Discrete-tijd-wachtlijnmodellen: veralgemeende bedieningsmechanismen en correlatie-effecten

Discrete-Time Queueing Models: Generalized Service Mechanisms and Correlation Effects

Bart Feyaerts

Promotoren: prof. dr. ir. Sabine Wittevrongel, prof. dr. ir. Herwig Bruneel
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. Herwig Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2013-2014

UNIVERSITEIT
GENT

ISBN 978-90-8578-713-6
NUR 919, 992
Wettelijk depot: D/2014/10.500/59

UNIVERSITEIT GENT

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Telecommunicatie en Informatieverwerking

Promotoren: prof. dr. ir. Sabine Wittevrongel, prof. dr. ir. Herwig Bruneel

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Telecommunicatie en Informatieverwerking
Sint-Pietersnieuwstraat 41, B-9000 Gent, België

Proefschrift tot het bekomen van de graad van
Doctor in de Ingenieurswetenschappen
Academiejaar 2013-2014

Dankwoord

Hierbij wens ik mijn dank uit te spreken aan iedereen die mij al dan niet rechtstreeks gesteund heeft bij het schrijven van dit doctoraat. Mijn naam mag dan misschien als enige onder de titel prijken maar dit werk had niet tot stand kunnen komen zonder de steun van vele anderen. In de eerste plaats denk ik daarbij natuurlijk aan mijn promotoren; Sabine Wittevrongel, die zich tot de laatste snik heeft ingespannen om dit werk naar een hoger niveau te tillen en Herwig Bruneel, om mij de kans te geven om deel uit te maken van deze fantastische vakgroep. Speciale dank gaat ook naar Stijn De Vuyst, om mij de voorbije jaren steeds met raad en daad bij te staan en aan wie ik meer dank verschuldigd ben dan ik hier kan neerschrijven.

Ook wens ik alle andere collega's en ex-collega's van vakgroep TELIN (en onderzoeksgroep SMACS in het bijzonder) te bedanken voor de goede sfeer en de aangename werkomgeving. Hoewel ik slechts het genoegen heb gehad om maar met een fractie van hen samen te werken, op conferentie te gaan of een al dan niet informeel gesprek te voeren, hebben zij de voorbije jaren voor mij mee gekleurd. Daarbij wens ik vooral ook mijn bureaugenoten expliciet te bedanken, met name (in alfabetische orde) Adriaan, Hannah, Huynh, Jasper, Julie, Kurt en Lennert, die de eindfase van dit werk bijzonder hebben weten te verlichten.

Als laatste, maar zeker niet als minste, wil ik ook mijn familie en vrienden bedanken. Niet zozeer vanwege een academische inbreng, maar vooral om mijn geest en gemoed gezond te houden en het mij niet al te kwalijk te nemen dat mijn dagelijkse leven als onderzoeker slechts zelden spectaculaire verhalen opleverde. In het bijzonder wil ik nog mijn vrouw Sofie bedanken om er voor mij te zijn, om te zijn wie je bent. Dankjewel!

Sint-Niklaas, April 2014
Bart Feyaerts

Table of Contents

Samenvatting

We leven in een tijdperk waarin communicatienetwerken tot zelfs de kleinste delen van ons dagelijkse leven zijn doorgesijpeld. 's Ochtendsvroeg worden we wakker met onze wekker afgestemd op ons favoriete radiostation, tijdens het ontbijt checken we de laatste tweets en statusupdates van onze vrienden en familieleden. Terwijl we op het perron staan te wachten, zoeken we op of de trein vertraging heeft en wanneer we eindelijk het station verlaten, lezen we het nieuws op onze tablet, smartphone, bril of zelfs polshorloge. Na het werk bellen we onze geliefde om te vertellen over onze dag, onderweg pikken we de boodschappen op die we online hebben besteld en 's avonds ontspannen we ons door naar een film op aanvraag te kijken.

De meeste mensen vinden deze netwerktoepassingen vanzelfsprekend en stellen er strenge eisen aan: er mag geen ruis op de radio zitten, webpagina's moeten snel laden, de spraak via de telefoonlijn moet goed zijn van kwaliteit en gestreamde media moeten vlot afspelen. Gezien vele van deze toepassingen tegelijkertijd afhankelijk zijn van dezelfde fysieke bronnen, houdt het tegemoetkomen aan die kwaliteitsvereisten verschillende technologische uitdagingen in. Om deze uitdagingen aan te gaan, is een diepgaand begrip van de onderdelen van communicatienetwerken onontbeerlijk om de prestaties van communicatiesystemen te bestuderen en om accurate voorspellingen te doen over de infrastructuur nodig om bepaalde prestatiedoelen te bereiken.

In dit proefschrift stel ik de resultaten voor van mijn onderzoeksactiviteiten binnen de onderzoeksgroep SMACS[1] (vakgroep TELIN[2], Universiteit Gent) gedurende de voorbije jaren. Dit onderzoek omvat de analytische studie van verschillende modellen voor wachtlijnsystemen binnen het domein van telecommunicatienetwerken en operationeel onderzoek. Analytische studies zoals deze kunnen dan overgedragen worden naar reële problemen, zoals degene die we eerder hebben aangehaald. Daarnaast bieden ze ook het nodige inzicht om geïnformeerde beslissingen te nemen om de vooropgestelde prestatiedoelen te bereiken.

De term *wachtlijnsysteem* behelst alle soorten systemen waarin gebruikers aankomen en hun beurt moeten afwachten om één of andere dienst te ontvangen. In de wetenschappelijke en toegepast wiskundige discipline van de *wachtlijntheorie* worden dergelijke systemen omgevormd tot wiskundige

[1] Stochastische Modellering en Analyse van Communicatiesystemen

[2] Telecommunicatie en Informatieverwerking

modellen die op hun beurt geanalyseerd kunnen worden om kennis te vergaren over de prestaties van het systeem. Doorgaans is men geïnteresseerd in stochastische eigenschappen zoals het gemiddeld aantal gebruikers in het systeem, de gemiddelde tijdsvertraging die gebruikers ondervinden en de probabiliteit dat die vertraging een bepaalde drempelwaarde overschrijdt.

In klassieke wachtlijnsystemen worden gebruikers bediend zodra de bedieningseenheid beschikbaar is en in de volgorde waarin ze toekomen in het systeem. In sommige gevallen is het bedieningsproces echter gewijzigd om specifieke prestatiedoelen te bereiken. Zulke wachtlijnsystemen met een gewijzigd bedieningsproces vormen het eerste thema van dit proefschrift, getiteld *veralgemeende bedieningsmechanismen*. Een eerste onderwerp binnen deze categorie betreft systemen waarin de gebruikers ingedeeld kunnen worden in twee aparte klassen: een kleine groep vertragingsgevoelige gebruikers met hoge prioriteit en een merendeel aan gebruikers met lage prioriteit en minder strikte vertragingsvereisten. Dit scenario is courant binnen de meeste communicatienetwerken, waar sommige gegevensstromen zoals gestreamde media minder vertraging kunnen verdragen dan de grote meerderheid aan internetverkeer bestaande uit surfverkeer, e-mails, het downloaden van bestanden, enz. In zulke gevallen is het wenselijk om de vertraging van de prioritaire gebruikers te verminderen of zelfs te minimaliseren, maar nog steeds voldoende doorvoer te verlenen aan niet-prioritaire gebruikers. Een ander onderwerp binnen deze categorie behandelt systemen waarbij het voordelig is om meerdere gebruikers na elkaar te bedienen, ook wanneer dit betekent dat gebruikers wellicht moeten wachten terwijl de bedieningseenheid eigenlijk werkloos is. Dit kan voorkomen wanneer de kostprijs om de bedieningseenheid te starten na een periode van inactiviteit beduidend groter is dan de kostprijs om individuele gebruikers te bedienen. In dergelijke gevallen moet er meestal een afweging worden gemaakt tussen de bijkomende vertraging en de winst in energie-efficiëntie of werkingskosten door meerdere gebruikers na elkaar te bedienen en de bedieningseenheid te stoppen wanneer die geen werk meer heeft.

De abstractie van een reëel wachtlijnsysteem naar een wiskundig model gaat doorgaans gepaard met een zekere vereenvoudiging die ervoor moet zorgen dat de analyse van het model haalbaar is. Zeker in geval van stochastische processen zoals het aankomst- en bedieningsproces, kan een accuraat model erg lastig zijn om te analyseren, terwijl een te verregaande vereenvoudiging naar een model van een beperkte moeilijkheidsgraad erg onnauwkeurige resultaten kan opleveren. In realiteit omvatten zulke processen doorgaans enige correlatie die een niet te onderschatten impact kan hebben op de prestatie van het systeem. Wanneer deze correlatie accuraat wordt gemodelleerd, creëert dit vaak een aanzienlijke uitdaging voor de wachtlijntheoreticus die het systeem analyseert. Het tweede thema van dit proefschrift is gewijd aan zulke *correlatie-effecten* die kunnen optreden in pakketgebaseerde telecommunicatienetwerken zoals het internet. Wanneer het aankomstproces van een wachtlijnsysteem meer bepaald overeenkomt met de uitgang van een bestandsserver of een media streaming server, is er vaak tijdscorrelatie aanwezig in het aankomstproces vanwege segmentatie van grote datastructuren. We presenteren twee verwante modellen voor een dergelijk aankomstproces en analyseren hoe deze correlatie de prestaties van het systeem beïnvloedt. In het eerste geval beschouwen we een zeer algemeen model dat een breed gamma aan reële aankomstprocessen kan voorstellen terwijl we het wachtlijnsysteem zelf vrij eenvoudig houden. In het tweede geval ligt de focus minder op correlatie binnen het aankomstproces, maar nemen we een realistisch model voor de uitgangslijn met gecorreleerde uitgangsonderbrekingen.

Dit proefschrift is als volgt opgebouwd. Deel I doet dienst als een inleiding en is specifiek bedoeld voor lezers die niet geheel vertrouwd zijn of slechts beperkte ervaring hebben met wachtlijntheorie in discrete tijd. We lichten toe wat wachtlijntheorie is in Hoofdstuk 1, geven een overzicht van dit proefschrift in Hoofdstuk 2 en introduceren enkele basiseigenschappen van wachtlijnsystemen in Hoofdstuk 3. We maken de lezer dan vertrouwd met de wiskunde achter probabiliteitsgenererende functies in Hoofdstuk 4 en presenteren een elementair wachtlijnmodel, notaties en basistechnieken in Hoofdstuk 5.

In Deel II bestuderen we twee wachtlijnmodellen waarbij geprobeerd wordt om de werking van wachtlijnsystemen zodanig aan te passen dat specifieke prestatiedoelen behaald kunnen worden. Hoofdstuk 6 onderzoekt een planningstechniek voor communicatiesystemen waarbij aan een gespecificeerd deel van het verkeer voorrang moet worden verleend. In Hoofdstuk 7 bestuderen we een mechanisme om het aantal keren dat de bedieningseenheid wordt geactiveerd en gedeactiveerd te verminderen.

Deel III is gewijd aan het modelleren van tijdscorrelatie in aankomst- en bedieningsprocessen en aan het analyseren van de impact van deze correlatie op de prestaties van het systeem. In Hoofdstuk 8 analyseren we de vertraging die structuren, genaamd *sessies*, bestaande uit meerdere pakketten ondervinden wanneer ze zich door een wachtlijnsysteem bewegen. Hoofdstuk 9 beschouwt een enigszins minder complex aankomstproces, maar introduceert tijdsgecorreleerde onderbrekingen van de uitgangslijn.

Summary

We live in an era where communication networks have seeped into even the smallest parts of everyday life. In the early morning we wake up with our alarm clocks tuned in to our favorite radio station, during breakfast we catch up on the latest tweets and status updates of our friends and family. Waiting on the platform, we look up if the train is going to be delayed and when we have finally left the station, we read the news on our tablets, smartphones, glasses or even wrist watches. After work, we phone our loved ones to tell about our day, en route we pick up the groceries we have ordered online and at night, we relax by watching a movie on demand.

Most people take these network applications for granted and set strict demands on them: there must not be any static on the radio, web pages should load fast, speech quality on the phone should be good and streamed media should play smoothly. Seeing as many of the applications rely on the same physical resources simultaneously, meeting the *quality of service* requirements comes with some technological challenges. Rising to these challenges, a profound understanding of the components in communication networks is essential to study the performance of communication systems and to make accurate projections on the infrastructure required to meet certain performance goals.

In this dissertation, I present the results of my research activities at the SMACS[3] research group (TELIN[4] department, Ghent University) during the past couple of years. This research consists of the analytical study of various models for queueing systems within the area of telecommunication networks as well as operations research. Analytical studies like these can then be ported to real-life problems, such as the ones illustrated above, and provide the insight necessary to make informed decisions in order to meet the intended performance goals.

The term *queueing system* applies to any kind of system at which customers arrive and have to wait their turn in order to receive some kind of service. In the scientific and applied mathematical discipline called *queueing theory*, these systems are converted into mathematical models that can then be analyzed in order to obtain knowledge about the system's performance. Usually, one is interested in stochastic properties such as the average number

[3] Stochastic Modeling and Analysis of Communication Systems

[4] Telecommunications and Information Processing

of customers in the system, the mean delay experienced by the customers and the probability that this delay exceeds a certain threshold.

In classical queueing systems, customers are served as soon as the service unit is available and in the order they have arrived to the system. In some cases however, the service process has been modified in order to reach specific performance goals. Such queueing systems with a modified service process constitute the first theme in this dissertation, called *generalized service mechanisms*. A first topic addressed in this category concerns systems where customers can be classified into two distinct classes: a small class of high-priority delay-sensitive customers and a majority of low-priority customers with less stringent delay requirements. This scenario is common in most communication networks, where some data streams such as streamed media are less delay-tolerant than the bulk of the Internet traffic comprised of web browsing, e-mails, file downloads, etc. In such cases it is desirable to reduce or even minimize the delay of the high-priority customers, while still offering sufficient throughput to the low-priority customers. Another topic in this category concerns systems where it is advantageous to serve multiple customers in a row, even if this means that customers might have to wait when the service unit is in fact idle. This can occur when the cost of initializing the service unit after a period of inactivity is considerably higher than the cost for serving individual customers. In such cases a trade-off must usually be made between the additional delay bestowed upon the customers and the profit made in terms of energy efficiency or operating costs by serving multiple customers in a row and switching the service unit off when it becomes idle.

The abstraction of a real-life queueing system into a mathematical model usually involves some sort of simplification in order to ensure that the analysis of the model is feasible. Especially in the case of stochastic processes such as the arrival and the service process, an accurate model can be quite tedious to analyze, whereas oversimplification to a model of low to moderate difficulty can yield quite inaccurate results. In real life, such processes usually incorporate some kind of correlation that can have a considerable impact on the system's performance. When modelled accurately, this correlation creates a considerable challenge for the queueing theorist analyzing the system. The second theme of this dissertation is devoted to such *correlation effects* that can occur in packet-based telecommunication networks such as the Internet. More specifically, when the arrival process of a queueing system corresponds to the output of a file server or a media streaming server, there is usually time correlation in the packet arrival process due to fragmentation of large data structures. We present two related models for such an arrival process and analyze how this correlation affects the system's performance. In the first case, the model is very general and can represent a vast range of real-life arrival processes, while the queueing system itself is kept rather simple. In the second case, we focus less on the correlation in the arrival process but in contrast we incorporate a realistic output line

model with correlated output interruptions.

This dissertation is structured as follows. Part I serves as an introduction and is of particular interest for readers who are not entirely familiar or have only limited experience with queueing theory in discrete time. We explain what queueing theory actually is in Chapter 1, we give an overview of this dissertation in Chapter 2 and we introduce some elementary properties of queueing systems in Chapter 3. We then familiarize the reader with the mathematics of probability generating functions in Chapter 4 and introduce a basic queueing model, notations and techniques in Chapter 5.

In Part II, we introduce two queueing models aimed at modifying the operation of queueing systems in order to reach certain specific performance goals. Chapter 6 investigates a scheduling technique for communication systems where a specified portion of the traffic should be prioritized. In Chapter 7, we study a mechanism for reducing the number of times the service unit is activated and deactivated.

Part III is devoted to modelling time correlation in arrival and service processes and to analyzing the impact of this correlation on the system's performance. In Chapter 8 we analyze the delay experienced by multi-packet entities, referred to as *sessions*, when they pass through a queueing system. Chapter 9 then considers a slightly less complex arrival process, but introduces time-correlated interruptions of the output line.

Part I

Introduction

Chapter 1

Queueing Theory

We start this dissertation with a brief introductory chapter on queueing theory, its history, common notations and some specifics of discrete-time models.

1.1 Queueing theory, systems and models

Although often unaware, we are all confronted with numerous examples of queueing *systems* throughout our everyday life: standing on the platform waiting for a train, waiting in line at the counter of a store, sitting in the waiting room of the doctor's office or experiencing delays when trying to order concert tickets online, etc. In each of these situations, customers gather in order to receive a certain service but have to wait because such a service takes time, only a limited number of customers can be served simultaneously or resources required for a service are not present.

Together, the whole of the gathering customers, the queue and the service unit along with the processes describing them define a queueing system.

Queueing *theory* is the scientific and mathematical field that researches such queueing systems and their behavior. This research is usually aimed at studying various performance measures such as the moments of the system content and the customer delay, assuming certain system parameters. As the majority of queueing research in this work is performed in a (digital) communication network setting, we will refer to the individual elements

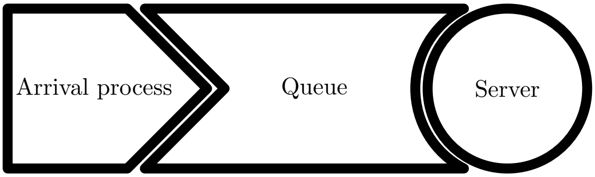

Figure 1.1: Illustration of a generic queueing model.

arriving to a queueing system as *packets*, and we will use the word *server* as a shorthand for the service unit.

In order to perform such queueing analyses, the real-life queueing systems under study are converted into an abstract mathematical queueing *model*. This model translates real-life aspects of the system (e.g. the arrival process) into mathematical structures (e.g. a certain probability distribution) that constitute an acceptable representation of the real-life counterpart. Throughout this dissertation, we will illustrate the structure of the queueing models under study by means of a modular schematic such as Figure 1.1. This schematic not only depicts the main constituting parts of a queueing model, but also shows how individual queueing models can be combined in order to create queueing networks. The study of such queueing networks however is beyond the scope of this dissertation.

1.2 A brief history

In the early 1900s, the Danish scientist Agner Krarup Erlang (1878-1929), working for the Copenhagen Telephone Company (KTAS), was frequently presented the classic problem of dimensioning the number of circuits and operators needed to provide an acceptable telephone service. Note that in those days, human operators were needed to connect callers with callees using telephone switchboards and jack plugs. He was one of the first to perform detailed studies about telephone traffic, resulting in his 1909 publication [32] where he presented the observation that random telephone traffic follows a Poisson distribution. Later on, studying the cost versus quality trade-off, he composed his renowned formulae for call loss and waiting times [33], which were soon adopted by telephone companies all over the world.

Erlang is considered to be the founding father of what is now called queueing theory and since its genesis with Erlang's 1909 paper, queueing theory itself has evolved along with network technologies, leading to new approaches, new techniques, new demands and new goals. Without going into technical detail, one can still understand that there is quite a difference between dimensioning an analog circuit-switched telephone network and dimensioning a digital packet-switched backbone network that supports various services of various types (e.g. telephony, television broadcast, the Internet, ...).

Although queueing theory is predominantly applied to the performance analysis of communication systems, it has also found its way into other application domains. Some examples are: transportation [50, 110] (e.g. traffic congestion control, public transport scheduling), production processes [52, 99] (e.g. stock management, process planning, machine breakdowns and repairs), healthcare [53, 98] (e.g. emergency planning), etc.

1.3 Kendall notation

The standard classification method for describing queueing models is referred to as the Kendall notation, named after the English scientist David George Kendall. In [70], he introduced the shorthand notation $A/S/c$ as a concise description of the main characteristics of a queueing system. The A describes the distribution of the interarrival times between the individual packets; from this parameter one can often identify whether the corresponding queueing model is a discrete-time model or a continuous-time model. Common arrival process notations are:

A	Type	Description
G/GI	C, D	Independent and Arbitrarily distributed interarrival times (GI = General Independent)
Geo, Geo^X	D	Geometrically distributed interarrival times
M, M^X	C	Poisson arrival process (M = Memoryless)
MAP	C	Markovian arrival process
D(B)MAP	D	Discrete (Batch) Markovian arrival process
PH	C	Phase-type distributed interarrival times
E$_k$	C	Erlang-k distributed interarrival times

Table 1.1: An overview of common arrival process descriptors. The first column contains the Kendall notation, note that the superscript X is a placeholder for a batch size. The second column denotes whether the arrival process is continuous-time (C), discrete-time (D) or can be either of the two.

The parameter S describes the service times of the individual packets. Similar to the descriptor of the interarrival times, the service process descriptor S can often be used to distinguish between discrete-time and continuous-time queueing models. Common notations for the service process are given in Table 1.2.

Finally, the third parameter, c, denotes the number of servers in the queueing system.

Over the years, the Kendall notation has been extended and adapted to fit the needs of authors studying more complex queueing systems. A

S	Type	Description
G/GI	C, D	Arbitrarily distributed and independent service times
D	C, D	Deterministic (fixed) service times, usually 1 slot per packet in the discrete-time case
Geo	D	Geometrically distributed service times
E_k	C	Erlang-k distributed service times

Table 1.2: An overview of common service process descriptors. The first column contains the Kendall notation. The second column denotes whether the service process is continuous-time (C), discrete-time (D) or can be either of the two.

common extended Kendall notation is given by $A/S/c/K/N/D$, where K denotes the queue capacity, i.e. the maximal number of packets that can reside in the queue at any time. The parameter N then stands for the size of the population from which the packets in the system stem and thus denotes the maximal number of packets that can be in the system at any time. This parameter can have a significant effect on the effective arrival rate. Finally, D denotes the queueing discipline; this usually is either *FIFO* (First in, first out) a.k.a. *FCFS* (First come, first served), where packets are served in the order in which they entered the queue or *LIFO* (Last in, first out) a.k.a. *LCFS* (Last come, first served), where the newest arrivals are first served. The final three parameters can be omitted in case they correspond with their default values (i.e. $K = \infty$, $N = \infty$ and $D =$ FIFO). This yields the notation $A/S/c$, conform with the short Kendall notation defined above.

When analyzing discrete-time queueing systems, it is usually more meaningful to describe the number of arrivals per slot, rather than the interarrival times between individual packets. In order to visualize this alternative convention, the delimiter / is replaced by $-$, yielding the notation $A - S - c - K - N - D$ or an abbreviated form thereof. The parameter A then no longer refers to the interarrival times, but denotes the distribution of the number of packet arrivals per slot.

1.4 Discrete-time queueing models

Given the nature of telephony, Erlang used continuous-time models as a mathematical representation of the telephone networks he studied. Therefore, it comes as no surprise that the majority of queueing literature studies continuous-time queueing models. Discrete-time queueing models received only little attention until the publication [88] of Torben Meisling in 1958 put discrete-time queues on the radar of queueing theorists. Since then, discrete-time queueing gained interest [13, 74, 93, 105], particularly because of its applicability in computer systems and digital communication systems.

In discrete-time queueing models, time is assumed to be divided into intervals of fixed length referred to as *slots*, which act as the smallest distinguishable time units. This assumption raises questions about how this slotted time translates to the various events that occur in queueing systems. More specifically, we need to specify when exactly arrivals can occur, when exactly services can start, when exactly services can end, when exactly we observe the system, ...

Throughout this dissertation, we will therefore always assume the timing conventions described below and depicted in Figure 1.2. Packets can arrive

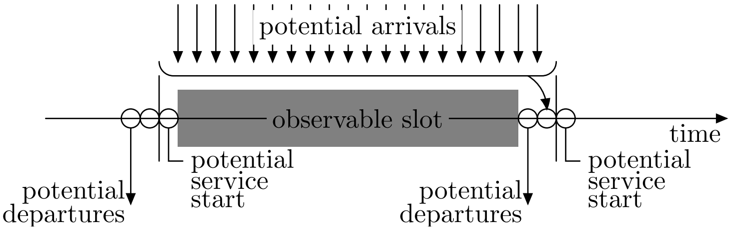

Figure 1.2: Illustration of the timing conventions adopted in this dissertation.

to the system at any time during a slot, but they are only inserted at the very end af their arrival slot. Services can only start at the very start of slots and last for an integer number of slots with departures at the very end of the final slot of the service. As such, all changes in the system state only occur in infinitesimally small intervals surrounding slot boundaries. Conversely, the system state is unchangeable during actual slots, i.e. excluding the infinitesimal intervals. Therefore, we observe the system only *during* slots (illustrated by the gray area), such that we are certain that the system state has stabilized. For the sake of clarity, we will however frequently refer to the system state *at the beginning* of a slot, merely to emphasize that we observe the system before any departures or arrivals that can occur at the end of the slot.

Chapter 2

Topics Addressed in this Dissertation

2.1 Generalized service mechanisms

In a classical discrete-time queue operating under the FIFO-policy, packets are appended to the queue when they arrive to the system. Meanwhile, the service unit drains the queue by pulling the oldest packet from the queue's head, provided that the service unit is empty and the queue is not. In some scenarios, there might however be specific performance goals that cannot be accomplished using this approach. In such cases, *generalized service mechanisms* can be applied to the system, altering how packets are inserted in the queue, how they move through the queue and/or how they are pulled from the queue by the service unit. We consider two such scenarios.

First, we consider a system where packets belong to one of two possible classes. A small portion of the packets should be transmitted with as little delay as possible, such as packets containing real-time data or streamed media, this is the high-priority class. The low-priority class covers the majority of the packets which is delay-tolerant and requires best-effort treatment only. The FIFO-policy makes no distinction between these classes and can therefore not impose such a delay differentiation. Absolute Priority is another scheduling discipline and blocks the passage of low-priority packets as long as there are high-priority packets in the system. We introduce the Reservation discipline, which achieves delay differentiation by letting some high-priority packets overtake low-priority packets. We compare the performance of this milder scheduling discipline with the two extremes of FIFO and Absolute Priority.

A second type of systems with a generalized service mechanism are systems where the activation of the service unit after a period of inactivity comes at a considerable cost. This scenario is less likely to occur in communication networks, but can be particularly interesting for production processes. It might then be more cost-effective to postpone the activation of the service unit - if idle - until a specified number of customers has arrived. In order to prevent excessive delays if the arrival rate is low, a timer can be installed to guarantee a maximal waiting time for the first customer. We study this double-threshold policy and compare its effects on the system's functioning and the customer delay with the effects of applying only one of these thresholds.

2.2 Correlation effects

The analysis of a queueing system usually starts with the construction of a mathematical model corresponding to that system. A model that accurately grasps the characteristics of the queueing system usually yields results that are more reliable than the results obtained by poorly constructed or oversimplified models. One of those characteristics is correlation in one of the system's subprocesses and this correlation often has a considerable impact both on the analysis of the system and the system's performance. We present two related models for correlated arrival processes tailored to accurately represent the output of file servers and media streaming servers. We apply these models to queueing systems with unreliable output lines and analyze the system state and the delay of individual packets as well as entire multi-packet structures representing large files or complete streams. In a first case, we consider a complex arrival model that can represent a vast range of real-life arrival processes in the context of a rather simple queueing system. In a second case, we consider a more complex output line model with correlated output interruptions, on top of a slightly less complex arrival process. In both cases we investigate the effect of these correlated processes on the system's performance.

2.3 Dissertation outline

In Chapter 1, we have elaborated on the history of queueing theory, setting a scene for the remainder of this dissertation. Next, we will discuss some elementary properties of queueing systems in Chapter 3 and provide a mathematical background for probability generating functions in Chapter 4. We illustrate some basic analysis techniques and common notations in Chapter 5, where we analyze a basic queueing model.

Part II considers two queueing models with *generalized service mechanisms.* In Chapter 6 we consider the reservation-based scheduling technique, which is aimed at prioritizing a specified portion of the traffic, without choking the non-prioritized traffic. We determine the distributions for the system state and the packet delay of either packet class, as well as the tail probabilities of the packet delay. We compare our results with FIFO-scheduling and Absolute Priority scheduling, which are the two extremes in priority scheduling. Chapter 7 studies a queueing system where service unit activation is postponed until a specified number of customers is waiting in the queue or until the first customer has been waiting for a certain amount of time. We analyze the operation of the queueing system and the delay of customers, depending on when they arrive in the system and present an approximation technique. We compare our findings with those of other policies where only the number of customers in the queue or the waiting time of the first customer triggers the activation of the service unit.

In Part III, we study *correlation effects* caused by correlated arrival and service processes. In Chapter 8 we introduce sessions as multi-packet entities that produce a variable number of arrivals over a variable number of consecutive slots. We extend previous research on this subject by analyzing the delay incurred by these sessions as they pass through the system and study the effects of the various parameters that control the time-correlation induced by session-based arrival processes. In Chapter 9, we consider not only a correlated arrival process, but also output line interruptions with time correlation. More specifically, the arrival process generates packet trains of geometric length and the accessibility of the output line is governed by a Markovian process with an arbitrary state space. We present a technique for studying the system state as well as the packet and train delay, allowing us to analyze the effects of both sources of correlation on the system's performance.

Finally, we conclude this dissertation with a summary of the main contributions presented here and some subjects that were left untouched throughout this work.

2.4 Overview of publications

The research presented in this dissertation has resulted in several journal papers and conference contributions. The listing below offers a detailed view of these papers.

Publications in international journals

1. Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel and Sabine Wittevrongel. Analysis of discrete-time buffers with heterogeneous session-based arrivals and general session lengths. *Computers & Operations Research*, 39 (12) 2905–2914, December 2012. ISSN 0305-0548. doi: 10.1016/j.cor.2011.11.023.

2. Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel and Sabine Wittevrongel. The impact of the NT-policy on the behaviour of a discrete-time queue with general service times. *Journal of Industrial and Management Optimization*, 10 (1) 131–149, January 2014. doi: 10.3934/jimo.2014.10.131.

3. Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel and Sabine Wittevrongel. Performance analysis of buffers with train arrivals and correlated output interruptions. Accepted for publication in *Journal of Industrial and Management Optimization.*

4. Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel and Sabine Wittevrongel. Delay analysis of a discrete-time $GI - GI - 1$ queue with reservation-based priority scheduling. Submitted for publication in *European Journal of Operational Research.*

Papers in proceedings of international conferences

1. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. Analysis of a discrete-time priority queue with place reservations and geometric service times. In *Proceedings of the Sixth Conference on Design, Analysis, and Simulation of Distributed Systems, (DASD 2008)*, pages 140–147, Edinburgh, Scotland, United Kingdom, June 2008.

2. Bart Feyaerts and Sabine Wittevrongel. Performance analysis of a priority queue with place reservation and general transmission times. In *Proceedings of the 5th European Performance Engineering Workshop (EPEW 2008)*, volume 5261 of *Lecture Notes in Computer Science*, pages 197–211. Palma de Mallorca, Spain, September 2008. ISBN 978-3-540-87411-9.

3. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. Session delay in file server output buffers with general session lengths. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC 2010)*, pages 1–5, Cape Town, South-Africa, May 2010. doi: 10.1109/ICC.2010.5502624.

4. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. Analysis of a discrete-time queueing system with an NT-policy. In *Proceedings of the 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010)*, volume 6148 of *Lecture Notes in Computer Science*, pages 29–43, Cardiff, United Kingdom, June 2010. ISBN 3-642-13567-6, 978-3-642-13567-5.

5. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. Analysis of a discrete-time NT-policy queue with general service times. In *Proceedings of the 7th International Conference on Queueing Theory and Network Applications (QTNA 2012)*, paper 16–1, pages 1–9, Kyoto, Japan, August 2012.

6. Bart Feyaerts, Sabine Wittevrongel, Stijn De Vuyst and Herwig Bruneel. Discrete-time queues with train arrivals and Markovian server interruptions. In *Proceedings of the 8th International Conference on Queueing Theory and Network Applications (QTNA 2013)*, pages 83–89, Taichung, Taiwan, July/August 2013.

Abstracts and other presentations

1. Bart Feyaerts and Sabine Wittevrongel. Delay analysis of a place reservation queue. In *Book of Abstracts of the 9th FirW PhD Symposium*, pages 84–85, Ghent, Belgium, December 2008.

2. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. The $Geo^{X,X}/G/1$ queue with the reservation discipline. In *Booklet of Abstracts of the 23rd Belgian Conference on Operations Research (ORBEL '09)*, page 95, Leuven, Belgium, February 2009.

3. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. Modelling data traffic performance in file servers : session-based arrivals. In *Proceedings of the 24th Belgian Conference on Operations Research (ORBEL 24)*, pages 154–155, Liège, Belgium, January 2010.

4. Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel and Herwig Bruneel. On the NT-policy for discrete-time queues. In *Proceedings of the 25th Belgian Conference on Operations Research (ORBEL 25)*, pages 40–41, Ghent, Belgium, February 2011.

Chapter 3

Elementary Properties

This chapter outlines some basic properties from queueing and probability theory that we will repeatedly use throughout this dissertation.

3.1 Steady state

Under the right conditions, the impact of the initial state of a queueing system will wither over time. The system then enters a *steady state* (also called *regime* or *equilibrium*) where the system state distribution at random instants is stochastically identical. Given that steady-state performance measures are not related to any specific initial state, it is especially interesting to investigate the regime behavior of queueing systems. A queueing system can only reach this equilibrium if it complies with the *equilibrium condition* that states that the average number of arrivals per time unit is not greater than the maximal average number of possible departures per time unit. Usually even, the average number of arrivals must be strictly less than this maximum. When the system reaches equilibrium, the actual average number of departures per time unit will be equal to the average number of arrivals per time unit.

Note that in case of a finite buffer capacity, this condition is met by definition. In case of infinite buffers however, the equilibrium condition is indispensable. Otherwise, the actual queue length as well as the packet delay will increase to infinity.

3.2 Little's law

One of the most important theorems in queueing theory is *Little's law*, named after John D. C. Little, the professor who first proved the validity of the theorem in a continuous-time setting in 1961 [81, 82]. Little's law states that the average number of packets in a system in equilibrium ($\mathrm{E}[u]$) is equal to the product of the average arrival rate (λ) and the average time spent by the packets in the system ($\mathrm{E}[d]$). This can be expressed formally as

$$\mathrm{E}[u] = \lambda\,\mathrm{E}[d]\,. \tag{3.1}$$

This result, however simple it may seem, is very important and remarkable in that it is independent of the distribution of any aspect of the system. Furthermore, it can not only be applied to systems in their entirety, but also to subsystems, or even partial arrival flows.

It has been proved that Little's law is also applicable to a large class of discrete-time systems [45]. Specifically, Little's law can be applied to synchronized discrete-time systems, i.e. systems where service is synchronized to slot boundaries. The service of a packet then starts at (just after) a slot boundary and ends at (just before) a subsequent slot boundary and is expressed as an integer number of slots.

Although this theorem is widely applicable, we will refrain ourselves from exploiting it as a means to obtain either $\mathrm{E}[u]$ or $\mathrm{E}[d]$. Rather, we will usually use it as a test to ensure the validity of our results.

3.3 PASTA / BASTA

In continuous-time systems where arrivals are governed by a Poisson process, the fraction of packets that arrive to a system in a certain state P generally equals the fraction of time the system resides in that same state P. This equality is referred to as the *PASTA* property (*Poisson arrivals see time averages*) [122]. The discrete-time counterpart is referred to as the *BASTA* property (*Bernoulli arrivals see time averages*) [7].

3.4 Law of total expectation

In probability theory, the *law of total expectation* states that for any two random variables X and Y, the expected value of X can be calculated as the expected value of the conditional expected value of X given Y, i.e.

$$\mathrm{E}[X] = \mathrm{E}[\mathrm{E}[X|Y]]\,. \tag{3.2}$$

Note that $\mathrm{E}[X|Y]$ is in fact a random variable dependent on Y, such that the outer $\mathrm{E}[\ldots]$ operator denotes the mean of that random variable and performs the weighted sum $\sum_y \mathrm{Prob}[Y = y]\,\mathrm{E}[X|Y = y]$.

In practice, this means that when $\mathrm{E}[X]$ is hard to determine directly, one might try to find a random variable Y for which the different conditional means $\mathrm{E}[X|Y = y]$ are less challenging to calculate. We will make frequent use of this technique and therefore we introduce the shorthand

$$\mathrm{E}[X\,\{Y = y\}] \triangleq \mathrm{Prob}[Y = y]\,\mathrm{E}[X|Y = y]\,. \tag{3.3}$$

Chapter 4

Probability Generating Functions

Throughout this work, we will make frequent use of probability generating functions (*pgf*s) as a means of describing distributions. The pgf $X(z)$, where z is a complex number, of a nonnegative discrete random variable X, defined as

$$X(z) \triangleq \mathrm{E}\big[z^X\big] = \sum_{n=0}^{\infty} \mathrm{Prob}[X = n]\, z^n, \tag{4.1}$$

comprises the same information about X as its probability mass function (*pmf*) $x(n) \triangleq \mathrm{Prob}[X = n]$, albeit more succinct. Note that in (4.1), the notation $\mathrm{E}[\ldots]$ denotes the expected value of the random variable within brackets. In this chapter we will present some properties of pgfs that we will use repeatedly over this dissertation, both for univariate distributions and for multivariate distributions.

4.1 Multivariate pgfs

Similarly to (4.1), the joint pgf $Y(z_1, \ldots, z_N)$ of the $N \geq 1$ nonnegative discrete random variables $Y_1, \ldots, Y_N$ can be defined as

$$\begin{aligned} Y(z_1, \ldots, z_N) &\triangleq \mathrm{E}\big[{z_1}^{Y_1} \ldots {z_N}^{Y_N}\big] \\ &= \sum_{n_1, \ldots, n_N = 0}^{\infty} \mathrm{Prob}[Y_1 = n_1, \ldots, Y_N = n_N]\, {z_1}^{n_1} \ldots {z_N}^{n_N}. \end{aligned} \tag{4.2}$$

The *marginal pgf* $Y_i(z)$ ($i \in \{1, \ldots, N\}$) of Y_i can be obtained from (4.2) by substitution of $z_i = z$ and $z_n = 1$ for all values of $n \neq i$.

A pgf can also be constructed as the weighted sum of *partial pgfs*, where a certain random variable is assumed to have a fixed value. For example, let the partial pgfs $F_j(z_1, \ldots, z_{N-1})$ ($j \in \mathbb{N}$) be defined as

$$\begin{aligned} F_j(z_1, \ldots, z_{N-1}) &\triangleq \mathrm{E}\big[{z_1}^{Y_1} \ldots {z_{N-1}}^{Y_{N-1}} \{Y_N = j\}\big] \\ &= \sum_{n_1, \ldots, n_{N-1}=0}^{\infty} \mathrm{Prob}[Y_1 = n_1, \ldots, Y_{N-1} = n_{N-1}, Y_N = j]\, {z_1}^{n_1} \ldots {z_{N-1}}^{n_{N-1}}. \end{aligned} \tag{4.3}$$

The joint pgf $Y(z_1, \ldots, z_N)$ of $Y_1, \ldots, Y_N$ can then be found from the law of total expectation as

$$Y(z_1, \ldots, z_N) \triangleq \mathrm{E}\big[{z_1}^{Y_1} \ldots {z_N}^{Y_N}\big] = \sum_{j=0}^{\infty} F_j(z_1, \ldots, z_{N-1}) {z_N}^j. \tag{4.4}$$

4.2 Normalization condition

The normalization condition states that every probability distribution must be normalized. Specifically for discrete random variables, this means that the sum of all probabilities must be equal to 1. This can be expressed in terms of the previously defined pgfs $X(z)$ and $Y(z_1, \ldots, z_N)$ as

$$X(1) = 1, \qquad \text{and} \qquad Y(1, \ldots, 1) = 1. \tag{4.5}$$

Note that partial pgfs do not necessarily obey the normalization condition, rather we get that

$$\begin{aligned} F_j(1, \ldots, 1) &= \sum_{n_1, \ldots, n_{N-1}=0}^{\infty} \mathrm{Prob}[Y_1 = n_1, \ldots, Y_{N-1} = n_{N-1}, Y_N = j] \\ &= \mathrm{Prob}[Y_N = j] \leq 1. \end{aligned} \tag{4.6}$$

4.3 Radius of convergence

Both definitions (4.1) and (4.2) contain a power series that may or may not converge. It can be shown that the power series in (4.1) converges at least for all $z \in \mathbb{C}$ with $|z| \leq 1$. Likewise, the power series in (4.2) converges at least for all $z_1, \ldots, z_N \in \mathbb{C}$ with $max\left(|z_1|, \ldots, |z_N|\right) \leq 1$. This is induced by the fact that probabilities are limited to $[0, 1]$ and distributions respect the normalization condition.

Within this radius of convergence, pgfs are analytic functions, and therefore they must not have any singularities in this area.

4.4 Probability generating property

According to the definitions (4.1) and (4.2), it is possible to construct a pgf from the corresponding pmf. It is also possible to invert this procedure and to extract values of the pmf from the pgf. Seeing Prob$[x = n]$ as the coefficient of z^n in the Taylor expansion of $X(z)$ at $z = 0$, Prob$[x = n]$ can be found as

$$\text{Prob}[X = n] = \frac{1}{n!} \frac{\mathrm{d}^n X(z)}{\mathrm{d}z^n}\bigg|_{z=0}, \qquad n \geq 0. \tag{4.7}$$

Although it might look somewhat more complex, inversion of multivariate pgfs can be done similarly:

$$\begin{aligned}&\text{Prob}[Y_1 = n_1, \dots, Y_N = n_N] \\ &\quad= \left(\prod_{i=1}^{N} \frac{1}{n_i!}\right) \frac{\partial^{\sum_{i=1}^{N} n_i} Y(z_1, \dots, z_N)}{\partial {z_1}^{n_1} \dots \partial {z_N}^{n_N}}\Bigg|_{z_1=0,\dots,z_N=0}, \qquad n_1, \dots, n_N \geq 0.\end{aligned} \tag{4.8}$$

From these formulas the following particular property of practical use can be obtained:

$$\text{Prob}[X = 0] = X(0), \quad \text{and} \quad \text{Prob}[Y_1 = 0, \dots, Y_N = 0] = Y(0, \dots, 0). \tag{4.9}$$

Actual application of formulas (4.7) and (4.8) is computationally extremely challenging and thus practically unfeasible, especially for large values of $n_1, \dots, n_N$. The inversion can however be performed using approximate techniques, as described in [1, 2].

