_j and have the following properties:

- $\mathbf{G}_j$ is idempotent, i.e. $\mathbf{G}_j^2 = \mathbf{G}_j$.
- $\mathbf{G}_1 + \mathbf{G}_2 + \ldots + \mathbf{G}_s = \mathbf{I}$, with $\mathbf{I}$ the $m \times m$ identity matrix.
- $\mathbf{G}_j\mathbf{G}_{j'} = \mathbf{0}$ whenever $j \neq j'$ $(1 \leq j, j' \leq s)$.

In general, the matrices $\mathbf{G}_j$ need to be calculated from the transformation matrix $\mathbf{P}$, for which $\mathbf{J} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}$. Specifically, if $\mathbf{P}$ is partitioned conformably as

$$\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1} = \begin{bmatrix}\mathbf{P}_1 & \mathbf{P}_2 & \cdots & \mathbf{P}_s\end{bmatrix} \begin{bmatrix}\mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_s\end{bmatrix} \begin{bmatrix}\mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \vdots \\ \mathbf{Q}_s\end{bmatrix}, \tag{B.3}$$

with $\mathbf{J}_j$ the Jordan segment corresponding with eigenvalue ξ_j, then the projectors $\mathbf{G}_j$ are

$$\mathbf{G}_j = \mathbf{P}_j\mathbf{Q}_j \qquad (j = 1, \ldots, s)\ . \tag{B.4}$$

We also note that the columns of $\mathbf{P}_j$ span the space of the right eigenvectors of $\mathbf{A}$ corresponding to ξ_j while the rows of $\mathbf{Q}_j$ span the space of its left eigenvectors.

C
APPENDIX: INVERTING A PGF

Consider a random variable x. The locations of singularities of the corresponding generating function $X(z)$ allow obtaining (an approximation of) the tail behavior. Moreover, the singularity with lowest norm (the so-called dominant singularity) determines the tail of the corresponding pmf $\Pr[x = n]$, when n is large enough. Consequently, the tail is completely characterized by the lowest-norm singularity z_x of $X(z)$ and the behaviour of $X(z)$ in the neighbourhood of this singularity. It is generally known that $z_x \in [1, \infty]$. For $z \in [0, z_x[$, $X(z)$ is a positive-real strictly-increasing function, so the inverse function $X^{-1}(z)$ can be defined on the real interval $[X(0), X(z_x)[$.

Following theorem (see [58]) allows characterization of the tail of the distribution of a random variable x from its generating function:

Theorem C.1. *Let $X(z)$ be the generating function of a random variable x, with dominant singularity z_x. Let $\beta \in \mathbb{R} \setminus \{0, 1, 2, \ldots\}$. If for $z \to \ z_x$*

$$X(z) \sim c_x \cdot (1 - z/z_x)^{\beta},$$

then the distribution $x(n)$ satisfies

$$x(n) \sim \frac{c_x n^{-\beta-1} z_x^{-n}}{\Gamma(-\beta)},$$

for $n \to \infty$, with $\Gamma(.)$ the Gamma function.

Basically, if the dominant singularity of a generating function (z_x) and the behaviour of the generating function in the neighbourhood of this dominant singularity (β and c_x) are identified, this theorem expresses the tail of the corresponding distribution ($x(n)$ for large n).

E.g. a dominant pole of multiplicity 1 ($\beta = -1$) in the interval $]1,\infty[$ leads to an exponential tail, whereas, if the dominant singularity is 1 ($z_x = 1$), a power-law tail is encountered.

BIBLIOGRAPHY

[1] J.F.C. Kingman. The first Erlang century - and the next. *Queueing Systems*, 63(1-4):3–12, 2009.

[2] W. Feller. *An introduction to probability theory and its applications*, volume vol. I. John Wiley & Sons, New York, 1950.

[3] D. Fiems and H. Bruneel. A note on the discretization of Little's result. *Operations Research Letters*, 30(1):17–18, 2002.

[4] H. Takagi. *Queueing analysis: a foundation of performance evaluation*, volume I, II and III. North-Holland, 1991-1993.

[5] M.F. Neuts. *Matrix-geometric solutions in stochastic models*. John Hopkins University Press, Baltimore, 1981.