4.5 Moment generating property

From (4.1), factorial moments of the random variable X can be calculated as

$$\mathrm{E}\left[\frac{X!}{(X-n)!}\right] = \frac{\mathrm{d}^n X(z)}{\mathrm{d}z^n}\bigg|_{z=1}, \qquad n \geq 0. \tag{4.10}$$

Choosing $n = 1$, we get

$$\mathrm{E}[X] = \frac{\mathrm{d}X(z)}{\mathrm{d}z}\bigg|_{z=1} = X'(1). \tag{4.11}$$

For $n = 2$, this becomes

$$\mathrm{E}[X\,(X-1)] = X''(1), \tag{4.12}$$

such that the variance $\mathrm{Var}[X]$ of X can be found as

$$\mathrm{Var}[X] = \mathrm{E}\big[X^2\big] - \mathrm{E}[X] = X''(1) + X'(1) - X'(1)^2. \tag{4.13}$$

Moments of a single random variable from a multivariate pgf can be obtained similarly, starting from the corresponding marginal pgf.

4.6 Linear combination of random variables

The use of pgfs will allow us to write and perform stochastic calculations more concisely and elegantly than when using pmfs, especially when we want to determine the distribution of a random variable which is a linear combination of other random variables. For instance, let $Y_1, \ldots, Y_N$ be a set of nonnegative discrete random variables with joint pgf $Y(z_1, \ldots, z_N)$ and marginal pgfs $Y_i(z)$ ($i \in \{1, \ldots, N\}$) and let X be a linear combination of these random variables, i.e.

$$X = \sum_{i=1}^{N} c_i Y_i, \tag{4.14}$$

for the tuple of nonnegative coefficients $(c_1, \ldots, c_N)$.

If we want to determine the pmf of X, we find

$$\mathrm{Prob}[X = m] = \sum_{m_1,\ldots,m_{N-1}=0}^{\infty} \mathrm{Prob}\Bigg[Y_1 = m_1, \ldots, Y_{N-1} = m_{N-1}, \\ c_N Y_N = m - \sum_{i=1}^{N-1} c_i m_i\Bigg]. \tag{4.15}$$

Alternatively, calculating the pgf of X, we get

$$X(z) \triangleq \mathrm{E}\big[z^X\big] = \mathrm{E}\Big[z^{\sum_{i=1}^{N} c_i Y_i}\Big] = \mathrm{E}\Bigg[\prod_{i=1}^{N} z^{c_i Y_i}\Bigg] = Y(z^{c_1}, \ldots, z^{c_N}). \tag{4.16}$$

In the special case where the random variables $Y_1, \ldots, Y_N$ are independent, both expressions can be simplified to

$$\mathrm{Prob}[X = m] = \sum_{m_1,\ldots,m_{N-1}=0}^{\infty} \left(\prod_{i=1}^{N-1} \mathrm{Prob}[Y_i = m_i]\right) \\ \left(\mathrm{Prob}\Bigg[c_N Y_N = m - \sum_{i=1}^{N-1} c_i m_i\Bigg]\right), \tag{4.17}$$

and

$$X(z) = \prod_{i=1}^{N} Y_i(z^{c_i}). \tag{4.18}$$

Now we consider the case where the discrete random variable X is the sum of N nonnegative discrete random variables Y_i ($i \in \{1, \ldots, N\}$) which are independent and identically distributed (*iid*) random variables with common pgf $Y(z)$. We assume that the number N of random variables in the sum is in fact a nonnegative discrete random variable with pgf $N(z)$ and that this random variable is independent of the random variables $Y_1, \ldots, Y_N$. The calculation of the pmf of X then becomes

$$\begin{aligned}\text{Prob}[X = m] = \sum_{n=0}^{\infty} \text{Prob}[N = n] \sum_{m_1,\ldots,m_{n-1}=0}^{\infty} \left(\prod_{i=1}^{n-1} \text{Prob}[Y_i = m_i] \right) \\ \left(\text{Prob}\left[Y_n = m - \sum_{i=1}^{n-1} m_i \right] \right). \quad (4.19)\end{aligned}$$

For the calculation of the pgf $X(z)$ on the other hand, we can exploit the law of total expectation and get

$$\begin{aligned}X(z) &= \sum_{n=0}^{\infty} \text{Prob}[N = n] \, \text{E}\left[z^{\sum_{i=1}^{N} Y_i} \middle| N = n \right] \\ &= \sum_{n=0}^{\infty} \text{Prob}[N = n] \, Y(z)^n = N(Y(z)). \quad (4.20)\end{aligned}$$

4.7 Tail distribution

The previous section has shown that when a random variable is a linear combination of other random variables, it can be quite beneficial to analyze its distribution using pgfs rather than pmfs. Not only can this approach yield a neater analysis, but the moment generating property allows us to find the moments of the random variable quite easily from its pgf. Some interesting performance measures however, rely on the pmf of a random variable of which the pgf has been calculated. In such cases, we could invoke the probability generating property, or a numerical approximation technique as mentioned above. If we are only interested in the tail distribution of a random variable X, i.e. the portion of the distribution of X where X has large values ($\text{Prob}[X = n]$ for large n), we can also resort to the complex residue technique, which is known for its rather accurate results [13, 14]. The tail distribution is especially interesting if we want to determine the probability that a random variable X exceeds a certain threshold X_T, where X_T is rather large.

The complex residue technique is based on the fact that the pgf $X(z)$ of X is actually the z-transform of the corresponding pmf $x(n)$. In particular, based on the inversion formula for z-transforms and assuming for the sake of argument that $X(z)$ is meromorphic, it follows that $x(n)$ can be written as a weighted sum of negative nth powers of the poles of $X(z)$. Due to the fact that $X(z)$ is a pgf and thus converges for all z with modulus $|z| \leq 1$, all poles of $X(z)$ are outside of the unit circle and the contribution of the pole of $X(z)$ with the smallest modulus (assuming there is only one such pole) will have the most impact on $x(n)$. This pole is therefore referred to as the *dominant pole*, noted z_d. Furthermore, in order to guarantee nonnegative values for $x(n)$, the dominant pole must be real and positive [14]. In general, this dominant pole can not be calculated analytically, numerical techniques such as the Newton-Raphson method or the Illinois [30] method on the other hand do provide the possibility to obtain z_d. Once the dominant pole z_d is found and if $X(z)$ has only a single dominant pole z_d of multiplicity 1, the tail distribution of X can be approximated as

$$\mathrm{Prob}[X = n] \approx -\theta {z_d}^{-n-1}, \qquad \text{for sufficiently large } n, \tag{4.21}$$

where θ is the complex residue of $X(z)$ for $z = z_d$:

$$\theta \triangleq \mathrm{Res}_{z_d} X(z) = \lim_{z \to z_d} (z - z_d)\, X(z). \tag{4.22}$$

The probability for X to exceed a certain threshold X_T can then be approximated as

$$\mathrm{Prob}[X > X_T] \approx -\sum_{n=X_T+1}^{\infty} \theta {z_d}^{-n-1} = -\frac{\theta {z_d}^{-X_T-1}}{z_d - 1}. \tag{4.23}$$

Chapter 5

The $GI - GI - 1$ Model

In this section, we will present and analyze the $GI - GI - 1$ model, i.e. a queueing model where both the arrival process and the server process are *iid*. This means that the numbers of arrivals during subsequent slots are statistically independent, and the same goes for the service times of subsequent packets. Note that the abbreviation GI stands for *general and independent*.

In the sections that follow, we will use this system as a base on which we will build more complex systems, by taking out specific features and replacing them with more advanced alternatives. Therefore, any notation, assumption, . . . in the analysis below, will be adopted in all of the next sections, unless otherwise stated. Analysis of this $GI - GI - 1$ model will allow us to present some basic techniques of discrete-time queueing theory and will serve as a roadmap for future analyses.

Note that the analysis presented here follows and summarizes the analysis presented in [11].

5.1 Model description

We consider a synchronized discrete-time queueing system with infinite storage capacity and one service unit. A graphical representation of this model is depicted in Figure 5.1. Packets arrive according to an arrival process defined by the pgf $A(z)$ of the number of arriving packets per slot. We assume the arrival process to be *iid*, such that the number of arrivals in

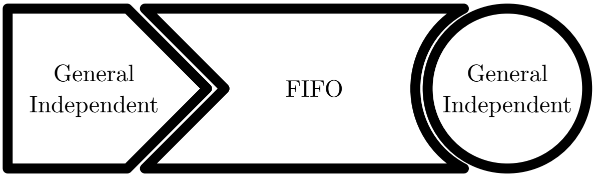

Figure 5.1: Illustration of the $GI - GI - 1$ model.

one slot does not affect the number of arrivals in another slot. We define $\lambda \triangleq A'(1)$, i.e. the mean number of arrivals in a random slot, also referred to as the *arrival rate*. The actual number of arrivals during a random slot k is denoted as a_k.

The system operates under the FIFO policy, such that packets are stored in arrival order in a buffer with infinite capacity, until they are pulled out by the server. The number of slots required to successfully serve a packet is dictated by the server process with pgf $S(z)$. Like the arrival process, the server process is *iid* as well, such that service times of different packets are statistically independent. The mean length of a service is denoted as $\mu = S'(1)$ and when inverted, it yields the *service rate*, i.e. $1/\mu$ is the mean number of packets transmitted in a single slot.

The load of the system, defined as $\rho \triangleq \lambda\mu$, represents the ratio of the speed at which packets are inserted in the queue versus the speed at which packets can leave the queue. For the system to be *stable*, i.e. to obey the equilibrium condition, it is required that $\rho < 1$, such that in the long run, the output line can handle all arriving packets. For $\rho \geq 1$, the queue would eventually continue to grow without ever being depleted completely. Packets would then face ever increasing delays, such that the system would appear to be clogged.

As mentioned in 1.4, services are synchronized to slot boundaries, i.e. they start at the beginning of a slot and last an integral and strictly positive number of slots. If no service is started at the beginning of a slot (e.g. because the system is empty at the start of the slot), the server will be idle during the entire slot.

5.2 Markovian state description

The goal of the analysis, is to determine the distribution of the delay experienced by the packets in the system. In order to do so, we will require specific information about the condition of the system at certain points in time. This information is stored in random variables that constitute a vector for each of these epochs, the system state vector. This vector should be Markovian, i.e. given the vector $\underline{\mathbf{v}}_k$ at an epoch k, the distribution of the variables in the subsequent vector $\underline{\mathbf{v}}_{k+1}$ is independent of the preceding vectors $\underline{\mathbf{v}}_{k-1}, \underline{\mathbf{v}}_{k-2}, \ldots$.

In case of the $GI-GI-1$ model, we will require the number of packets u_k in the system at the beginning of a random slot k, as well as the number of slots h_k left until service completion of the packet in the server at the beginning of slot k. If there is no packet in service at the beginning of slot k, i.e. the system is empty at that point (or $u_k=0$), we impose that $h_k=0$. The vector $\langle h_k, u_k\rangle$ then forms the system state vector at the beginning of slot k.

Note that if $h_k=1$, the packet in service will leave the system at the end of slot k and the server will attempt to pull a new packet from the queue. If $h_k \neq 1$, no departure will take place, and the queue content will simply grow according to the arrivals during slot k. These considerations can be formulated as:

- if $h_k=0$:

$$\begin{aligned} h_{k+1} &= \begin{cases} 0, & \text{if } a_k = 0, \\ s, & \text{if } a_k > 0, \end{cases} \\ u_{k+1} &= a_k, \end{aligned} \tag{5.1}$$

- if $h_k=1$:

$$\begin{aligned} h_{k+1} &= \begin{cases} 0, & \text{if } u_k = 1 \text{ and } a_k = 0, \\ s, & \text{if } u_k > 1 \text{ or } a_k > 0, \end{cases} \\ u_{k+1} &= u_k - 1 + a_k, \end{aligned} \tag{5.2}$$

- if $h_k>1$:

$$\begin{aligned} h_{k+1} &= h_k - 1, \\ u_{k+1} &= u_k + a_k. \end{aligned} \tag{5.3}$$

In these equations, the variable s was used to represent a sample of the service time distribution. These equations are called the *system equations*, as they describe the system state evolution from slot to slot.

5.3 Buffer analysis

Let us now consider the distribution of the system state vector at the beginning of a random slot k. More specifically, we define $P_k(x,z)$ as the joint pgf of the system state at the beginning of slot k as

$$P_k(x,z) \triangleq \mathrm{E}\left[x^{h_k} z^{u_k}\right], \tag{5.4}$$

Given that an empty system at the beginning of slot k implies that $h_k = 0$, we have that

$$P_k(x,0) = P_k(0,0) = p_{0,k}, \quad \forall x, \tag{5.5}$$

where we introduced $p_{0,k}$ to denote the probability for the system to be empty at the beginning of slot k.

Exploiting the system equations derived earlier, we can calculate the joint system state pgf $P_{k+1}(x,z)$ at the beginning of slot $k+1$ as

$$\begin{aligned} P_{k+1}(x,z) &\triangleq \mathrm{E}\big[x^{h_{k+1}} z^{u_{k+1}}\big] \\ &= \mathrm{E}\big[x^{h_{k+1}} z^{a_k} \{h_k = 0\}\big] + \mathrm{E}\big[x^{h_{k+1}} z^{u_k - 1 + a_k} \{h_k = 1\}\big] \\ &\quad + \mathrm{E}\big[x^{h_{k+1}} z^{u_k + a_k} \{h_k > 1\}\big] \\ &= P_k(0,0)\left(A(0) + S(x)\,\mathrm{E}[z^{a_k} \{a_k > 0\}]\right) \\ &\quad + A(0)\mathrm{Prob}[h_k = 1, u_k = 1] \\ &\quad + S(x)\,\mathrm{E}\big[z^{u_k - 1 + a_k} \{h_k = 1, u_k - 1 + a_k > 0\}\big] \\ &\quad + A(z)\,\mathrm{E}\big[x^{h_k - 1} z^{u_k} \{h_k > 1\}\big], \end{aligned} \tag{5.6}$$

In our further analysis, the case where $h_k = 1$, plays a special role, therefore we introduce the partial pgf $Q_k(z)$ as

$$Q_k(z) \triangleq \mathrm{E}\big[z^{u_k - 1} \{h_k = 1\}\big] = \sum_{n=1}^{\infty} \mathrm{Prob}[h_k = 1, u_k = n]\, z^{n-1}, \tag{5.7}$$

such that

$$Q_k(0) = \mathrm{Prob}[h_k = 1, u_k = 1]. \tag{5.8}$$

This definition helps us to determine $P_{k+1}(x,z)$ as

$$\begin{aligned} P_{k+1}(x,z) &= A(0)\left(1 - S(x)\right) P_k(0,0) + S(x)A(z)P_k(0,0) + A(0)Q_k(0) \\ &\quad + S(x)\left(A(z)Q_k(z) - A(0)Q_k(0)\right) \\ &\quad + \frac{A(z)}{x}\left(P_k(x,z) - xzQ_k(z) - P_k(0,0)\right) \\ &= A(0)\left(1 - S(x)\right)\left(P_k(0,0) + Q_k(0)\right) + A(z)\frac{xS(x) - 1}{x} P_k(0,0) \\ &\quad + A(z)\left(S(x) - z\right) Q_k(z) + \frac{A(z)}{x} P_k(x,z). \end{aligned} \tag{5.9}$$

This equation expresses the system state pgf at the beginning of slot $k+1$ in terms of expressions belonging to slot k. Assuming we would know the initial system state distribution, i.e. at the beginning of slot $k = 0$, iteration of (5.9) would allow us to determine the system state distribution at the beginning of any subsequent slot. However, we are not concerned with the system state distribution at the beginning of a certain slot, rather we want to investigate the steady-state behavior of the system.

The steady-state counterparts of the expressions derived earlier can be found by taking the limit for $k \to \infty$. Note that this implies that corresponding expressions for k and $k+1$ will converge. In what follows, we will leave out the time index k for expressions in steady state. Specifically for the system state, the pgfs $P_k(x,z)$ and $P_{k+1}(x,z)$ will converge into $P(x,z)$, which can be calculated as

$$\begin{aligned} P(x,z) &\triangleq \lim_{k\to\infty} P_k(x,z) = \lim_{k\to\infty} P_{k+1}(x,z) \\ &= \frac{1}{x-A(z)} \left[A(0)x\left(1-S(x)\right)\left(p_0+Q(0)\right)\right. \qquad (5.10) \\ &\qquad \left. +A(z)\left(xS(x)-1\right)p_0 + xA(z)\left(S(x)-z\right)Q(z)\right], \end{aligned}$$

with $Q(z)$ and p_0, the steady-state counterparts of $Q_k(z)$ and $p_{0,k}$ respectively. Note that (5.10) still contains two unkown parameters $Q(0)$ and p_0 and that the function $Q(z)$ is yet undetermined. These will be calculated in the remainder of this section.

First, we note that the property in (5.5) also translates to its steady-state counterpart. This leads to two separate methods to obtain $P(1,0)$

$$P(1,0) = P(0,0) = p_0, \qquad \text{steady-state counterpart of (5.5)}, \qquad (5.11)$$

$$= \frac{A(0)Q(0)}{1-A(0)}, \qquad \text{from (5.10)}, \qquad (5.12)$$

such that

$$p_0 = A(0)\left[p_0 + Q(0)\right]. \qquad (5.13)$$

The next step involves the property of pgfs that states that all pgfs are bounded when all arguments are in the unit disk. More specifically, $P(x,z)$ should be bounded for $x = A(z)$ with $|z| \leq 1$. Note that $|z| \leq 1$ is a sufficient condition such that $|A(z)| \leq 1$ since $A(z)$ is a pgf. Substitution in (5.10) would however cause the denominator to become 0, leading to an unbounded result, unless of course the numerator is 0 as well and de l'Hôpital's rule can be applied. Expressing that the numerator of (5.10) becomes 0 for $x = A(z)$ allows us to determine $Q(z)$ as

$$Q(z) = \frac{S(A(z))\left(A(z)-1\right)}{A(z)\left(z-S(A(z))\right)} p_0. \qquad (5.14)$$

Finally, we determine p_0 from the normalization property of pgfs that states that any pgf should return 1 when all arguments are equal to 1. Specifically we first determine $P(1,z)$ as

$$P(1,z) = \frac{A(z)\left(1-z\right)Q(z)}{1-A(z)}, \qquad (5.15)$$

such that, again using de l'Hôpital's rule

$$P(1,1) = \frac{Q(1)}{\lambda} = 1. \qquad (5.16)$$

Note that $Q(z)$ is a partial pgf and therefore does not satisfy the normalization property. Rather, we can determine $Q(1)$ from (5.14) as

$$Q(1) = \frac{\lambda}{1-\lambda\mu} p_0. \tag{5.17}$$

Substitution of (5.17) in (5.16) and application of the normalization property $P(1,1) = 1$ then gives

$$p_0 = 1 - \lambda\mu. \tag{5.18}$$

Note that $Q(1) = \text{Prob}[h = 1]$ corresponds to the fraction of slots in which there is a departure, and thus $Q(1)$ can also be considered to be the actual departure rate, i.e. the mean number of departures in a single slot. This means that the actual average number of departures per slot is equal to the average number of arrivals per slot. As such, this illustrates the statement in Section 3.1.

Equations (5.13), (5.14) and (5.18) then allow us to transform (5.10) to the closed-form expression

$$P(x,z) = (1-\lambda\mu)\left[1 - xz\frac{(1-A(z))\,(S(x)-S(A(z)))}{(x-A(z))\,(z-S(A(z)))}\right]. \tag{5.19}$$

The pgf $U(z)$ of the system content at the beginning of a random steady-state slot can either be found by substitution of $x = 1$ in (5.19) or by substitution of (5.14) and (5.18) in (5.15) as

$$U(z) = (1-\lambda\mu)\left[1 - z\frac{1-S(A(z))}{z-S(A(z))}\right] = (1-\lambda\mu)\,S(A(z))\frac{z-1}{z-S(A(z))}. \tag{5.20}$$

The mean system content at the beginning of steady-state slots can then be found as described in Section 4.5. More specifically, we determine the first order derivative of $U(z)$ and evaluate it for $z \to 1$ such that after multiple applications of de l'Hôpital's theorem, we get

$$\text{E}[u] = U'(1) = \lambda\mu + \frac{A''(1)\mu + \lambda^2 S''(1)}{2\,(1-\lambda\mu)}. \tag{5.21}$$

This result is the discrete-time counterpart of the *Pollaczek-Khintchine* formula for determining the mean system content in an $M/G/1$ queue [71, 101].

Another interesting measure is the unfinished work w at the beginning of a random steady-state slot. This is the total number of slots needed to remove all u packets, present at the beginning of that particular slot, from the system. If the system is already empty at the beginning of that slot, of course $w = 0$. These considerations yield

$$w = \begin{cases} 0, & \text{if } u = 0, \\ h + \sum_{i=1}^{u-1} s_i, & \text{if } u > 0, \end{cases} \tag{5.22}$$

where the s_i's denote the service times of the $u-1$ packets in the queue. The pgf $W(z)$ of the random variable w can then be found as

$$\begin{aligned} W(z) &\triangleq \mathrm{E}[z^w] = \mathrm{Prob}[u=0] + \mathrm{E}[z^w \{u>0\}] \\ &= p_0 + \mathrm{E}\Big[z^h S(z)^{u-1}\{u>0\}\Big] = p_0 + \frac{P(z,S(z))-p_0}{S(z)} \\ &= (1-\lambda\mu)\,A(S(z))\frac{z-1}{z-A(S(z))}. \end{aligned} \tag{5.23}$$

5.4 Packet delay analysis

We can now proceed to determine the packet delay distribution for packets that arrive during an arbitrary slot in the steady state. The delay d of a random steady-state packet $\mathcal{P}$ that arrives to the system in the course of slot $\mathcal{S}$ is defined as the integer number of slots starting immediately after $\mathcal{S}$, up until the end of the slot during which $\mathcal{P}$ leaves the system. This delay consists of three parts:

- the unfinished work $(w_{\mathcal{S}}-1)^+ \triangleq \max(0, w_{\mathcal{S}}-1)$ at the end of slot $\mathcal{S}$, related to the packets in the system at the beginning of $\mathcal{S}$;
- the total service time of all $\chi_{\mathcal{P}}$ packets that have arrived during slot $\mathcal{S}$ as well, but are to be served before $\mathcal{P}$;
- the service time of $\mathcal{P}$ itself.

The delay of $\mathcal{P}$ can then be found as

$$d = (w_{\mathcal{S}}-1)^+ + \sum_{i=1}^{\chi_{\mathcal{P}}+1} s_i, \tag{5.24}$$

where the s_i's now denote the service times of the packets mentioned in the two last items of the enumeration above.

Note that $\mathcal{S}$ is *not* a random slot; in fact it is the arrival slot of the randomly chosen packet $\mathcal{P}$. At the very least this implies that the number of arrivals $a_{\mathcal{S}}$ during $\mathcal{S}$ must be greater than or equal to 1. Moreover, the probability that a randomly chosen packet arrives during a slot with α arrivals in total is proportional to the number of arrivals α in that slot. Therefore we find the distribution of $a_{\mathcal{S}}$ as

$$\mathrm{Prob}[a_{\mathcal{S}}=\alpha] = \frac{\alpha}{\lambda}\mathrm{Prob}[a_k=\alpha]\,, \qquad \alpha \geq 1. \tag{5.25}$$

Given that packet arrivals are *iid*, the BASTA-property holds and the system state distribution at the beginning of $\mathcal{S}$ is stochastically identical to the system state distribution at the beginning of a truly random steady-state slot. As a result, the distribution of $w_{\mathcal{S}}$ in turn is identical to that of w. Given the fact that $\mathcal{P}$ was chosen randomly, its position within all $a_{\mathcal{S}}$

arrivals during $\mathcal{S}$ is uniformly distributed and the pgf $X(z)$ of $\chi_{\mathcal{P}}$ can be found (see e.g. [90]) from

$$\text{Prob}[\,\chi_{\mathcal{P}} = n|\, a_{\mathcal{S}} = \alpha] = \frac{1}{\alpha}, \qquad n \geq 0,\ \alpha \geq 1, \tag{5.26}$$

as

$$\begin{aligned} X(z) &\triangleq \text{E}[z^{\chi_{\mathcal{P}}}] = \sum_{n=0}^{\infty} \text{Prob}[\chi_{\mathcal{P}} = n]\, z^n \\ &= \sum_{\alpha=1}^{\infty} \text{Prob}[a_{\mathcal{S}} = \alpha] \sum_{n=0}^{\alpha-1} \text{Prob}[\,\chi_{\mathcal{P}} = n|\, a_{\mathcal{S}} = \alpha]\, z^n \\ &= \frac{1}{z-1} \sum_{\alpha=1}^{\infty} \frac{1}{\alpha} \text{Prob}[a_{\mathcal{S}} = \alpha]\,(z^{\alpha} - 1) = \frac{A(z) - 1}{\lambda\,(z-1)}, \end{aligned} \tag{5.27}$$

with mean

$$\text{E}[\chi_{\mathcal{P}}] = X'(1) = \frac{A''(1)}{2\lambda}. \tag{5.28}$$

The pgf $D(z)$ of the packet delay can then be found as

$$\begin{aligned} D(z) &\triangleq \text{E}\big[z^d\big] = \text{E}\Big[z^{(w_{\mathcal{S}}-1)^+}\Big]\, \text{E}\Big[S(z)^{\chi_{\mathcal{P}}+1}\Big] \\ &= \frac{W(z) + (z-1)\, W(0)}{z} S(z) X(S(z)) \\ &= (1 - \lambda\mu)\, S(z) \frac{(z-1)\,(1 - A(S(z)))}{\lambda\,(1 - S(z))\,(z - A(S(z)))}. \end{aligned} \tag{5.29}$$

The mean packet delay can be determined by evaluation of the first derivative of (5.29) for $z = 1$. After multiple applications of de l'Hôpital's theorem, we get

$$\text{E}[d] = D'(1) = \mu + \frac{A''(1)\mu + \lambda^2 S''(1)}{2\lambda\,(1 - \lambda\mu)}. \tag{5.30}$$

Application of Little's theorem (3.1) confirms the validity of this expression. Similar to (5.21), (5.30) is the discrete-time version of the Pollaczek-Khintchine formula for the expected delay in an $M/G/1$ queue.

Finally, we will approximate the tail distribution of the packet delay using the complex residue technique. It can be shown that the dominant pole z_d of the packet delay pgf $D(z)$ must be a zero of $z - A(S(z))$, such that $z_d = A(S(z_d))$. The complex residue θ of $D(z)$ for $z = z_d$ can then be found as

$$\begin{aligned} \theta &\triangleq \text{Res}_{z_d} D(z) = \lim_{z \to z_d} (z - z_d)\, D(z) \\ &= \frac{(1 - \lambda\mu)\, S(z_d)(z_d - 1)^2}{\lambda\,(S(z_d) - 1)\,(1 - A'(S(z_d))S'(z_d))}. \end{aligned} \tag{5.31}$$

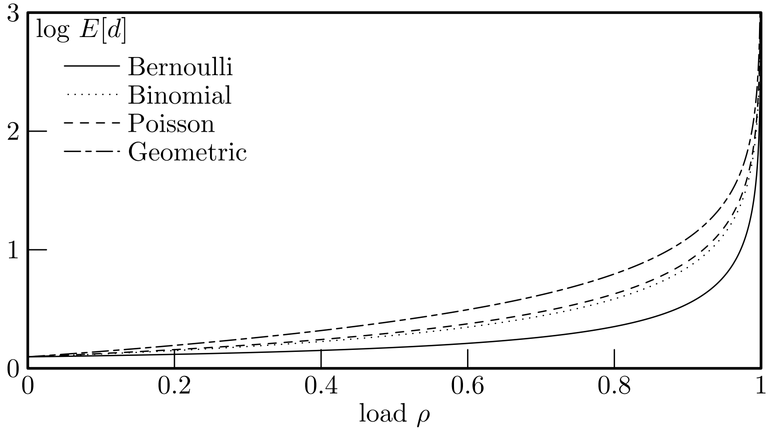

Figure 5.2: The mean packet delay vs. the system load ρ for various arrival distributions.

The tail probability of the packet delay d for sufficiently large values of n can then be approximated as

$$\text{Prob}[d = n] \approx -\theta {z_d}^{-n-1}. \tag{5.32}$$

5.5 Numerical examples

In the previous sections, we have calculated analytical results for some common performance measures, such as the expected value and the tail distribution of the packet delay. In this section, we illustrate these results by means of graphs in order to show the implications of the results calculated in the above.

First, we will investigate the mean packet delay $\text{E}[d]$. From (5.30), we see that the mean packet delay is influenced by the first and second order moments of both the packet arrival distribution and the service distribution. Intuitively one can assume that an increase in the mean and the variance of the number of arrivals per slot will lead to an increase in the mean packet delay and we can see this intuitive relation reflected in (5.30), although somewhat obfuscated.

To make matters more clear, we start off by investigating the impact of various packet arrival distributions on the mean packet delay. Therefore we consider four different distributions for the number of packet arrivals a_k in a random slot k: a Bernoulli distribution, a binomial distribution $B(n, p)$ with $n = 5$, a Poisson distribution and a geometric distribution. In Figure 5.2, the mean packet delay $\text{E}[d]$ is plotted for each of the packet arrival distributions as a function of the system load $\rho = \lambda\mu$. The service

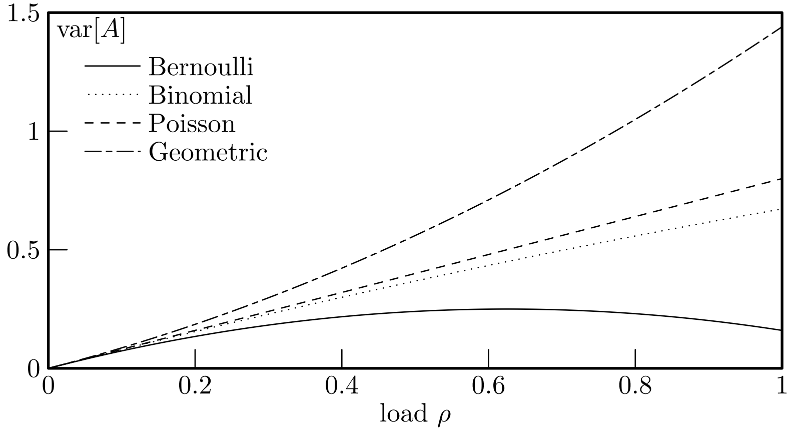

Figure 5.3: The variance of the arrival distributions vs. the system load.

times have a shifted geometric distribution with parameter 0.2, such that the expected service time is $\mu = 1.25$ slots per packet. The packet arrival rate is then for each value of ρ obtained as $\lambda = \rho/\mu$ and the parameters of the packet arrival distributions are adjusted in accordance. Figure 5.2 shows that for each of the packet arrival distributions, the mean packet delay E[d] increases as the system load increases. This increase in the mean packet delay is gently at first, but when the system load nears 1, the mean packet delay grows excessively, as the system approaches instability. For a fixed value of the system load, and thus a fixed value of the arrival rate λ, we can see that the Bernoulli distributed arrival process yields the lowest mean packet delay, whereas the geometrically distributed arrival process yields the highest mean packet delay.

This discrepancy can be explained by looking at the variance of the packet arrival distributions, as depicted in Figure 5.3 for the same system configuration. It can be shown - even symbolically - that for a fixed value of $\lambda \in]0, 1]$, the considered arrival distributions have a fixed order when ordered according to increasing values of their variance, namely: the Bernoulli dsitribution, the binomial distribution, the Poisson distribution and finally the geometric distribution. This is reflected in Figure 5.3, although the order only becomes clear when the curves start to fan out. Comparing Figure 5.2 with Figure 5.3 the effect of the difference in the arrival process variance on the mean packet delay becomes manifest. The observation that an increase in variance of the arrival process leads to an increase of the mean packet delay is in fact a typical result in queueing theory.

Next, we will discuss the impact of various service time distributions on the mean packet delay. In this case, we consider two standard service time distributions, more specifically a shifted Poisson distribution and a shifted

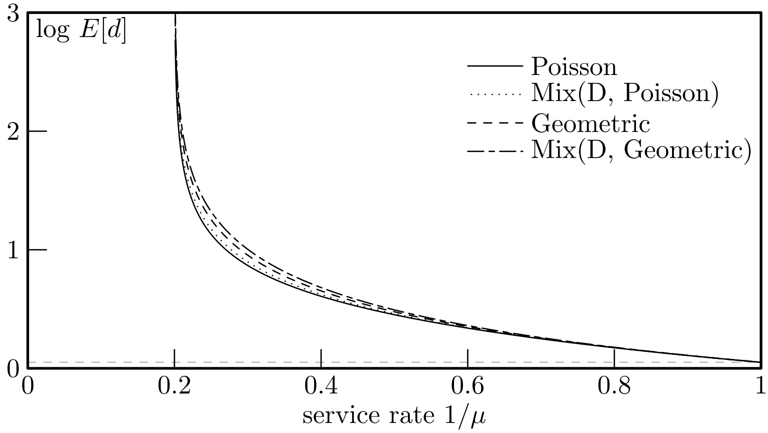

Figure 5.4: The mean packet delay vs. the service rate $1/\mu$ for various service time distributions.

geometric distribution, as well as two mixtures of a degenerate distribution and a shifted Poisson or geometric distribution. The weight of the degenerate distributions with pgf $S_{\text{degenerate}}(z) = z$ in both mixtures is equal to 0.25. Figure 5.4 shows the mean packet delay E$[d]$ for each of the service time distributions versus the service rate $1/\mu$. The number of packet arrivals per slot is Poisson distributed with arrival rate $\lambda = 0.2$ packets per slot. The equilibrium condition is illustrated by both the void for $1/\mu <= 0.2$ and the exorbitant values of the mean packet delay when the service rate drops and approaches 0.2. When the service rate further increases, the mean packet delay decreases and approaches $1+\frac{A''(1)}{2\lambda(1-\lambda)}$ when the service rate approaches 1. This minimal value is plotted in Figure 5.4 as a dashed gray line and corresponds to the mean packet delay in a system with service times of exactly 1 slot per packet. Similar to the previous charts, the difference in variance between the service time distributions causes the curve corresponding to the shifted Poisson service times to be lower than the curve corresponding to the shifted geometric service times.

Finally, we take a look at the packet delay tail distribution. As can be expected from (5.32), the approximation yields a linear curve when plotted on a logarithmic scale. More specifically, the decay rate of the packet delay is given by $1/z_d$, such that the downward slope of the curve will be equal to $-\log(z_d)$.

First, we show the effect of the packet arrival rate on the tail distribution of the packet delay. In Figure 5.5, we show the pmf of the packet delay on a logarithmic scale for a $GI - GI - 1$ model with Poisson distributed packet arrivals and shifted geometric service times with service rate $\mu = 1.25$. The different curves plotted in Figure 5.5 correspond to different values of the

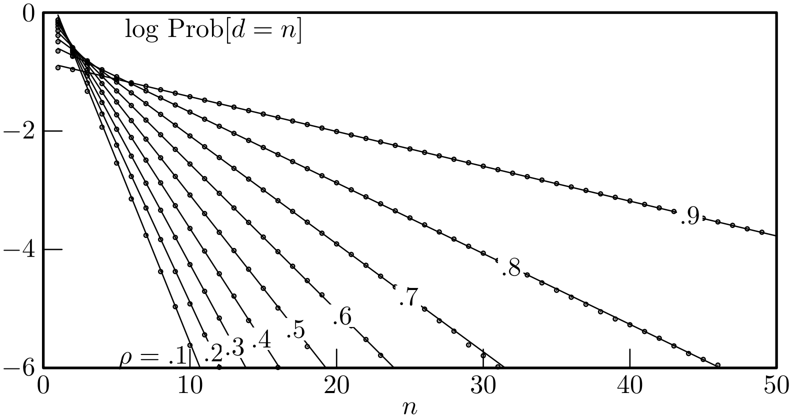

Figure 5.5: The packet delay pmf, simulated (dots) and approximated (lines) using (5.32) for various values of the system load ρ.

system load ρ and a corresponding arrival rate of $\lambda = \rho/\mu$. For low values of the system load, the packet delay decay rate is rather large, implicating that the probability Prob[$d = n$] for the delay to span n slots rapidly decreases when n increases, resulting in a steep curve. In practice, this means that long delays are rare and most packets will experience only a short delay. When the system load on the other hand is high, the decay rate is small, and long delays become abundant.

The effect of the packet service rate is depicted in Figure 5.6, where the packet delay pmf is plotted on a logarithmic scale for a system with Poisson distributed arrivals with rate $\lambda = 0.2$ and shifted geometric service times, for various values of the service rate. As a higher service rate corresponds to smaller service times, we see that the slope of the curves increases when the service rate increases.

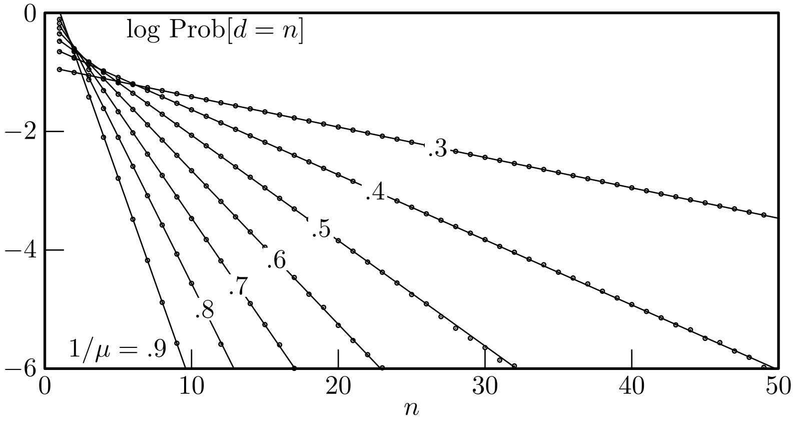

Figure 5.6: The packet delay pmf, simulated (dots) and approximated (lines) using (5.32) for various values of the service rate $1/\mu$.

Part II

Generalized Service Mechanisms

Chapter 6

The Reservation Discipline

6.1 Introduction

Although modern packet-based communication networks support an increasingly diverse spectrum of applications, each with their own set of Quality of Service (QoS) requirements, network traffic can roughly be divided into two types, according to its delay sensitivity. On one side we have delay-sensitive traffic, usually generated by real-time applications (e.g. multimedia streaming, telephony, gaming, . . .), characterized typically by stringent requirements in terms of mean delay and delay variance (also known as *jitter*), but rather tolerant to packet loss. Conversely, we also have delay-tolerant traffic (e.g. www, ftp, e-mail, . . .), that can withstand higher delays and jitter, but in turn is less resilient to packet loss. As both types of traffic usually have to share network resources, the challenge thus arises to manipulate the network traffic such that both types of traffic can coexist, with delay-sensitive traffic streams meeting their delay requirements without crippling delay-tolerant traffic.

Over the years, many solutions have been proposed to tackle this issue, varying in complexity and efficiency, with the most intuitive being the *Absolute Priority* (AP) scheduling discipline. In a queueing system operating under the AP discipline, traffic is divided into two classes according to its priority, and the low-priority packets are only served if no high-priority packets are present in the queue. Due to its intuitivity, AP has been researched extensively, both for uncorrelated packet arrivals (e.g. [87, 92, 106, 113, 115]) and for correlated arrival processes (e.g. [4, 64, 112]). Several methodologies have been used to study the effects of the AP discipline, such as the *Supplementary Variables* technique [113], the *Large Deviations* method [64] and *matrix-analytic* methods [67]. The main drawback of AP is that in a system operating under AP, low-priority traffic can be completely obstructed in case of a high load of high-priority traffic. This effect is called *starvation* of the low-priority traffic. Variations on the AP scheduling discipline aimed at overcoming this problem, have been proposed and studied. In [108], a probabilistic priority scheme is discussed that assigns a small probability to each class by which the server may provide service to a lower-priority class, even if higher-priority packets are available. Another idea is to allow packets to be promoted to a higher-priority class under certain well-specified conditions, or conversely, to degrade to a lower class [80, 86]. In [22], time is divided in fixed frames and a 'frame-bound' priority for a certain class is established by favorably reordering the packets that arrived within the same frame. Priority systems usually are either strictly preemptive (i.e. the service to low-priority packets is cancelled or interrupted as soon as a high-priority packet is available) or strictly nonpreemptive (in which case service in progress to a lower-priority packet is allowed to be completed before service to a higher-priority packet is started). In [73, 116] however a hybrid method is studied where a lower-priority service is only preempted by an incoming higher-priority packet if the service of the former did not advance beyond a certain threshold. The scheduling disciplines mentioned above are all aimed at a hierarchically differentiated service, where a certain class of traffic is granted explicit priority over other packet stream classes. Other popular approaches to differentiated service are *proportional-rate* (so-called *fair-queueing*) and *proportional-delay* methods.

The goal of proportional-rate schedulers is to guarantee for each traffic class a pre-defined fraction of the available output bandwidth. Implementations usually have a separate queue for each class and the server visits these queues periodically or according to a more involved rule. *Generalized Processor Sharing* (GPS) achieves optimality in this respect, but this is a theoretical approach which assumes the work in the queue to be infinitely divisible. A suboptimal but practical approximation of GPS that can handle atomic packets of a certain size is *Weighted Fair Queuing* (WFQ). Although WFQ allows for a realistic implementation, it is computationally rather complex since it needs to recalculate virtual finish times for all queues each time a packet is either enqueued or dequeued somewhere. Many modifications exist: *Start-time Fair Queueing* (SFQ), *Self-clocked Fair Queueing* (SCFQ), *Worst-case Fair Weighted Fair Queueing* (WF^{2}Q), *Frame-based Fair Queueing* (FFQ), *Weighted Round-Robin* (WRR), *Deficit Round-Robin* (DRR), all of which are widely used and have their specific advantages in certain situations [55, 103]. The theoretical performance of these complicated algorithms however is poorly understood, although there is a large body of research on polling systems [79, 111], their queueing-theoretic counterpart.

Recently, some studies have also been performed on the delay differentiation of probabilistic implementations of GPS in a discrete-time setting. In [18], the server selects a queue to pull a packet from with a probability depending on both queue levels. A more in-depth analysis in case of a fixed probability of choosing either queue is given in [114].

Unlike proportional-rate scheduling, proportional-delay scheduling is concerned with fairly sharing the queueing delay incurred to the packet streams [119, 123]. Related to proportional-delay methods are the deadline-based disciplines, but these are not class-based strictly speaking. Deadline-based systems require each packet to have a strict deadline by which that packet should have been transmitted. Here, the *Earliest Deadline First* (EDF) scheduler is known to achieve the smallest possible overall lateness of the packets. Many specific scheduling disciplines have been proposed and studied which combine elements of rate, delay, queue and even loss fairness in order to achieve specific objectives in terms of service differentiation [65, 107]. For example, the *Discriminatory Processor Sharing* (DPS) discipline continuously reevaluates the GPS weights in order to assign a higher rate to longer queues [56] or longer jobs [57].

The *Reservation*-based scheduling discipline, introduced by Burakowski and Tarasiuk in [15] and studied in this chapter, offers yet another approach towards priority scheduling. Under the Reservation discipline, a dummy packet referred to as a reservation serves as a placeholder for a future high-priority packet. When a high-priority packet is inserted, it takes the place of the reservation which is reinserted at the queue's tail afterwards. This easy-to-implement discipline thereby effectively prevents starvation of low-priority traffic. In the original paper [15], the mean delays of both traffic streams were roughly estimated by means of a continuous-time model with Poisson arrivals. The distribution of the packet delays in a discrete-time buffer with reservation-based scheduling and general uncorrelated arrivals has been studied in the case of constant packet service times equal to exactly one slot [25, 26]. This chapter reproduces my research on the Reservation discipline for general uncorrelated arrivals in case of geometric service times [36] and general independent service times [34, 35].

6.2 The Reservation discipline

The Reservation discipline achieves its priority scheduling by a combination of two mechanisms that both affect the way packets are inserted into the queue. Firstly, the Reservation discipline reorders all packets arriving in the same slot such that high-priority packets will be inserted into the queue before the low-priority packets. Secondly, after reordering the high-priority packets are not simply appended to the queue, rather they (one by one) replace a dummy packet $\mathcal{R}$, referred to as a reservation, which is then reinserted at the queue's tail. Therefore, the first high-priority packet to be

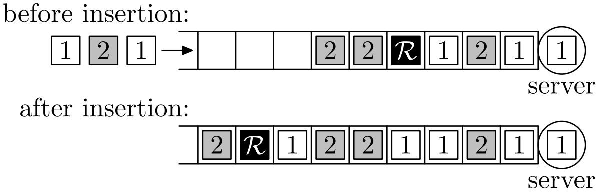

Figure 6.1: Example of how the Reservation discipline handles packet insertion.

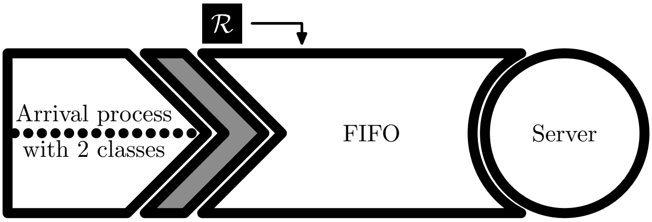

Figure 6.2: Illustration of the reservation system.

inserted at the end of a slot, can possibly jump over a large portion of the queue and since the reservation is immediately repositioned at the end of the queue, the remainder of the high-priority packets gain less profit. The low-priority packets are then inserted behind the reservation.

The net effect of this insertion mechanism is illustrated in Figure 6.1, where high-priority packets are displayed as white squares with label 1 and low-priority packets are depicted as gray squares with label 2. The reservation is shown as a black square with label $\mathcal{R}$.

Due to this twofold modification to the standard FIFO scheduling discipline, every low-priority packet can be outpaced by a high-priority packet, albeit only once. In general, this will present only a minor setback for the low-priority traffic, but can provide a nonnegligible gain for certain high-priority packets. The Reservation discipline can therefore be expected to have the most impact on the high-priority delay in cases where there is much low-priority traffic in relation to high-priority traffic. Note that, once inserted, the priority level of the actual data packets has no further significance for the queueing system.

6.3 Mathematical model

We reuse the $GI - GI - 1$ model, as described in Chapter 5, with the application of an alternate arrival process along with the insertion mechanism described above. In this chapter, the arrival process A consists of two sub-processes A_1 and A_2, where A_1 models the high-priority traffic and A_2

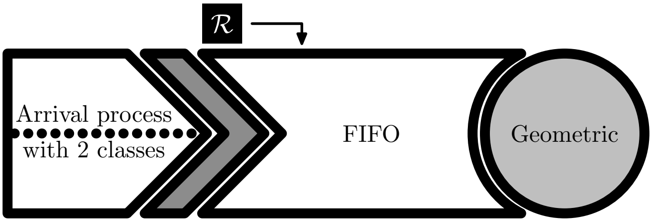

Figure 6.3: Illustration of the reservation system with geometric service times.

corresponds to low-priority traffic. This is depicted in Figure 6.2, where a block was added between the arrival process and the queue to indicate the Reservation discipline and the fact that it only interferes with how packets are inserted in the queue. Packets are classified according to the arrival process they originated from, class 1 referring to the high-priority traffic and class 2 being the low-priority traffic. The number of arrivals of class j during slot k is denoted by $a_{j,k}$ (where $j = 1, 2$); the total number of arrivals during slot k is then given by $a_{T,k} = a_{1,k} + a_{2,k}$. The numbers of arrivals of either class are supposed to be *iid* from slot to slot, but correlation between the arrivals of both classes during a single slot is accepted. This allows us to model the entire arrival process by the joint pgf

$$A(z_1, z_2) \triangleq \mathrm{E}[{z_1}^{a_{1,k}} {z_2}^{a_{2,k}}] \,. \tag{6.1}$$

Note that we omitted the time index k for the pgf $A(z_1, z_2)$, based on the assumption that the arrival process is *iid* from slot to slot. We can omit this index for the random variables $a_{1,k}$ and $a_{2,k}$ as well, but in contexts where we want to refer to specific slots, we will specify this index as a means of clarity. As a shorthand, we introduce the marginal pgfs $A_1(z)$ and $A_2(z)$ and the pgf $A_T(z)$ of the sum $a_{1,k} + a_{2,k}$ as

$$A_1(z) \triangleq A(z, 1), \qquad A_2(z) \triangleq A(1, z), \qquad A_T(z) \triangleq A(z, z). \tag{6.2}$$

The mean number of class-j arrivals is defined as $\lambda_j \triangleq A'_j(1)$, such that the total arrival rate is $\lambda_T \triangleq A'_T(1) = \lambda_1 + \lambda_2$.

We will analyze the model described above for geometric service times and general independent service times separately in the next two sections.

6.4 Geometric service times

In this section, we assume that the service times are distributed according to a shifted geometric distribution as illustrated in Figure 6.3. This implies

$$S(z) \triangleq \frac{\sigma z}{1 - (1 - \sigma)\, z}, \tag{6.3}$$

where $\sigma = 1/\mu$ $(0 < \sigma \leq 1)$ is the service rate.

6.4.1 System equations

In view of the delay analysis, the system state vector should contain information about the system content and the position of $\mathcal{R}$ at the beginning of a slot. Therefore, we introduce m_k as the position of $\mathcal{R}$ within the queue at the beginning of slot k. Note that we consider the server to be at position 0, such that m_k must always be strictly positive, with $m_k = 1$ denoting that $\mathcal{R}$ is at the head of the queue The system content u_k at the beginning of k must not include $\mathcal{R}$, since it is no data packet and is ever present in the queue. Following the definitions of m_k and u_k, we get that

$$m_k = 1, \text{ if } u_k = 0, \qquad \text{and} \qquad 1 \leq m_k \leq u_k, \text{ if } u_k > 0. \tag{6.4}$$

Note that, contrary to the $GI - GI - 1$ model, we do not include a random variable to monitor the progress of the active service. This is typical of systems with geometric service times, where packets in service have a fixed probability of continuing service. We therefore introduce a Bernoulli variable r_k that equals 1 with probability σ, corresponding to the end of a service; when r_k is equal to 0, a service in progress during slot k does not end in slot k, which occurs with probability $1 - \sigma$. The random variable r_k does not depend on the actual value of k and is controlled by the pgf $R(z) \triangleq 1 - \sigma + \sigma z$. As a shorthand we introduce the complementary pgf $\bar{R}(z) \triangleq \sigma + (1 - \sigma)\, z$.

Assuming we know the system state vector $\langle m_k, u_k \rangle$ at the beginning of slot k, we can construct its slot $k + 1$ counterpart as follows. If there is no class-1 arrival during slot k, any arriving class-2 packet will be appended to the queue, without affecting the position of $\mathcal{R}$. If there is at least one class-1 arrival during slot k, $\mathcal{R}$ is seized and reinserted at the tail of the queue. After all class-1 packets arriving in slot k have been inserted, any class-2 arrival will be appended to the queue. A departure only occurs at the end of slot k, if the system is not empty at the beginning of k and if the packet in service terminates its service at the end of slot k. These considerations lead to the following set of system equations:

$$m_{k+1} = \begin{cases} (m_k - 1 - r_k)^+ + 1, & \text{if } a_{1,k} = 0, \\ (u_k - r_k)^+ + a_{1,k}, & \text{if } a_{1,k} > 0, \end{cases} \tag{6.5}$$

$$u_{k+1} = (u_k - r_k)^+ + a_{1,k} + a_{2,k}, \tag{6.6}$$

with $(\cdot)^+ \triangleq \max(0, \cdot)$ as introduced before.

6.4.2 Buffer analysis

We start our buffer analysis by defining the joint pgf $P_k(y,z)$ of the system state vector at the beginning of slot k as

$$P_k(y,z) \triangleq \mathrm{E}\big[y^{m_k-1} z^{u_k}\big] . \tag{6.7}$$

We choose to add the -1 in the exponent of y such that $P_k(0,0)$ corresponds to the probability of having an empty system at the beginning of slot k, since $u_k = 0$ implies that $m_k = 1$. Additionally, $P_k(0,z)$ is the partial pgf of the system content at the beginning of slot k with $\mathcal{R}$ at the queue's head.

From the system equations and based on the fact that $(a_{1,k}, a_{2,k})$ are statistically independent of (m_k, u_k), we find the pgf $P_{k+1}(y,z)$ of the system state vector at the beginning of slot $k+1$ as

$$\begin{aligned}
P_{k+1}(y,z) &\triangleq \mathrm{E}\big[y^{m_{k+1}-1} z^{u_{k+1}}\big] \\
&= A_1(0)\,\mathrm{E}[z^{a_{2,k}} | a_{1,k} = 0] \\
&\qquad \cdot \Big(\sigma\,\mathrm{E}\Big[y^{(m_k-2)^+} z^{(u_k-1)^+}\Big] + (1-\sigma)\,\mathrm{E}\big[y^{m_k-1} z^{u_k}\big]\Big) \\
&\quad + \frac{1-A_1(0)}{y}\,\mathrm{E}[(yz)^{a_{1,k}} z^{a_{2,k}} | a_{1,k} > 0] \\
&\qquad \cdot \Big(\sigma\,\mathrm{E}\Big[(yz)^{(u_k-1)^+}\Big] + (1-\sigma)\,\mathrm{E}[(yz)^{u_k}]\Big) \\
&= \frac{A(0,z)}{yz}\big(\sigma y\,(z-1)\,p_{0,k} + \sigma\,(y-1)\,P_k(0,z) + \bar{R}(yz) P_k(y,z)\big) \\
&\quad + \frac{A(yz,z) - A(0,z)}{y^2 z}\big(\sigma\,(yz-1)\,p_{0,k} + \bar{R}(yz) P_k(1,yz)\big),
\end{aligned} \tag{6.8}$$

where $p_{0,k}$ denotes the probability of the system to be empty at the beginning of slot k. Note that

$$\begin{aligned}
p_{0,k} &\triangleq \mathrm{Prob}[u_k = 0, m_k = 1] = \mathrm{Prob}[u_k = 0] \\
&= P_k(0,0) = P_k(1,0) = P_k(y,0), \quad \forall y.
\end{aligned} \tag{6.9}$$

Assuming the equilibrium condition $\lambda_T < \sigma$ holds, the functions $P_k(y,z)$ and $P_{k+1}(y,z)$ for $k \to \infty$ converge to the same limiting function $P(y,z)$. Taking this limit of (6.8), we get the pgf $P(y,z)$ of the system state at the beginning of a random steady-state slot as

$$\begin{aligned}
P(y,z) &= \frac{A(0,z)}{yz}\big(\sigma y\,(z-1)\,p_0 + \sigma\,(y-1)\,P(0,z) + \bar{R}(yz) P(y,z)\big) \\
&\quad + \frac{A(yz,z) - A(0,z)}{y^2 z}\big(\sigma\,(yz-1)\,p_0 + \bar{R}(yz) P(1,yz)\big) \\
&= \frac{\sigma A(0,z)}{yz - \bar{R}(yz) A(0,z)}\left(y\,(z-1)\,p_0 + (y-1)\,P(0,z)\right) \qquad (6.10) \\
&\quad + \frac{A(yz,z) - A(0,z)}{y\,\big(yz - \bar{R}(yz) A(0,z)\big)}\big(\sigma\,(yz-1)\,p_0 + \bar{R}(yz) P(1,yz)\big) .
\end{aligned}$$

This result will be further expanded once we have obtained closed-form expressions for the unknowns p_0, $P(0,z)$ and $P(1,z)$.

The marginal pgf $U(z)$ of the system content u at the beginning of a random steady-state slot can be found as

$$\begin{aligned} U(z) &\triangleq \mathrm{E}[z^u] = P(1,z) \\ &= \frac{1}{z-\bar{R}(z)A(0,z)} \left[\sigma A_T(z)\,(z-1)\,p_0 + (A_T(z)-A(0,z))\,\bar{R}(z)U(z)\right] \\ &= \frac{(z-1)\,S(A_T(z))}{z-S(A_T(z))} p_0. \end{aligned} \tag{6.11}$$

The mean system content at the beginning of a random steady-state slot can then be found as

$$\mathrm{E}[u] = U'(1) = \lambda_T\mu + \frac{\lambda'_T\mu + \lambda_T^2\mu'}{2p_0} = \frac{\lambda_T}{\sigma} + \frac{\lambda'_T}{2\sigma p_0} + \frac{\lambda_T^2\,(1-\sigma)}{\sigma^2 p_0}, \tag{6.12}$$

where we introduced the shorthand notations $\lambda'_T \triangleq A''_T(1)$ and $\mu' \triangleq S''(1) = 2\,(1-\sigma)\,/\sigma^2$.

Note that the system content is not in any way affected by the Reservation discipline: $\mathcal{R}$ is not included in the system content and once inserted, no more distinction is made by the system between packets of either class. Therefore it could be expected that (6.11) and (6.12) would be similar to (5.20) and (5.21). Application of the normalization condition to (6.11) allows us to determine the empty system probability as

$$p_0 = 1 - \frac{\lambda_T}{\sigma} = 1 - \lambda_T\mu, \tag{6.13}$$

similar to the corresponding result in the $GI-GI-1$ model.

Substitution of (6.11) in (6.10) yields

$$\begin{aligned} P(y,z) = {} & \frac{\sigma A(0,z)}{yz-\bar{R}(yz)A(0,z)}\,(y\,(z-1)\,p_0 + (y-1)\,P(0,z)) \\ & + \frac{(yz-1)\,(A(yz,z)-A(0,z))}{y\,\big(yz-\bar{R}(yz)A(0,z)\big)}\left(\sigma + \bar{R}(yz)\frac{S(A_T(yz))}{yz-S(A_T(yz))}\right)p_0. \end{aligned} \tag{6.14}$$

For $y = \bar{R}(yz)A(0,z)/z$, the denominator in (6.14) becomes 0, presumably causing a singularity. After some calculations, this equality becomes $y = S(A(0,z))/z$. Assuming such values of y exist within the open unit disk for values of z also in the open unit disk (i.e. $|S(A(0,z))/z|\,,|z| < 1$), the numerator in (6.14) must become 0 as well, because $P(y,z)$ is a pgf and thus analytic for every y and z both in the open unit disk. Note that this assumption is realistic, as we show in Section 6.7, there always exists a non-empty subset $\aleph$ of the open unit disk such that

$$z \in \aleph \Rightarrow \left|\frac{S(A(0,z))}{z}\right| < 1. \tag{6.15}$$

This yields an additional relation that allows us to find $P(0,z)$ as

$$P(0,z) = \frac{p_0}{z - S(A(0,z))}\left[(z-1)\,S(A(0,z)) + z^2\frac{1-S(A(0,z))}{S(A(0,z))}\phi(z)\right], \tag{6.16}$$

where we introduced the shorthand

$$\phi(z) = \frac{A(0,z) - A(S(A(0,z)),z)}{A(0,z) - A_T(S(A(0,z)))}. \tag{6.17}$$

Finally, we can get a closed-form expression for $P(y,z)$ by substituting (6.16) in (6.14), resulting in

$$\begin{aligned} P(y,z) = \frac{p_0}{yz - S(A(0,z))}\Bigg[&(z-1)\,S(A(0,z))\frac{yz - S(A(0,z))}{z - S(A(0,z))} \\ &+ z\,(yz-1)\,\frac{S(A(0,z)) - S(A_T(yz))}{A(0,z) - A_T(yz)}\,\frac{A(yz,z) - A(0,z)}{yz - S(A_T(yz))} \\ &+ z^2\,(y-1)\,\frac{1 - S(A(0,z))}{z - S(A(0,z))}\phi(z)\Bigg]. \end{aligned} \tag{6.18}$$

6.4.3 Packet delay analysis

The actual goal of the Reservation discipline, is to decrease the mean delay for class-1 packets, without interfering too much with the mean class-2 packet delay. It therefore makes sense to analyze the packet delay for the two classes separately, by selecting a random steady-state class-j ($\in \{1,2\}$) packet $\mathcal{P}_j$ and investigating its delay d_j. Similar to the delay analysis in the $GI-GI-1$ model, we note that due to the *iid* nature of the arrival process, the BASTA-property holds. Thus although the arrival slot $\mathcal{S}$ of packet $\mathcal{P}_j$ is not a random slot, the system state distribution at the beginning of $\mathcal{S}$ is identical to the system state distribution at the beginning of a random steady-state slot, governed by the joint pgf $P(y,z)$.

In contrast, the numbers of arrivals $(a_{1,\mathcal{S}}, a_{2,\mathcal{S}})$ during slot $\mathcal{S}$ does not have the same distribution as the numbers of arrivals (a_1, a_2) during a random steady-state slot. Not only do we have that $a_{j,\mathcal{S}} \geq 1$, but the probability that the randomly selected packet $\mathcal{P}_j$ belongs to slot $\mathcal{S}$ is proportional to the number of class-j arrivals during $\mathcal{S}$. Similar to (5.25), we then find

$$\text{Prob}[a_{1,\mathcal{S}} = \alpha_1, a_{2,\mathcal{S}} = \alpha_2] = \frac{\alpha_j}{\lambda_j}\text{Prob}[a_1 = \alpha_1, a_2 = \alpha_2]\,. \tag{6.19}$$

Since all $a_{1,\mathcal{S}}$ class-1 packets will be inserted before any of the $a_{2,\mathcal{S}}$ class-2 packets, we only need to keep track of the relative position of $\mathcal{P}_j$ among all $a_{j,\mathcal{S}}$ class-j packets. Therefore we define $\chi_{\mathcal{P}_j}$ as the number of class-j

packets arriving during $\mathcal{S}$ that are to be served before $\mathcal{P}_j$. Similar to (5.27), the pgf X_j of $\chi_{\mathcal{P}_j}$ can then be found as

$$X_j(z) \triangleq \mathrm{E}[z^{\chi_{\mathcal{P}_j}}] = \frac{A_j(z)-1}{\lambda_j\,(z-1)}, \tag{6.20}$$

with mean

$$\mathrm{E}\big[\chi_{\mathcal{P}_j}\big] = X_j'(1) = \frac{\lambda_j'}{2\lambda_j}, \tag{6.21}$$

where $\lambda_j' \triangleq A_j''(1)$.

Delay of class-1 packets

The delay d_1 of a random class-1 packet $\mathcal{P}_1$ is fully determined by

- the remaining service time of the packet in service at the beginning of $\mathcal{S}$, if any;
- the total service time of the data packets in the queue (i.e. excluding the packet in the server, if any) at the beginning of $\mathcal{S}$, that have to be served before $\mathcal{P}_1$;
- the total service time of the $\chi_{\mathcal{P}_1}$ class-1 packets arriving along with $\mathcal{P}_1$, that have to be served before $\mathcal{P}_1$;
- the service time of $\mathcal{P}_1$ itself.

Let n_1 be the sum of the number of packets mentioned in the second and third item, i.e. the total number of packets in front of $\mathcal{P}_1$ at the actual moment of its insertion, excluding the one in the server, if any. If $\mathcal{P}_1$ is the first class-1 arrival in slot $\mathcal{S}$ (i.e. $\chi_{\mathcal{P}_1} = 0$), it will replace the reservation $\mathcal{R}$ at position $m_{\mathcal{S}}$ in the queue. On the other hand, if $\mathcal{P}_1$ is not the first class-1 packet arriving in slot $\mathcal{S}$ (i.e. $\chi_{\mathcal{P}_1} > 0$), it will be appended to the queue in arrival order, such that all $(u_{\mathcal{S}} - 1)^+$ data packets, present in the queue itself at the beginning of $\mathcal{S}$, will be served before $\mathcal{P}_1$. This translates to

$$n_1 = \begin{cases} m_{\mathcal{S}} - 1, & \text{if } \chi_{\mathcal{P}_1} = 0, \\ (u_{\mathcal{S}} - 1)^+ + \chi_{\mathcal{P}_1}, & \text{if } \chi_{\mathcal{P}_1} > 0. \end{cases} \tag{6.22}$$

The delay d_1 of $\mathcal{P}_1$ can thus be found as

$$d_1 = \begin{cases} \sum_{i=1}^{n_1+1} s_i, & \text{if } u_{\mathcal{S}} = 0, \\ \sum_{i=1}^{r_{\mathcal{S}}+n_1+1} s_i, & \text{if } u_{\mathcal{S}} > 0, \end{cases} \tag{6.23}$$

where $r_{\mathcal{S}} = 0$ corresponds with a departure during slot $\mathcal{S}$ and $r_{\mathcal{S}} = 1$ denotes that the packet in service at the beginning of $\mathcal{S}$ does not leave the system during slot $\mathcal{S}$. The s_i denote service times of individual packets; due to the memoryless property of the geometric distribution, we do not need to make any distinction between the packet in service at the beginning of slot $\mathcal{S}$, if any, and the other packets.

The pgf $D_1(z)$ of the class-1 packet delay d_1 can then be calculated as

$$\begin{aligned} D_1(z) \triangleq \mathrm{E}\big[z^{d_1}\big] &= \mathrm{E}\Big[S(z)^{n_1+1}\,\{u_{\mathcal{S}}=0\}\Big] + \frac{S(z)}{z}\,\mathrm{E}\Big[S(z)^{n_1+1}\,\{u_{\mathcal{S}}>0\}\Big] \\ &= p_0 S(z) X_1(S(z)) + X_1(0)\frac{S(z)^2}{z}\,(P(S(z),1)-p_0) \\ &\quad + \frac{S(z)}{z}\,(X_1(S(z))-X_1(0))\,(U(S(z))-p_0) \\ &= \frac{p_0 S(z)}{\lambda_1}\Bigg[\frac{1-A_1(S(z))}{1-S(z)} - (1-A_1(0))\,\frac{S(z)}{z} \\ &\qquad - \frac{1-A_T(S(z))}{z-A_T(S(z))}\left(\frac{S(z)-A_1(S(z))}{1-S(z)} + A_1(0)\right) \\ &\qquad + (1-z)\,\frac{1-A_1(0)}{z-A_1(0)}\left(\frac{A_1(0)-A_1(S(z))}{z-A_T(S(z))} - \frac{A_1(0)S(z)}{zS(A_1(0))}\phi(1)\right)\Bigg]. \end{aligned} \tag{6.24}$$

From (6.24) we can find the expected class-1 delay after some mathematical elaboration as

$$\mathrm{E}[d_1] = \frac{1}{\sigma}\left(2 + \frac{\lambda_T'}{2\sigma p_0} + \frac{\lambda_1' - 2\lambda_T}{2\lambda_1}\right) - \frac{p_0}{\lambda_1}\frac{A_1(0)}{S(A_1(0))}\,(1-\phi(1)) + \frac{\lambda_T\mu'}{2p_0}, \tag{6.25}$$

where $\lambda_T' \triangleq A_T''(1)$.

Delay of class-2 packets

Similar to n_1, we define n_2 as the total number of data packets in front of a random steady-state class-2 packet $\mathcal{P}_2$ at the actual moment of its insertion, excluding the one in the server, if any. This includes

- the $(u_{\mathcal{S}}-1)^+$ data packets in the queue at the beginning of $\mathcal{S}$, excluding the one in the server, if any;
- the $a_{1,\mathcal{S}}$ class-1 packets arriving during slot $\mathcal{S}$;
- the $\chi_{\mathcal{P}_2}$ class-2 packets arriving along with $\mathcal{P}_2$, that have to be served before $\mathcal{P}_2$;

or more concise

$$n_2 = (u_{\mathcal{S}}-1)^+ + a_{1,\mathcal{S}} + \chi_{\mathcal{P}_2}. \tag{6.26}$$

Note that $\mathcal{P}_2$ is preceded in the queue by the reservation $\mathcal{R}$, which is in fact no data packet and therefore does not contribute to $\mathcal{P}_2$'s delay. As long as $\mathcal{P}_2$ has not yet entered the server, $\mathcal{R}$ can however be replaced by a class-1 packet, which in turn *does* contribute to the delay of $\mathcal{P}_2$. Therefore, we first define v as the sum of the remaining service time minus 1 slot of the packet in service at the beginning of $\mathcal{S}$, if any, and the total service time of all n_2 data packets that arrive during $\mathcal{S}$ and have to be served before

$\mathcal{P}_2$. The random variable v thus represents the minimal number of slots $\mathcal{P}_2$ spends in the queue before entering the server and can be found as

$$v = \begin{cases} \sum_{i=1}^{n_2} s_i, & \text{if } u_{\mathcal{S}} = 0, \\ \sum_{i=1}^{r_{\mathcal{S}}+n_2} s_i, & \text{if } u_{\mathcal{S}} > 0. \end{cases} \tag{6.27}$$

Note that due to the fact that $\chi_{\mathcal{P}_2}$ depends on $a_{2,\mathcal{S}}$, we have that $a_{1,\mathcal{S}}$ and $\chi_{\mathcal{P}_2}$ are stochastically correlated, with their joint pgf following from (6.19):

$$\begin{aligned} \mathrm{E}[x^{a_{1,\mathcal{S}}} y^{\chi_{\mathcal{P}_2}}] &= \mathrm{E}\left[x^{a_{1,\mathcal{S}}} \sum_{n=0}^{a_{2,\mathcal{S}}-1} \frac{1}{a_{2,\mathcal{S}}} y^n\right] = \mathrm{E}\left[\frac{x^{a_{1,\mathcal{S}}}\,(y^{a_{2,\mathcal{S}}}-1)}{a_{2,\mathcal{S}}\,(y-1)}\right] \\ &= \mathrm{E}\left[\frac{x^{a_1}\,(y^{a_2}-1)}{\lambda_2\,(y-1)}\right] = \frac{A(x,y)-A_1(x)}{\lambda_2\,(y-1)}, \end{aligned} \tag{6.28}$$

such that the pgf $V(z)$ of v can be found as

$$\begin{aligned} V(z) &\triangleq \mathrm{E}[z^v] = \mathrm{E}[S(z)^{n_2}\,\{u_{\mathcal{S}}=0\}] + \frac{S(z)}{z}\,\mathrm{E}[S(z)^{n_2}\,\{u_{\mathcal{S}}>0\}] \\ &= \mathrm{E}\Big[S(z)^{a_{1,\mathcal{S}}+\chi_{\mathcal{P}_2}}\Big]\left(p_0 + \frac{U(S(z))-p_0}{z}\right) \\ &= \frac{\sigma p_0}{\lambda_2}\,\frac{z}{S(z)}\,\frac{A_T(S(z))-A_1(S(z))}{z-A_T(S(z))}. \end{aligned} \tag{6.29}$$

Note that the expression $U(S(z))$ in the above calculation bears some special meaning. If we consider the amount of work in the system during an arbitrary steady-state slot, instead of the number of packets, it is clear that due to the memoryless nature of the service time distribution the pgf of this quantity is given by $U(S(z))$, which yields

$$U(S(z)) = p_0 A_T(S(z)) \frac{z-1}{z-A_T(S(z))}. \tag{6.30}$$

The delay of $\mathcal{P}_2$ then consists of these v slots augmented with 1 service time (i.e. of $\mathcal{P}_2$ itself) if $\mathcal{R}$ is not replaced during the v slots or augmented with 2 service times if a class-1 packet does arrive before $\mathcal{P}_2$ enters the server. Introducing the Bernoulli variable γ_n that is equal to 0 if and only if no class-1 packet arrives during n consecutive slots, we have that

$$\mathrm{Prob}[\gamma_n = 0] = A_1(0)^n, \qquad \text{and} \qquad \mathrm{Prob}[\gamma_n = 1] = 1 - A_1(0)^n, \tag{6.31}$$

with pgf

$$\Gamma_n(z) \triangleq \mathrm{E}[z^{\gamma_n}] = z + (1-z)\,A_1(0)^n. \tag{6.32}$$

The class-2 packet delay d_2 can then simply be calculated as

$$d_2 = v + s_{\mathcal{P}} + \gamma_v s_*, \tag{6.33}$$

where $s_{\mathcal{P}}$ is the service time of $\mathcal{P}_2$ and s_* is the service time of the class-1 packet that seized $\mathcal{R}$, if any. Both these service times are identically distributed as random service times and thus have pgf $S(z)$. We then find

$$\begin{aligned} D_2(z) &\triangleq \mathrm{E}\big[z^{d_2}\big] = \mathrm{E}\big[z^{v+s_{\mathcal{P}}+\gamma_v s_*}\big] = S(z)\,\mathrm{E}[z^v S(z)^{\gamma_v}] \\ &= S(z)\,[S(z)V(z) + (1-S(z))\,V(zA_1(0))] \\ &= \frac{\sigma p_0 z S(z)}{\lambda_2}\left[\frac{A_T(S(z)) - A_1(S(z))}{z - A_T(S(z))}\right. \\ &\qquad\left. + (1-S(z))\,\frac{A_1(0)}{S(zA_1(0))}\,\frac{A_T(S(zA_1(0))) - A_1(S(zA_1(0)))}{zA_1(0) - A_T(S(zA_1(0)))}\right]. \end{aligned} \tag{6.34}$$

The mean class-2 packet delay is then

$$\mathrm{E}[d_2] = \frac{1}{\sigma}\left(2 + \frac{\lambda_T'}{2\sigma p_0} + \frac{\lambda_T' - \lambda_1'}{2\lambda_2} - V(A_1(0))\right) + \frac{\lambda_T \mu'}{2p_0}. \tag{6.35}$$

Delay of a random packet

Additionally, we can determine the distribution of the delay d of a random steady-state packet, thus without selecting the class in advance. The probability for a random steady-state packet $\mathcal{P}$ to belong to a specific class-j can be simply found as

$$\mathrm{Prob}[\mathcal{P} \text{ belongs to class } j] = \frac{\lambda_j}{\lambda_T}. \tag{6.36}$$

The pgf $D(z)$ of the delay d of $\mathcal{P}$ is then the weighted sum of the delay pgfs of both classes

$$D(z) = \sum_{j=1}^{2} \frac{\lambda_j}{\lambda_T} D_j(z). \tag{6.37}$$

More importantly, the mean packet delay $\mathrm{E}[d]$ of a random steady-state packet can then be calculated as

$$\mathrm{E}[d] = D'(1) = \frac{\lambda_1}{\lambda_T}\,\mathrm{E}[d_1] + \frac{\lambda_2}{\lambda_T}\,\mathrm{E}[d_2] = \frac{1}{\sigma} + \frac{\lambda_T'}{2\lambda_T \sigma p_0} + \frac{\lambda_T \mu'}{2p_0}. \tag{6.38}$$

Note that (6.38) is identical to (5.30) after substitution of $S(z)$ as in (6.3). Therefore, it comes as no surprise that (6.12) and (6.38) comply with Little's law, which for this system translates to

$$\mathrm{E}[u] = \lambda_T\,\mathrm{E}[d] = \lambda_1\,\mathrm{E}[d_1] + \lambda_2\,\mathrm{E}[d_2]\,. \tag{6.39}$$

Tail distributions

From (6.24) and (6.34), it can be shown that the tail distributions of the packet delay for both packet classes are governed by the same dominant

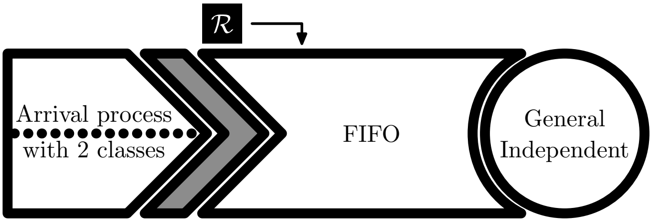

Figure 6.4: Illustration of the reservation system with general independent service times.

pole z_d. This dominant pole is the smallest real positive zero larger than 1 of the factor $z - A_T(S(z))$ in the denominators. The complex residues θ_j ($j \in \{1,2\}$) of the pgfs $D_j(z)$ for $z = z_d$ can then be found as

$$\theta_1 = \frac{p_0\,(z_d-1)\,S(z_d)}{\lambda_1\,(1-A_T'(S(z_d))S'(z_d))}\left[\frac{S(z_d)-A_1(S(z_d))}{1-S(z_d)}\right. \\ \left. + \frac{A_1(0)\,(z_d-1)+(1-A_1(0))\,A_1(S(z_d))}{z_d-A_1(0)}\right], \tag{6.40}$$

and

$$\theta_2 = \frac{\sigma p_0 z_d S(z_d)}{\lambda_2}\frac{z_d - A_1(S(z_d))}{1-A_T'(S(z_d))S'(z_d)}. \tag{6.41}$$

The tail distribution of the class-j packet delay can then be approximated as

$$\text{Prob}[d_j = n] \approx -\theta_j {z_d}^{-n-1}. \tag{6.42}$$

6.5 General service times

In this section we let go of the restriction for the service times to be geometrically distributed. Instead, we assume a *general iid* server process, just like in the $GI-GI-1$ model, such that service times are independent and identically distributed for subsequent packets. The corresponding system is illustrated in Figure 6.4.

6.5.1 System equations

In the previous section, the geometric server process allowed us to discard all information about the progress of the active service from the system state vector. In case of general independent service times however, the remaining service time h_k at the beginning of slot k of a packet in service is required in order to determine the distributions of the system content u_{k+1} and the remaining service time h_{k+1} at the beginning of slot $k+1$. We therefore expand the system state vector at the beginning of slot k from the previous section to the vector $\langle h_k, m_k, u_k \rangle$. Where m_k and u_k are the position of the reservation $\mathcal{R}$ and the system content at the beginning of slot k, as defined in the previous section.

In order to construct the system equations, we first combine the observations from the $GI-GI-1$ model and the reservation model for geometric service times. Only in case the system is empty ($u_k = 0$) at the beginning of slot k, $h_k = 0$, by definition. The reservation $\mathcal{R}$ is then at the first position in the queue, such that $m_k = 1$, and will move backward in the queue according to the number of class-1 arrivals during slot k. If there are arrivals during slot k, one of these packets will be allowed access to the server at the beginning of slot $k+1$. In case the system is not empty at the beginning of slot k, the packet in the server receives one more slot of service during slot k, such that a departure occurs if $h_k = 1$. The reservation will only move backward in the queue if class-1 packets arrive during slot k. If a departure occurs at the end of slot k, all packets will shift one position closer, except for $\mathcal{R}$ if it was already at position 1. These observations yield the following sets of system equations:

- if $h_k = 0$:

$$\begin{aligned} h_{k+1} &= \begin{cases} 0, & \text{if } a_{T,k} = 0, \\ s, & \text{if } a_{T,k} > 0, \end{cases} \\ m_{k+1} &= \begin{cases} 1, & \text{if } a_{1,k} = 0, \\ a_{1,k}, & \text{if } a_{1,k} > 0, \end{cases} \\ u_{k+1} &= a_{T,k}, \end{aligned} \tag{6.43}$$

- if $h_k = 1$:

$$\begin{aligned} h_{k+1} &= \begin{cases} 0, & \text{if } u_k = 0 \text{ and } a_{T,k} = 0, \\ s, & \text{if } u_k > 0 \text{ or } a_{T,k} > 0, \end{cases} \\ m_{k+1} &= \begin{cases} (m_k - 2)^+ + 1, & \text{if } a_{1,k} = 0, \\ u_k - 1 + a_{1,k}, & \text{if } a_{1,k} > 0, \end{cases} \\ u_{k+1} &= u_k - 1 + a_{T,k}, \end{aligned} \tag{6.44}$$

- if $h_k > 1$:

$$\begin{aligned} h_{k+1} &= h_k - 1, \\ m_{k+1} &= \begin{cases} m_k, & \text{if } a_{1,k} = 0, \\ u_k + a_{1,k}, & \text{if } a_{1,k} > 0, \end{cases} \\ u_{k+1} &= u_k + a_{T,k}. \end{aligned} \tag{6.45}$$

6.5.2 Buffer analysis

The system state pgf $P_k(x,y,z)$ at the beginning of slot k is defined as

$$P_k(x,y,z) \triangleq \mathrm{E}\left[x^{h_k} y^{m_k-1} z^{u_k}\right]. \tag{6.46}$$

Note that again, we refer to $m_k - 1$ rather than to m_k itself; in this way, since $u_k = 0$ if and only if both $h_k = 0$ and $m_k = 1$, $P_k(0,0,0)$ is the probability of having an empty system at the beginning of slot k. Using the law of total expectation, we can then determine $P_{k+1}(x,y,z)$ as follows

$$\begin{aligned} P_{k+1}(x,y,z) &\triangleq \mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}}\right] \\ &= \mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}} \{h_k = 0\}\right] \\ &\quad + \mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}} \{h_k = 1\}\right] \\ &\quad + \mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}} \{h_k > 1\}\right], \end{aligned} \tag{6.47}$$

according to the three cases for the system equations distinguished in the previous section.