[6] M.F. Neuts. *Structured stochastic matrices of M/G/1 type and their applications*. Probability, pure and applied. Marcel Dekker, 1989.

[7] S. Blake, D. Black, M. Carlson, E. Davies, Z Wang, and W. Weiss. An architecture for differentiated services. IETF RFC 2475, 1998.

[8] B. Carpenter and K. Nichols. Differentiated Services in the Internet. *Proceedings of the IEEE*, 90:1479–1494, (2002.

[9] J. Walraevens, T. Demoor, T. Maertens, and H. Bruneel. Stochastic queueing-theory approach to human dynamics. *Phys. Rev. E*, 85:021139, 2012.

[10] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. Partially shared buffers with full or mixed priority. *Journal of Industrial and Management Optimization*, 7(3):735–751, 2011.

[11] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Performance analysis of a priority queue : expedited forwarding PHB in DiffServ. *AEU-International Journal of Electronics and Communications*, 65(3):190–197, 2011.

[12] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Influence of real-time queue capacity on system contents in Diffserv's expedited forwarding per-hop-behavior. *Journal of Industrial and Management Optimization*, 6(3):587–602, 2010.

[13] J. Walraevens, T. Demoor, D. Fiems, and H. Bruneel. Uncovering the evolution from finite to infinite high-priority capacity in a priority queue. In *2013 International Conference on Computing, Networking and Communications (IEEE ICNC), San Diego*, 2013.

[14] D. Fiems, S. Andreev, T. Demoor, H. Bruneel, Y. Koucheryavy, and K. De Turck. Analytic evaluation of power saving in cooperative communication. In *Conference on Future Internet Communications (CFIC), Coimbra, Portugal*, 2013.

[15] T. Demoor, S. Andreev, K. De Turck, H. Bruneel, and D. Fiems. On the effect of combining cooperative communication with sleep mode. In *9th Annual Conference on Wireless On-demand Network Systems and Services (WONS), Courmayeur, Italy*, 2012.

[16] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. The impact of queue capacities on asymptotics in priority queues. In *International conference on Stochastic Modelling and Simulation, Chennai, India*, pages 29–29, 2011.

[17] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Tail behaviour of a finite-/infinite-capacity priority queue. In *3rd Madrid conference on Queueing Theory, Toledo, Spain*, pages 31–32, 2010.

[18] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. The preemptive repeat hybrid server interruption model. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010), Cardiff, Wales. Lecture Notes in Computer Science*, volume 6148, pages 59–71. Springer, Springer, 2010.

[19] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Time and space priority in a partially shared priority queue. In *5th International conference on Queueing Theory and Network Applications, Beijing, China*, pages 125–131. Association for Computing Machinery (ACM), 2010.

[20] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Mixed finite-/infinite-capacity priority queue with general class-1 service times. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2009), Madrid, Spain. Lecture Notes in Computer Science*, volume 5513, pages 264–278. Springer, 2009.

[21] T. Demoor, J. Walraevens, D. Fiems, S. De Vuyst, and H. Bruneel. Modelling queue sizes in an expedited forwarding DiffServ router with service differentiation. In *4th International conference on Queueing Theory and Network Applications, Singapore, Singapore.* Association for Computer Machinery (ACM), 2009.

[22] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Mixed finite-/infinite-capacity priority queue with interclass correlation. In *Analytical and Stochastic Modeling Techniques and Applications (ASMTA), Nicosia, Cyprus. Lecture Notes in Computer Science*, volume 5055, pages 61–74. Springer-Verlag, 2008.

[23] T. Demoor, D. Fiems, J. Walraevens, and H. Bruneel. Controlling delay and loss in a DiffServ router with expedited forwarding PHB. In *23rd National Conference of the Belgian Operations Research Society*, pages 98–98, 2009.

[24] T. Demoor, J. Walraevens, D. Fiems, and H. Bruneel. Performance analysis of a two-class priority queue with finite high-priority queue capacity. In *22nd National Conference of the Belgian Operations Research Society*, pages 54–56, 2008.