The first term on the right hand side of (6.47) can then be calculated by application of (6.43). Again using the law of total expectation, we get

$$\begin{aligned} &\mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}} \{h_k = 0\}\right] = \mathrm{E}\left[x^0 y^0 z^0 \{h_k = 0, a_{1,k} = 0, a_{2,k} = 0\}\right] \\ &\quad + \mathrm{E}\left[x^{s^*} z^{a_{2,k}} \{h_k = 0, a_{1,k} = 0, a_{2,k} > 0\}\right] \\ &\quad + \mathrm{E}\left[x^{s^*} y^{a_{1,k}-1} z^{a_{T,k}} \{h_k = 0, a_{1,k} > 0\}\right], \\ &= \left(A_T(0) + S(x)\left(A(0,z) - A_T(0)\right) + \frac{S(x)}{y}\left(A(yz,z) - A(0,z)\right)\right) p_{0,k} \\ &= \left(\left(1 - S(x)\right) A_T(0) + \frac{S(x)}{y}\left(\left(y-1\right) A(0,z) + A(yz,z)\right)\right) p_{0,k}, \end{aligned} \tag{6.48}$$

where $p_{0,k} \triangleq \mathrm{Prob}[h_k = 0] = P_k(0,0)$ is the probability that the system is empty at the beginning of slot k. The second term on the right hand side of (6.47) can be found similarly by using (6.44) as

$$\begin{aligned} &\mathrm{E}\left[x^{h_{k+1}} y^{m_{k+1}-1} z^{u_{k+1}} \{h_k = 1\}\right] \\ &= \mathrm{E}\left[x^0 y^0 z^0 \{h_k = 1, u_k = 1, a_{1,k} = 0, a_{2,k} = 0\}\right] \\ &\quad + \mathrm{E}\left[x^{s^*} y^{(m_k-2)^+} z^{u_k-1} \{h_k = 1, u_k > 1, a_{1,k} = 0, a_{2,k} = 0\}\right] \\ &\quad + \mathrm{E}\left[x^{s^*} y^{(m_k-2)^+} z^{u_k-1+a_{2,k}} \{h_k = 1, a_{1,k} = 0, a_{2,k} > 0\}\right] \\ &\quad + \mathrm{E}\left[x^{s^*} y^{u_k+a_{1,k}-2} z^{u_k-1+a_{T,k}} \{h_k = 1, a_{1,k} > 0\}\right] \\ &= \left(1 - S(x)\right) A_T(0) Q_k(0,0) + \frac{S(x)}{y} A(0,z)\left(Q_k(y,z) + \left(y-1\right) Q_k(0,z)\right) \\ &\quad + \frac{S(x)}{y}\left(A(yz,z) - A(0,z)\right) Q_k(1,yz), \end{aligned} \tag{6.49}$$

where we introduced the function $Q_k(y,z)$ as

$$\begin{aligned} Q_k(y,z) &\triangleq \mathrm{E}\big[y^{m_k-1}z^{u_k-1}\,\{h_k=1\}\big] \\ &= \sum_{i=1}^{\infty}\sum_{j=1}^{\infty}\mathrm{Prob}[h_k=1, m_k=i, u_k=j]\,y^{i-1}z^{j-1}. \end{aligned} \tag{6.50}$$

We then find the third term on the right hand side of (6.47) from (6.45) as

$$\begin{aligned} &\mathrm{E}\big[x^{h_{k+1}}y^{m_{k+1}-1}z^{u_{k+1}}\,\{h_k>1\}\big] \\ &= \mathrm{E}\big[x^{h_k-1}y^{m_k-1}z^{u_k+a_{2,k}}\,\{h_k>1, a_{1,k}=0\}\big] \\ &\quad + \mathrm{E}\big[x^{h_k-1}y^{u_k+a_{1,k}-1}z^{u_k+a_{T,k}}\,\{h_k>1, a_{1,k}>0\}\big] \\ &= \frac{A(0,z)}{x}\left(P_k(x,y,z)-p_{0,k}-xzQ_k(y,z)\right) \\ &\quad + \frac{A(yz,z)-A(0,z)}{xy}\left(P_k(x,1,yz)-p_{0,k}-xyzQ_k(1,yz)\right). \end{aligned} \tag{6.51}$$

Substitution of (6.48), (6.49) and (6.51) into (6.47) then allows us to express $P_{k+1}(x,y,z)$ in terms of $P_k(x,y,z)$:

$$\begin{aligned} P_{k+1}(x,y,z) &= \frac{1}{x}A(0,z)P_k(x,y,z) + A_T(0)\left(1-S(x)\right)\left(p_{0,k}+Q_k(0,0)\right) \\ &\quad + \frac{xS(x)-1}{xy}\left(\left(y-1\right)A(0,z)+A(yz,z)\right)p_{0,k} \\ &\quad + \frac{y-1}{y}S(x)A(0,z)Q_k(0,z) + \frac{S(x)-yz}{y}A(0,z)Q_k(y,z) \\ &\quad + \frac{S(x)-yz}{y}\left(A(yz,z)-A(0,z)\right)Q_k(1,yz) \\ &\quad + \frac{1}{xy}\left(A(yz,z)-A(0,z)\right)P_k(x,1,yz). \end{aligned} \tag{6.52}$$

Given the fact that if the system is empty at the beginning of slot k, the reservation $\mathcal{R}$ is at the head of the queue (i.e. $m_k=1$) and by definition $h_k=0$, we get that

$$\begin{aligned} p_{0,k} \triangleq \mathrm{Prob}[u_k=0] &= \mathrm{Prob}[h_k=0, m_k=1, u_k=0] \\ &= P_k(0,0,0) = P_k(x,y,0), \quad \forall x,y. \end{aligned} \tag{6.53}$$

Similarly, if there is exactly one packet in the system at the beginning of slot k, this packet must reside in the server and the queue itself is empty, which again means that $\mathcal{R}$ is at the queue's head.

$$Q_k(0,0) = \mathrm{Prob}[h_k=1, m_k=1, u_k=1] = Q_k(y,0), \quad \forall y. \tag{6.54}$$

Assuming system stability (i.e. $\lambda_T\mu<1$), expressions (6.46) and (6.52) will converge for $k\to\infty$ and the system will reach a steady state. Omitting the time index k for expressions corresponding to this steady state, we determine the system state pgf in steady-state as

$$\begin{aligned}P(x,y,z) = \frac{1}{y\,(x-A(0,z))}\,&[xy\,(1-S(x))\,A_T(0)\,(p_0+Q(0,0))\\&+(xS(x)-1)\,((y-1)\,A(0,z)+A(yz,z))\,p_0\\&+x\,(y-1)\,S(x)A(0,z)Q(0,z)+x\,(S(x)-yz)\,A(0,z)Q(y,z)\\&+x\,(S(x)-yz)\,(A(yz,z)-A(0,z))\,Q(1,yz)\\&+(A(yz,z)-A(0,z))\,P(x,1,yz)]\,. \qquad (6.55)\end{aligned}$$

Substitution of $y = 1$ then yields

$$\begin{aligned}P(x,1,z) = \frac{1}{x-A_T(z)}\,&[x\,(1-S(x))\,A_T(0)\,(p_0+Q(0,0))\\&+(xS(x)-1)\,A_T(z)p_0+x\,(S(x)-z)\,A_T(z)Q(1,z)]\,. \quad (6.56)\end{aligned}$$

Note the similarity between (6.55) and the initial expression for the steady-state pgf of the system state of the $GI-GI-1$ model (5.10). In fact, substitution of $y = 1$ yields an expression without knowledge about $\mathcal{R}$'s position in the queue. The Reservation discipline only interferes with how packets are inserted into the queue and has no effect whatsoever on the system content or the service times. Similar to the argumentation in the previous section, it therefore was to be expected that substitution of $y = 1$ results in expressions similar to the ones in the $GI-GI-1$ model.

Moreover, results obtained for the $GI-GI-1$ model can be adopted and transformed according to the parameters of the current system. This yields for the empty system probability

$$p_0 = 1 - A_T'(1)S'(1) = 1 - \lambda_T\mu. \qquad (6.57)$$

The system content pgf follows as

$$U(z) \triangleq \mathrm{E}[z^u] = Q(1,z) = \frac{S(A_T(z))\,(A_T(z)-1)}{A_T(z)\,(z-S(A_T(z)))}p_0, \qquad (6.58)$$

with mean

$$\mathrm{E}[u] = U'(1) = (1-p_0) + \frac{\lambda_T'\mu+\lambda_T^2\mu'}{2p_0}, \qquad (6.59)$$

where $\lambda_T' \triangleq A_T''(1)$ and $\mu' \triangleq S''(1)$.

Substitution of $z = 0$ in (6.56) and using the property that $Q(y,0) = Q(0,0), \forall y$ (see (6.53)), it then follows that

$$p_0 = \frac{A_T(0)}{x-A_T(0)}\,((x-1)\,p_0+xQ(0,0))\,, \quad \forall x. \qquad (6.60)$$

After substitution of $x = 1$, we get

$$p_0 = \frac{A_T(0)Q(0,0)}{1 - A_T(0)}, \tag{6.61}$$

such that

$$A_T(0)\left(p_0 + Q(0,0)\right) = p_0. \tag{6.62}$$

Since $P(x, y, z)$ is a pgf, it must be bounded for all points (x, y, z) for which $|x| \leq 1$, $|y| \leq 1$ and $|z| \leq 1$. In particular, this should hold for $(A(0, z), y, z)$ with $|y| \leq 1$ and $|z| \leq 1$. Mind that, due to the fact that $A(z_1, z_2)$ is a pgf, we have that $|A(0, z)| \leq 1$ for $|z| \leq 1$. However, if we were to substitute $x = A(0, z)$ in (6.55), the denominator would become 0 such that the corresponding numerator should vanish as well. This way, we can then apply de l'Hôpital's theorem to find a bounded value for $P(A(0, z), y, z)$. With (6.58) and (6.62), this consideration leads to an additional relation which allows us to determine $Q(y, z)$ as

$$\begin{aligned} Q(y,z) = {} & \frac{S(A(0,z))}{yz - S(A(0,z))} \frac{A(0,z)-1}{A(0,z)} \left(y + \frac{A(yz,z) - A(0,z)}{A(0,z) - A_T(yz)} \right) p_0 \\ & + \frac{(y-1)\, S(A(0,z))}{yz - S(A(0,z))} Q(0,z) - \frac{A(yz,z) - A(0,z)}{A(0,z) - A_T(yz)} Q(1,yz). \end{aligned} \tag{6.63}$$

The same argumentation can be applied to the partial pgf $Q(y, z) \triangleq \mathrm{E}\left[y^{m-1} z^{u-1} \{h = 1\}\right]$ with respect to the denominator $yz - S(A(0, z))$. Just like any normal pgf, partial pgfs must also be bounded for arguments on the unit disk (which in this case corresponds to when $|y| \leq 1$, $|z| \leq 1$), such that there must not be any singularities in the open unit disk. Provided we can find a point $(S(A(0, z))/z, z)$ in the unit disk, a bounded value for $Q(S(A(0, z))/z, z)$ can only be found if the numerator becomes 0, yielding

$$Q(0,z) = \frac{A(0,z) - 1}{A(0,z)\left(z - S(A(0,z))\right)} \left(S(A(0,z)) - z\phi(z)\right) p_0, \tag{6.64}$$

where we recycled $\phi(z)$ from (6.17). We note again that it *is* possible to find a non-empty subset $\aleph$ of the open unit disk that contains a z for which $|S(A(0, z))/z| \leq 1$. We will come back on this in 6.7.

At this point, we have determined all of the unknowns in (6.55). Substitution of these results then yields the closed-form expression

$$\begin{aligned}P(x,y,z) &= p_0 \left\{ 1 - xz \frac{1 - A(0,z)}{x - A(0,z)} \frac{S(x) - S(A(0,z))}{z - S(A(0,z))} \right\} \\ &+ \frac{p_0}{yz - S(A(0,z))} \left\{ \frac{xz\,(yz - S(x))}{A(0,z) - A_T(yz)} \frac{A(yz,z) - A(0,z)}{yz - S(A_T(yz))} \right. \\ &\quad \left(S(A(0,z)) \frac{1 - A(0,z)}{x - A(0,z)} - S(A_T(yz)) \frac{1 - A_T(yz)}{x - A_T(yz)} \right) \\ &+ \frac{xz\,(1-x)}{x - A(0,z)} \frac{A(yz,z) - A(0,z)}{x - A_T(yz)} \left(S(A(0,z)) \frac{S(x) - S(A_T(yz))}{yz - S(A_T(yz))} - S(x) \right) \\ &\left. + xz^2\,(y-1) \frac{1 - A(0,z)}{x - A(0,z)} \frac{S(x) - S(A(0,z))}{z - S(A(0,z))} \phi(z) \right\}. \qquad (6.65)\end{aligned}$$

6.5.3 Packet delay analysis

The analysis of the packet delay is very similar to that presented in the previous section. In fact, the only difference is in the service time distribution, such that all argumentations that do not involve the actual service time distribution can be adopted without hesitation.

For instance, this implies that for a random steady-state class-j packet $\mathcal{P}_j$, the system state distribution at the beginning of its arrival slot $\mathcal{S}$, is stochastically identical to that of a random steady-state slot, governed by $P(x,y,z)$. The pmf of the numbers of arrivals $(a_{1,\mathcal{S}}, a_{2,\mathcal{S}})$ during slot $\mathcal{S}$ is given by

$$\text{Prob}[a_{1,\mathcal{S}} = \alpha_1, a_{2,\mathcal{S}} = \alpha_2] = \frac{\alpha_j}{\lambda_j} \text{Prob}[a_1 = \alpha_1, a_2 = \alpha_2]\,. \qquad (6.66)$$

The pgf of the number of class-j packets $\chi_{\mathcal{P}_j}$ arriving during $\mathcal{S}$ and to be served before $\mathcal{P}$ is

$$X_j(z) \triangleq \text{E}[z^{\chi_{\mathcal{P}_j}}] = \frac{A_j(z) - 1}{\lambda_j\,(z-1)}. \qquad (6.67)$$

Delay of class-1 packets

Similar as in 6.4.3, the delay of a class-1 packet $\mathcal{P}_1$ can be determined from

- the remaining service time $h_{\mathcal{S}}$ of the packet in service during slot $\mathcal{S}$, if any;
- the total service time of all data packets in the queue (i.e. excluding the packet in the server, if any) at the beginning of $\mathcal{S}$, that have to be served before $\mathcal{P}_1$;
- the total service time of the $\chi_{\mathcal{P}_1}$ class-1 packets arriving along with $\mathcal{P}_1$, that have to be served before $\mathcal{P}_1$;
- the service time of $\mathcal{P}_1$ itself.

Note that this time, the service time distribution generally is not memoryless, such that similar to (6.23) we now get

$$d_1 = (h_{\mathcal{S}} - 1)^+ + \sum_{i=1}^{n_1+1} s_i, \tag{6.68}$$

where n_1 is the number of data packets in the queue (i.e. excluding the packet in the server, if any) and to be served before $\mathcal{P}$ at the exact time of its insertion in the queue and the s_is denote complete service times of the n_1 packets and $\mathcal{P}$ itself. The value of n_1 depends on the system state at the beginning of $\mathcal{S}$ the number $\chi_{\mathcal{P}_1}$ of class-1 arrivals during $\mathcal{S}$ and to be served before $\mathcal{P}$, specifically we have

$$\begin{cases} n_1 = m_{\mathcal{S}} - 1, & \text{if } \chi_{\mathcal{P}_1} = 0, \\ n_1 = (u_{\mathcal{S}} - 1)^+ + \chi_{\mathcal{P}_1}, & \text{if } \chi_{\mathcal{P}_1} > 0. \end{cases} \tag{6.69}$$

The pgf $D_1(z)$ of the class-1 packet delay d_1 can then be found as

$$\begin{aligned} D_1(z) &\triangleq \mathrm{E}\left[z^{d_1}\right] = \mathrm{E}\left[S(z)^{n_1+1}\middle| u_{\mathcal{S}} = 0\right] + \mathrm{E}\left[z^{h_{\mathcal{S}}-1} S(z)^{n_1+1}\middle| u_{\mathcal{S}} > 0\right] \\ &= p_0 S(z) X_1(S(z)) + X_1(0)\frac{S(z)}{z}\left(P(z, S(z), 1) - p_0\right) \\ &\quad + \frac{1}{z}\left(X_1(S(z)) - X_1(0)\right)\left(P(z, 1, S(z)) - p_0\right) \\ &= \frac{p_0}{\lambda_1} S(z) \left\{ \frac{1 - A_1(S(z))}{1 - S(z)} + (z-1)\frac{1 - A_1(0)}{z - A_1(0)}\frac{A_1(S(z)) - A_1(0)}{z - A_T(S(z))} \right. \\ &\qquad + \frac{A_T(S(z)) - 1}{z - A_T(S(z))}\frac{S(z) - A_1(S(z)) + (1 - S(z))\, A_1(0)}{1 - S(z)} \\ &\qquad \left. + \frac{(1 - A_1(0))^2 \left[S(A_1(0)) - S(z) + (S(z) - 1)\, \phi(1)\right]}{(z - A_1(0))\,(1 - S(A_1(0)))} \right\}. \end{aligned} \tag{6.70}$$

The mean class-1 packet delay can then be found as

$$\mathrm{E}[d_1] = \mu\left(2 + \frac{\lambda_T' \mu}{2p_0} + \frac{\lambda_1' - 2\lambda_T}{2\lambda_1} - \frac{p_0}{\lambda_1}\frac{1 - A_1(0)}{1 - S(A_1(0))}\left(1 - \phi(1)\right)\right) + \frac{\lambda_T \mu'}{2p_0}. \tag{6.71}$$

Delay of class-2 packets

When a class-2 packet is inserted in the queue, it is always inserted behind the reservation $\mathcal{R}$, such that the class-2 packet delay distribution depends on the possibility of $\mathcal{R}$ to be taken by a class-1 packet. Therefore, we first consider the number of slots v a random class-2 packet $\mathcal{P}_2$ would have to wait in the queue if the reservation is not taken. This number depends on

- the remaining service time $h_{\mathcal{S}}$ of the packet in service during slot $\mathcal{S}$, if any;

- the $(u_{\mathcal{S}}-1)^+$ data packets in the queue at the beginning of $\mathcal{S}$, excluding the one in the server, if any;
- the $a_{1,\mathcal{S}}$ class-1 packets arriving during slot $\mathcal{S}$;
- the $\chi_{\mathcal{P}_2}$ class-2 packets arriving along with $\mathcal{P}_2$, that have to be served before $\mathcal{P}_2$;

such that v can then be calculated as

$$v = (h_{\mathcal{S}}-1)^+ + \sum_{i=1}^{(u_{\mathcal{S}}-1)^+ + a_{1,\mathcal{S}} + \chi_{\mathcal{P}_2}} s_i. \tag{6.72}$$

As explained in the analysis for geometric service times, the random variables $a_{1,\mathcal{S}}$ and $\chi_{\mathcal{P}_2}$ are correlated, with joint pgf

$$\mathrm{E}[x^{a_{1,\mathcal{S}}} y^{\chi_{\mathcal{P}_2}}] = \frac{A(x,y)-A_1(x)}{\lambda_2\,(y-1)}, \tag{6.73}$$

such that the pgf $V(z)$ of v can be found as

$$\begin{aligned}
V(z) &\triangleq \mathrm{E}[z^v] = \mathrm{E}\Big[z^{(h_{\mathcal{S}}-1)^+ + \sum_{i=1}^{(u_{\mathcal{S}}-1)^+ + a_{1,\mathcal{S}} + \chi_{\mathcal{P}_2}} s_i}\Big] \\
&= \mathrm{E}\Big[S(z)^{a_{1,\mathcal{S}}+\chi_{\mathcal{P}_2}}\{u_{\mathcal{S}}=0\}\Big] + \frac{1}{zS(z)}\,\mathrm{E}\Big[z^{h_{\mathcal{S}}} S(z)^{u_{\mathcal{S}}+a_{1,\mathcal{S}}+\chi_{\mathcal{P}_2}}\{u_{\mathcal{S}}>0\}\Big] \\
&= \mathrm{E}\Big[S(z)^{a_{1,\mathcal{S}}+\chi_{\mathcal{P}_2}}\Big]\,S(z)\left(p_0 + \frac{P(z,1,S(z))-p_0}{zS(z)}\right) \\
&= \frac{p_0}{\lambda_2}\frac{1-z}{1-S(z)}\frac{A_T(S(z))-A_1(S(z))}{z-A_T(S(z))}.
\end{aligned} \tag{6.74}$$

If the reservation $\mathcal{R}$ is not seized during these v slots (i.e. there is no class-1 arrival in any of v subsequent slots), $\mathcal{P}_2$ will enter the server as planned. If a class-1 packet does arrive before $\mathcal{P}_2$ is in service, the total waiting time of $\mathcal{P}_2$ is augmented with one service time such that

$$d_2 = v + s_{\mathcal{P}} + \gamma_v s_*, \tag{6.75}$$

where γ_n is a Bernoulli random variable that is 0 with probability $A_1(0)^n$, which corresponds to the reservation not being seized. Additionally, $s_{\mathcal{P}}$ is the service time of $\mathcal{P}_2$ and s_* is the service time of the class-1 packet that seized $\mathcal{R}$, if any. The pgf $D_2(z)$ of the class-2 packet delay d_2 follows as

$$\begin{aligned}
D_2(z) &\triangleq \mathrm{E}\big[z^{d_2}\big] = S(z)\left\{S(z)V(z) + (1-S(z))\,V(zA_1(0))\right\} \\
&= \frac{p_0}{\lambda_2}S(z)\left\{S(z)\frac{1-z}{1-S(z)}\frac{A_T(S(z))-A_1(S(z))}{z-A_T(S(z))}\right. \\
&\qquad \left. + (1-S(z))\,\frac{1-zA_1(0)}{1-S(zA_1(0))}\frac{A_T(S(zA_1(0)))-A_1(S(zA_1(0)))}{zA_1(0)-A_T(S(zA_1(0)))}\right\},
\end{aligned} \tag{6.76}$$

and the expected class-2 packet delay then becomes

$$\mathrm{E}[d_2] = \mu \left(2 + \frac{\lambda_T' \mu}{2p_0} + \frac{\lambda_T' - \lambda_1'}{2\lambda_2} - V(A_1(0)) \right) + \frac{\lambda_T \mu'}{2p_0}. \tag{6.77}$$

Delay of a random packet

Again, the pgf $D(z)$ of the delay d of a random steady-state packet (i.e. the packet can be of either class) can be found from the weighted sum of the class specific delay pgfs as

$$D(z) = \sum_{j=1}^{2} \frac{\lambda_j}{\lambda_T} D_j(z), \tag{6.78}$$

with mean

$$\mathrm{E}[d] = D'(1) = \frac{\lambda_1}{\lambda_T} \mathrm{E}[d_1] + \frac{\lambda_2}{\lambda_T} \mathrm{E}[d_2] = \mu \left(1 + \frac{\lambda_T'}{2\lambda_T p_0} \right) + \frac{\lambda_T \mu'}{2p_0}. \tag{6.79}$$

As expected, the results (6.59) and (6.79) comply with Little's law.

Tail distributions

Similar to the case of geometric arrivals, the packet delay pgfs $D_1(z)$ and $D_2(z)$ share the same dominant pole z_d, defined by the equation $z_d - A_T(S(z_d)) = 0$. Assuming we know z_d, we can then calculate the complex residues θ_j ($j \in \{1, 2\}$) of the pgfs $D_j(z)$ for $z = z_d$ as

$$\theta_1 = \frac{p_0 \, (z_d - 1) \, S(z_d)}{\lambda_1 \, (1 - A_T'(S(z_d)) S'(z_d))} \left[\frac{S(z_d) - A_1(S(z_d))}{1 - S(z_d)} \right. \\ \left. + \frac{A_1(0) \, (z_d - 1) + (1 - A_1(0)) \, A_1(S(z_d))}{z_d - A_1(0)} \right], \tag{6.80}$$

and

$$\theta_2 = \frac{p_0}{\lambda_2} S(z_d)^2 \frac{1 - z_d}{1 - S(z_d)} \frac{z_d - A_1(S(z_d))}{1 - A_T'(S(z_d)) S'(z_d)}. \tag{6.81}$$

From these residues, the tail distribution of the class-j packet delay can then be approximated as

$$\mathrm{Prob}[d_j = n] \approx -\theta_j {z_d}^{-n-1}. \tag{6.82}$$

6.6 Relation to the $GI - GI - 1$ model

As depicted in Figure 6.2, the Reservation discipline imposes some restrictions and modifications to the $GI - GI - 1$ model:

- the packets generated by the arrival process can be categorized into two classes reflecting the priority level;
- a reorder unit ensures that during each slot all newly generated high-priority packets are inserted before any newly generated low-priority packets;
- a reservation $\mathcal{R}$ is inserted in the queue as a placeholder for future high-priority packets;
- when a high-priority packet is inserted in the queue, it replaces the reservation $\mathcal{R}$ in the queue, the reservation in turn moves to the tail of the queue.

As such, a normal FIFO queue fed by a two-class arrival stream can adopt the Reservation discipline by the insertion of a reservation and some minor changes to the way data packets are inserted to the queue. Most other scheduling disciplines aimed at service differentiation require more profound modifications, such as additional queues, decision units to select a certain queue or packet according to some predefined settings.

Although the modifications required by the aforementioned differentiated service discipline essentially change the way packets traverse through the system, the effects of these schedulers can be suppressed by grouping all packets into a single class, such that one of λ_1 or λ_2 is equal to 0. In case of the Reservation discipline, this would eliminate the need for packet reordering and cause all packets to be inserted to the queue either in front of $\mathcal{R}$ (when $\lambda_T = \lambda_1$) or behind the reservation (when $\lambda_T = \lambda_2$). We now explore the two possibilities $\lambda_2 = 0$ and $\lambda_1 = 0$ and study how this setting affects some of the expressions obtained in 6.5.

First, we consider the case where there are only high-priority packets, such that $\lambda_T = \lambda_1$ and $\lambda_2 = 0$. Given that the number of low-priority arrivals during any slot k is $a_{2,k} = 0$, the joint pgf $A(z_1, z_2)$ breaks down to

$$\begin{aligned} A(z_1, z_2) \triangleq \mathrm{E}\big[{z_1}^{a_1} {z_2}^{0}\big] &= \mathrm{E}[{z_1}^{a_1}] = A_1(z_1) \\ &= \mathrm{E}[{z_1}^{a_T}] = A_T(z_1), \qquad \forall z_2. \end{aligned} \tag{6.83}$$

Furthermore, we know that $\mathcal{R}$ will always be at the queue's tail, such that $m_k = (u_k - 1)^+ + 1$ at the beginning of any slot k, such that the joint system state pgf $P(x, y, z)$ can be found as

$$\begin{aligned} P(x,y,z) &= \lim_{k\to\infty} P_k(x,y,z) = \lim_{k\to\infty} \mathrm{E}\Big[x^{h_k} y^{(u_k-1)^+} z^{u_k}\Big] \\ &= \lim_{k\to\infty} \frac{P_k(x,1,yz) + (y-1)\, p_{0,k}}{y} = \frac{P(x,1,yz) + (y-1)\, p_0}{y} \\ &= p_0 \left[1 - xz \frac{(1 - A_1(yz))\,(S(x) - S(A_1(yz)))}{(x - A_1(yz))\,(yz - S(A_1(yz)))}\right]. \end{aligned} \tag{6.84}$$

Note that it is redundant to keep track of the position of $\mathcal{R}$, since we determine its position directly from the system content. Furthermore, given the homogeneous nature of the arrival stream, the Reservation discipline will have no effect on the ordering of the packets. Removing the information concerning $\mathcal{R}$'s position in (6.84) corresponds to the substitution of $y = 1$, yielding the joint system state pgf (5.19) from the $GI - GI - 1$ model, where $A(z)$ has been replaced by $A_1(z)$. For the packet delay pgf of the high-priority packets $D_1(z)$, substitution of $A(z_1, z_2) = A_1(z_1)$ yields

$$D_1(z) = p_0 S(z) \frac{(z-1)\,(1 - A_1(S(z)))}{\lambda_1\,(1 - S(z))\,(z - A_1(S(z)))}, \tag{6.85}$$

which in turn corresponds to the packet delay pgf (5.29) of the $GI - GI - 1$ model.

In the case where $\lambda_T = \lambda_2$ (and thus $\lambda_1 = 0$ and $A(z_1, z_2) = A_2(z_2)$, $\forall z_1$), all packets have low priority, such that every arriving packet will be appended to the queue at some position behind the reservation. Therefore, the $\mathcal{R}$ will be at position $m_k = 1$ at the beginning of every slot k, such that the joint system state pgf becomes $P(x, y, z) = P(x, 0, z)$, which can be obtained as

$$P(x, y, z) = P(x, 0, z) = p_0 \left[1 - xz \frac{(1 - A_2(z))\,(S(x) - S(A_2(z)))}{(x - A_2(z))\,(z - S(A_2(z)))} \right], \tag{6.86}$$

which corresponds to the system state pgf (5.19) of the $GI - GI - 1$ model with $A_2(z)$ as the pgf of the number of packet arrivals per slot. For the packet delay, we note that $A_1(z) = A(z, 1) = A_2(1) = 1$, such that we quickly find

$$D_2(z) = p_0 S(z) \frac{(z-1)\,(1 - A_2(S(z)))}{\lambda_2\,(1 - S(z))\,(z - A_2(S(z)))}. \tag{6.87}$$

From these observations, it can be understood that for either $\lambda_2 = 0$ or $\lambda_1 = 0$, the Reservation discipline has no effect on the functioning of the underlying system, and therefore has no effect on the system's performance. The model described in 6.4 will then break down to a $GI - Geo - 1$ queueing model and the model in 6.5 essentially becomes a $GI - GI - 1$ model.

In our analysis, we mentioned that substitution of $y = 1$ in (6.65) removes the information about the location of the reservation, resulting in

$$P(x, 1, z) = p_0 \left[1 - xz \frac{(1 - A_T(z))\,(S(x) - S(A_T(z)))}{(x - A_T(z))\,(z - S(A_T(z)))} \right], \tag{6.88}$$

which also has the same structure as the corresponding expression (5.19) for the $GI - GI - 1$ model. This expression however, does not allow for determination of the packet delay distribution, since the packet delay depends on the actual position of $\mathcal{R}$ at the moment of insertion. In case there is only one packet class, the reservation remains either fixed at the queue's head (for $\lambda_1 = 0$) or is always positioned at the tail of the queue (for $\lambda_2 = 0$). In such cases, there is no need to include the position of $\mathcal{R}$ in the system state pgf, such that substitution of $y = 1$ only removes redundant information and determination of the packet delay distributions remains possible.

6.7 On the existence of $\aleph$

In course of the calculation of the system state pgfs for both the geometrically distributed service times case and the general independent service times case, we argued that a non-empty subset $\aleph$ exists, allowing us to proceed with the calculations. In this section, we prove that such a subset $\aleph$ of the open unit disk does in fact exist, thus validating our calculations.

Our approach extends a technique used in [26], where an open *annulus* $\aleph = \{z : r < |z| < 1\}$, with $r = A(0,0)/(A(0,0) + 1 - A_1(0))$, was defined. Note that, by definition, this annulus is a subset of the open unit disk, and except for the case where $\lambda_1 = 0$, the set $\aleph$ is non-empty. It was then proven that

$$z \in \aleph \Rightarrow \left| \frac{A(0,z)}{z} \right| < 1, \tag{6.89}$$

which also validates the technique used to determine (6.63) in 6.5.2. As mentioned before, the Reservation discipline has no effect on the overall buffer behavior if $\lambda_1 = 0$, such that the calculation of the system state pgf need not be done as described in Sections 6.4 and 6.5.

In order to validate the technique used to construct the expressions (6.16) in 6.4.2 and (6.64) in 6.5.2, we note that for any $|z| \leq 1$, we find that

$$\begin{aligned} |A(0,z)| &= \left| \sum_{n=0}^{\infty} \text{Prob}[a_1 = 0, a_2 = n]\, z^n \right| \leq \sum_{n=0}^{\infty} \text{Prob}[a_1 = 0, a_2 = n]\, |z|^n \\ &\leq \sum_{n=0}^{\infty} \text{Prob}[a_1 = 0, a_2 = n]\, |z| = A_1(0)\, |z| \leq |z| \leq 1. \end{aligned} \tag{6.90}$$

Using this result and an identical approach we find, again for $|z| \leq 1$, that

$$\begin{aligned} |S(A(0,z))| = \left| \sum_{n=1}^{\infty} s(n) A(0,z)^n \right| &\leq \sum_{n=1}^{\infty} s(n) |A(0,z)|^n \\ &\leq \sum_{n=1}^{\infty} s(n)\, |A(0,z)| = |A(0,z)|\,, \end{aligned} \tag{6.91}$$

such that

$$|z| \leq 1 \Rightarrow \left| \frac{S(A(0,z))}{z} \right| \leq \left| \frac{A(0,z)}{z} \right|. \tag{6.92}$$

Combining (6.89) with (6.92) then yields

$$z \in \aleph \Rightarrow \left| \frac{S(A(0,z))}{z} \right| < 1. \tag{6.93}$$

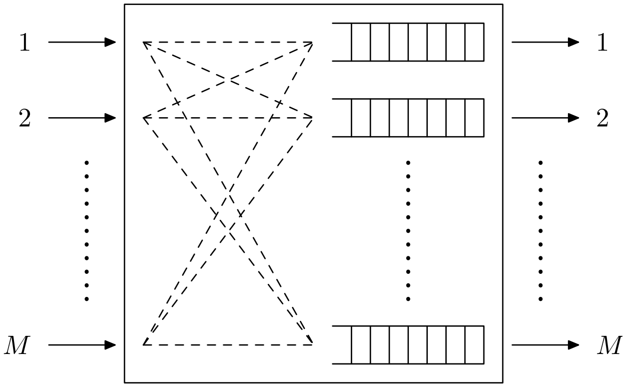

Figure 6.5: A schematic illustration of an $M \times M$ output buffering switch.

6.8 Numerical examples

We now illustrate the impact of the Reservation discipline on the steady-state delay distributions for the packets of either class. Meanwhile, we compare these results with their counterparts in FIFO-scheduled systems and systems operating under the Absolute Priority discipline.

Therefore, we consider a practical example of a non-blocking output buffering switch with M inlets and M outlets as shown in Figure 6.5. When a packet arrives on one of the M switch inlets, it is routed to the output buffer of the outlet corresponding to the packet's destination. We assume this internal routing to be independent for each individual packet and that each inlet feeds each outlet uniformly. In case of *iid* Bernoulli arrivals on each of the inlets with a combined arrival rate of λ_T, the total number of packet arrivals to any of the switch's outlet buffers per slot then follows the binomial distribution noted $\mathrm{B}(M, \lambda_T/M)$, with pgf

$$A_T(z) = \left(1 - \frac{\lambda_T}{M} + \frac{\lambda_T}{M} z\right)^M . \tag{6.94}$$

We now assume that each packet arriving on one of the inlets has a probability $\alpha \triangleq \lambda_1/\lambda_T$ to be a high-priority packet and a probability $1-\alpha = \lambda_2/\lambda_T$ to be a low-priority packet. The joint pgf $A(z_1, z_2)$ of the number of packet arrivals per slot of either class at each of the outlet buffers can then be found as

$$A(z_1, z_2) = A_T(\alpha z_1 + (1-\alpha)\, z_2) = \left(1 - \frac{\lambda_1}{M}\,(1 - z_1) - \frac{\lambda_2}{M}\,(1 - z_2)\right)^M . \tag{6.95}$$

Due to the Bernoulli distributed arrivals at each inlet, at most M packets can arrive during a single slot and there is a negative correlation between

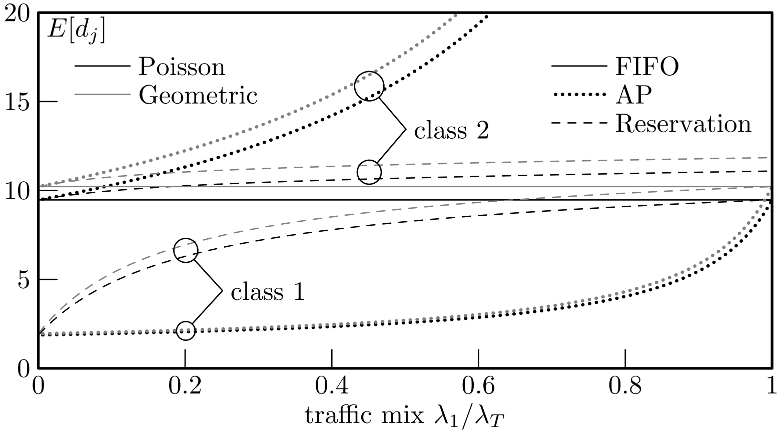

Figure 6.6: Mean packet delay for both packet classes as a function of λ_1/λ_T for different scheduling disciplines and different service time distributions.

the numbers of packet arrivals of either class per slot. For the queueing system's server, we consider shifted Poisson distributed service times, with pgf

$$S(z) = ze^{(\mu-1)(z-1)}. \tag{6.96}$$

First, we focus on the impact of the traffic mix λ_1/λ_T on the mean packet delay. Therefore, we consider an output buffer of a non-blocking 16×16 output buffering switch with a total arrival rate of $\lambda_T = 0.6$. For the service process, we consider both a server with shifted geometric service times as in Section 6.4 and a server with shifted Poisson distributed service times as an example of the model studied in Section 6.5. For both servers, the mean service length is set to $\mu = 1.5$, such that the system load in both cases is $\rho = 0.9$. In Figure 6.6, we have plotted the mean packet delay for packets of either class in this system in case of FIFO scheduling, AP scheduling and the Reservation discipline. Note that in case of FIFO scheduling, all packets are treated equal and no service differentiation is performed. Because of this and due to the specific nature of the arrival process, this yields that the delay distributions of packets of either class coincide with the delay distribution of an arbitrary packet. As such, we have for each of the servers that both packet classes result in one overlapping curve for the mean packet delay in the FIFO case. As expected, the curves pertaining to the mean packet delay $\mathrm{E}[d_j]$ of packets of class j in case of the Reservation discipline are contained between the FIFO curves and the corresponding AP curves. The class-1 packet delay is clearly reduced as compared to FIFO but not as much as would have been obtained with AP. Conversely, even for high partial class-1 loads, the class-2 packet delay in the

Reservation system remains neatly restricted to acceptable values not much different from FIFO, whereas the AP scheduled system suffers from packet starvation. We see that for very low partial class-1 loads, the differences in delay characteristics between AP scheduling and the Reservation discipline are negligible. In this case, most class-1 packets will enter the queue while the reservation is located at the queue's head, such that they can benefit plenty from the Reservation discipline. As the partial class-1 load increases, the reservation is more frequently seized and is less likely to be positioned at the beginning of the queue. Therefore, the gain in class-1 delay performance decreases more quickly for the Reservation discipline than for AP as λ_1/λ_T increases. For high partial class-1 loads, AP looses most of its effectiveness and the mean class-1 delay $\mathrm{E}[d_1]$ progressively increases and approaches the mean class-1 delay obtained for the Reservation discipline. Under the Reservation discipline, an increase in the partial class-1 load only causes an increase in the probability of a class-2 packet to be jumped over by a class-1 packet, whereas under AP there is an increase in the number of class-1 packets overtaking a class-2 packet. This results in a small increase for the mean class-2 packet delay for the Reservation discipline, as opposed to a progressive increase in the AP case.

Next, we look at the impact of the total arrival rate λ_T for fixed values of the traffic mix λ_1/λ_T and the mean service rate $1/\mu$. We consider the same 16 × 16 output buffering switch as before, but we now only focus on the system with shifted Poisson distributed service times. The traffic mix is assumed to be set at $\lambda_1/\lambda_T = 0.15$ and the mean service length is $\mu = 1.5$. The load ρ covers the interval $]0, 1[$ and for each value of ρ, the total arrival rate is then determined as $\lambda_T = \rho/\mu$. Figure 6.7 shows the mean packet delay for packets of either class for FIFO scheduling, AP scheduling and the Reservation discipline. Again, we see that the curves corresponding to the Reservation discipline are wedged between the FIFO curve and the curves representing AP scheduling. Remarkably, the curve for the mean class-1 delay for the AP case is limited for all values of the load, whereas all other curves increase excessively under high load conditions. This can be understood by the fact that under AP, the class-1 packets are allowed to consume all the available system capacity as they want, leaving the class-2 packets with whatever is left of this capacity. From a class-1 packet point of view, an AP scheduled system is then stable as long as $\lambda_1\mu < 1$, resulting in bounded values for $\mathrm{E}[d_1]$ even when the total system load is 1. In the Reservation case, we see that for low to moderate load conditions, the mean class-1 packet delay $\mathrm{E}[d_1]$ hardly differs from its AP counterpart. For high system loads however, more class-1 packets will be inserted closeby the queue's tail and the mean class-1 packet delay starts to increase progressively.

Similar effects can be seen when we plot the mean packet delay for packets of either class as a function of the service rate $1/\mu$, as depicted in Figure 6.8. Here, the total arrival rate at an arbitrary output buffer of the

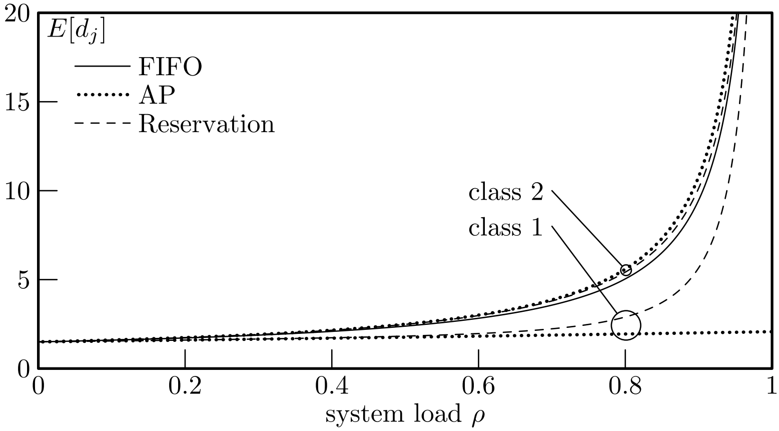

Figure 6.7: Mean packet delay for both packet classes as a function of the system load $\rho = \lambda_T \mu$ for different scheduling disciplines.

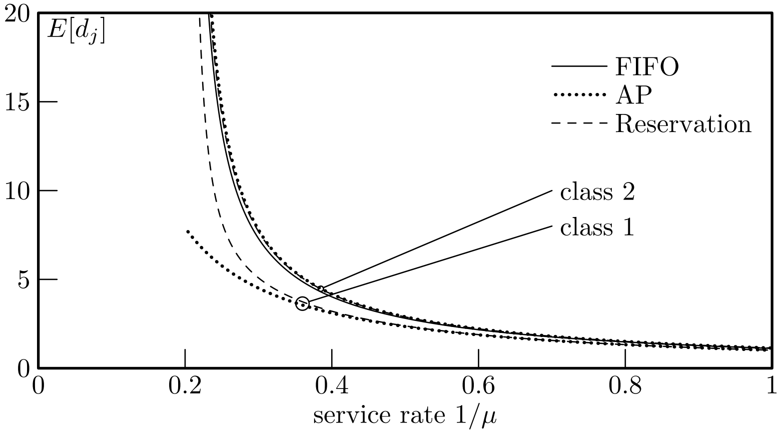

Figure 6.8: Mean packet delay for both packet classes as a function of the service rate $1/\mu$ for different scheduling disciplines.

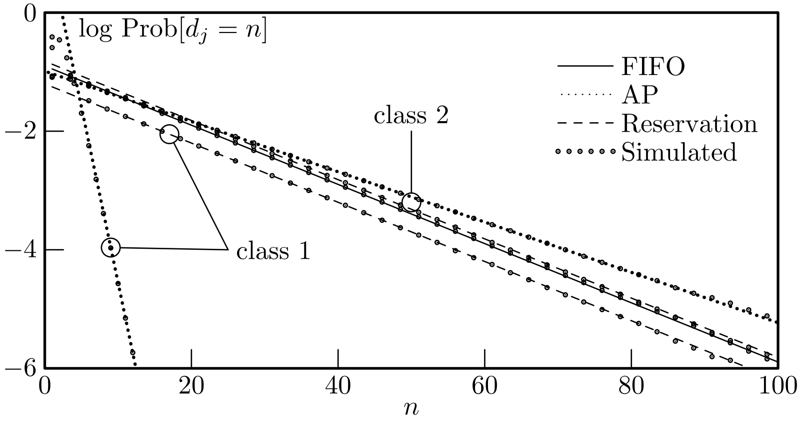

Figure 6.9: The packet delay pmf, simulated (dots) and approximated (lines) using (6.82) for both packet classes for different scheduling disciplines.

16×16 switch is fixed at $\lambda_T = 0.2$, with a traffic mix of $\lambda_1/\lambda_T = 0.15$. The service rate $1/\mu$ of the Poisson server then covers the interval $]1/\lambda_T, 1]$. Similar to our previous findings, we notice that all curves except for the mean class-1 delay in the AP system grow excessively when the service rate drops to $1/\lambda_T$ (because of which the system load becomes 1).

Finally, we take a look at the tail distributions of the packet delay. As before, we consider an output buffer of a non-blocking 16×16 switch with total arrival rate of $\lambda_T = 0.6$ and a traffic mix of $\lambda_1/\lambda_T = 0.15$. The service times are shifted Poisson distributed random variables with mean $\mu = 1.5$. In Figure 6.9, the pmf of the packet delay is plotted on a logarithmic scale for both packet classes and for the usual scheduling disciplines: FIFO, AP and the Reservation discipline. As could be expected from (6.82), both curves corresponding to the Reservation discipline are parallel to the FIFO curve, due to the fact that the packet delay pgfs $D_1(z)$ and $D_2(z)$ for these systems have the same dominant pole $z_d = A_T(S(z_d))$. The fact that both packet classes have the same delay decay rate is an inherent and unique feature of the reservation-based scheduling discipline. Many other service differentiation disciplines provide either a predefined throughput proportionality or an average delay for each packet class, whereas the Reservation discipline provides a proportionality of the *delay quantiles* of both packet streams. Specifically, for small enough α, if we define the delay quantiles d_j^α by $\text{Prob}\big[d_j > d_j^\alpha\big] = \alpha$, $j = 1, 2$ then the spacing $d_2^\alpha - d_1^\alpha$ is constant, i.e. independent of α. This could be seen as a property of *asymptotic delay fairness* between the packet streams where under no circumstance it can happen that any of the streams exhibits a faster or slower delay decay rate than the other. It is conceivable that this property can be advantageous in applications where the perceived fairness is mainly determined by the perception of very long delays. This is clearly not the case for the AP discipline, where we see that the class-1 delay curve drops much more steeper than the class-2 delay curve. This implies that high delays become much more unlikely much quicker for class-1 packets than for class-2 packets.

From the previous examples, it can be seen that the Reservation discipline achieves a reasonable degree of service differentiation without the side-effect of packet starvation. When used under appropriate conditions, such as a small relative class-1 load and a total load that is not very high, the gain in delay performance as experienced by class-1 packets only differs slightly from AP, whereas the drawback for the class-2 packets is negligible. In less than ideal conditions, the delay performance for both packet classes however tends more towards the performance seen in FIFO scheduling. In such circumstances, the performance of the Reservation discipline can be enhanced by the insertion of multiple reservations $\mathcal{R}$ instead of just one. The system would then be initialized with N reservations at positions 1 to N in the queue and an arriving class-1 packet would then seize the reservation closest to the server, after which a new reservation is appended at the queue's tail. The number of reservations N could then be tuned to meet the delay requirements, even in less than ideal circumstances. In Figure 6.10, we present simulation results for the packet delay pmf for either packet class on a logarithmic scale for FIFO scheduling, AP scheduling and the Reservation discipline with N reservations, for multiple values of N. The system parameters of Figure 6.10 are identical to the parameters for Figure 6.9. As expected, we see that the additional reservations indeed result in a larger distinction between the two traffic classes. This is illustrated by the fact that for increasing values of N, the tail probabilities move further apart, towards the tail probabilities of the AP system. Furthermore, we see that the decay rate of the delay tail distributions is independent of the number of reservations N, such that the tails of the curves are all parallel to each other. Analytical results for systems operating under the Reservation discipline with $N > 1$ reservations in case of deterministic service times of 1 slot per packet can be found in [27, 29]. For more general service processes however, the analysis of the system with multiple reservations has not been done. Note that to study the Reservation discipline with $N > 1$ reservations, the analysis method presented in this chapter will need to be further extended and a $(N + 2)$-dimensional state description will be required, as one will need to keep track of the positions of all N reservations. One can expect that such analysis will be quite involved.

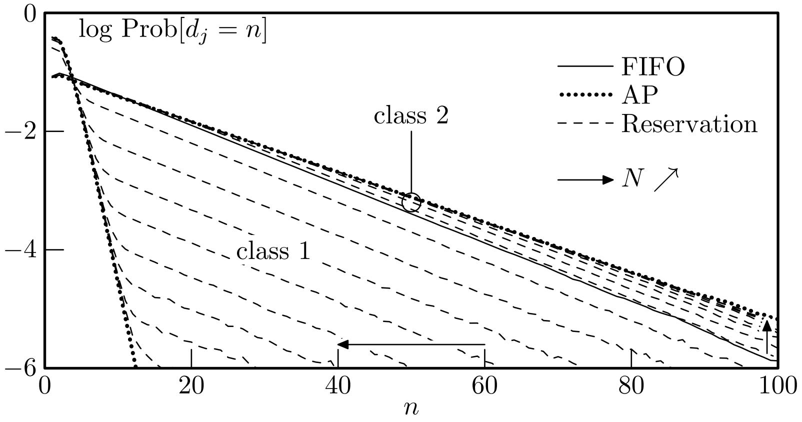

Figure 6.10: The packet delay pmf simulated for both packet classes for FIFO scheduling, Absolute Priority scheduling and the Reservation discipline with N reservations ($N \in 1, 3, 5, 7, 9, 11, 13, 15$).

Chapter 7

The NT-Policy

7.1 Introduction

In many application fields of queueing theory, the delay is generally considered to be the main performance parameter of queueing systems. This approach is common in wired computer networks and telecommunication systems in general, but in other applications other parameters must be taken into account as well. These other parameters can be financial or even ecological in nature, such as operating costs, resource usage, power consumption, ... We refer to these parameters as *cost*s which we want to reduce, while still offering an acceptable service. Especially in systems where the service unit must undergo some costly initialization procedure after an idle period (like powering up, recalibration, ...) or where leaving the server activated but idle is costly, it might be beneficial to shape the stream from the queue to the server such that work is clustered. Under low to moderate load conditions for example, the service unit of the unaltered $GI-GI-1$ model will exhibit frequent switch-overs between being active and being inactive. Application of a clustering mechanism would cause the duration of active and idle periods to be greater and the number of switch-overs to be smaller. Given that this clustering is more suitable in a general operations research context, and less in a telecommunication context, we will use *customers* as the operative word referring to the items passing through the queueing system, instead of *packets*.

Clustering is usually achieved by applying a threshold policy to the system that blocks the access to the service unit until a certain parameter reaches a certain threshold. Probably the most intuitive threshold policy is the so-called N-Policy, where a fixed number of N (> 1) customers have to accumulate in the queue before the server is activated. The service unit then starts serving all customers in succession until the system becomes empty, after which the server is deactivated and remains idle until the threshold N is reached again. This straightforward approach has the benefit of being easy to implement, but it also has the weakness that when N is chosen too large, relative to the customer arrival rate, then the time needed to reach the threshold can become excessive. The N-policy was first presented and studied in continuous time in [125], and various adaptations [69, 77, 117] have been developed since. In discrete time, batch arrival and service for N-policy queues has been studied in [6], a bi-level threshold mechanism is studied in [59] and differentiated service between the N accumulated customers and later arrivals is studied in [91].

Another intuitive threshold policy is the T-policy. Under this policy, a customer arriving to an empty system will have to wait a fixed number of T (> 1) slots until the server is activated. Just like with the N-policy, the server then starts serving customers, until the system becomes empty and the server is deactivated again. Obviously, this approach avoids unacceptable delays, even for extremely low arrival rates, but in general it achieves less efficient clustering. The T-policy was studied in [60, 118].

In order to incorporate the best of both policies without the weaknesses, the hybrid NT-policy was developed as a combination of the N-policy and the T-policy. Under the NT-policy, the server is activated as soon as one of the thresholds is reached, i.e. when N customers have accumulated in the queue, or when the first customer in the queue has been waiting for T slots, whichever happens first. In continuous time, the NT-policy has been studied in [3, 68, 76].

In this chapter, we will revisit my contributions concerning the analysis of the NT-policy in a single-server infinite-capacity discrete-time queueing system. In the next section, we give a detailed description of the NT-policy. Then, we present a general mathematical model that will allow for a detailed analysis of the system. In the two subsequent sections, we analyze the system content, the customer delay and the effects of the NT-policy, for both deterministic service times of exactly 1 slot as presented in my paper [37] and general independent service times as presented in my publications [41, 43].

7.2 The NT-policy

In this section we describe in full detail how the NT-policy works, not only for better understanding, but because a good insight in the functioning of the NT-policy will prove useful in our analysis. Figure 7.1 illustrates the

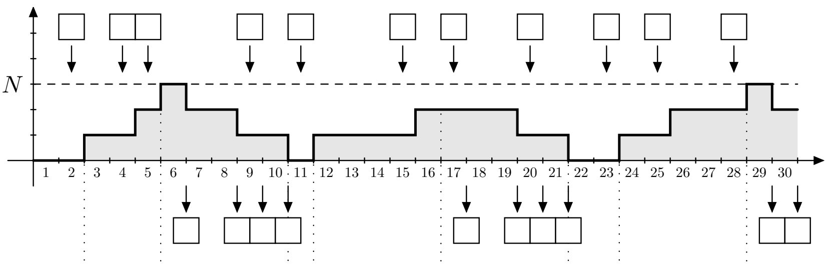

Figure 7.1: Example of how the NT-policy affects the queueing system.

behavior of a queueing system with variable service times, operating under the NT-policy, for $N = 3$ and $T = 5$. The initial situation depicted in Figure 7.1 is such that the system is empty and therefore, the server is deactivated. During slot 2, the first customer arrives to the empty system and a timer is initialized to monitor the time spent in the queue by that first customer. During slot 4, a second customer arrives and is added to the queue without consequence. But when the third customer arrives during slot 5, the N-threshold is reached and the server is activated such that the first service can start at the beginning of the next slot (slot 6 in this case). The server then keeps serving customers until the system becomes empty again at the end of slot 10, such that the customer arriving in slot 11 finds the queue empty and the server deactivated. Again a timer is started and when it reaches T in slot 16, the server is activated again, although only one more customer has arrived. Starting from slot 17 the server can then resume serving customers, only to be deactivated when the system becomes empty again at the end of slot 21.

From the argumentation above and Figure 7.1, it can be verified that the system operates in a non-periodic cyclic pattern, consisting of three phases delimited by the dotted lines. In what follows, we will refer to these phases as *empty*, *accumulating* customers and *serving* customers.

Note that throughout this chapter, we will assume $1 < N \leq T$, in order to guarantee that none of the thresholds is obsolete. Mind that for $N = 1$, the N-threshold will be reached immediately on insertion of a customer into an empty queue, such that the system in fact becomes a traditional queueing system, i.e. without a threshold policy. Furthermore, our analysis is limited to a Bernoulli arrival process, such that at most 1 customer can arrive during a slot and the interarrival times are geometrically distributed. This implies that the minimum number of slots to reach the N-threshold, after insertion of the first customer, is $N - 1$ slots. Choosing $N > T$ essentially makes it impossible to reach the N-threshold before the T-threshold, such that the N-threshold is obsolete and in fact we are left with a T-policy system.

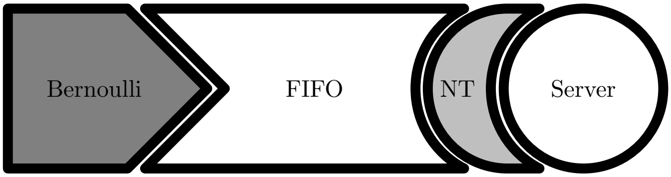

Figure 7.2: Illustration of the NT-policy system.

7.3 Mathematical model

The mathematical model used in our analysis, corresponds to a special case of the $GI-GI-1$-model with a Bernoulli arrival process and where access to the server is controlled by the NT-policy, as depicted in Figure 7.2. Customers therefore arrive at a fixed rate of λ customers per slot such that every slot exactly 1 arrival occurs with probability λ and with probability $1-\lambda$ there is no arrival. The pgf $A(z)$ of the number of arrivals a_k during slot k is hence given by

$$A(z) \triangleq \mathrm{E}[z^{a_k}] = 1 - \lambda + \lambda z. \tag{7.1}$$

For later use, we introduce the random variable A_n as the total number of arrivals in n consecutive slots. The pgf $A_n(z)$ of this random variable can then be found as

$$A_n(z) = A(z)^n = (1-\lambda+\lambda z)^n = \sum_{j=0}^{n} \binom{n}{j} (1-\lambda)^{n-j} (\lambda z)^j. \tag{7.2}$$

Related to this, we also define the random variable c_n as the number of slots needed to collect n arrivals, such that

$$\mathrm{Prob}[c_n = t] = \binom{t-1}{n-1} \lambda^n (1-\lambda)^{t-n}, \qquad t \geq n. \tag{7.3}$$

Note that c_n has a negative binomial distribution, such that its pgf $C_n(z)$ is given by

$$C_n(z) \triangleq \mathrm{E}[z^{c_n}] = \left(\frac{\lambda z}{1-(1-\lambda)\,z} \right)^n. \tag{7.4}$$

We will analyze this queueing system for deterministic service times equal to exactly one slot per customer and general independent service times separately in the next two sections. Different types of analysis methods will be used in these sections.

7.4 Deterministic service times

In this section, we limit our analysis to service times of exactly one slot per customer, as depicted in Figure 7.3. Although this assumption is rather

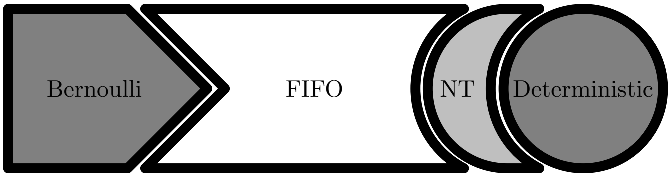

Figure 7.3: Illustration of the NT-policy system with deterministic service times.

restrictive, it will allow us to become acquainted with the NT-policy, before we study the NT-policy in a system with general service times.

7.4.1 System equations

From the description of the NT-policy in the previous sections, it is clear that the system's behavior highly depends on which phase the system is in. Let us therefore introduce the random variable $\phi_k \in \{0, 1, 2\}$ to denote the phase of the system at the beginning of a random slot k. Note that phase transitions can only occur at slot boundaries, such that the phase of the system remains unaltered during a slot. The random variable ϕ_k can only take the values 0, 1 and 2, referring to the empty, the accumulating and the serving phase respectively.

During the accumulating phase, we need to monitor the time spent in the queue by the first customer. Thus, the random variable t_k is defined as the integer number of slots the first customer has been in the queue at the *end* of a random slot k in the accumulating phase. Due to the fact that phase transitions only occur at slot boundaries, t_k can already be determined at the *beginning* of slot k. Thus we have that $1 \leq t_k \leq T$ for any accumulating phase slot k. If $t_k = T$, then for sure the system proceeds to the serving phase in slot $k + 1$. We extend the definition of t_k such that $t_k = 0$ during either the empty or the serving phase.

As usual, we also require the system content u_k at the beginning of a random slot k to complete the system state vector. During the empty and the accumulating phase, the system content can only grow following the arrival process, whereas during each serving phase slot there will be a departure.

The system equations that relate the system state vector $\langle \phi_k, t_k, u_k \rangle$ at slot k with its slot $k + 1$ counterpart follow from the description of the NT-policy as

- if $\phi_k = 0$:

$$\begin{aligned}
\phi_{k+1} &= \begin{cases} 0, & \text{if } a_k = 0, \\ 1, & \text{if } a_k > 0, \end{cases} \\
t_{k+1} &= \begin{cases} 0, & \text{if } a_k = 0, \\ 1, & \text{if } a_k > 0, \end{cases} \\
u_{k+1} &= a_k,
\end{aligned} \tag{7.5}$$

- if $\phi_k = 1$:

$$\begin{aligned}
\phi_{k+1} &= \begin{cases} 1, & \text{if } t_k < T \text{ and } u_k + a_k < N, \\ 2, & \text{if } t_k = T \text{ or } u_k + a_k = N, \end{cases} \\
t_{k+1} &= \begin{cases} t_k + 1, & \text{if } t_k < T \text{ and } u_k + a_k < N, \\ 0, & \text{if } t_k = T \text{ or } u_k + a_k = N, \end{cases} \\
u_{k+1} &= u_k + a_k,
\end{aligned} \tag{7.6}$$

- if $\phi_k = 2$:

$$\begin{aligned}
\phi_{k+1} &= \begin{cases} 0, & \text{if } u_k = 1 \text{ and } a_k = 0, \\ 2, & \text{if } u_k > 1 \text{ or } a_k > 0, \end{cases} \\
t_{k+1} &= 0, \\
u_{k+1} &= u_k - 1 + a_k.
\end{aligned} \tag{7.7}$$

7.4.2 Buffer analysis

Due to the finite state space, the system lends itself to be analyzed entirely based on state probabilities, rather than the pgf approach we adopt in the other sections of this work. These state probabilities are defined as

$$p_0 \triangleq \text{Prob}[\phi_k = 0]\,, \tag{7.8}$$

$$p_{1,m,n} \triangleq \text{Prob}[\phi_k = 1, t_k = m, u_k = n]\,, \quad 1 \leq n < N, n \leq m \leq T, \tag{7.9}$$

$$p_{2,n} \triangleq \text{Prob}[\phi_k = 2, u_k = n]\,, \qquad 1 \leq n \leq N. \tag{7.10}$$

The state probabilities can be interpreted as the frequency of occurrence of the different states over an infinite time span. For our analysis, it is however beneficial to consider the relative frequency of occurrence of the different states within a single cycle. Note that the state $\langle 1,1,1 \rangle$ is bound to occur exactly once per cycle, as it corresponds to the first slot of the accumulating phase. The states $\langle 0,0,0 \rangle$ and $\langle 2,0,n \rangle$ on the other hand can occur multiple times per cycle. For a given value of m the states $\langle 1,m,n \rangle$ for different values of n are mutually exclusive within a cycle, such that most of them will even never occur during a random cycle at all. We therefore define the coefficients corresponding to the relative frequency of occurrence of the different states within a single cycle as

$$q_0 \triangleq \frac{p_0}{p_{1,1,1}}, \qquad q_{1,m,n} \triangleq \frac{p_{1,m,n}}{p_{1,1,1}}, \qquad q_{2,n} \triangleq \frac{p_{2,n}}{p_{1,1,1}}. \tag{7.11}$$

In order to determine the coefficients in (7.11), we recall the system equations and the observations leading towards the system equations. If the system is empty at the beginning of slot k it will move to the accumulating phase upon the arrival of a customer, which has a probability λ to occur. So we find that

$$p_{1,1,1} = \lambda p_0 \Leftrightarrow q_0 = \frac{1}{\lambda}. \tag{7.12}$$

A certain accumulating phase state $\langle 1, m, n\rangle$ $(1 < m)$ can only be reached if there were n customers in the queue already in the previous slot and no arrival occurred, or if the n'th customer arrived during that previous slot. This observation leads to

$$\begin{aligned} q_{1,m,n} &= (1-\lambda)\, q_{1,m-1,n} + \lambda q_{1,m-1,n-1} \\ &= \binom{m-1}{n-1} \lambda^{n-1}(1-\lambda)^{m-n}, \qquad\qquad 1 \le n \le m, \end{aligned} \tag{7.13}$$

where we silently assumed that $q_{1,m,0} = 0, \forall m$ since these coefficients relate to system states that can never occur. Remarkably, the coefficients $q_{1,m,n}$ are independent of either threshold N and T. Therefore, although state $\langle 1, T+1, n\rangle$ can never occur in the system, we can calculate $q_{1,T+1,n}$ $(n < N)$ from (7.13) to account for state transitions due to the T-threshold. This allows us to express the serving phase coefficients $q_{2,n}$ for $n < N$ as

$$\begin{aligned} q_{2,n} &= q_{1,T+1,n} + (1-\lambda)\, q_{2,n+1} + \lambda q_{2,n} \\ &= \frac{q_{1,T+1,n}}{1-\lambda} + q_{2,n+1} = \frac{1}{1-\lambda} \sum_{j=n}^{N-1} q_{1,T+1,j} + q_{2,N}. \end{aligned} \tag{7.14}$$

The state $\langle 2, 0, N\rangle$ can either be reached from the accumulating phase if the N-threshold is reached or from state $\langle 2, 0, N\rangle$ itself, if an arrival occurs. Thus, we have

$$q_{2,N} = \lambda \sum_{m=N-1}^{T} q_{1,m,N-1} + \lambda q_{2,N} = \frac{\lambda}{1-\lambda} \sum_{m=N-1}^{T} q_{1,m,N-1}. \tag{7.15}$$

Finally, we calculate the sum of the coefficients for all possible system states as defined in (7.11).

$$q_0 + \sum_{n=1}^{N-1} \sum_{m=n}^{T} q_{1,m,n} + \sum_{n=1}^{N} q_{2,n} = \frac{1}{p_{1,1,1}} \left(p_0 + \sum_{n=1}^{N-1} \sum_{m=n}^{T} p_{1,m,n} + \sum_{n=1}^{N} p_{2,n} \right) \tag{7.16}$$

Application of the normalization condition then allows to find the probability $p_{1,1,1}$ as

$$p_{1,1,1} = \left(q_0 + \sum_{n=1}^{N-1} \sum_{m=n}^{T} q_{1,m,n} + \sum_{n=1}^{N} q_{2,n} \right)^{-1}. \tag{7.17}$$

With $p_{1,1,1}$ now determined, we can determine all other system state probabilities from (7.11) and the closed-form expressions (7.12)-(7.15).

7.4.3 Phase durations and cycle length

Before we proceed to the delay analysis, we first determine how many slots there are in a random cycle and in the phases that constitute that cycle. Therefore we define the random variables Φ_i ($i \in \{0,1,2\}$) as the phase i duration, i.e. the number of slots in a random phase i, with pgf $\Phi_i(z)$.

Empty phase

The empty phase starts once the last customer leaves the system such that it becomes empty again and ends as soon as a new customer arrives. The empty phase therefore consists of as many slots as needed to collect a single arrival, given by the shifted geometrically distributed random variable c_1. We therefore find the pmf of the empty phase duration as

$$\text{Prob}[\Phi_0 = t] = \lambda (1-\lambda)^{t-1}, \qquad\qquad t \geq 1, \tag{7.18}$$

with pgf

$$\Phi_0(z) \triangleq \text{E}\big[z^{\Phi_0}\big] = \frac{\lambda z}{1-(1-\lambda)\,z}. \tag{7.19}$$

The mean empty phase pgf then follows as

$$\text{E}[\Phi_0] = \Phi_0'(1) = \frac{1}{\lambda}. \tag{7.20}$$

Accumulating phase

In order to reach the N-threshold, there must be $N-1$ arrivals during the accumulating phase, given the fact that at the beginning of the accumulating phase there is already 1 customer in the queue. Note that this takes c_{N-1} slots, with pmf $\text{Prob}[c_{N-1} = t] = \lambda\, q_{1,t,N-1}$ ($t \geq N-1$). If the N-threshold is not reached when the timer reaches $T-1$, the accumulating phase is ended due to the T-threshold. In short, the accumulating phase takes as many slots as needed to collect $N-1$ (more) arrivals, with a maximum of T slots. Thus, we find the pmf of the accumulating phase duration as

$$\text{Prob}[\Phi_1 = t] = \begin{cases} \lambda\, q_{1,t,N-1}, & N-1 \leq t \leq T-1, \\ \sum_{n=1}^{N-1} q_{1,T,n}, & t = T. \end{cases} \tag{7.21}$$

The pgf of the accumulating phase duration is then given by

$$\Phi_1(z) \triangleq \mathrm{E}\big[z^{\Phi_1}\big] = \lambda \sum_{m=N-1}^{T-1} q_{1,m,N-1} z^m + z^T \sum_{n=1}^{N-1} q_{1,T,n}, \tag{7.22}$$

such that we find the mean number of slots in the accumulating phase as

$$\mathrm{E}[\Phi_1] = \Phi_1'(1) = \lambda \sum_{m=N-1}^{T-1} m\ q_{1,m,N-1} + T \sum_{n=1}^{N-1} q_{1,T,n}. \tag{7.23}$$

An interesting measure of the NT-policy system is the probability ω that a transition from the accumulating to the serving phase within a cycle occurs due to the N-threshold. This probability can be calculated as

$$\begin{aligned} \omega &\triangleq \mathrm{Prob}[N \text{ customers have accumulated during } \Phi_1] \\ &= \lambda \sum_{m=N-1}^{T} q_{1,m,N-1}. \end{aligned} \tag{7.24}$$

Serving phase

The determination of the serving phase duration is less straightforward due to the fact that the number of customers in the queue at the beginning of the serving phase is unknown and that additional customers, arriving in the course of the serving phase itself, must also be served. The number of customers being served during a serving phase is therefore not fully determined a priori. In order to resolve the issue of the additional customers, we introduce the random variable Δ as the number of slots needed to reduce the number of customers in the system by 1. Note that after every serving phase slot with no arrival, the system content will decrease with 1 due to the single slot service times. The system content will however remain unchanged if a customer does arrive. Therefore, Δ corresponds to the number of slots until a non-arrival slot, with pmf

$$\mathrm{Prob}[\Delta = t] = \lambda^{t-1}\left(1-\lambda\right), t \geq 1, \tag{7.25}$$

and pgf

$$\Delta(z) \triangleq \mathrm{E}\big[z^{\Delta}\big] = \frac{(1-\lambda)\, z}{1-\lambda z}. \tag{7.26}$$

The mean number of slots needed to reduce the system content by 1 is then given by

$$\mathrm{E}[\Delta] = \Delta'(1) = \frac{1}{1-\lambda}. \tag{7.27}$$

The initial number of customers in the queue at the beginning of the serving phase is correlated to the preceding accumulating phase duration.

Note that if $\Phi_1 < T$, the serving phase was triggered by the N-threshold, whereas for $\Phi_1 = T$, there could be less than N customers in the queue. To account for this correlation, we first determine the joint pgf $\Phi_{1,2}(x, y)$ of Φ_1 and the subsequent Φ_2 as

$$\begin{aligned}\Phi_{1,2}(x,y) &\triangleq \mathrm{E}\big[x^{\Phi_1} y^{\Phi_2}\big] \\ &= \lambda \Delta(y)^N \sum_{m=N-1}^{T-1} q_{1,m,N-1} x^m + x^T \sum_{n=1}^{N-1} q_{1,T,n} \Delta(y)^n A(\Delta(y)). \quad (7.28)\end{aligned}$$

The pgf $\Phi_2(z)$ and the expected value $\mathrm{E}[\Phi_2]$ of the duration of a random serving phase can then be found from (7.28) as

$$\begin{aligned}\Phi_2(z) &= \Phi_{1,2}(1,z) \\ &= \lambda \Delta(z)^N \sum_{m=N-1}^{T-1} q_{1,m,N-1} + \sum_{n=1}^{N-1} q_{1,T,n} \Delta(z)^n A(\Delta(z)), \qquad (7.29)\end{aligned}$$

and

$$\mathrm{E}[\Phi_2] = \Phi_2'(1) = \frac{1}{1-\lambda} \left(\lambda N \sum_{m=N-1}^{T-1} q_{1,m,N-1} + \sum_{n=1}^{N-1} (n+\lambda)\, q_{1,T,n} \right). \quad (7.30)$$

Cycle length

The total length Q of an arbitrary cycle can then be found as the sum of the durations of the three constituting phases. Note that we must take the correlation between the accumulating and serving phases into account, such that the pgf $Q(z)$ of the cycle length is given by

$$\begin{aligned}Q(z) &\triangleq \mathrm{E}\big[z^{\Phi_0+\Phi_1+\Phi_2}\big] = \Phi_0(z)\Phi_{1,2}(z,z) \\ &= \frac{\lambda z}{1-(1-\lambda)\,z} \left(\lambda \Delta(z)^N \sum_{m=N-1}^{T-1} q_{1,m,N-1} z^m \right. \\ &\qquad \left. + z^T \sum_{n=1}^{N-1} q_{1,T,n} \Delta(z)^n A(\Delta(z)) \right). \qquad (7.31)\end{aligned}$$

The mean cycle length could be found by taking the first derivative of (7.31) for $z = 1$, however this would be needlessly cumbersome. A better approach comes from the argumentation concerning the definition of the state coefficients $q_{\dots}$. Given that the state $\langle 1, 1, 1 \rangle$ occurs exactly once per cycle, $p_{1,1,1}$ serves as the rate at which cycles succeed each other and the mean cycle length can be found as

$$\mathrm{E}[Q] = \frac{1}{p_{1,1,1}} = q_0 + \sum_{n=1}^{N-1} \sum_{m=n}^{T} q_{1,m,n} + \sum_{n=1}^{N} q_{2,n}. \qquad (7.32)$$

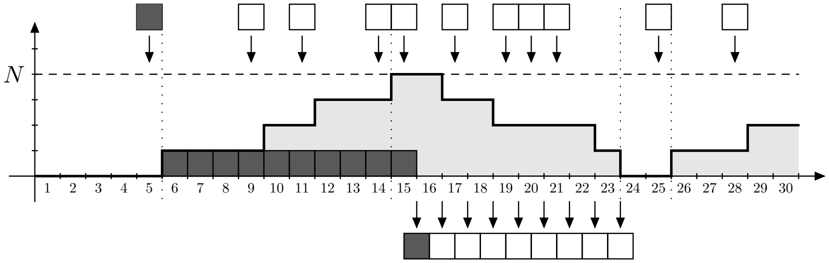

Figure 7.4: Illustration of d_0. The darkened squares represent $\mathcal{C}_0$.

Phase probabilities

Finally, we define the phase probabilities $p_i \triangleq \text{Prob}[\phi_k = i]$ $(i \in \{0, 1, 2\})$ that describe the probability of the system to be in a certain phase. Note that the probability p_0 was defined this way earlier. A phase probability p_i can be understood as the fraction of time slots the system is in phase i and can be found as

$$p_i = \frac{\mathrm{E}[\Phi_i]}{\mathrm{E}[Q]}. \tag{7.33}$$

This is particularly helpful to find $p_0 = (\lambda\,\mathrm{E}[Q])^{-1}$. The probability for the server to be active can be found more efficiently from the assumption that the system is stable and reaches equilibrium. As explained in section 3.1, in equilibrium, the actual average number of departures per slot must be equal to the arrival rate. Given that departures exclusively occur every serving phase slot, the actual departure rate is equal to p_2, such that $p_2 = \lambda$. From p_0 and p_2, p_1 can then be found from the normalization condition as $p_1 = 1 - p_0 - p_2$.

7.4.4 Customer delay analysis

Given that the system behaves very differently over the various phases of a cycle, we will perform the delay analysis for the three phases separately. Let $\mathcal{C}_i$ $(i \in \{0, 1, 2\})$ therefore be a random customer that arrives during phase i of a cycle and $\mathcal{S}$ be the arrival slot of that customer. Even though $\mathcal{S}$ is not a random slot, the BASTA property (see Section 3.3) yields that the system state distribution at the beginning of $\mathcal{S}$ is stochastically identical to that of a random phase i slot. This will allow us to find the distribution of the delay d_i experienced by $\mathcal{C}_i$ from the system state distribution at the beginning of slot $\mathcal{S}$.

Empty phase

If a customer $\mathcal{C}_0$ arrives during the empty phase, it will initiate an accumulating phase at the beginning of slot $\mathcal{S} + 1$. Once the accumulating phase

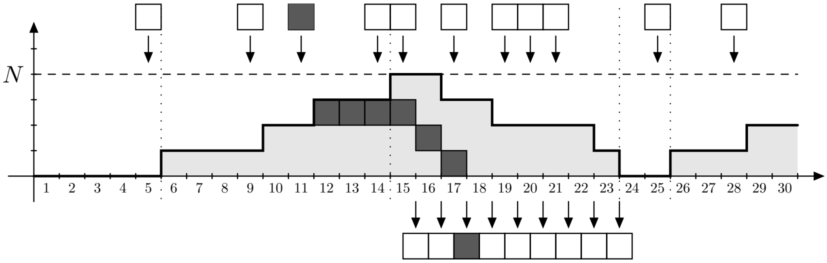

Figure 7.5: Illustration of d_1. The darkened squares represent $\mathcal{C}_1$.

is terminated, $\mathcal{C}_0$ will be the first customer to be served and will leave the system 1 slot later, as depicted in Figure 7.4. Thus, we find the empty phase customer delay d_0 as

$$d_0 = \Phi_1 + 1, \tag{7.34}$$

with pgf

$$D_0(z) \triangleq \mathrm{E}\big[z^{d_0}\big] = z\Phi_1(z). \tag{7.35}$$

Accumulating phase

As shown in Figure 7.5, a customer $\mathcal{C}_1$ that arrives during the accumulating phase will first have to wait an unknown number of slots until the server becomes active and all previous arrivals have left the system until it can enter the server. At the beginning of the arrival slot $\mathcal{S}$ of $\mathcal{C}_i$, the system finds itself in state $\langle 1, t_\mathcal{S}, u_\mathcal{S}\rangle$, i.e. there are $u_\mathcal{S}$ customers in the queue and the first customer has already been waiting for $t_\mathcal{S} - 1$ slots. Thus, starting from slot $\mathcal{S}+1$, the server will only be activated once $N - u_\mathcal{S} - 1$ more customers have arrived, or $T - t_\mathcal{S}$ more slots have passed. This time span corresponds to the accumulating phase of an $N'T'$-policy system with $N' \triangleq N - u_\mathcal{S}$ and $T' \triangleq T - t_\mathcal{S}$, such that

$$d_1 = \Phi_1^{(N',T')} + u_\mathcal{S} + 1, \tag{7.36}$$

where $\Phi_1^{(N',T')}$ denotes the accumulating phase duration of the corresponding $N'T'$-policy system. Note that the distribution of $\Phi_1^{(N',T')}$ can only be determined from (7.21) if $1 < N - u_\mathcal{S} \leq T - t_\mathcal{S}$. If however $u_\mathcal{S} = N - 1$ or $t_\mathcal{S} = T$, then slot $\mathcal{S}$ is the final slot of the current accumulating phase, such that $\Phi_1^{(N',T')} = 0$. Application of this argumentation on (7.36) then yields the distribution of the accumulation phase customer delay d_1 as

$$\mathrm{Prob}[d_1 = t + u_\mathcal{S} + 1] = \begin{cases} 1, & N' = 1, T' = 0, t = 0, \\ \lambda\, q_{1,t,N'-1}, & 0 < N' - 1 \leq t \leq T' - 1, \\ \sum_{n=1}^{N'-1} q_{1,T',n}, & t = T'. \end{cases} \tag{7.37}$$

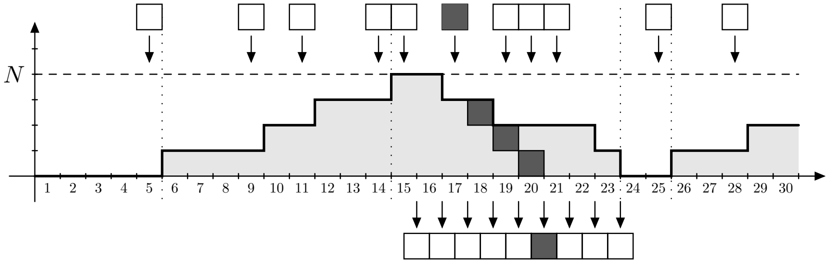

Figure 7.6: Illustration of d_2. The darkened squares represent $\mathcal{C}_2$.

The pgf $D_1(z)$ of d_1 then follows as

$$\begin{aligned} D_1(z) &\triangleq \mathrm{E}\left[z^{d_1}\right] \\ &= \frac{p_{1,1,1}}{p_1}\left[\sum_{n=1}^{N-1} q_{1,T,n} z^{n+1} + z^N \sum_{m=N}^{T-1} q_{1,m,N-1}\right. \qquad (7.38) \\ &\left.+ \sum_{n=1}^{N-2} z^{n+1} \sum_{m=n}^{T-1} q_{1,m,n}\left(\lambda \sum_{t=N-n-1}^{T-m-1} q_{1,t,N-n-1} z^t + z^{T-m} \sum_{k=1}^{N-n-1} q_{1,T-n,k}\right)\right]. \end{aligned}$$

Serving phase

A customer $\mathcal{C}_2$ arriving in course of the serving phase is only delayed by the service of customers, as illustrated in Figure 7.6. The customers adding to the delay d_2 of $\mathcal{C}_2$ are $\mathcal{C}_2$ itself and all customers in the queue at the start of slot $\mathcal{S}$, excluding the one in the server because it will leave the system at the end of slot $\mathcal{S}$. This yields

$$d_2 = u_{\mathcal{S}}, \qquad (7.39)$$

with pgf

$$D_2(z) \triangleq \mathrm{E}\left[z^{d_2}\right] = \sum_{n=1}^{N} \frac{p_{2,n}}{p_2} z^n = \frac{p_{1,1,1}}{\lambda} \sum_{n=1}^{N} q_{2,n} z^n. \qquad (7.40)$$

Customer delay

The delay distribution of a random customer $\mathcal{C}$, regardless of the phase during which $\mathcal{C}$ enters the system, can be found as the weighted sum

$$\mathrm{Prob}[d = t] = p_0 \mathrm{Prob}[d_0 = t] + p_1 \mathrm{Prob}[d_1 = t] + p_2 \mathrm{Prob}[d_2 = t], \qquad (7.41)$$

with corresponding pgf

$$D(z) \triangleq \mathrm{E}\left[z^d\right] = p_0 D_0(z) + p_1 D_1(z) + p_2 D_2(z). \qquad (7.42)$$

7.4.5 Relation to the $GI - GI - 1$ model

As mentioned in section 7.2, the considered NT-policy system becomes a traditional queueing system for $N = 1$, and behaves as if there is no threshold policy controlling the access to the server. Simple substitution of $N = 1$ in the above expressions should however not be performed without extra care, since the above analysis expects every cycle to have an accumulating phase during which the queue is not empty and the server is inactive. In fact, if $N = 1$, there are only two possible system states: the empty system, or the system with one customer in service, with corresponding probabilities

$$p_0 = \text{Prob}[u = 0]\,, \qquad\qquad p_2 = \text{Prob}[u = 1]\,. \tag{7.43}$$

Due to the deterministic service times, an arrival in slot k causes the system to be in the non-empty state during slot $k + 1$, such that we can determine the above probabilities as

$$p_0 = 1 - \lambda, \qquad\qquad p_2 = \lambda. \tag{7.44}$$

The system content pgf $U(z)$ then follows as

$$U(z) = 1 - \lambda + \lambda z = A(z). \tag{7.45}$$

Since every customer is served in the slot following its arrival slot we find the customer delay pgf $D(z)$ as

$$D(z) = z. \tag{7.46}$$

It can be easily verified that (7.45) and (7.46) satisfy the results (5.20) and (5.29) for the $GI - GI - 1$ model with $A(z) = 1 - \lambda + \lambda z$ and $S(z) = z$.

7.4.6 Numerical examples

Before we expand the analysis of the NT-policy to general service times, we first illustrate the system we have analyzed in the above by means of various numerical examples. We will also compare the main results with corresponding results related to either the N-policy or the T-policy.

First, we focus on the mean phase durations and the cycle length as a function of the arrival rate λ. Note that due to the deterministic service times, we have that the system load $\rho = \lambda\mu = \lambda$. Figure 7.7 shows the mean phase durations and mean cycle length on a logarithmic scale for an NT-policy system with parameters set to $N = 41$ and $T = 100$. The mean cycle length of the corresponding N-policy and T-policy systems are plotted as well. The background is filled according to the phase probabilities p_0 (dark grey), p_1 (light gray) and p_2 (white), such that the portion of a vertical cut in a certain background color is equal to the phase probability of the corresponding phase. We see that the mean empty phase duration

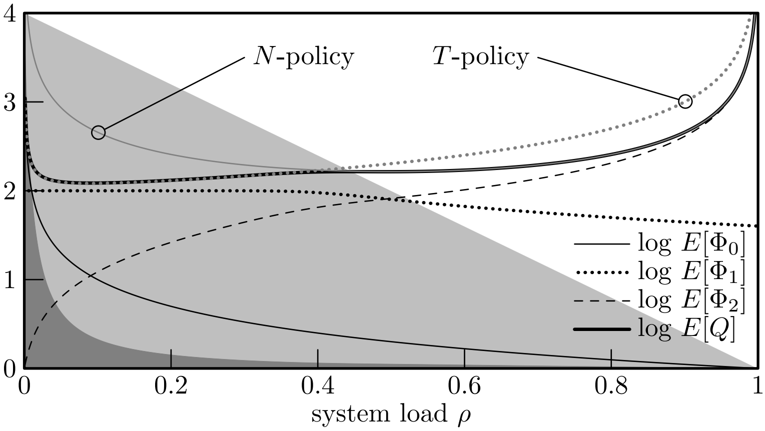

Figure 7.7: The mean phase durations and the mean cycle length (on a logarithmic scale) as a function of the system load $\rho = \lambda$.

drops down as the load ρ increases as $\mathrm{E}[\Phi_0] = 1/\lambda$ according to (7.20). The evolution of the mean accumulating phase length $\mathrm{E}[\Phi_1]$ as a function of the arrival rate λ is not as predictable from its formula (7.23), but is intuitively clear. For low values of λ the probability to reach the N-threshold before the timer expires is small, leading to an expected value of the accumulating phase duration around T slots. For $\lambda > (N-1)/T$, it becomes more and more likely that the N-threshold is actually met, such that the $\mathrm{E}[\Phi_1]$ drops to $(N-1)/\lambda$ for $\lambda \to 1$. Similar to the accumulating phase, the effect of λ on the expected serving phase duration is more clear from intuition than from the corresponding formula (7.30). For very small values of λ, only few customers are in the queue at the beginning of Φ_2 and few customers will arrive during the serving phase, resulting in small values of $\mathrm{E}[\Phi_2]$. As the arrival rate increases, more customers will be in the queue at the beginning of the serving phase and there will be more arrivals during Φ_2, resulting in an increasing mean serving phase duration. As λ continues to increase beyond $(N-1)/T$, the number of initially accumulated customers will saturate at N, whereas the number of additional arrivals during Φ_2 will continue to grow. When looking at the mean cycle length $\mathrm{E}[Q]$, we see that the NT-policy in general yields cycles which are not longer than under either the N-policy or the T-policy. This was to be expected based on the design of the NT-policy.