[25] A. Cobham. Priority assignment in waiting line problems. *Journal of the American Operations Research Society*, 2(1):70–76, 1954.

[26] R. Miller. Priority queues. *Annals of Mathematical Statistics*, 31:86–103, 1960.

[27] N. Jaiswal. *Priority queues.* Academic Press, New York, 1968.

[28] M. Sidi and A. Segall. Structured priority queueing systems with applications to packet-radio networks. *Performance Evaluation*, 3(4):265–275, 1983.

[29] T. Takine, B. Sengupta, and T. Hasegawa. An analysis of a discrete-time queue for broadband ISDN with priorities among traffic classes. *IEEE Transactions on Communications*, 42(2-4):1837–1845, 1994.

[30] J. Walraevens, B. Steyaert, and H. Bruneel. Performance analysis of a single-server ATM queue with a priority scheduling. *Computers & Operations Research*, 30(12):1807–1829, 2003.

[31] M. Mehmet Ali and X. Song. A performance analysis of a discrete-time priority queueing system with correlated arrivals. *Performance Evaluation*, 57(3):307–339, 2004.

[32] J. Walraevens. *Discrete-time queueing systems with priorities.* PhD thesis, Ghent University, 2004.

[33] A. Kapadia, M. Kazmi, and A. Mitchell. Analysis of a finite capacity non preemptive priority queue. *Computers & Operations Research*, 11(3):337 – 343, 1984.

[34] K. Avrachenkov, N. Vilchevsky, and G. Shevlyakov. Priority queueing with finite buffer size and randomized push-out mechanism. *Perform. Eval.*, 61(1):1–16, June 2005.

[35] J. Van Velthoven, B. Van Houdt, and C. Blondia. The impact of buffer finiteness on the loss rate in a priority queueing system. *Lecture Notes in Computer Science*, 4054:211–225, 2006.

[36] K. Al-Begain, A. Dudin, A. Kazimirsky, and S. Yerima. Investigation of the $M(2)/G(2)/1/\infty, N$ queue with restricted admission of priority customers and its application to HSDPA mobile systems. *Computer Networks*, 53(8):1186–1201, 2009.

[37] S. Asmussen. *Applied Probability and queues*. Springer-Verlag, New York, 2003.

[38] H. Bruneel and B. Kim. *Discrete-time models for communication systems including ATM*. Kluwer Academic Publisher, Boston, 1993.

[39] B. Vinck and H. Bruneel. A note on the system contents and cell delay in FIFO ATM-buffers. *Electronics Letters*, 31(12):952–954, 1995.

[40] D. Bertsimas and D. Nakazato. The distributional Little's law and its applications. *Operations Research*, 43:298–310, 1995.

[41] P. Whittle. Equilibrium distributions for an open migration process. *Journal of Applied Probability*, 5(3):567–571, 1968.

[42] D. Fiems. *Analysis of discrete-time queueing systems with vacations*. PhD thesis, Ghent University, 2004.

[43] T. Takine. A nonpreemptive priority MAP/G/1 queue with two classes of customers. *Journal of Operations Research Society of Japan*, 39(2):266–290, 1996.

[44] M. Kramer. Waiting times in a queueing system with capacity constraints and preemptive priorities. *Operations-Research-Spektrum*, 9(1):33–39, 1987.

[45] C. Blondia. A finite capacity multi-queueing system with priorities and with repeated server vacations. *Queueing Systems*, 5(4):313–330, 1989.

[46] U. Gupta, S. Samanta, R. Sharma, and M. Chaudhry. Discrete-time single-server finite-buffer queues under discrete Markovian arrival process with vacations. *Performance Evaluation*, 64(1):1–19, Jan 2007.

[47] E. Falkenberg. On the asymptotic behaviour of the stationary distribution of markov chains of m/g/1-type. *Communications in Statistics. Stochastic Models*, 10(1):75–97, 1994.

[48] J. Abate and W. Whitt. Asymptotics for M/G/1 low-priority waiting-time tail probabilities. *Queueing Systems*, 25(1-4):173–233, 1997.

[49] I. Adan, M. Mandjes, W. Scheinhardt, and E. Tzenova. On a generic class of two-node queueing systems. *Queueing Systems*, 61(1):37–63, 2009.