Now we focus on the phase probabilities as depicted in the background of Figure 7.7. These probabilities are important as they serve as a weight factor in the calculation of the distribution of the delay of an arbitrary customer, as shown in (7.41) and (7.42). We see that the probability of a random slot to be part of an empty phase is high for extremely low arrival

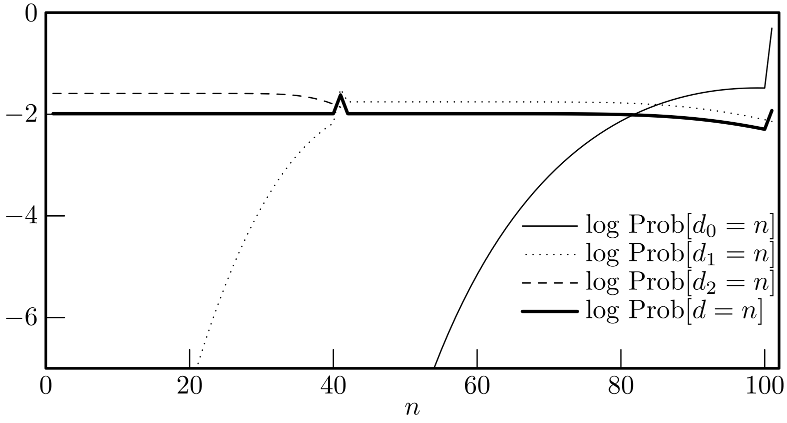

Figure 7.8: The customer delay pmfs.

rates, but it very soon drops as the arrival rate gets higher; for the greater part of the graph, p_0 is even negligible. The probability p_1 runs a very different course: if the arrival rate is very low, the majority of time will be spent in the empty phase, but as the arrival rate increases, the system will be empty for fewer slots per cycle and both p_1 and p_2 will increase. Since the accumulation phase duration is limited to a maximum of T slots, only the serving phase will become longer due to an increasing arrival rate, therefore p_1 will decrease again, while p_2 continues to rise proportionally to the arrival rate.

Next, we take a look at the delay probabilities for the case where $\lambda = 0.4$, $N = 41$ and $T = 100$. In Figure 7.8 we show the delay pmf for customers arriving during each phase individually, and for random customers as well. Note that each of the random variables d_i $(i \in \{0, 1, 2\})$ has its own range of support: $d_0 \in [N, T + 1]$, $d_1 \in [2, T + 1]$ and $d_2 \in [1, N]$. Even though the individual curves are very different, they do add up to a smooth curve for d with two remarkable outliers. The peak at $n = T + 1$ originates from the pmf of d_0 and corresponds to cycles where the timer expires. As such, this peak accumulates all cycles where the T-threshold is met, regardless of the number of customers in the system at the end of the accumulating phase. The peak at $n = N$ is caused by a peak in the pmf of the accumulating phase customer delay d_1. This is illustrated in Figure 7.9, where we show the portions of the pmf of d_1 on a log scale, split up according to the event that triggered the transition from the accumulating phase to the serving phase. From this graph, we can clearly see that the peak at $n = N$ comes from the cases where the N-threshold has been reached. In such cases, the last arriving customer in the accumulating phase will be the Nth customer in the queue and therefore will have a delay of exactly N slots. Moreover,

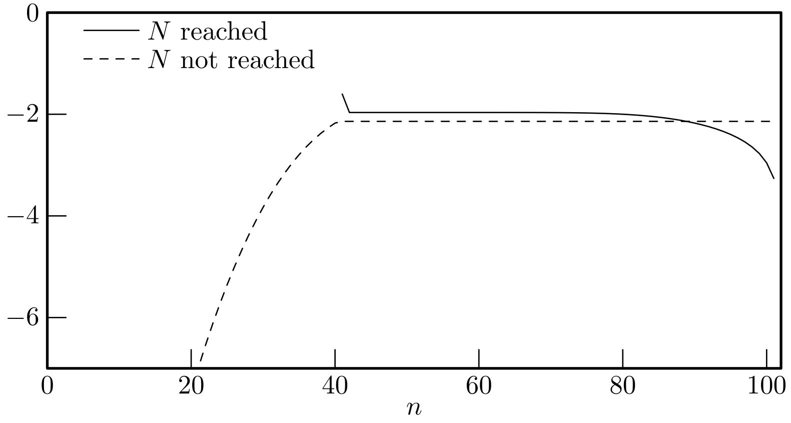

Figure 7.9: Portions of the pmf of the accumulating phase customer delay d_1 split up according to the transition from Φ_1 to Φ_2.

if the $(N-1)$th customer did arrive in the slot preceding to the arrival slot of the Nth customer, its delay will also be N slots: the final accumulating phase slot and the $N-1$ slots required to serve all customers up to the $(N-1)$th. A similar argumentation holds for any customer arriving during an uninterrupted series of consecutive slots in which every slot features an arrival, including the arrival of the Nth customer. The peak at $n = N$ therefore accumulates the delays of all such customers.

Finally, we look at the effect of the arrival rate (the system load) on the mean customer delay. For each phase, Figure 7.10 shows the mean customer delay multiplied with the corresponding phase probabilities, for a system where $N = 41$ and $T = 100$. Also plotted are the overall mean customer delay for the NT-policy, the N-policy and the T-policy. For extremely low load conditions, the mean customer delay is dominated by the mean empty phase customer delay. Under these conditions, the accumulating phase is likely to span the full T slots, such that the mean customer delay approaches $T+1$ for $\rho \to 0$. For moderate load conditions, the mean customer delay stabilizes under the influence of the accumulating phase. For $\lambda > (N-1)/T$, the relative importance of the serving phase starts to dominate the mean customer delay. Again we see the hybrid nature of the NT-policy reflected when comparing with the N-policy and the T-policy.

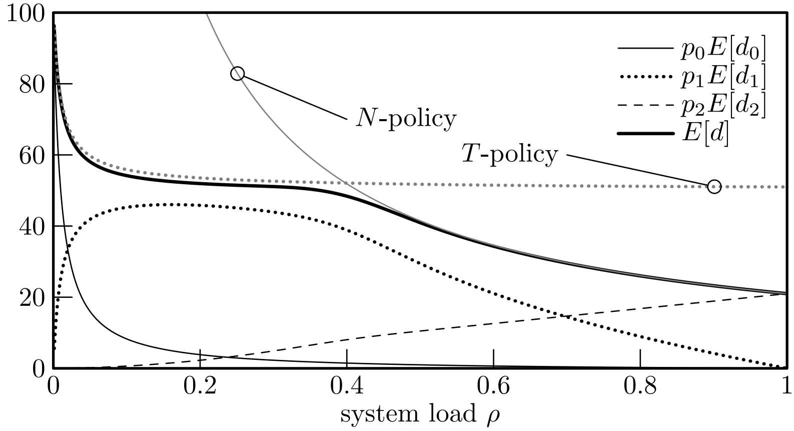

Figure 7.10: Weighted mean customer delays as a function of the system load $\rho = \lambda$.

7.5 General service times

In this section, we broaden the scope of our research to an NT-policy system with general service times, as illustrated in Figure 7.11. The approach presented here will differ significantly from the approach in other sections, in that we will not construct any system equations in order to get some general system state pgf that we can refer to. Rather, we will analyze the system phase per phase and determine the desired expressions more directly. We can expect our results to be similar to those of the previous section, especially for the empty and the accumulating phase, given that the server's characteristics manifest only during the serving phase.

Adopting the notations from the $GI - GI - 1$ model, the pgf of the *iid* service times is given by $S(z)$ with mean μ. Furthermore, we assume the system load $\rho \triangleq \lambda\mu$ to be smaller than 1, such that the system is stable.

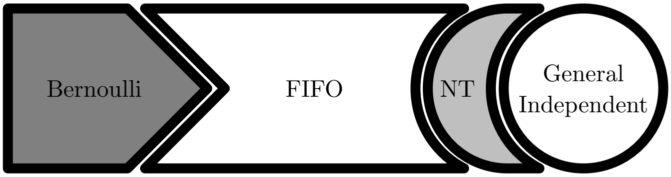

Figure 7.11: Illustration of the NT-policy system with general independent service times.

7.5.1 Phase durations and cycle length

Similar to the previous section, we first focus on the cyclic behavior of the NT-policy.

Empty phase

The empty phase starts once the last customer leaves the system such that it becomes empty again and ends as soon as a new customer arrives. The empty phase therefore consists of as many slots as needed to collect a single arrival, given by the random variable c_1. We therefore find that

$$\text{Prob}[\Phi_0 = t] = \lambda(1-\lambda)^{t-1}, t \geq 1, \tag{7.47}$$

$$\Phi_0(z) \triangleq \text{E}\left[z^{\Phi_0}\right] = \frac{\lambda z}{1-(1-\lambda)\,z}, \tag{7.48}$$

with mean

$$\text{E}[\Phi_0] = \Phi_0'(1) = \frac{1}{\lambda}. \tag{7.49}$$

Accumulating phase

The accumulating phase is initiated by a customer arriving to an empty system and is terminated when the queue contains N customers or when the first customer has been in the queue for T slots, whichever happens first. In order to differentiate between these two possibilities, we will add $\langle N \rangle$ to expressions specific to the case where the N-threshold is reached and $\left\langle \overline{N} \right\rangle$ for expressions where the timer expires before N customers have accumulated.

The N-threshold can only be reached if $N-1$ customers arrive over T slots or less, subsequent to the arrival slot of the first customer. The accumulating phase duration is then equal to the number of slots needed to accumulate these $N-1$ additional arrivals, given by the random variable c_{N-1} , such that

$$\begin{aligned}\text{Prob}\left[\Phi_1^{\langle N \rangle} = t\right] &= \text{Prob}[c_{N-1} = t] \\ &= \binom{t-1}{N-2}\lambda^{N-1}(1-\lambda)^{t-N+1}, \quad N-1 \leq t \leq T,\end{aligned} \tag{7.50}$$

$$\begin{aligned}\Phi_1^{\langle N \rangle}(z) \triangleq \text{E}\left[z^{\Phi_1^{\langle N \rangle}}\right] &= \sum_{t=N-1}^{T} \text{Prob}\left[\Phi_1^{\langle N \rangle} = t\right] z^t \\ &= (\lambda z)^{N-1} \sum_{j=0}^{T-N+1} \binom{j+N-2}{N-2} ((1-\lambda)\,z)^j.\end{aligned} \tag{7.51}$$

Given the fact that this is a partial pgf encompassing all cases where the N-threshold is reached, we can determine the probability ω to reach the N-threshold as

$$\omega \triangleq \mathrm{Prob}[c_{N-1} \leq T] = \Phi_1^{\langle N \rangle}(1) = \lambda^{N-1} \sum_{j=0}^{T-N+1} \binom{j+N-2}{N-2} (1-\lambda)^j. \tag{7.52}$$

The N-threshold will not be reached if the number of arrivals A_T during the T slots following the arrival slot of the first customer, is less than $N-1$. The pgf of the number of arrivals during the accumulating phase when it does not reach the N-threshold, can therefore be determined as

$$A_T^{\langle \overline{N} \rangle}(z) = \sum_{j=0}^{N-2} \mathrm{Prob}[A_T = j]\, z^j = \sum_{j=0}^{N-2} \binom{T}{j} (1-\lambda)^{T-j} (\lambda z)^j. \tag{7.53}$$

The probability $\overline{\omega}$ of the system not reaching the N-threshold can then be found as

$$\overline{\omega} \triangleq \mathrm{Prob}[A_T < N-1] = A_T^{\langle \overline{N} \rangle}(1) = \sum_{j=0}^{N-2} \binom{T}{j} (1-\lambda)^{T-j} \lambda^j. \tag{7.54}$$

Before we determine the actual distribution of the accumulating phase duration, we introduce Ψ as the number of customers in the system at the end of an accumulating phase. The joint pgf of Φ_1 and Ψ follows from (7.51) and (7.53) as

$$\begin{aligned} \mathrm{E}\big[x^{\Phi_1} y^{\Psi}\big] &= \mathrm{E}\big[x^{\Phi_1} y^{\Psi} \{\langle N \rangle\}\big] + \mathrm{E}\big[x^{\Phi_1} y^{\Psi} \{\langle \overline{N} \rangle\}\big] \\ &= \Phi_1^{\langle N \rangle}(x) y^N + x^T y A_T^{\langle \overline{N} \rangle}(y). \end{aligned} \tag{7.55}$$

The marginal pgfs of Φ_1 and Ψ can be determined by setting the appropriate argument to 1, yielding

$$\Phi_1(z) \triangleq \mathrm{E}\big[z^{\Phi_1}\big] = \Phi_1^{\langle N \rangle}(z) + \overline{\omega} z^T, \tag{7.56}$$

$$\Psi(z) \triangleq \mathrm{E}\big[z^{\Psi}\big] = \omega z^N + z A_T^{\langle \overline{N} \rangle}(z), \tag{7.57}$$

and the mean accumulating phase duration is given by

$$\mathrm{E}[\Phi_1] = \Phi_1'(1) = \overline{\omega} T + \Phi_1^{\langle N \rangle \prime}(1). \tag{7.58}$$

Serving phase

During the serving phase, the server is active and serves all customers in the queue until it becomes empty again, after which the server is deactivated and the system moves to the empty phase of a new cycle. The customers

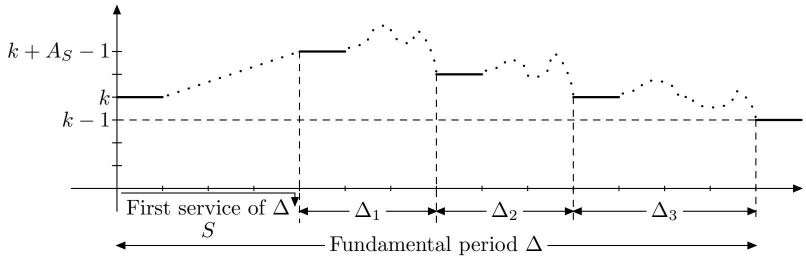

Figure 7.12: Illustration of a fundamental period, with $A_S = 3$ arrivals during the service of a primary customer and k customers in the system at the beginning of the fundamental period.

served during the serving phase include not only the Ψ customers in the queue at the beginning of the serving phase, but also every customer that arrives while the server is active.

In order to account for these additional customers, we resort to the notion of a *fundamental period* [13, 105] to denote the number of slots needed to reduce the system content by 1 customer. More specifically, due to arrivals and departures, the system content may fluctuate during a fundamental period, but it is terminated as soon as the system content becomes less than it was at the beginning of the period. Although the definition and the duration of such a fundamental period is independent of the scheduling policy of the queue, it can be understood most intuitively in a LIFO setting, as depicted in Figure 7.12. In Figure 7.12, a service S is started for a (*primary*) customer, during which A_S (*secondary*) customers arrive to the system and add to the system content. After the service of the primary customer, each of the secondary customers initiates a (*secondary*) fundamental period Δ_i, such that the total duration of the (*primary*) fundamental period Δ becomes

$$\Delta = S + \sum_{i=1}^{A_S} \Delta_i. \tag{7.59}$$

Due to the fact that different arrivals and service times are stochastically independent, the same goes for fundamental periods, allowing us to determine the pgf of a fundamental period as

$$\begin{aligned}\Delta(z) &\triangleq \mathrm{E}\big[z^{\Delta}\big] = \mathrm{E}\Big[z^S \Delta(z)^{A_S}\Big] = \mathrm{E}\Big[z^S A(\Delta(z))^S\Big] \\ &= S(zA(\Delta(z))) = S(z\,(1 - \lambda + \lambda\Delta(z))).\end{aligned} \tag{7.60}$$

Although (7.60) is an implicit expression, an explicit expression can be found for the mean fundamental period duration as

$$\mathrm{E}[\Delta] = \Delta'(1) = \mu\,(1 + \lambda\,\mathrm{E}[\Delta]) = \frac{\mu}{1 - \lambda\mu} = \frac{\mu}{1 - \rho}. \tag{7.61}$$

The serving phase duration can then be found as the sum of the fundamental periods related to the Ψ customers in the system at the beginning of the serving phase, such that

$$\Phi_2(z) \triangleq \mathrm{E}\big[z^{\Phi_2}\big] = \Psi(\Delta(z)), \tag{7.62}$$

with mean

$$\mathrm{E}[\Phi_2] = \Phi_2'(1) = \frac{\mu\Psi'(1)}{1-\rho}. \tag{7.63}$$

Cycle length

The length of a cycle is the sum of the durations of its constituting phases. However, we can not simply add the durations of the individual phases, because in the former, we did not account for the correlation between the accumulating and the serving phase duration. We therefore first introduce $\Phi_{1,2}(x,y)$ as the joint pgf of the accumulating and the serving phase duration,

$$\begin{aligned}\Phi_{1,2}(x,y) &\triangleq \mathrm{E}\big[x^{\Phi_1}y^{\Phi_2}\big] = \sum_{j=1}^{N-1} \mathrm{E}\big[x^{\Phi_1}y^{\Phi_2}\{\Psi=j\}\big] + \mathrm{E}\big[x^{\Phi_1}y^{\Phi_2}\{\langle N\rangle\}\big] \\ &= \mathrm{E}\Big[x^{\Phi_1}\Delta(y)^{\Psi}\big\{\langle\overline{N}\rangle\big\}\Big] + \Phi_1^{\langle N\rangle}(x)\Delta(y)^N \\ &= x^T\Delta(y)A_T^{\langle\overline{N}\rangle}(\Delta(y)) + \Phi_1^{\langle N\rangle}(x)\Delta(y)^N. \end{aligned} \tag{7.64}$$

The pgf $Q(z)$ of the cycle length can then be found as

$$\begin{aligned} Q(z) &\triangleq \mathrm{E}\big[z^{\Phi_0+\Phi_1+\Phi_2}\big] = \Phi_0(z)\Phi_{1,2}(z,z) \\ &= \Phi_0(z)\left(z^T\Delta(z)A_T^{\langle\overline{N}\rangle}(\Delta(z)) + \Phi_1^{\langle N\rangle}(z)\Delta(z)^N\right), \end{aligned} \tag{7.65}$$

with mean

$$\mathrm{E}[Q] = Q'(1) = \frac{1}{\lambda} + \varpi T + \Phi_1^{\langle N\rangle\prime}(1) + \frac{\mu}{1-\rho}\left(\overline{\omega} + A_T^{\langle\overline{N}\rangle\prime}(1) + \omega N\right). \tag{7.66}$$

Phase probabilities

For later use, we define the phase probabilities p_i ($i \in \{0,1,2\}$) as the probability for the system to be in phase i during a random slot:

$$p_i \triangleq \text{Prob[system is in phase } i] = \frac{\mathrm{E}[\Phi_i]}{\mathrm{E}[Q]}. \tag{7.67}$$

Although this definition is sufficient for the determination of any of the three phase probabilities p_i, a more efficient approach is advised in order to determine p_2. In equilibrium, the mean departure rate of any queueing system is equal to the mean arrival rate. In this system, departures only take place during the serving phase and do so at a rate of μ^{-1} customers per slot, such that $p_2 = \lambda\mu = \rho$.

7.5.2 System content and customer delay

In this section we will look at the three different phases separately to analyze both the system content u_i ($i \in \{0, 1, 2\}$) at the beginning of a random slot in phase i and the customer delay d_i of a random customer $\mathcal{C}_i$ that arrives in course of phase i. This analysis will result in an expression for the partial pgfs $U_i(z)$ and $D_i(z)$ of u_i and d_i respectively. From these, we can then find the pgf $U(z)$ of the system content at the beginning of a random slot and the pgf $D(z)$ of the customer delay of a random customer as

$$U(z) = \sum_{i=0}^{2} p_i U_i(z) \qquad \text{and} \qquad D(z) = \sum_{i=0}^{2} p_i D_i(z). \tag{7.68}$$

Note that the BASTA property holds, as the arrivals are generated by a Bernoulli process. As a result, an arbitrary customer $\mathcal{C}$, arriving during slot $\mathcal{S}$, will perceive the system to be in a state that is stochastically indistinguishable from the state at the beginning of a random slot.

Empty phase

The empty phase is characterized by the system being empty, and therefore $u_0 = 0$ and $U_0(z) = 1$. When a customer $\mathcal{C}_0$ arrives during the empty phase, the empty phase will immediately be terminated and that customer will be the first to get served once the accumulating phase has ended. Therefore, that customer will stay in the queue for exactly Φ_1 slots and reside in the server during the entirety of its service. This gives us

$$d_0 = \Phi_1 + S, \tag{7.69}$$

with pgf

$$D_0(z) \triangleq \mathrm{E}\left[z^{d_0}\right] = S(z)\Phi_1(z). \tag{7.70}$$

Accumulating phase

The accumulating phase is started as soon as an arrival occurs during an empty phase, and during the accumulating phase, the system content evolves from that 1 single customer to an unknown number of customers, over an unknown number of slots. In order to deal with these unknowns, we will first analyze the system as if it were a T-policy system, then we will convert the resulting expressions to make them fit for the NT-policy system. For clarity and to prevent confusion, we will decorate expressions belonging

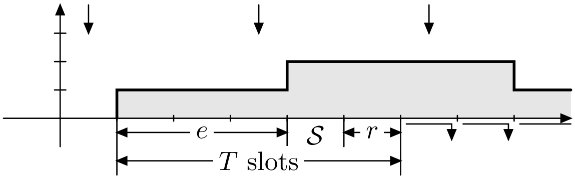

Figure 7.13: The accumulating phase customer delay in a T-policy system - definition of e and r.

to the underlying T-policy system with a superscript (T) and expressions pertaining to the actual NT-policy system with a superscript (NT).

Selecting an arbitrary accumulating phase slot $\mathcal{S}$ in the T-policy system, we introduce the auxiliary random variables e and r as the contiguous number of slots in that accumulating phase, respectively preceding and succeeding slot $\mathcal{S}$, as illustrated in Figure 7.13. Given that in a T-policy system, the accumulating phase duration of every cycle is equal to exactly T slots, the expression $T = e + r + 1$ always holds and we can find the joint pgf of e and r as

$$\begin{aligned}\mathrm{E}[x^e y^r] &= \sum_{t=0}^{T-1} \mathrm{Prob}[e = t, r = T - t - 1]\, x^t y^{T-t-1} = \frac{1}{T}\sum_{t=0}^{T-1} x^t y^{T-t-1} \\ &= \frac{x^T - y^T}{T\,(x-y)}.\end{aligned} \tag{7.71}$$

The joint pgf $P^{(T)}(x,z)$ of the remaining number of accumulating phase slots r and the system content $u_{\mathcal{S}}$ at the beginning of slot $\mathcal{S}$ can be calculated from (7.71) as

$$\begin{aligned}P^{(T)}(x,z) &\triangleq \mathrm{E}[x^r z^{u_{\mathcal{S}}}] = \mathrm{E}\big[x^{T-e-1} z^{1+A_e}\big] = x^{T-1} z\, \mathrm{E}\left[\left(\frac{A(z)}{x}\right)^e\right] \\ &= z\frac{\left(A(z)\right)^T - x^T}{T\left(A(z) - x\right)}.\end{aligned} \tag{7.72}$$

Due to the memoryless arrival process and the *iid* server process, different cycles are uncorrelated and so are different accumulating phases, both in the T-policy and in the NT-policy system. Additionally, the system state evolution during the accumulating phase in the NT-policy system cannot be distinguished from its T-policy counterpart, until possibly the queue length becomes N. Where the NT-policy system will terminate the accumulating phase upon arrival of an Nth customer, this event will remain inconsequential in the T-policy system. The NT-policy counterpart of (7.72) can therefore be obtained by removing the terms corresponding to $u_{\mathcal{S}} \geq N$ and ensuring normalization, such that

$$P^{(NT)}(x,z) = \mathrm{E}[x^r z^{u_{\mathcal{S}}} \,|\, u_{\mathcal{S}} < N] = \frac{\hat{P}^{(T)}(x,z)}{\hat{P}^{(T)}(1,1)}, \tag{7.73}$$

where

$$\hat{P}^{(T)}(x,z) = \sum_{k=1}^{N-1} \Big(\left[z^k\right] P^{(T)}(x,z) \Big) z^k. \tag{7.74}$$

The notation $\left[z^k\right] f(z)$ is known as the *coefficient extractor* notation [46] and denotes the coefficient of z^k in the function $f(z)$. Due to the Bernoulli arrivals we can determine these coefficients directly, without the need of the probability generating property of pgfs or approximate techniques as described in Section 4.4. By reverting some of the steps in (7.71) and (7.72), we get

$$\begin{aligned} P^{(T)}(x,z) &= \frac{z}{T} \sum_{t=0}^{T-1} x^{T-t-1} A(z)^t = \frac{z}{T} \sum_{t=0}^{T-1} x^{T-t-1} \sum_{n=0}^{t} \binom{t}{n} (1-\lambda)^{t-n} (\lambda z)^n \\ &= \frac{1}{T} \sum_{n=1}^{T} \lambda^{n-1} z^n \sum_{t=n}^{T} \binom{t-1}{n-1} x^{T-t} (1-\lambda)^{t-n}, \end{aligned} \tag{7.75}$$

such that the desired coefficients follow as

$$\left[z^k\right] P^{(T)}(x,z) = \frac{\lambda^{k-1}}{T} \sum_{t=k}^{T} \binom{t-1}{k-1} x^{T-t} (1-\lambda)^{t-k}, \quad 1 \le k \le T. \tag{7.76}$$

We can then find $\hat{P}^{(T)}(x,z)$ as

$$\hat{P}^{(T)}(x,z) = \frac{1}{T} \sum_{k=1}^{N-1} \lambda^{k-1} z^k \sum_{t=k}^{T} \binom{t-1}{k-1} x^{T-t} (1-\lambda)^{t-k}. \tag{7.77}$$

From $P^{(NT)}(x,z)$, the pgf $U_1(z)$ of the system content u_1 at the beginning of a random accumulating phase slot follows as

$$U_1(z) \triangleq \mathrm{E}[z^{u_1}] = P^{(NT)}(1,z). \tag{7.78}$$

In (7.73), the random variable r should be understood as the number of slots following $\mathcal{S}$ until timer expiration, even if the accumulating phase under question is terminated before the T-threshold is reached. Therefore, the underlying equality $T = e + r + 1$ remains valid for every accumulating phase slot $\mathcal{S}$ and more specifically, the first accumulating phase slot ($e = 0$ and $u_{\mathcal{S}} = 1$) is uniquely characterized by $r = T - 1$. Thus, the probability $\mathrm{Prob}[r = T-1, u_{\mathcal{S}} = 1]$, which can be found as the coefficient of $x^{T-1}z$ in $P^{(NT)}(x,z)$, denotes the fraction of accumulating phase slots that are the

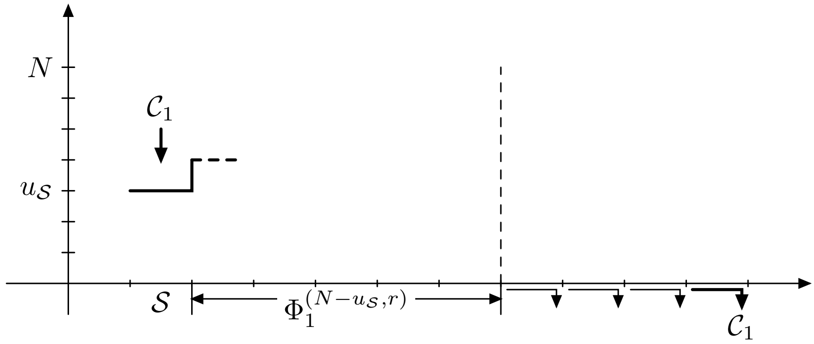

Figure 7.14: The accumulating phase customer delay.

first slot of their accumulating phase. Conversely, since every accumulating phase has exactly one first slot, the mean accumulating phase duration can be found as the inverse of this fraction, yielding

$$\mathrm{E}[\Phi_1] = \frac{1}{[x^{T-1}z]\, P^{(NT)}(x,z)} = T\hat{P}^{(T)}(1,1). \tag{7.79}$$

Now assume that slot $\mathcal{S}$ is no longer a random slot, but rather it is the arrival slot of a random customer $\mathcal{C}_1$ that arrives during the accumulating phase. Note that due to the BASTA property, the system state at the beginning of slot $\mathcal{S}$ is stochastically identical to that of a random accumulating phase slot. Therefore, the distribution of the system content $u_\mathcal{S}$ at the start of slot $\mathcal{S}$ will equal the distribution of u_1. Once the server becomes active in the next serving phase, the $u_\mathcal{S}$ previously arrived customers will be served first before $\mathcal{C}_1$ is allowed in the server. The delay experienced by $\mathcal{C}_1$ therefore consists of the remaining accumulating phase duration and the total service time of $u_\mathcal{S}+1$ customers. As illustrated in Figure 7.14, the remaining accumulating phase duration has the same distribution as the full accumulating phase duration $\Phi_1^{(N-u_\mathcal{S},r)}$ of an $N'T'$-policy system with $N' \triangleq N - u_\mathcal{S}$ and $T' \triangleq r$. The accumulating phase customer delay d_1 can therefore be found as

$$d_1 = \Phi_1^{(N-u_\mathcal{S},r)} + \sum_{j=1}^{u_\mathcal{S}+1} S_j, \tag{7.80}$$

where S_j is the service time of the jth customer in the queue. The pgf

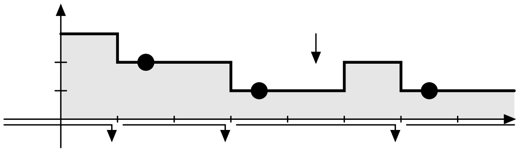

Figure 7.15: The serving phase customer delay - definition of v.

$D_1(z)$ of the accumulating phase customer delay d_1 then follows as

$$\begin{aligned} D_1(z) &\triangleq \mathrm{E}\big[z^{d_1}\big] = \mathrm{E}\Big[\Phi_1^{(N-u_{\mathcal{S}},r)}(z)S(z)^{u_{\mathcal{S}}+1}\Big] \\ &= \sum_{t=0}^{T-1}\sum_{j=1}^{N-1} \mathrm{Prob}[r=t,u_{\mathcal{S}}=j]\,\Phi_1^{(N-j,t)}(z)S(z)^{j+1} \\ &= \sum_{t=0}^{T-1}\sum_{j=1}^{N-1} \Big(\big[x^t z^j\big]\,P^{(NT)}(x,z)\Big)\,\Phi_1^{(N-j,t)}(z)S(z)^{j+1}. \qquad (7.81) \end{aligned}$$

The coefficients $\big[x^t z^j\big]\,P^{(NT)}(x,z)$ can be found from (7.73) and (7.77) as

$$\big[x^t z^j\big]\,P^{(NT)}(x,z) = \frac{1}{T\hat{P}^{(T)}(1,1)}\binom{T-t-1}{j-1}\lambda^{j-1}(1-\lambda)^{T-t-j}, \qquad (7.82)$$

for $1 \le j \le N-1$ and $0 \le t \le T-j$.

Serving phase

As we have not composed any system equations, direct analysis of the system content at the beginning of a random serving phase slot is somewhat cumbersome. Therefore we provide a workaround for which we make use of the *embedded points* approach, i.e. we will perform the analysis for specific well-chosen epochs for which the analysis is more feasible and then use the corresponding results to extract general results. Specifically, we will first focus on the system content v at the beginning of the first slot of a random service, as indicated by the dots in Figure 7.15. At the beginning of the first service time of a serving phase, exactly Ψ customers are in the system. At the beginning of a subsequent service time, the queue has grown according to the A_S arrivals during the previous service time and one customer has left the system. The last service of a serving phase is characterized by $v = 1$ in combination with the fact that no arrival occurs during that service time. Therefore, subsequent values v_n and v_{n+1} of the random variable v are related through the piecewise equation

$$v_{n+1} = \begin{cases} \Psi, & \text{if } v_n = 1 \text{ and } A_S = 0 \\ v_n + A_S - 1, & \text{otherwise.} \end{cases} \qquad (7.83)$$

For ease of computation and notation, we introduce the auxiliary random variables $\tilde{v} \triangleq v-1$ with pgf $\tilde{V}(z) = V(z)/z$ and similarly $\tilde{v}_n \triangleq v_n - 1$. The pgf $\tilde{V}_{n+1}(z)$ of $\tilde{v}_{n+1}$ can then be found as

$$\begin{aligned}\tilde{V}_{n+1}(z) &\triangleq \mathrm{E}\big[z^{\tilde{v}_{n+1}}\big] = \mathrm{E}\big[z^{v_{n+1}-1}\big] \\ &= \mathrm{E}\big[z^{\Psi-1}\{v_n = 1, A_S = 0\}\big] + \mathrm{E}\big[z^{v_n + A_S - 2}\{\neg\,(v_n = 1, A_S = 0)\}\big] \\ &= \frac{1}{z}\left[S(A(z))\tilde{V}_n(z) + (\Psi(z) - 1)\,S(A(0))\tilde{V}_n(0)\right]. \qquad (7.84)\end{aligned}$$

Taking the limit for $n \to \infty$, we get the pgf $\tilde{V}(z)$ of the system content at the beginning of a random service, reduced with 1 as

$$\tilde{V}(z) \triangleq \mathrm{E}\big[z^{\tilde{v}}\big] = \lim_{n\to\infty} \tilde{V}_n(z) = \tilde{V}(0) S(A(0)) \frac{\Psi(z) - 1}{z - S(A(z))}, \qquad (7.85)$$

where $\tilde{V}(0)$ can be found from the normalization condition $\tilde{V}(1) = 1$ as

$$\tilde{V}(0) = \frac{1 - \lambda\mu}{\Psi'(1) S(A(0))}, \qquad (7.86)$$

such that we finally get

$$\tilde{V}(z) = \frac{1-\lambda\mu}{\Psi'(1)} \frac{\Psi(z) - 1}{z - S(A(z))}. \qquad (7.87)$$

Now we consider a random serving phase slot $\mathcal{S}$ and note that it is part of some service time $S_{\mathcal{S}}$. The service time $S_{\mathcal{S}}$ is not random but follows from the selection of $\mathcal{S}$. As such, the probability that $S_{\mathcal{S}}$ consists of n slots is proportional to both the relative occurrence of service times of length n and the length n itself, since $\mathcal{S}$ could be any of the n slots such that

$$\mathrm{Prob}[S_{\mathcal{S}} = n] = \frac{ns(n)}{\mu}, \qquad n \geq 1. \qquad (7.88)$$

Furthermore, we redefine the random variables e and r respectively to denote the numbers of slots of $S_{\mathcal{S}}$ elapsed before and remaining after slot $\mathcal{S}$ and similarly to (7.71) we get

$$\begin{aligned}\mathrm{E}[x^e y^r] &= \sum_{n=1}^{\infty} \mathrm{Prob}[S_{\mathcal{S}} = n] \sum_{t=0}^{n-1} \mathrm{Prob}[e = t, r = n-t-1 \,|\, S_{\mathcal{S}} = n]\, x^t y^{n-t-1} \\ &= \sum_{n=1}^{\infty} \frac{s(n)}{\mu} \sum_{t=0}^{n-1} x^t y^{n-t-1} = \frac{S(x) - S(y)}{\mu\,(x-y)}. \qquad (7.89)\end{aligned}$$

The system content u_2 at the beginning of a random serving phase slot $\mathcal{S}$ then follows as the sum of the system content v at the beginning of the corresponding service $S_{\mathcal{S}}$ and the number of customers A_e that have arrived during the first e slots of that service time. This yields

$$u_2 = v + A_e = \tilde{v} + A_e + 1, \tag{7.90}$$

with pgf

$$\begin{aligned} U_2(z) &\triangleq \mathrm{E}[z^{u_2}] = \mathrm{E}\big[z^{\tilde{v}+A_e+1}\big] = z\tilde{V}(z)\,\mathrm{E}[A(z)^e] \\ &= z\frac{1-\lambda\mu}{\mu\Psi'(1)}\frac{\Psi(z)-1}{A(z)-1}\frac{S(A(z))-1}{z-S(A(z))}. \end{aligned} \tag{7.91}$$

Next, we assume that $\mathcal{S}$ is no longer a random slot, but rather the arrival slot of a randomly selected customer $\mathcal{C}_2$ that arrives during the serving phase. Again we note that this does not affect the system state distribution at the beginning of $\mathcal{S}$, as the BASTA property is in effect. Thus, the distribution of the system content $u_{\mathcal{S}}$ at the start of slot $\mathcal{S}$ will be identical to the distribution of u_2. The delay d_2 of $\mathcal{C}_2$ consists of the remaining service time r of the customer in service during slot $\mathcal{S}$, the total service time of all $u_{\mathcal{S}} - 1$ customers in the queue at the beginning of slot $\mathcal{S}$ and the service time of $\mathcal{C}_2$ itself, such that we get

$$d_2 = r + \sum_{j=1}^{u_{\mathcal{S}}} S_j, \tag{7.92}$$

with pgf

$$\begin{aligned} D_2(z) &\triangleq \mathrm{E}\big[z^{d_2}\big] = \mathrm{E}[z^r S(z)^{u_{\mathcal{S}}}] = S(z)\tilde{V}(S(z))\,\mathrm{E}[A(S(z))^e z^r] \\ &= S(z)\frac{1-\lambda\mu}{\mu\Psi'(1)}\frac{1-\Psi(S(z))}{A(S(z))-z}. \end{aligned} \tag{7.93}$$

7.5.3 Relation to the $GI - GI - 1$ model

Again, we will compare our results for the system content pgf and the delay pgf to the corresponding results found for the $GI - GI - 1$ model. For $N = 1$, their is no accumulation phase and the NT-policy system we have considered effectively becomes a traditional FIFO queueing system. Note that simple substitution of $N = 1$ in the above expressions is discouraged, since they were obtained especially for $N > 1$.

In case $N = 1$, the accumulating phase no longer exists, such that $\Phi_1 = 0$ with pgf $\Phi_1(z) = 1$. From (7.67) and the alternative method for finding p_2, we then find

$$p_0 = 1 - \lambda\mu, \qquad p_1 = 0, \qquad p_2 = \lambda\mu.$$

The pgfs of the empty phase system content and customer delay can be found as

$$U_0(z) = 1, \qquad \text{and} \qquad D_0(z) = S(z). \tag{7.94}$$

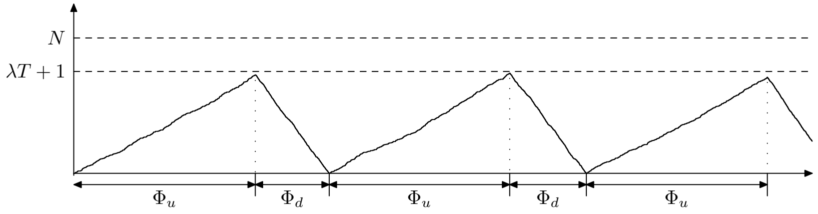

Figure 7.16: The evolution of the system content in an NT-policy system over 4000 slots with $N = 400$, $T = 1000$, $\lambda = 0.3$ and shifted geometric service times with mean $\mu = 1.5$.

For the serving phase system content and customer delay pgfs, we note that the serving phase starts as soon as the first customer has entered the system, i.e. $\Psi = 1$ with pgf $\Psi(z) = z$, yielding

$$U_2(z) = z\frac{1-\lambda\mu}{\lambda\mu}\frac{S(A(z))-1}{z-S(A(z))}, \tag{7.95}$$

$$D_2(z) = S(z)\frac{1-\lambda\mu}{\mu}\frac{1-S(z)}{A(S(z))-z}. \tag{7.96}$$

Taking the weighted sum over the different phases, we get

$$\begin{aligned} U(z) = \sum_{i=0}^{2} p_i U_i(z) &= (1-\lambda\mu)\left(1 + z\frac{S(A(z))-1}{z-S(A(z))}\right) \\ &= (1-\lambda\mu)\, S(A(z))\frac{z-1}{z-S(A(z))}, \end{aligned} \tag{7.97}$$

and

$$\begin{aligned} D(z) = \sum_{i=0}^{2} p_i D_i(z) &= (1-\lambda\mu)\, S(z)\left(1 + \lambda\frac{1-S(z)}{A(S(z))-z}\right) \\ &= (1-\lambda\mu)\, S(z)\frac{z-1}{z-A(S(z))}. \end{aligned} \tag{7.98}$$

Note that (7.97) is identical to the corresponding expression (5.20) in the $GI-GI-1$ model, and (7.98) can be found from (5.29) by substitution of $A(z) = 1-\lambda+\lambda z$.

7.5.4 Fluid flow approximation

Especially for very large values of N and T, the determination of some performance measures, such as the mean cycle length or the mean customer delay, can be computationally tedious due to the multiple summations in the intermediate expressions. Therefore, we now explore a *fluid flow* approximation technique that yields very simple approximate expressions that pose no computational challenge but still offer acceptable precision. The approximation is particularly suitable for situations where the arrival rate λ is small, the service times have low variance, both thresholds N and T are high and when the system tends to reach one of the thresholds much more than the other. In such cases the step function of the system content versus time, can be approximated tightly by a sawtooth function, as depicted in Figure 7.16, due to the variance in the arrival and service process being small and the scale of the system reducing the relative importance of any random variations. Note that the final condition is automatically satisfied if the average number of arrivals during T slots differs much from $N-1$, the number of arrivals needed in the accumulating phase to reach the N-threshold, i.e. $\lambda T \lll\ggg N-1$. Figure 7.16 shows the system content evolution over 4000 slots in a system that satisfies all of these conditions. This graph also illustrates the underlying idea of a fluid flow approximation: the upward and downward portions of the sawtooth curve almost seem to be straight lines on a macroscopic scale, while on a microscopic scale the curve consists of stepwise increments and decrements. The concept of a fluid flow approximation, is to treat some discrete variable as if it were continuous, thus omitting small scale details that have little influence on the large scale behavior.