[50] D.P. Kroese, W.R.W. Scheinhardt, and P.G. Taylor. Spectral properties of the tandem Jackson network, seen as a quasi-birth-and-death process. *Annals of Applied Probability*, 14(4):2057–2089, 2004.

[51] Y. Sakuma and M. Miyazawa. On the effect of finite buffer truncation in a two-node Jackson network. *Journal of Applied Probability*, 42(1):199–222, 2005.

[52] N. Bean and G. Latouche. Approximations to quasi-birth-and-death processes with infinite blocks. *Adv. in Appl. Probab.*, 42(4):1102–1125, 12 2010.

[53] M. Miyazawa and Y.Q. Zhao. The stationary tail asymptotics in the GI/G/1-type queue with countably many background states. *Advances in Applied Probability*, 36(4):1231–1251, 2004.

[54] M. Miyazawa. A Markov renewal approach to M/G/1 type queues with countably many background states. *Queueing Systems*, 46(1-2):177–196, 2004.

[55] Q. He, H. Li, and Y. Zhao. Light-tailed behavior in qbd processes with countably many phases. *Stochastic Models*, 25(1):50–75, 2009.

[56] M. Miyazawa. Tail decay rates in double QBD processes and related reflected random walks. *Mathematics of Operations Research*, 34(3):547–575, 2009.

[57] F. Guillemin and J.S.H. van Leeuwaarden. Rare event asymptotics for a random walk in the quarter plane. *Queueing Systems*, 67(1):1–32, 2011.

[58] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2008.

[59] M. Drmota. Systems of functional equations. *Random Structures & Algorithms*, 10(1-2):103–124, 1997.

[60] P. Kravanja. *On Computing Zeros of Analytic Functions and Related Problems in Structured Numerical Linear Algebra*. PhD thesis, Katholieke Universiteit Leuven, 1999.

[61] P. Kravanja, M. Van Barel, O. Ragos, M. Vrahatis, and F. Zafiropoulos. ZEAL: A mathematical software package for computing zeros of analytic functions. *Computer Physics Communications*, 124(2-3):212–232, FEB 2000.

[62] C. Blondia. A discrete time batch Markovian arrival process as B-ISDN traffic model. *Belgian Journal of Operations Research, Statistics and Computer Science*, 32:3–23, 1993.

[63] H. Kröner, G. Hébuterne, P. Boyer, and A. Gravey. Priority management in atm switching nodes. *IEEE Journal on Selected Areas in Communications*, 9(3):418–427, 1991.

[64] Y. Wang, C. Liu, and C. Lu. Loss behavior in space priority queue with batch markovian arrival process - discrete-time case. *Performance Evaluation*, 41:269–293, 2000.

[65] G. Hwang and B. Choi. Performance analysis of the $DAR(1)/D/c$ priority queue under partial buffer sharing policy. *Computers & Operations Research*, 31:2231–2247, 2004.

[66] Y. Wang, J. Wang, and F. Tsai. Analysis of discrete-time space priority queue with fuzzy threshold. *Journal of Industrial and Management Optimization*, 5:467–479, 2009.

[67] D. Fiems, J. Walraevens, and H. Bruneel. Performance of a partially shared priority buffer with correlated arrivals. In *Proceedings of the 20th International Teletraffic Congress (ITC20), LNCS*, volume 4516, pages 582–593, Ottawa, 2007.

[68] H. Radha, Y. Chen, Parthasarathy K., and R. Cohen. Scalable internet video using MPEG-4. *Signal Processing: Image Communication*, 15:95–126, 1999.

[69] J. Zhao, B. Li, X. Cao, and I. Ahmad. A matrix-analytic solution for the DBMAP/PH/1 priority queue. *Queueing Systems*, 53(3):127–145, 2006.

[70] C. Blondia and O. Casals. Statistical multiplexing of VBR sources - a matrix-analytic approach. *Performance Evaluation*, 16(1-3):5–20, 1992.

[71] D.A. Bini, G. Latouche, and B. Meini. *Numerical methods for structured Markov chains*. Oxford University Press,.

[72] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.