In the upward portions, the system is in the accumulating phase and the system content gradually builds up at a rate of λ customers per slot. During the serving phase, departures occur according to the service process, while customers keep coming at the same rate; this interaction causes the curve to decrease with a slope of $\mu^{-1}-\lambda$. At this scale, cycles consist of only two phases, a build-up phase and a build-down phase and the apex of the sawtooth will always approach some fixed value u_a. The expected build-up phase duration Φ_u and the build-down phase duration Φ_d can be approximated as

$$\mathrm{E}[\Phi_u] \approx \frac{u_a}{\lambda}, \qquad \text{and} \qquad \mathrm{E}[\Phi_d] \approx \frac{\mu u_a}{1-\lambda\mu}, \tag{7.99}$$

such that the approximated mean cycle length becomes

$$\mathrm{E}[Q] \approx \mathrm{E}[\Phi_u] + \mathrm{E}[\Phi_d] \approx \frac{u_a}{\lambda\,(1-\lambda\mu)}. \tag{7.100}$$

The approximated mean system content can be found as half the height of the sawtooth and the approximated mean customer delay then follows from Little's theorem, such that

$$\mathrm{E}[u] \approx \frac{u_a}{2}, \qquad \text{and} \qquad \mathrm{E}[d] \approx \frac{u_a}{2\lambda}. \tag{7.101}$$

The height of the apex of the sawtooth depends on the relation between N, T and λ. More specifically, if $\lambda T \ll N-1$, it generally takes much more

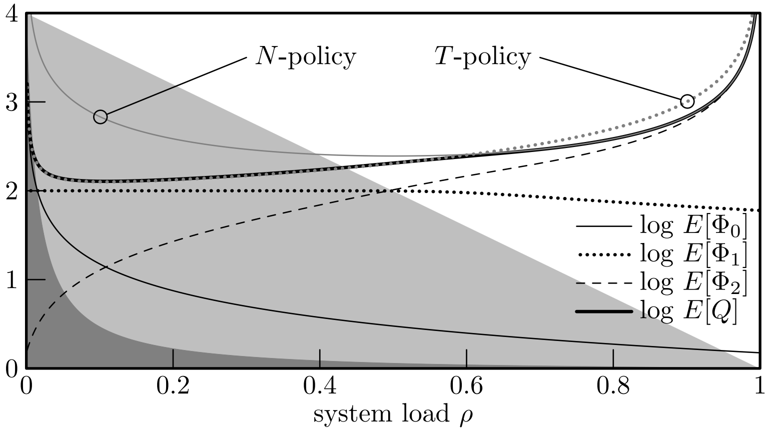

Figure 7.17: The mean phase sojourn times and the mean cycle length as a function of the system load $\rho = \lambda\mu$.

than T slots to reach the N-threshold, such that most of the accumulating phases are terminated due to timer expiration. In that case, the system will contain on average $u_a = 1 + \lambda T$ customers when the serving phase is started. If on the other hand we have that $\lambda T \gg N - 1$, the system will usually accumulate N customers well before timer expiration, such that $u_a = N$. In summary, we have

$$u_a = \begin{cases} 1 + \lambda T, & \lambda T \ll N - 1, \\ N, & \lambda T \gg N - 1. \end{cases} \tag{7.102}$$

7.5.5 Numerical examples

We end this section by means of some more numerical examples regarding the NT-policy. Just like in Section 7.4, we will compare the results of the NT-policy with results obtained for the N-policy and the T-policy.

Again, we first focus on the mean phase durations and the mean cycle length as functions of the system load. In Figure 7.17 we show the mean phase durations and the mean cycle length on a logarithmic scale for $N = 41$, $T = 100$ and shifted Poisson service times with mean $\mu = 1.5$. The background colors illustrate the distribution of the phase probabilities, such that for each vertical cut, the dark grey portion corresponds to the empty phase probability p_0, the light grey portion then corresponds to the accumulating phase probability p_1 and the serving phase probability p_2 is illustrated by the white background. For very small values of the system load, the mean cycle length is dominated by the contribution of the empty phase. Under these load conditions, the accumulating phase practically always lasts for T

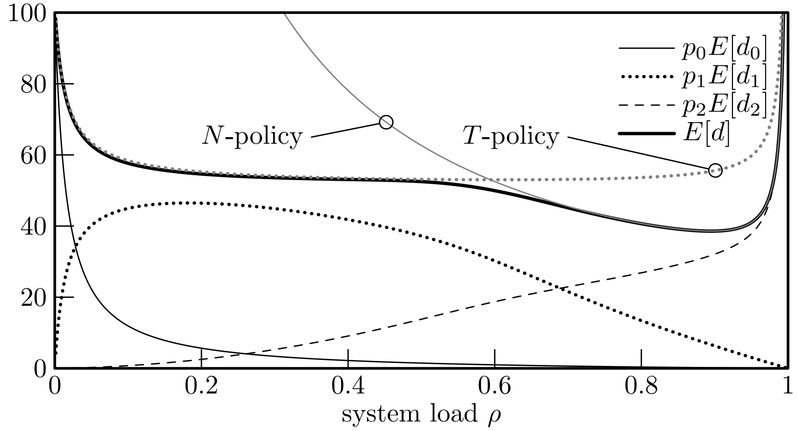

Figure 7.18: The mean customer delay as a function of the system load $\rho = \lambda\mu$.

slots during which only few customers will accumulate such that the serving phase duration will be negligible. When the load increases, the empty phase duration quickly drops and the serving phase duration starts to increase. The mean accumulating phase duration is virtually unchanged until the arrival rate $\lambda = \rho/\mu$ approaches $(N-1)/T$. When the system load keeps on increasing, the number of secondary arrivals during a fundamental period increases, and the mean serving phase duration becomes dominant. When comparing with both the N-policy and the T-policy, we see that the mean cycle length of the NT-policy serves as a lower bound for the mean cycle length in both primitive policies. For $\lambda < (N-1)/T$, the mean cycle length in the NT-policy coincides with its T-policy counterpart and for higher arrival rates the NT-policy mimics the N-policy. When $\lambda \approx (N-1)/T$, both thresholds N and T have a similar chance of being the first to be reached and the differences between the three policies are minimal

Next, we study how the system load affects the mean customer delay. Figure 7.18 shows the mean customer delay for a system with $N = 41$, $T = 100$ and shifted Poisson service times with mean $\mu = 1.5$. For each phase, the mean customer delay is multiplied with the corresponding phase probability, such that the sum of these weighed means corresponds to the mean customer delay for a customer arriving in a random slot. For comparison, the mean customer delay in corresponding N-policy and T-policy systems is plotted as well. As we could expect, Figure 7.18 is quite similar to Figure 7.17. For a very low system load, the mean customer delay - just like the cycle length - is dominated by the contribution of the empty phase. As the system load increases to $\mu(N-1)/T$, the weight of the empty phase contribution quickly drops and the mean customer delay is mainly shaped

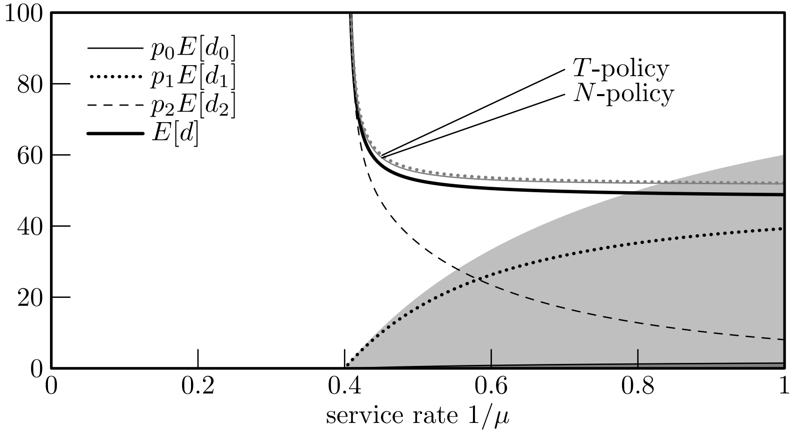

Figure 7.19: The mean customer delay as a function of the service rate $1/\mu$.

by the accumulating phase portion. When the load increases even more, the serving phase dominates the mean customer delay, growing excessively as the system load approaches 1. Again we see that for the lower values of the system load, the mean customer delay of the NT-policy matches the mean customer delay in the T-policy system and for higher values of ρ, the mean customer delay coincides with its N-policy counterpart.

Note the general similarity between the curves related to the mean phase and cycle durations on the one hand and the weighted and total mean customer delays on the other hand. Therefore we will from now on only focus on the mean customer delay. In what follows we will investigate the influence of other system parameters, such as μ, N and T, all of which have an effect on the mean customer delays whereas, for example, $\mathrm{E}[\Phi_0]$ only depends on λ. The curves displayed are then the counterparts of the curves displayed in Figure 7.18, i.e. for each phase the mean customer delay weighted according to the corresponding phase probability and the mean delay for a customer arriving in a random slot for the three policies considered. As before, the background of the charts will symbolize the phase probabilities.

The effect of the service rate $1/\mu$ on the mean customer delay is portrayed in Figure 7.19, for a system where $\lambda = 0.4$, $N = 41$, $T = 100$ and service times are shifted Poisson distributed. When the service rate $1/\mu$ is hardly greater than the arrival rate λ, the system load will be close to 1, leading to excessive delays. In such cases, the serving phase probability will approach 1, such that the mean customer delay is completely dominated by the serving phase contribution. As the service rate increases, service times decrease, leading to shorter delays, but also decreasing the weight of the serving phase, especially in favor of the accumulating phase. In this scenario, the relative importance of the empty phase is negligible.

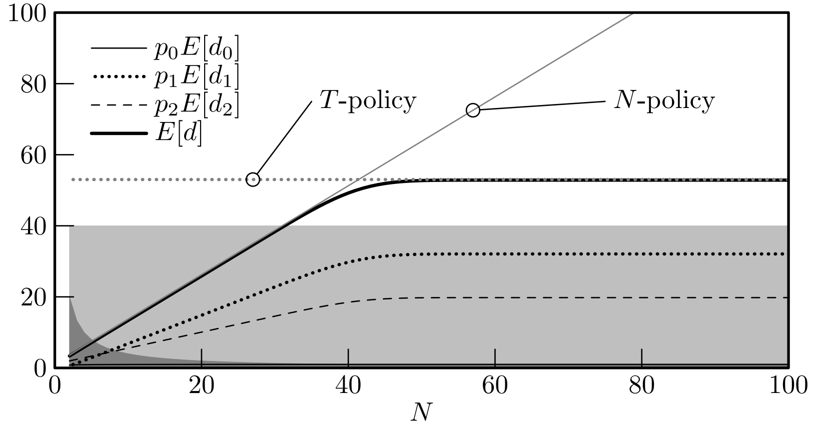

Figure 7.20: The mean customer delay as a function of N.

We now illustrate the impact of the N-threshold on the mean customer delay in Figure 7.20 for a system with $\lambda = 0.4$, $T = 100$ and shifted Poisson service times with mean $\mu = 1.5$. Note that the serving phase probability $p_2 = \rho$ is independent of N, such that the serving phase has a fixed weight on the mean customer delay. For low values of the N-threshold, the accumulating phase will be quite short and not many customers will be in the system at the beginning of the serving phase, resulting in a rather small mean customer delay. As long as N is sufficiently smaller than $\lambda T + 1$, an increase in N will cause an increase in the accumulating phase length and in the number of customers in the system at the beginning of the serving phase and as a result, the mean customer delay will increase as well. When N becomes significantly larger than $\lambda T + 1$, the N-threshold will become practically unreachable before timer expiration, and the system will start to behave like a T-policy system, independent of the actual value of N.

Now we investigate the effect of the time threshold T on the mean customer delay, as illustrated in Figure 7.21 for a system where $\lambda = 0.4$, $N = 41$ and service times are shifted Poisson distributed with mean $\mu = 1.5$. Again, we clearly see that the actual value of T has no influence on the serving phase probability p_2. Similar to low values of N, low values of T will result in a short mean accumulating phase duration and by consequence a small number of customers in the queue at the beginning of the serving phase, such that the mean customer delay is rather small. When T becomes larger, there will be more time for primary customers to accumulate during the accumulating phase, such that the mean customer delay will increase accordingly. For values of T sufficiently larger than $(N-1)/\lambda$, it will become ever so probable that N customers can accumulate before the timer expires. The system will then behave more and more like an N-policy system and

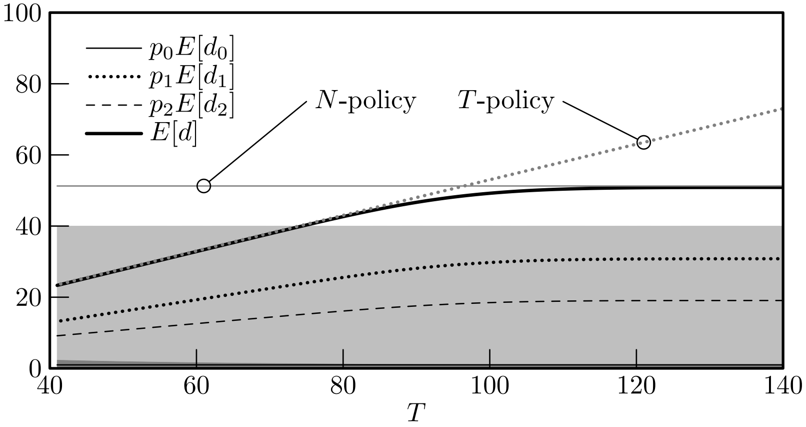

Figure 7.21: The mean customer delay as a function of T.

the T-threshold will no longer affect the mean customer delay.

Finally, we illustrate the accuracy of the fluid flow approximation of the mean customer delay E$[d]$ in Figure 7.22 for a system with $N = 41$, $T = 500$ and shifted Poisson service times with mean $\mu = 1.5$. The curves plotted are the curves for the mean customer delay for the NT-policy system as calculated in Section 7.5.2 and as approximated using (7.101). For the N-policy and the T-policy the calculated curve for E$[d]$ has been plotted as well. For the fluid flow approximation, the NT-policy system is basically reduced to a single threshold policy, according to the relation between λT and $N-1$. In case $\lambda T \ll N-1$, it is assumed that only the timer threshold T is relevant and if $\lambda T \gg N-1$ only the N threshold is retained. Therefore, it comes as no surprise that the approximated curve closely mimics the original single threshold policy curves in the applicable regions.

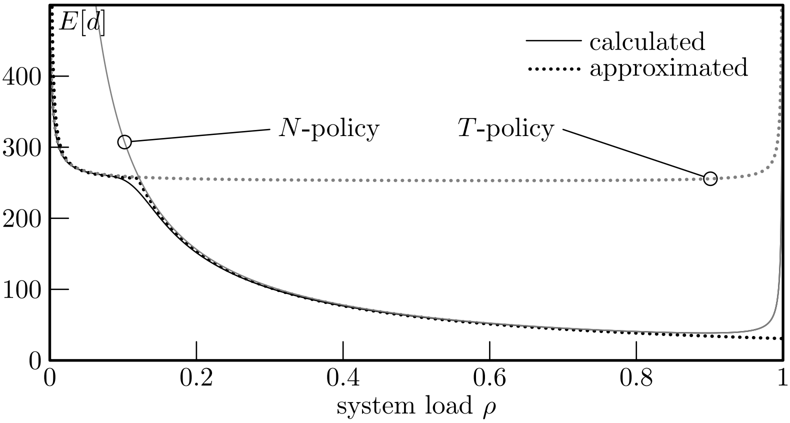

Figure 7.22: The mean customer delay as approximated by (7.101) as a function of the system load $\rho = \lambda\mu$.

Part III

Correlation Effects

Chapter 8

Session-Based Arrivals and Bernoulli Output Line Interruptions

8.1 Introduction

The main goal of queueing theory in general is to obtain knowledge and understanding about queueing systems in the form of mathematical expressions, algorithms, graphs, . . . Depending on the characteristics of the queueing system, some models may be more appropriate than others for inferring accurate and realistic results. In general, a model tailored to match the key properties and inner workings of a queueing system offers better results than a general purpose model, where only some basic parameters can be fitted.

Such a general purpose model is the $GI-GI-1$ model described earlier, for which the arrival process and the server process can be configured as any *iid* process. In many realistic situations however, the assumption of *iid* arrivals is very much inadequate, as there is usually some sort of correlation between subsequent arrivals. This is especially the case when considering information packet arrival streams in telecommunication systems. Furthermore, it is known, as well as it is intuitively clear, that the performance of a queueing system degrades as the (positive) correlation in the arrival process increases [78, 83]. In general, the term *correlation* - apart from its mathematical meaning - is mainly understood as a measure of burstiness or some sort of dependence. In the Internet specifically, many reasons can be found for correlation, both technical and non-technical, e.g.

- most web pages require additional files to be downloaded, such as images, css stylesheets, javascript files, . . . ;
- most data is too large to be sent as a single packet, such as large files, multimedia streams, . . . ;
- some Internet applications require parallel connections, such as ftp;
- Internet usage shows temporal trends that are dependent on socio-cultural and geographical factors [16].

In this chapter, we will study queues with session-based arrivals, a complex arrival model devised to represent a particular type of short-term time correlation in the arrival process.

A simple class of arrival processes incorporating correlation are the so-called *On/Off*-processes [4, 31, 124, 126]. These models assume that a finite number of users is responsible for generating information packets. Each user can either reside in the On-state, during which the user is active and thus generates packets, or the user can be in the Off-state meaning that the user is idle and does not generate any packets.

Related to the On/Off-processes, but much more flexible are the family of *Markovian Arrival Processes* (MAPs in short) [85, 94]. In a MAP, a source governed by a Markov process generates packets at a rate that depends on the state of that background Markov process. As an extension, *Batch Markovian Arrival Processes* (BMAPs) [19, 84] and their discrete-time counterpart D-BMAPs [5], were introduced in order to allow for batch arrivals, instead of (at most) one per arrival instant. The family of MAPs and all their varieties (see [96] for an overview) share the advantages that they are versatile and allow for a tractable analysis. They can either be used to model the entire arrival stream to the buffer [102] or to model one of many multiplexed flows [104, 109].

Other traffic models that have been studied with respect to the related buffer performance are e.g. (discrete) autoregressive arrivals [72] and semi-Markov processes [47, 75]. Specifically designed to study the effect of traffic correlation on very long time scales are also the self-similar or *long range dependent* (LRD) traffic models [54, 97, 100].

Although the arrival models presented in the former are well-suited to model correlated arrivals and even correlated traffic flows, they usually lack the possibility to perceive the individual packets as part of a whole, higher-level entity, commonly referred to as a *message*. In cases where results about such messages are desired, appropriate arrival models, aware of the natural grouping of packets are generally required. To that end, *dispersed messages* were introduced [20] as collectives of constant numbers of consecutive packets in an independent arrival process. Although this model was a good first step towards message modelling, the definition of dispersed messages is very limitative, not allowing multiple simultaneous or variable-length messages.

Both these limitations were overcome by the introduction of *train arrivals* [12, 17, 21, 120], where messages are referred to as *trains*. Such a train is a group of packets that arrive in consecutive slots at a rate of exactly one packet per slot. This definition offers a versatile notion of messages, allowing various distributions for the number of new trains per slot, for the number of packets per train and even allowing for multiple simultaneous trains. Most research involving train arrivals assumes the generation of new trains to be *iid*, although some research has been committed to (Markovian) correlated models for train generation [23, 24, 66].

In this chapter however, we focus on *session-based arrivals*, an extension of the train arrival model. With session-based arrivals, the messages are called *sessions*, which are defined similarly to trains, except that a session can generate a variable, yet strictly positive number of packets per slot. Previous research on session-based arrivals [61, 62, 121] has resulted in expressions for the system content and the packet delay. This chapter revisits my contributions about the session delay [38, 39] and the session-based arrival model with multiple heterogeneous session types [40].

Note that the messages-based arrival processes are particularly relevant when studying queueing systems in close proximity of the origin of the messages, e.g. the outgoing buffer of a file server. At points located further away from the message source, intersecting streams and other network effects usually blur the correlation between the individual packets of a message.

8.2 Session-based arrivals

Session-based arrival streams generate packets as part of larger entities referred to as sessions. These sessions usually span over multiple consecutive slots and produce packets at a variable rate of one or more packets per slot. Session-based arrival streams can be grouped into different classes or types according to their characteristics in terms of session *incidence*, *bandwidth* and *length*, each of which can be described by a discrete probability distribution. The session incidence distribution describes the number of newly initiated sessions per slot. Note that we assume this distribution to be *iid* from slot to slot and from session type to session type, such that all sessions are initiated independently from each other, whether or not they belong to the same session type. The bandwidth of a session is the variable yet strictly positive number of packets generated by a session during a single slot. We will assume that the session bandwidth distribution is *iid* from slot to slot and from session to session. Finally, the number of slots during which a session generates packets is called the session length, which we assume to be *iid* from session to session. The definitions of the session bandwidth and session length are illustrated in Figure 8.1. Here, the grayed session is initiated in slot 4 and has a length of 8 slots during which it produces 20 packets at a variable rate.

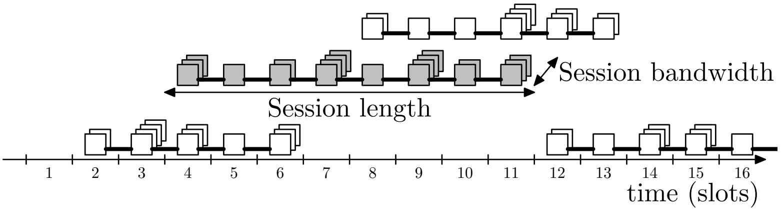

Figure 8.1: Example of how the packets arrive to a system with session-based arrivals.

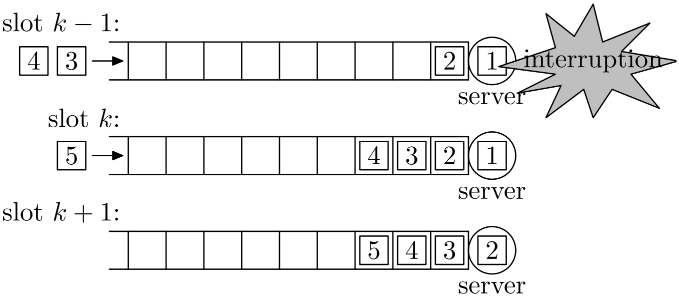

Figure 8.2: The effect of an output line interruption.

In order to get familiar with the techniques required to analyze the session delay, we will first assume homogeneous sessions (i.e. only 1 session type) in Section 8.7.1. Afterwards, we will expand the analysis to the heterogeneous case in Section 8.7.2.

8.3 Output line interruptions

The service offered to the packets arriving in the system consists of transmitting these packets over an output line, which we consider to be subject to interruptions. When such an interruption occurs, no packets can be transmitted. This is illustrated in Figure 8.2, where the packet with label 1 is ready for transmission in slot $k-1$, but due to an interruption it can only leave the system during slot k, such that the packet labeled 2 can be transmitted no sooner than in slot $k+1$. Note that the net effect of these output line interruptions is essentially the same as if the packets have prolonged transmission times and the output line is always accessible. For clarity, the term *transmission time* denotes the number of slots needed to transmit a packet over an accessible output line. The term *effective transmission time* also incorporates the time lost due to output line interruptions.

These output line interruptions allow us to model the unreliable nature of communication networks.

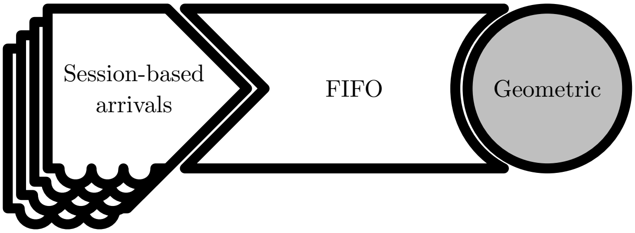

Figure 8.3: Illustration of a system with session-based arrivals and geometric output line interruptions.

8.4 Mathematical model

In this chapter, we will investigate the system depicted in Figure 8.3, more specifically a FIFO queueing system fed by a session-based arrival process with deterministic transmission times of 1 slot per packet and independent Bernoulli output line interruptions. This means that during each random slot, the output line is accessible with a fixed probability σ and with a probability $1-\sigma$, the output line is interrupted. In combination with the single-slot transmission times, this yields shifted geometrically distributed effective transmission times with mean $\mu = 1/\sigma$. Hence the shifted geometric server in Figure 8.3.

For the session-based arrival process, we consider T types of sessions, each characterized by their incidence, bandwidth and length distributions, which in turn are described by their pgfs. These pgfs are $B_t(z)$ for the number of new sessions of type $t \in \{1, \ldots, T\}$ in a random slot, $P_t(z)$ for the number of packets generated by a session of type t during a random slot and $L_t(z)$ is the pgf of the length af a random session of type t. As mentioned before, we assume for each session type the numbers of new sessions per slot to be *iid*, as well as the lengths of these sessions and the numbers of packets per slot generated by these sessions.

This specific queueing system has already been studied in [121], yielding expressions for the pgf, the mean value and the tail distributions of both the system content and the packet delay. My personal contribution to the analysis of this particular model, consists mainly of the study of the session delay. Therefore, we will not revisit the previous work into full detail, but we will summarize the most relevant results. The attentive reader may notice some notational differences between the expressions mentioned here and those in previous work, caused by adapting the original expressions to the notations used in this dissertation.

The remainder of this chapter is structured as follows. In Section 8.5, we elaborate on the arrival process and define some relevant variables that will aid us in our further analysis. Section 8.6 summarizes the main results from [121]. We then focus on the session delay, first for homogeneous sessions (i.e.

$T = 1$) in Section 8.7.1, and then for arbitrary values of T in Section 8.7.2. Finally, we illustrate our results by means of some numerical examples in Section 8.8.

8.5 The packet arrival process

First, we look at the arrival process at session level and define $a_{n,k}(t)$ as the number of active sessions of type t in their nth slot during slot k. Considering the fact that the number of sessions of type t in their first slot is determined by the corresponding session incidence distribution, whereas the number of sessions of type t in their non-first slot is determined by the number of sessions of type t in the previous slot that continue to be active, we find

$$a_{1,k}(t) = b_k(t), \qquad \text{and} \qquad a_{n,k}(t) = \sum_{i=1}^{a_{n-1,k-1}(t)} c^i_{n-1,k}(t), \quad n > 1, \tag{8.1}$$

where $b_k(t)$ is the number of sessions of type t started in slot k and the variables $c^i_{n-1,k}(t)$ are 1 if and only if the ith active session of type t, in its $(n-1)$th slot during slot $k-1$, will continue during slot k and 0 otherwise. The distributions of the continuity variables $c^i_{n-1,k}(t)$ can be derived from the session length distributions. Therefore we first define $\pi_t(n)$ as the probability that a session of type t, that has been active for n slots will remain active for at least one more slot:

$$\pi_t(n) \triangleq \frac{1 - \sum_{i=1}^{n} \ell_t(i)}{1 - \sum_{i=1}^{n-1} \ell_t(i)}, \tag{8.2}$$

with $\ell_t(i)$ being the pmf of the length distribution of sessions of type t. The pgf $C_{n-1,t}(z)$ of the variables $c^i_{n-1,k}(t)$ can then be found as

$$C_{n-1,t}(z) \triangleq \mathrm{E}\Big[z^{c^i_{n-1,k}(t)}\Big] = 1 - \pi_t(n-1) + z\pi_t(n-1), \quad n > 1. \tag{8.3}$$

At packet level, we do not need to distinguish between packets pertaining to sessions of different types and thus we define m_k as the total number of packets arriving during slot k, given by

$$m_k = \sum_{t=1}^{T} \sum_{n=1}^{\infty} \sum_{i=1}^{a_{n,k}(t)} p^i_{n,k}(t), \tag{8.4}$$

where $p^i_{n,k}(t)$ is the number of packets generated by the ith session of type t that was in its nth slot during slot k. The packet arrival rate λ, i.e. the mean number of packet arrivals in an arbitrary steady-state slot, can then be calculated as

$$\lambda = \mathrm{E}[m] = \sum_{t=1}^{T} B'_t(1) L'_t(1) P'_t(1). \tag{8.5}$$

The system load ρ then becomes

$$\rho = \frac{\lambda}{\sigma}. \tag{8.6}$$

8.6 Summary of previous work: system equations, buffer analysis and packet delay analysis

For the system state, we not only need to keep track of the system content, the number of active sessions and the progress of each of those sessions, but also of the type of each session, yielding the system state vector $\langle \underline{\mathbf{a}}_{1,k-1}, \ldots, \underline{\mathbf{a}}_{T,k-1}, u_k \rangle$. Here, we introduced the infinite dimensional vectors $\underline{\mathbf{a}}_{t,k} \triangleq \langle a_{1,k}(t), a_{2,k}(t), \ldots \rangle$ containing the number of active sessions of a specific type t, grouped by the number of slots they have been active for up until slot k. The transition of the vectors $\underline{\mathbf{a}}_{t,k-1}$ from slot to slot is described by (8.1) and the transition of the system content at the beginning of a slot can be described by

$$u_{k+1} = (u_k - r_k)^+ + m_k, \tag{8.7}$$

where r_k is a Bernoulli variable that is 1 with probability σ and 0 with probability $1-\sigma$, thus modelling Bernoulli output line interruptions. From (8.1) and (8.7), it can be seen that the set $\{\langle \underline{\mathbf{a}}_{1,k-1}, \ldots, \underline{\mathbf{a}}_{T,k-1}, u_k \rangle\}$ of system state vectors constitutes a Markov chain.

With $\underline{\mathbf{x}}_t = (x_{1,t}, x_{2,t}, \ldots)$, the steady-state joint pgf $Q(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z)$ of the system state can be found as

$$Q(\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_T, z) = \frac{1}{z} \left(\prod_{t=1}^{T} B_t(x_{1,t} P_t(z)) \right) \Big\{ \sigma\,(z-1)\,p_0 \\ + \bar{R}(z) Q(\underline{\mathbf{G}}_1(\underline{\mathbf{x}}_1, z), \ldots, \underline{\mathbf{G}}_T(\underline{\mathbf{x}}_T, z), z) \Big\}, \tag{8.8}$$

where p_0 is the steady-state empty system probability, the function $\bar{R}(z) = \sigma + (1-\sigma)\,z$ pertains to the output line state and the vector functions $\underline{\mathbf{G}}_t(\underline{\mathbf{x}}_t, z)$ are defined as

$$\underline{\mathbf{G}}_t(\underline{\mathbf{x}}_t, z) \triangleq \left(G_{1,t}(\underline{\mathbf{x}}_t, z), G_{2,t}(\underline{\mathbf{x}}_t, z), \ldots \right), \qquad 1 \le t \le T, \tag{8.9}$$

with

$$G_{n,t}(\underline{\mathbf{x}}_t, z) \triangleq C_{n,t}(x_{n+1,t} P_t(z)), \qquad n \ge 1, 1 \le t \le T. \tag{8.10}$$

The probability p_0 that the system is empty at the beginning of a random steady-state slot can be determined as

$$p_0 = 1 - \frac{1}{\sigma} \sum_{t=1}^{T} B_t'(1) L_t'(1) P_t'(1) = 1 - \frac{\lambda}{\sigma}, \tag{8.11}$$

and the mean steady-state system content is

$$\begin{aligned}\mathrm{E}[u] = -\frac{1}{2}\sum_{t=1}^{T} B_t'(1)P_t'(1)L_t''(1) + \frac{\rho\,(2-\lambda)}{2\,(1-\rho)} \\ + \frac{1}{2\,(1-\rho)\,\sigma}\Bigg[\sum_{t=1}^{T}\left(\mathrm{Var}[p_t] - P_t'(1)\right)B_t'(1)L_t'(1) \\ + \sum_{t=1}^{T}\Big(\mathrm{Var}[b_t]\,L_t'(1)^2 + \mathrm{Var}[\ell_t]\,B_t'(1)\Big)\,P_t'(1)^2\Bigg], \quad (8.12)\end{aligned}$$

where $\mathrm{Var}[b_t]$, $\mathrm{Var}[p_t]$ and $\mathrm{Var}[\ell_t]$ are the variances of the incidence, bandwidth and length distribution of sessions of type t respectively. The mean packet delay $\mathrm{E}[d_{\mathcal{P}}]$ then follows from Little's theorem as

$$\mathrm{E}[d_{\mathcal{P}}] = \frac{\mathrm{E}[u]}{\lambda}. \quad (8.13)$$

For later use, we also define $\mathrm{E}[a_n(t)]$ as the mean number of active sessions of type t in their nth slot during an arbitrary steady-state slot, which can be found as

$$\mathrm{E}[a_n(t)] = \frac{\partial}{\partial x_{n,t}} Q(\underline{\mathbf{1}},\ldots,\underline{\mathbf{1}},1) = B_t'(1)\left(1 - \sum_{i=1}^{n-1}\ell_t(i)\right) = B_t'(1)\Lambda_t(n), \quad (8.14)$$

where $\underline{\mathbf{1}}$ is an infinite dimensional vector with all elements equal to 1 and where we introduced $\Lambda_t(n) \triangleq \sum_{j=n}^{\infty}\ell_t(j)$ for convenience. This auxiliary function has two interesting properties which we will exploit later on:

$$\Lambda_t(1) = \sum_{j=1}^{\infty}\ell_t(j) = 1, \qquad \text{and} \qquad \sum_{n=1}^{\infty}\Lambda_t(n) = L_t'(1). \quad (8.15)$$

The mean total number of active sessions $\mathrm{E}[a(t)]$ of a certain type t in an arbitrary slot can then be found as

$$\mathrm{E}[a(t)] = \sum_{n=1}^{\infty}\mathrm{E}[a_n(t)] = \sum_{n=1}^{\infty}\frac{\partial}{\partial x_{n,t}}Q(\underline{\mathbf{1}},\ldots,\underline{\mathbf{1}},1) = B_t'(1)L_t'(1). \quad (8.16)$$

8.7 Session delay analysis

Given the complex nature of the session-arrival process, we first analyze the session delay for homogeneous sessions (i.e. $T = 1$) in Section 8.7.1. The method developed there will then serve as a guideline for the study of the session delay for heterogeneous sessions in Section 8.7.2.

8.7.1 Homogeneous sessions $(T = 1)$

In the case where there is only 1 single session type, the indices $_t$ in previous expressions, referring to the session type are redundant and have therefore been omitted in this section.

Let $\mathcal{M}$ be a randomly selected steady-state session and let $\mathcal{S}$ be the slot during which that session $\mathcal{M}$ is started, i.e. the slot during which the first packet of $\mathcal{M}$ arrives to the system. The delay $d_{\mathcal{M}}$ of session $\mathcal{M}$ is then defined as the integer number of slots between the end of slot $\mathcal{S}$ and the end of the slot during which the session's final packet effectively leaves the system. As can be expected, the determination of the session delay poses quite a challenge, therefore we will limit ourselves to the determination of the mean session delay, which can be calculated using the law of total expectation as

$$\mathrm{E}[d_{\mathcal{M}}] = \sum_{\ell=1}^{\infty} \mathrm{E}\big[d_{\mathcal{M}|\ell}\big] \, \mathrm{Prob}[\text{session } \mathcal{M} \text{ has length } \ell] \,, \tag{8.17}$$

where $d_{\mathcal{M}|\ell}$ denotes the delay of a session of length ℓ, such that the session's final slot can be defined as $\bar{\mathcal{S}} \triangleq \mathcal{S} + \ell - 1$. This conditional delay $d_{\mathcal{M}|\ell}$ consists of the total transmission time of the $u_{\mathcal{S}+1}$ packets in the queue at the beginning of slot $\mathcal{S}+1$, the total transmission time of all $m_{\mathcal{S}+i}$ packets arriving during slots $\mathcal{S} + i, i \in \{1, \ldots, \ell - 1\}$ except for the $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}$ packets arriving during slot $\bar{\mathcal{S}}$ but after the session's final packet. Therefore we get

$$\mathrm{E}\big[d_{\mathcal{M}|\ell}\big] = \frac{1}{\sigma} \left(\mathrm{E}[u_{\mathcal{S}+1}] + \sum_{i=1}^{\ell-1} \mathrm{E}[m_{\mathcal{S}+i}] - \mathrm{E}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}\big] \right) . \tag{8.18}$$

Note that, although $\mathcal{S}$ is not an arbitrary slot, the system state at the beginning of $\mathcal{S}$ has the same distribution as the system state at the beginning of a random steady-state slot. This is because sessions start independently from slot to slot. Due to the fact that by definition at least 1 session is started during slot $\mathcal{S}$, this can not be said of the system state at the beginning of slot $\mathcal{S}+1$, such that $\mathrm{E}[u_{\mathcal{S}+1}] \neq \mathrm{E}[u_{\mathcal{S}}] = \mathrm{E}[u]$. The pmf of the number $b_{\mathcal{S}}$ of new sessions in slot $\mathcal{S}$ can be found by considering that this probability is proportional to the number of new sessions in that slot, or

$$\mathrm{Prob}[b_{\mathcal{S}} = \beta] = \frac{\beta}{B'(1)} \mathrm{Prob}[b_k = \beta] \,, \qquad \beta \geq 1, \tag{8.19}$$

similar to (5.25). The joint pgf $Q_{\mathcal{S}+1}(x_1, x_2, \ldots, z)$ of the system state at the beginning of slot $\mathcal{S}+1$ can be found as

$$\begin{aligned} Q_{\mathcal{S}+1}(x_1, x_2, \ldots, z) \triangleq \mathrm{E}\left[\left(\prod_{n=1}^{\infty} {x_n}^{a_{n,\mathcal{S}}} \right) z^{u_{\mathcal{S}+1}} \right] &= \frac{x_1}{B'(1)} \frac{\partial}{\partial x_1} Q(x_1, x_2, \ldots, z) \\ &= \frac{x_1 P(z) B'(x_1 P(z))}{B'(1) B(x_1 P(z))} Q(x_1, x_2, \ldots, z), \end{aligned} \tag{8.20}$$

such that

$$\mathrm{E}[u_{\mathcal{S}+1}] = \frac{\partial}{\partial z} Q_{\mathcal{S}+1}(1,1,\ldots,1) = \mathrm{E}[u] + (1+\mathbb{B})\,P'(1), \tag{8.21}$$

where we introduced the shorthand $\mathbb{B} \triangleq \frac{B''(1)}{B'(1)} - B'(1)$ for convenience.

Given that all sessions share the same *iid* bandwidth distribution, we find that

$$\mathrm{E}[m_{\mathcal{S}+i}] = P'(1)\,\mathrm{E}\left[\sum_{n=1}^{\infty} a_{n,\mathcal{S}+i}\right] = P'(1)\,\mathrm{E}[\alpha_{\mathcal{S}+i}]\,, \tag{8.22}$$

where we introduced the shorthand $\alpha_k \triangleq \sum_{n=1}^{\infty} a_{n,k}$. Although (8.18) does not make use of $\mathrm{E}[m_{\mathcal{S}}]$ directly, we will require it later on. We could calculate $\mathrm{E}[m_{\mathcal{S}}]$ using the fact that $\mathrm{Prob}[a_{1,\mathcal{S}} = j] = \frac{j}{B'(1)}\mathrm{Prob}[a_1 = j]$, similar to (6.19), or we can calculate it directly from $Q_{\mathcal{S}+1}(x_1, x_2, \ldots, z)$ as

$$\begin{aligned}\mathrm{E}[m_{\mathcal{S}}] &= P'(1)\sum_{n=1}^{\infty}\mathrm{E}[a_{n,\mathcal{S}}] = P'(1)\sum_{n=1}^{\infty}\frac{\partial}{\partial x_n}Q_{\mathcal{S}+1}(1,1,\ldots,1)\\ &= \lambda + P'(1)\,(1+\mathbb{B})\,. \end{aligned} \tag{8.23}$$

For the mean number of packet arrivals in the subsequent $\ell-1$ slots $\mathcal{S}+i, i \in \{1,\ldots,\ell-1\}$, we know that there will always be at least one session that has been active for exactly $i+1$ slots, such that

$$\mathrm{E}[a_{1,\mathcal{S}+i}] = B'(1), \tag{8.24}$$

$$\begin{aligned}\mathrm{E}[a_{i+1,\mathcal{S}+i}] &= 1 + \pi(i)\,(\mathrm{E}[a_{i,\mathcal{S}+i-1}] - 1)\\ &= 1 + \frac{B''(1)}{B'(1)}\Lambda(i+1), \qquad 1 \le i \le \ell-1, \end{aligned} \tag{8.25}$$

$$\begin{aligned}\mathrm{E}[a_{n,\mathcal{S}+i}] &= \pi(n-1)\,\mathrm{E}[a_{n-1,\mathcal{S}+i-1}]\\ &= B'(1)\Lambda(n), \qquad 1 < n \ne i+1. \end{aligned} \tag{8.26}$$

Substitution of (8.24), (8.25) and (8.26) in (8.22) then yields

$$\mathrm{E}[m_{\mathcal{S}+i}] = \lambda + P'(1)\,(1 + \mathbb{B}\Lambda(i+1))\,, \qquad 1 \le i \le \ell-1. \tag{8.27}$$

Note that (8.23) is in fact consistent with (8.27), allowing us to expand the range for i to $0 \le i \le \ell-1$ in the latter.

In order to determine $\mathrm{E}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}\big]$, we note that due to the random order of arrivals during a slot, $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}$ only depends on the total number $m_{\bar{\mathcal{S}}}$ of arrivals during slot $\bar{\mathcal{S}}$ and the total number of packets generated by session $\mathcal{M}$ during slot $\bar{\mathcal{S}}$, which we will refer to as $p_{\bar{\mathcal{S}}}^{\mathcal{M}}$. The relation between $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}$, $m_{\bar{\mathcal{S}}}$ and $p_{\bar{\mathcal{S}}}^{\mathcal{M}}$ can thus be expressed as

$$\mathrm{Prob}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}} = x | m_{\bar{\mathcal{S}}} = m, p_{\bar{\mathcal{S}}}^{\mathcal{M}} = p\big] = \frac{\binom{m-x-1}{p-1}}{\binom{m}{p}}, \quad 0 \le x \le m-p. \tag{8.28}$$

Therefore, we can calculate $\mathrm{E}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}\big]$ as

$$\begin{aligned}\mathrm{E}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}\big] &= \sum_{j=1}^{\infty}\sum_{m=j}^{\infty}\sum_{p=1}^{m-j+1}\sum_{x=0}^{m-p} x\mathrm{Prob}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}} = x, m_{\bar{\mathcal{S}}} = m, p_{\bar{\mathcal{S}}}^{\mathcal{M}} = p, \alpha_{\bar{\mathcal{S}}} = j\big] \\ &= \sum_{j=1}^{\infty}\sum_{m=j}^{\infty}\sum_{p=1}^{m-j+1} \frac{m-p}{p+1}\mathrm{Prob}\big[m_{\bar{\mathcal{S}}} = m, p_{\bar{\mathcal{S}}}^{\mathcal{M}} = p, \alpha_{\bar{\mathcal{S}}} = j\big] \\ &= \sum_{j=1}^{\infty}\mathrm{E}\left[\left.\frac{m_{\bar{\mathcal{S}}} - p_{\bar{\mathcal{S}}}^{\mathcal{M}}}{p_{\bar{\mathcal{S}}}^{\mathcal{M}}+1}\right| \alpha_{\bar{\mathcal{S}}} = j\right]\mathrm{Prob}[\alpha_{\bar{\mathcal{S}}} = j]\,. \end{aligned} \tag{8.29}$$

In order to compute the conditional mean in (8.29), we will make use of a helper function $\Omega_j(x,y)$, defined as the conditional joint pgf of the random variables $p_{\bar{\mathcal{S}}}^{\mathcal{M}}$ and $m_{\bar{\mathcal{S}}}$, conditioned on $\alpha_{\bar{\mathcal{S}}} = j$. We can calculate $\Omega_j(x,y)$ as

$$\begin{aligned}\Omega_j(x,y) &\triangleq \mathrm{E}\Big[x^{p_{\bar{\mathcal{S}}}^{\mathcal{M}}} y^{m_{\bar{\mathcal{S}}}}\,|\alpha_{\bar{\mathcal{S}}} = j\Big] = \mathrm{E}\Big[x^{p_{\bar{\mathcal{S}}}^{\mathcal{M}}} y^{p_{\bar{\mathcal{S}}}^{\mathcal{M}}+\sum_{i=1}^{j-1} p_{\bar{\mathcal{S}}}^{i}}\,|\alpha_{\bar{\mathcal{S}}} = j\Big] \\ &= P(xy)P(y)^{j-1}, \end{aligned} \tag{8.30}$$

where $p_{\bar{\mathcal{S}}}^{i}$ is the number of packets generated by a session i during slot $\bar{\mathcal{S}}$. Note that this number is independent of the total number of active sessions during $\bar{\mathcal{S}}$ and the number of packets generated by any other session during slot $\bar{\mathcal{S}}$. We can now determine the conditional mean in (8.29) as

$$\begin{aligned}\mathrm{E}\left[\left.\frac{m_{\bar{\mathcal{S}}} - p_{\bar{\mathcal{S}}}^{\mathcal{M}}}{p_{\bar{\mathcal{S}}}^{\mathcal{M}}+1}\right| \alpha_{\bar{\mathcal{S}}} = j\right] &= \left.\left(\frac{\partial}{\partial y}\int_0^1 \Omega_j(x,y)\mathrm{d}x\right)\right|_{y=1} - 1 + \int_0^1 \Omega_j(x,1)\mathrm{d}x \\ &= (j-1)\,P'(1)\int_0^1 P(x)\mathrm{d}x, \end{aligned} \tag{8.31}$$

such that (8.29) eventually becomes

$$\begin{aligned}\mathrm{E}\big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}}\big] &= P'(1)\,(\mathrm{E}[\alpha_{\bar{\mathcal{S}}}] - 1)\int_0^1 P(x)\mathrm{d}x \\ &= (\lambda + P'(1)\mathbb{B}\Lambda(\ell))\int_0^1 P(x)\mathrm{d}x, \end{aligned} \tag{8.32}$$

where we made use of the property $P'(1)\,\mathrm{E}[\alpha_{\bar{\mathcal{S}}}] = \mathrm{E}[m_{\bar{\mathcal{S}}}] = \mathrm{E}[m_{\mathcal{S}+\ell-1}]$.

We can then find the mean session delay conditioned on the session length by substitution of (8.21), (8.27) and (8.32) in (8.18), such that

$$\begin{aligned} \mathrm{E}\big[d_{\mathcal{M}|\ell}\big] &= \frac{1}{\sigma}\Bigg[\mathrm{E}[u] + (1+\mathbb{B})\,P'(1) + (\ell-1)\,(\lambda + P'(1)) \\ &\qquad + P'(1)\mathbb{B}\sum_{i=1}^{\ell-1}\Lambda(i+1) - (\lambda + P'(1)\mathbb{B}\Lambda(\ell))\int_0^1 P(x)\mathrm{d}x\Bigg] \\ &= \frac{\mathrm{E}[u]}{\sigma} + \frac{P'(1)}{\sigma}\Bigg[1 + (\ell-1)\,(1+B'(1)L'(1)) + \mathbb{B}\sum_{i=1}^{\ell}\Lambda(i) \\ &\qquad - (B'(1)L'(1) + \mathbb{B}\Lambda(\ell))\int_0^1 P(x)\mathrm{d}x\Bigg], \end{aligned} \tag{8.33}$$

where we made use of $\Lambda(1) = 1$ and $\lambda = B'(1)L'(1)P'(1)$. Substituting (8.33) in (8.17) finally yields the unconditional mean session delay

$$\begin{aligned} \mathrm{E}[d_{\mathcal{M}}] = \frac{\mathrm{E}[u]}{\sigma} + \frac{P'(1)}{\sigma}\Bigg[1 + (L'(1)-1)\,(1+B'(1)L'(1)) + \mathbb{B}\sum_{i=1}^{\infty}\Lambda(i)^2 \\ - \left(B'(1)L'(1) + \mathbb{B}\sum_{n=1}^{\infty}\ell(n)\Lambda(n)\right)\int_0^1 P(x)\mathrm{d}x\Bigg]. \end{aligned} \tag{8.34}$$

8.7.2 Heterogeneous sessions ($T > 1$)

When considering multiple session types, we will not only condition the mean session delay on the session length, but also on the session type, such that the mean session delay $\mathrm{E}[d_{\mathcal{M}}]$ can be calculated as

$$\begin{aligned} \mathrm{E}[d_{\mathcal{M}}] = \sum_{t=1}^{T}\mathrm{Prob}[\text{session is of type } t] \\ \sum_{\ell=1}^{\infty}\mathrm{E}\big[d_{\mathcal{M}_t|\ell}\big]\,\mathrm{Prob}[\text{session } \mathcal{M}_t \text{ has length } \ell], \end{aligned} \tag{8.35}$$

where $\mathcal{M}_t$ denotes an arbitrary steady-state session of type t.

Just as in Section 8.7.1, the delay of a session $\mathcal{M}_t$, started during slot $\mathcal{S}$, is defined as the integer number of slots between the end of slot $\mathcal{S}$ and the end of the slot during which the session's final packet effectively leaves the system. The delay of an individual session is not affected directly by the fact that there are multiple session types, this allows us to calculate the mean delay $\mathrm{E}\big[d_{\mathcal{M}_t|\ell}\big]$ of a session $\mathcal{M}_t$ of type t having a duration of ℓ slots as

$$\mathrm{E}\big[d_{\mathcal{M}_t|\ell}\big] = \frac{1}{\sigma}\left(\mathrm{E}\big[u_{\mathcal{S}+1|t}\big] + \sum_{i=1}^{\ell-1}\mathrm{E}\big[m_{\mathcal{S}+i|t}\big] - \mathrm{E}\Big[\chi_{\mathcal{S}}^{\mathcal{M}_t}\Big]\right), \tag{8.36}$$

where the index extension $|t$ denotes that the corresponding random variable is conditioned on the type of the selected session $\mathcal{M}_t$.

As argued before, $\mathcal{S}$ is not an arbitrary slot, but it is the first slot of the randomly chosen session $\mathcal{M}_t$ of type t. The probability that there are $b_{\mathcal{S}}(t)$ newly initiated sessions during slot $\mathcal{S}$ is therefore proportional to $b_{\mathcal{S}}(t)$, such that we can find the pmf of $b_{\mathcal{S}}(t)$ as

$$\mathrm{Prob}[b_{\mathcal{S}}(t)=\beta]=\frac{\beta}{B_t'(1)}\mathrm{Prob}[b_k(t)=\beta]\,,\qquad \beta\geq 1, \tag{8.37}$$

just like (5.25) and (8.19). Conversely, slot $\mathcal{S}$ does have the same system state distribution as at the beginning of a random steady-state slot. From these observations, we can find the joint pgf $Q_{\mathcal{S}+1|t}(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z)$ of the system state at the beginning of slot $\mathcal{S}+1$, conditioned on t similar as in (8.20) as

$$\begin{aligned}Q_{\mathcal{S}+1|t}(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z) &\triangleq \mathrm{E}\left[\left(\prod_{\tau=1}^{T}\prod_{n=1}^{\infty}{x_{n,\tau}}^{a_{n,\mathcal{S}|t}(\tau)}\right)z^{u_{\mathcal{S}+1|t}}\right]\\ &=\frac{x_{1,t}}{B_t'(1)}\frac{\partial}{\partial x_{1,t}}Q(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z)\\ &=\frac{x_{1,t}P_t(z)B_t'(x_{1,t}P_t(z))}{B_t'(1)B_t(x_{1,t}P_t(z))}Q(\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_T,z).\end{aligned} \tag{8.38}$$

Hence, the mean system content at the beginning of slot $\mathcal{S}+1$ follows as

$$\mathrm{E}\big[u_{\mathcal{S}+1|t}\big]=\frac{\partial}{\partial z}Q_{\mathcal{S}+1|t}(\underline{\mathbf{1}},\ldots,\underline{\mathbf{1}},1)=\mathrm{E}[u]+(1+\mathbb{B}_t)\,P_t'(1), \tag{8.39}$$

where we introduced the shorthand $\mathbb{B}_t\triangleq\frac{B_t''(1)}{B_t'(1)}-B_t'(1)$ for convenience.

Next, we need to determine the number of packet arrivals during each of the slots $\mathcal{S}+i, 0\leq i\leq \ell-1$ during which the session $\mathcal{M}_t$ is active. For later use however, we first introduce the variables $\alpha_{\mathcal{S}+i|t}(\tau)$ as

$$\alpha_{\mathcal{S}+i|t}(\tau)\triangleq\sum_{n=1}^{\infty}a_{n,\mathcal{S}+i|t}(\tau),\qquad 1\leq\tau\leq T, \tag{8.40}$$

which can be combined into the vector $\underline{\boldsymbol{\alpha}}_{\mathcal{S}+i|t}\triangleq\big\langle\alpha_{\mathcal{S}+i|t}(1),\ldots,\alpha_{\mathcal{S}+i|t}(T)\big\rangle$. The mean number of packet arrivals during slot $\mathcal{S}+i$ then follows as

$$\mathrm{E}\big[m_{\mathcal{S}+i|t}\big]=\sum_{\tau=1}^{T}P_\tau'(1)\,\mathrm{E}\left[\sum_{n=1}^{\infty}a_{n,\mathcal{S}+i|t}(\tau)\right]=\sum_{\tau=1}^{T}P_\tau'(1)\,\mathrm{E}\big[\alpha_{\mathcal{S}+i|t}(\tau)\big]\,. \tag{8.41}$$

To obtain $\mathrm{E}\big[m_{S|t}\big]$ we note that the mean number of active sessions of type τ in their nth slot during slot $\mathcal{S}$, denoted by $\mathrm{E}\big[a_{n,\mathcal{S}|t}(\tau)\big]$, can be found from the system state pgf at the beginning of slot $\mathcal{S}+1$ (8.38). This yields

$$\begin{aligned}\mathrm{E}\big[m_{\mathcal{S}|t}\big] &= \sum_{\tau=1}^{T} P_\tau'(1) \sum_{n=1}^{\infty} \mathrm{E}\big[a_{n,\mathcal{S}|t}(\tau)\big] \\ &= \sum_{\tau=1}^{T} P_\tau'(1) \sum_{n=1}^{\infty} \frac{\partial}{\partial x_{n,\tau}} Q_{\mathcal{S}+1|t}(\underline{\mathbf{1}},\dots,\underline{\mathbf{1}},1) \\ &= \lambda + P_t'(1)\left(1+\mathbb{B}_t\right). \end{aligned} \tag{8.42}$$

For the remaining slots $\mathcal{S}+i$ ($i \in \{1,\dots,\ell-1\}$) we find that

$$\mathrm{E}\big[a_{1,\mathcal{S}+i|t}(\tau)\big] = B_\tau'(1), \tag{8.43}$$

$$\begin{aligned}\mathrm{E}\big[a_{i+1,\mathcal{S}+i|t}(t)\big] &= 1 + \pi_t(i)\left(\mathrm{E}\big[a_{i,\mathcal{S}+i-1|t}(t)\big] - 1\right) \\ &= 1 + \frac{B_t''(1)}{B_t'(1)} \Lambda_t(i+1) \end{aligned} \tag{8.44}$$

$$\begin{aligned}\mathrm{E}\big[a_{n,\mathcal{S}+i|t}(\tau)\big] &= \pi_\tau(n-1)\,\mathrm{E}\big[a_{n-1,\mathcal{S}+i-1|t}(\tau)\big] \\ &= B_\tau'(1)\Lambda_\tau(n), \qquad 1 < n, (n \neq i+1 \vee \tau \neq t). \end{aligned} \tag{8.45}$$

Note the difference between the expressions (8.44) and (8.45) on one hand and the corresponding expressions (8.25) and (8.26) from the previous section on the other hand. In each of the slots $\mathcal{S}+i, 1 \leq i \leq \ell-1$, we only know that there is at least 1 active session $\mathcal{M}_t$ of type t that is in its $(i+1)$th slot, which is reflected in (8.44). Any session active in any of those slots that is either of type $\tau \neq t$ or not in its $(i+1)$th slot during $\mathcal{S}+1$, is most certainly not the selected session $\mathcal{M}_t$, but rather a completely random session as is reflected by (8.45). Substitution of (8.43), (8.44) and (8.45) in (8.41) then yields

$$\mathrm{E}\big[m_{\mathcal{S}+i|t}\big] = \lambda + P_t'(1)\left(1 + \mathbb{B}_t \Lambda_t(i+1)\right), \qquad 1 \leq i \leq \ell-1, \tag{8.46}$$

where we applied the property $\sum_{n=1}^{\infty} \Lambda_t(n) = L_t'(1)$. Again we note that (8.42) and (8.46) are consistent, such that (8.46) holds true for $0 \leq i \leq \ell-1$.

Finally we need to determine the mean number of packets $\mathrm{E}\Big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}\Big]$ arriving during the final slot $\bar{\mathcal{S}}$ of session $\mathcal{M}_t$, but after the session's final packet. Since the order of the individual packet arrivals is purely random, $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$ only depends on the total number $m_{\bar{\mathcal{S}}|t}$ of packet arrivals during slot $\bar{\mathcal{S}}$ and the number of packets $p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$ generated by session $\mathcal{M}_t$ during slot $\bar{\mathcal{S}}$. The pmf of $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$, conditioned on $m_{\bar{\mathcal{S}}|t}$ and $p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$ can then be found as

$$\mathrm{Prob}\Big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t} = x | m_{\bar{\mathcal{S}}|t} = m, p_{\bar{\mathcal{S}}}^{\mathcal{M}_t} = p\Big] = \frac{\binom{m-x-1}{p-1}}{\binom{m}{p}}, \quad 0 \leq x \leq m-p, \tag{8.47}$$

such that the mean value of $\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$ can be found as

$$\begin{aligned}\mathrm{E}\Big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}\Big] &= \sum_{j_1,\ldots,j_T=0}^{\infty}\sum_{m=J}^{\infty}\sum_{p=1}^{m-J+1}\sum_{x=0}^{m-p} \\ &\qquad x\mathrm{Prob}\Big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}=x, m_{\bar{\mathcal{S}}|t}=m, p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}=p, \underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\Big] \\ &= \sum_{j_1,\ldots,j_T=0}^{\infty}\sum_{m=J}^{\infty}\sum_{p=1}^{m-J+1}\frac{m-p}{p+1}\mathrm{Prob}\Big[m_{\bar{\mathcal{S}}|t}=m, p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}=p, \underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\Big] \\ &= \sum_{j_1,\ldots,j_T=0}^{\infty}\mathrm{E}\left[\frac{m_{\bar{\mathcal{S}}|t}-p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}}{p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}+1}\middle|\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\right]\mathrm{Prob}\Big[\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\Big], \qquad (8.48)\end{aligned}$$

where $\underline{\mathbf{j}} = (j_1, \ldots, j_T)$ and $J = \sum_{\tau=1}^{T} j_\tau$. Similarly as in the previous section, the calculation of the conditional mean in (8.48) can be simplified by using the joint pgf of $m_{\bar{\mathcal{S}}|t}$ and $p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}$, conditioned on $\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}$. This pgf is given by

$$\begin{aligned}\Omega_{\underline{\mathbf{j}}|t}(x,y) &\triangleq \mathrm{E}\Big[x^{p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}}y^{m_{\bar{\mathcal{S}}|t}}\Big|\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\Big] \\ &= \mathrm{E}\Big[(xy)^{p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}}y^{\sum_{\tau=1}^{T}\sum_{i=1}^{j_\tau}p_{\bar{\mathcal{S}}}^{i}(\tau)-p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}}\Big|\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\Big] \\ &= \frac{P_t(xy)}{P_t(y)}\prod_{\tau=1}^{T}P_\tau(y)^{j_\tau}, \qquad (8.49)\end{aligned}$$

where $p_{\bar{\mathcal{S}}}^{i}(\tau)$ is the number of packets generated by a session i of type τ during slot $\bar{\mathcal{S}}$. Note that this number is independent of the total number of active sessions of any type during $\bar{\mathcal{S}}$ and the number of packets generated by any other session during slot $\bar{\mathcal{S}}$. By means of (8.49), the conditional mean in (8.48) then becomes

$$\begin{aligned}\mathrm{E}\left[\frac{m_{\bar{\mathcal{S}}|t}-p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}}{p_{\bar{\mathcal{S}}}^{\mathcal{M}_t}+1}\middle|\underline{\boldsymbol{\alpha}}_{\bar{\mathcal{S}}|t}=\underline{\mathbf{j}}\right] &= \left(\frac{\partial}{\partial y}\int_0^1\Omega_{\underline{\mathbf{j}}|t}(x,y)\mathrm{d}x\right)\Bigg|_{y=1} - 1 + \int_0^1\Omega_{\underline{\mathbf{j}}|t}(x,1)\mathrm{d}x \\ &= \left(\sum_{\tau=1}^{T}j_\tau P_\tau'(1) - P_t'(1)\right)\int_0^1 P_t(x)\mathrm{d}x. \qquad (8.50)\end{aligned}$$

Substitution in (8.48) therefore yields

$$\begin{aligned}\mathrm{E}\Big[\chi_{\bar{\mathcal{S}}}^{\mathcal{M}_t}\Big] &= \left(\sum_{\tau=1}^{T}P_\tau'(1)\,\mathrm{E}\big[\alpha_{\bar{\mathcal{S}}|t}(\tau)\big] - P_t'(1)\right)\int_0^1 P_t(x)\mathrm{d}x \\ &= (\lambda + P_t'(1)\mathbb{B}_t\Lambda_t(\ell))\int_0^1 P_t(x)\mathrm{d}x. \qquad (8.51)\end{aligned}$$

The mean delay of sessions of type t, conditioned on the session length can then be found by substitution of (8.39), (8.46) and (8.51) in (8.36) as

$$\mathrm{E}\big[d_{\mathcal{M}_t|\ell}\big] = \frac{1}{\sigma}\Bigg[\mathrm{E}[u] - \lambda + \ell\,(\lambda + P_t'(1)) + P_t'(1)\mathbb{B}_t \sum_{i=1}^{\ell} \Lambda_t(i) \\ - (\lambda + P_t'(1)\mathbb{B}_t\Lambda_t(\ell)) \int_0^1 P_t(x)\mathrm{d}x\Bigg], \tag{8.52}$$

such that the mean session delay of a random type t session becomes

$$\begin{aligned}\mathrm{E}[d_{\mathcal{M}_t}] &= \sum_{n=1}^{\infty} \ell_t(n)\,\mathrm{E}\big[d_{\mathcal{M}_t|\ell}\big] \\ &= \frac{1}{\sigma}\Bigg[\mathrm{E}[u] - \lambda + L_t'(1)\,(\lambda + P_t'(1)) + P_t'(1)\mathbb{B}_t \sum_{n=1}^{\infty} \ell_t(n) \sum_{i=1}^{n} \Lambda_t(i) \\ &\qquad - \left(\lambda + P_t'(1)\mathbb{B}_t \sum_{n=1}^{\infty} \ell_t(n)\Lambda_t(n)\right) \int_0^1 P_t(x)\mathrm{d}x\Bigg]. \end{aligned} \tag{8.53}$$

In order to remove the conditioning on the session type, we need to determine the probability that a randomly chosen session is of a specific type t. This probability corresponds to the portion of sessions of type t among all sessions, and can be calculated as

$$\text{Prob[random session is of type } t] = \frac{B_t'(1)}{\sum_{\tau=1}^{T} B_\tau'(1)}. \tag{8.54}$$

Finally we can determine the unconditional mean session delay as

$$\begin{aligned}\mathrm{E}[d_{\mathcal{M}}] &= \sum_{t=1}^{T} \text{Prob[random session is of type } t]\,\mathrm{E}[d_{\mathcal{M}_t}] \\ &= \frac{\mathrm{E}[u] - \lambda}{\sigma} + \frac{\sum_{t=1}^{T} B_t'(1)L_t'(1)\,(\lambda + P_t'(1))}{\sigma\sum_{\tau=1}^{T} B_\tau'(1)} \\ &\quad + \frac{1}{\sigma\sum_{\tau=1}^{T} B_\tau'(1)} \sum_{t=1}^{T} B_t'(1)\Bigg[P_t'(1)\mathbb{B}_t \sum_{n=1}^{\infty} \ell_t(n) \sum_{i=1}^{n} \Lambda_t(i) \\ &\qquad - \left(\lambda + P_t'(1)\mathbb{B}_t \sum_{n=1}^{\infty} \ell_t(n)\Lambda_t(n)\right) \int_0^1 P_t(x)\mathrm{d}x\Bigg]. \end{aligned} \tag{8.55}$$

8.8 Numerical examples

In the remainder of this chapter we will illustrate the effects of the correlation embedded in session-based arrival processes on the mean packet and

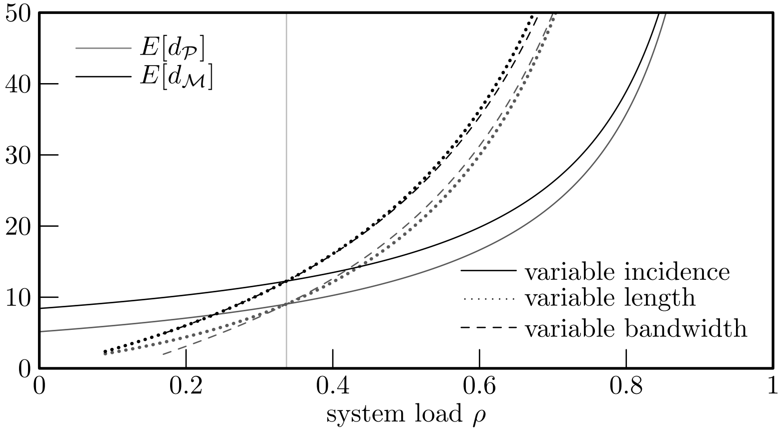

Figure 8.4: The mean packet and session delay as a function of the system load $\rho = \lambda/\sigma$.

mean session delay by means of some numerical examples. In the previous chapters, we typically illustrated the effect of the system load on the delay by varying the arrival rate λ while keeping the service rate fixed. With session-based arrivals however, the arrival rate itself depends on four system parameters: the number of session types T and the session incidence, bandwidth and length distributions.

Therefore, we start by illustrating the effect of the different characteristic distributions in Figure 8.4. This figure shows the mean packet delay (gray lines) and the mean session delay (black lines) as a function of the system load for homogeneous sessions ($T = 1$) and a transmission rate $\sigma = 0.95$. The session incidence distribution is a Poisson distribution, the session length is shifted geometrically distributed and the session bandwidth has a shifted binomial distribution $B(n, p)$ with $n = 50$. For each pair of curves, two of the three distributions are kept fixed, while the third one is variable. The default parameters for the distributions are such that $B'(1) = 0.04$, $L'(1) = 4$ and $P'(1) = 2$. The gray vertical line marks the point for which all parameters have their default values. We see that the different characteristic distributions of the session-based arrival process each have a unique effect on the mean packet and session delay. Due to the fact that new sessions start independently from each other, the session incidence distribution does not introduce any correlation. Therefore, in case of a change in the system load due to a change in the session arrival rate $B'(1)$, the net effect on the mean packet and session delay is less pronounced than if the system load change were caused by a change in either the mean session length or the mean session bandwidth. This is reflected in Figure 8.4 by the difference in the relative positioning of the curves left and right

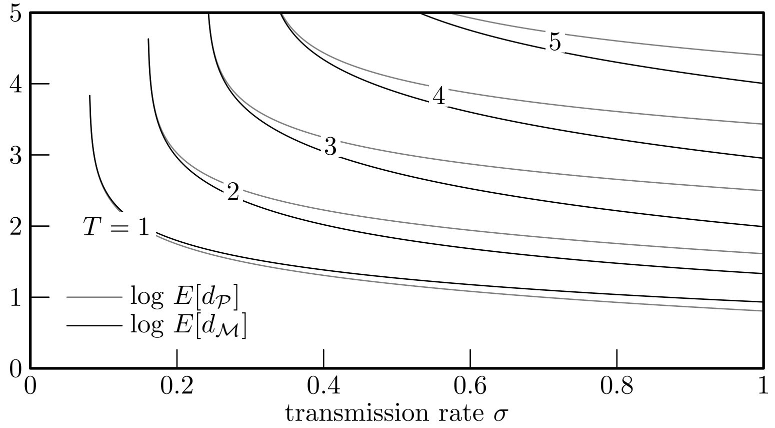

Figure 8.5: The mean packet and session delay as a function of the transmission rate σ.

of the grey vertical line.

Next, we illustrate the effect of the transmission rate σ on the mean packet and session delay on a logarithmic scale for various numbers of session types T in Figure 8.5. The number of new sessions of type t ($t \in \{1, \ldots, T\}$) in a random slot is Poisson distributed with mean $B_t'(1) = 10^{-t-1}$ and have a shifted geometrically distributed length with mean $L_t'(1) = 2 \times 10^{t-1}$. The bandwidth of each session has a shifted binomial distribution $B(n, p)$ with $n = 50$ with mean $P_t'(1) = 4$, identical for each session type. As such, sessions of type t generally are longer than sessions of type $t' < t$, but they occur less frequent. By construction, the session type t is identical for each configuration where $t \leq T$. Therefore, all differences in the mean delays for successive values of T are exclusively caused by the additional session types. Given that the mean packet arrival rate $\mathrm{E}[m]$ increases for successive values of T, the leftmost point for successive curves moves to the right while the curves move up. As expected, we see that the mean delays decrease when the transmission rate σ increases. Note however that Figure 8.5 exhibits the peculiar property that under the right circumstances, the mean session delay can be smaller than the mean packet delay. This rather counterintuitive effect is a direct result of the definition of both the mean packet delay and the mean session delay. Note that every session contributes equally to the mean session delay, whether it consists of few packets only or a vast number of packets. Therefore, the mean session delay is highly influenced by session types with a high incidence rate. Conversely, the mean packet delay is obtained by averaging the delay of any random packet, such that sessions that generate a very high number of packets can have a significant impact on the mean packet delay, even if such sessions do not occur very often.

This explains the counterintuitive result presented in Figure 8.5.

Chapter 9

Geometric Train Arrivals and Markovian Output Line Interruptions

9.1 Introduction

It has been argued before that tailor-made queueing models, specific to certain real-life queueing systems generally yield better, more accurate results then general-purpose queueing models. In Chapter 8, we introduced the session-based arrival process as a more realistic model to approximate the outbound server traffic in common network scenarios than the general independent arrival process feeding the $GI-GI-1$ model. The augmented level of detail, provided by the session-based arrivals came at the cost of a more challenging mathematical analysis of the key performance measures. In this chapter however, we raise the bar further, by imposing a more complex process governing the output line interruptions.

In literature, different types of queueing models with output line interruptions have been studied before. In [8, 10, 51, 58, 63] various models are studied, both for a single server setting as for multi-server applications, with interruptions characterized by a single parameter f denoting the fraction of the time the output line is accessible. In [9, 66], the output line interruptions are governed by an On/Off-process, characterized by two independent geometric distributions describing the duration of the accessible periods and interrupted periods respectively. A more complex output line interruption mechanism is studied in [28], where the arrival process is *iid* and the output

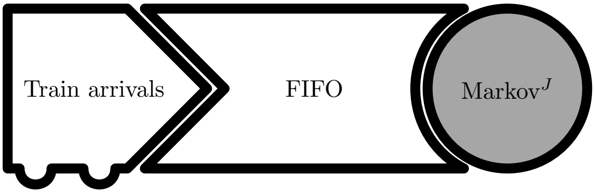

Figure 9.1: Illustration of a system with train arrivals and Markovian output line interruptions

line is governed by a two-state Markov process. The output line is then interrupted with a probability e_j, dependent on the state j of the Markov process, this allows for an intuitive representation of a good state 0 and a bad state 1, by choosing $0 \leq e_0 < e_1 \leq 1$.

In this chapter, representing my contributions [42, 44], we not only generalize this last model to a J-state Markov process governing the output line state, but we also assume that the packets are part of geometric trains. As described earlier, packet trains are multi-packet entities that during their lifetime generate exactly one packet per slot. As such, the model studied here will exhibit two sources of correlation: at the arrival side there is the time correlation induced by the train arrival process; at the departure side there is again time correlation due to the Markovian output line interruptions. This is unlike the existing research papers, where at least one of these processes was kept simple and uncorrelated.

9.2 Mathematical model

We consider the discrete-time queueing model illustrated in Figure 9.1, i.e. a FIFO-queue fed by a train arrival process generating packets with transmission times of exactly 1 slot per packet over an unreliable output line. The output line is prone to interruptions governed by a Markov process with J states, each with its own probability for the output line itself to be accessible or interrupted.

The packet trains arrive to the system according to an *iid* distribution, characterized by its pgf $B(z)$ and are assumed to have shifted geometric lengths, such that each individual packet has a fixed probability $1 - \gamma$ of being the last packet of its corresponding train. The total number of active trains a_k during a random slot k can therefore be calculated as

$$a_k = b_k + \sum_{i=1}^{a_{k-1}} c_k^i, \tag{9.1}$$

where b_k is the number of new trains started in slot k and c_k^i is a Bernoulli variable equal to 1 if and only if the ith train that was active during slot

$k-1$ continues in slot k. The random variables c_k^i therefore have the pgf

$$C(z) \triangleq 1-\gamma+\gamma z. \tag{9.2}$$

As mentioned before, this causes the trains to have a shifted geometric length ℓ, with pgf and mean given by

$$L(z)=\frac{(1-\gamma)\,z}{1-\gamma z}, \qquad \text{and} \qquad \mathrm{E}[\ell]=L'(1)=\frac{1}{1-\gamma}. \tag{9.3}$$

Note that, due to the fixed rate of 1 packet per train per slot, the number of active trains a_k during a random slot k is equal to the number of packet arrivals during that slot.

The output line interruptions are governed by a Markovian process with $J \geq 1$ states $j \in \{1,2,\ldots,J\}$. The actual state of this Markovian process, or (less verbose) *the output line state*, in a slot k is denoted by the random variable s_k. Transitions between different states occur at slot boundaries and are described by the transition matrix $\mathbf{H}$, with the individual transition probabilities as its elements:

$$[\mathbf{H}]_{jj'}=\sigma_{j'|j} \triangleq \mathrm{Prob}[\,s_k=j'\,|\,s_{k-1}=j\,]\,. \tag{9.4}$$

As a shorthand we also introduce σ_j as the probability for the Markovian process to stay in state j, i.e. $\sigma_j \triangleq \sigma_{j|j}$. Given that $\mathbf{H}$ is a stochastic matrix, corresponding to an irreducible Markov chain, the stationary probability vector $\underline{\boldsymbol{\pi}}$ can be found as a normalized left eigenvector of $\mathbf{H}$ corresponding to eigenvalue 1, i.e. $\underline{\boldsymbol{\pi}}=\underline{\boldsymbol{\pi}}\;\mathbf{H}$ and $\underline{\boldsymbol{\pi}}\cdot\underline{\mathbf{e_J}}=1$, where $\underline{\mathbf{e_J}}$ is a column vector of order J with all elements equal to 1. The jth element π_j of the row vector $\underline{\boldsymbol{\pi}}$ corresponds to the probability that the output line is in state j at the beginning of a random steady-state slot.

When the output line is in state j, the accessibility of the line is governed by a Bernoulli distribution with pgf

$$H_j(z) \triangleq 1-\eta_j+\eta_j z, \tag{9.5}$$

such that the line is accessible with probability η_j and interrupted with probability $1-\eta_j$. The J probabilities η_j can be collected in the diagonal matrix $\boldsymbol{\eta} \triangleq \mathrm{diag}\,(\eta_1,\eta_2,\ldots,\eta_J)$. This accessibility matrix $\boldsymbol{\eta}$ can be used to decompose the transition matrix $\mathbf{H}$ into $\mathbf{H_0}$ and $\mathbf{H_1}$, with

$$\mathbf{H_0} \triangleq \mathbf{H}\,(\mathbf{I_J}-\boldsymbol{\eta})\,, \qquad \text{and} \qquad \mathbf{H_1} \triangleq \mathbf{H}\;\boldsymbol{\eta}, \tag{9.6}$$

where $\mathbf{I_J}$ is the $J\times J$ identity matrix, such that $\mathbf{H_0}+\mathbf{H_1}=\mathbf{H}$. From $\mathbf{H_0}$ and $\mathbf{H_1}$, we can construct the matrix generating function $\mathbf{H}^*(z) \triangleq \mathbf{H_0}+\mathbf{H_1}z$ as the matrix counterpart of the functions $H_j(z)$.

An interesting metric concerning the Markovian process is the steady-state lag-κ correlation coefficient $\phi(\kappa)$ between the output line state in a random slot k and the output line state in slot $k-\kappa$, defined as

$$\phi(\kappa) \triangleq \lim_{k\to\infty} \rho_{s_k,s_{k-\kappa}} = \lim_{k\to\infty} \frac{\mathrm{E}[s_k s_{k-\kappa}] - \mathrm{E}[s_k]\,\mathrm{E}[s_{k-\kappa}]}{\sqrt{\mathrm{Var}[s_k]\,\mathrm{Var}[s_{k-\kappa}]}}. \tag{9.7}$$

The lag-κ correlation coefficient $\phi(\kappa)$ ($\in [-1,1]$) is a measure of the statistical dependence between the states of the Markovian process in two slots that are κ slots apart. Specifically for $\kappa = 1$, we find the correlation coefficient between the output line state in two subsequent slots as

$$\phi(1) = \frac{\sum_{j=1}^{J} j\pi_j \sum_{j'=1}^{J} j'\sigma_{j'|j} - \left(\sum_{j=1}^{J} j\pi_j\right)^2}{\sum_{j=1}^{J} j^2\pi_j - \left(\sum_{j=1}^{J} j\pi_j\right)^2}. \tag{9.8}$$

Note that the correlation coefficients only tell a part of the story and are unable to quantify every aspect of the correlated nature of the Markovian process. Especially for larger values of J, it can be understood that contributions from a few unique states can be overshadowed by the contributions of many other states.

9.3 Packet arrival process and system load

In the previous section, we already gave a short description of the arrival process, explaining how trains are initiated and how they evolve. In this section, we further focus on the arrival process and determine some interesting results such as the mean packet arrival rate and the system load.

From (9.1), the steady-state pgf of the number of packet arrivals in a random slot can be found implicitly as

$$A(z) = B(z)A(C(z)) = B(z)A(1-\gamma+\gamma z). \tag{9.9}$$

Recursive application of this equation results in the closed-form expression

$$A(z) = \prod_{i=0}^{\infty} B(1-\gamma^i+\gamma^i z) = \prod_{i=0}^{\infty} B(C_i(z)), \tag{9.10}$$

where we introduced the shorthand function $C_i(z) \triangleq 1-\gamma^i+\gamma^i z$. In [95] it has been proved that the infinite product in (9.10) converges for all $\gamma \in]0,1[$. Although this expression contains an infinite product, we can determine the mean packet arrival rate $\lambda \triangleq \mathrm{E}[a]$ explicitly from (9.10) as

$$\lambda \triangleq \mathrm{E}[a] = A'(1) = B'(1)\sum_{i=0}^{\infty} C_i'(1) = \frac{B'(1)}{1-\gamma} = B'(1)L'(1). \tag{9.11}$$

In order to determine the system load, we still need to determine the effective service rate, i.e. the mean number of packets that can actually leave the system per slot, taking into account the output line interruptions. Given that the transmission times are deterministic of exactly one slot per packet, the effective service rate is equal to the rate at which the output line is accessible, which can be calculated as

$$\text{Prob[output line is accessible]} = \sum_{j=1}^{J} \pi_j \eta_j = \underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}}. \tag{9.12}$$

Thus, the system load can be determined as

$$\rho = \frac{\lambda}{\underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}}}. \tag{9.13}$$

9.4 System equations and buffer analysis

Quite similar to the systems with session-based arrivals and geometric output line interruptions studied in Chapter 8, the system content at the beginning of slot $k+1$ can be expressed in terms of random variables pertaining to slot k as

$$u_{k+1} = (u_k - r_k)^+ + a_k, \tag{9.14}$$

where r_k is a random variable that represents the output line accessibility during slot k. Specifically, we have that

$$r_k = \begin{cases} 0, & \text{with probability } 1 - \eta_{s_k}, \\ 1, & \text{with probability } \eta_{s_k}. \end{cases} \tag{9.15}$$

It follows from equations (9.1), (9.2), (9.4), (9.14) and (9.15) that the set of vectors $\{\langle a_{k-1}, s_{k-1}, u_k \rangle\}$ forms a three-dimensional Markov chain, such that we choose $\langle a_{k-1}, s_{k-1}, u_k \rangle$ to be the system state vector at the beginning of a random slot k. The joint system state pgf $P_k(x, y, z)$ for a random slot k is then defined by

$$P_k(x, y, z) \triangleq \mathrm{E}\Big[x^{a_{k-1}} y^{(s_k - 1)} z^{u_k}\Big]. \tag{9.16}$$

Given the complex nature of the queueing system, we will not determine this pgf directly, rather we introduce the partial pgfs $P_{j,k}(x, z)$ ($j \in \{1, \ldots, J\}$) as

$$P_{j,k}(x, z) \triangleq \mathrm{E}[x^{a_{k-1}} z^{u_k} \{s_{k-1} = j\}]. \tag{9.17}$$

In view of our further matrix-based calculations, it will prove useful to combine these partial pgfs into a row vector of order J, thus yielding the vector generating function $\underline{\mathbf{P_k^*}}(x, z) \triangleq [P_{1,k}(x, z), \ldots, P_{J,k}(x, z)]$.

From the system equations, we can then determine the partial system state pgfs $P_{j,k+1}(x,z)$ as

$$\begin{aligned}
P_{j,k+1}(x,z) &\triangleq \mathrm{E}[x^{a_k} z^{u_{k+1}} \{s_k = j\}] = \mathrm{E}\Big[(xz)^{a_k} z^{(u_k - r_k)^+} \{s_k = j\}\Big] \\
&= B(xz)\,\mathrm{E}\Big[(C(xz))^{a_{k-1}} z^{(u_k - r_k)^+} \{s_k = j\}\Big] \\
&= B(xz)\Big((1-\eta_j)\,\mathrm{E}[(C(xz))^{a_{k-1}} z^{u_k} \{s_k = j\}] \\
&\qquad + \eta_j\,\mathrm{E}\Big[(C(xz))^{a_{k-1}} z^{(u_k - 1)^+} \{s_k = j\}\Big]\Big) \\
&= B(xz)\Big(\eta_j \frac{z-1}{z}\mathrm{Prob}[u_k = 0, s_k = j] \\
&\qquad + \hat{H}_j(z)\,\mathrm{E}[(C(xz))^{a_{k-1}} z^{u_k} \{s_k = j\}]\Big), \qquad (9.18)
\end{aligned}$$

where we introduced the shorthand $\hat{H}_j(z) \triangleq H_j(1/z) = 1 - \eta_j + \eta_j/z$. The partial expectation $\mathrm{E}[(C(xz))^{a_{k-1}} z^{u_k} \{s_k = j\}]$ in the right hand side of (9.18) can be calculated as

$$\begin{aligned}
\mathrm{E}[(C(xz))^{a_{k-1}} z^{u_k} \{s_k = j\}] &= \sum_{j'=1}^{J} \sigma_{j|j'}\,\mathrm{E}[(C(xz))^{a_{k-1}} z^{u_k} \{s_{k-1} = j'\}] \\
&= \sum_{j'=1}^{J} \sigma_{j|j'} P_{j',k}(C(xz), z) \\
&= \Big[\underline{\mathbf{P_k^*}}(C(xz), z)\ \mathbf{H}\Big]_j. \qquad (9.19)
\end{aligned}$$

The probability $\mathrm{Prob}[u_k = 0, s_k = j]$ can be found similarly as

$$\begin{aligned}
\mathrm{Prob}[u_k = 0, s_k = j] &= \sum_{j'=1}^{J} \mathrm{Prob}[u_k = 0, s_{k-1} = j', s_k = j] \\
&= \sum_{j'=1}^{J} \sigma_{j|j'} \mathrm{Prob}[u_k = 0, s_{k-1} = j'] \\
&= \Big[\underline{\mathbf{P_k^*}}(0,0)\ \mathbf{H}\Big]_j. \qquad (9.20)
\end{aligned}$$

Substitution of these results allows us to rewrite (9.18) more compactly as

$$P_{j,k+1}(x,z) = B(xz)\left[\frac{z-1}{z}\underline{\boldsymbol{\nu}_\mathbf{k}} + \underline{\mathbf{P_k^*}}(C(xz), z)\ \hat{\mathbf{H}}^*(z)\right]_j, \qquad (9.21)$$

where $\underline{\boldsymbol{\nu}_\mathbf{k}} \triangleq \underline{\mathbf{P_k^*}}(0,0)\mathbf{H_1}$ and $\hat{\mathbf{H}}^*(z) \triangleq \mathbf{H}^*(1/z) = \mathbf{H_0} + \mathbf{H_1}/z$. Collecting these partial pgfs into a row vector, we get the vector generating function

$$\underline{\mathbf{P_{k+1}^*}}(x,z) = B(xz)\left[\frac{z-1}{z}\underline{\boldsymbol{\nu}_\mathbf{k}} + \underline{\mathbf{P_k^*}}(C(xz), z)\ \hat{\mathbf{H}}^*(z)\right]. \qquad (9.22)$$

Taking the limit of (9.22) for $k \to \infty$ then yields the following expression for its steady-state counterpart $\underline{\mathbf{P}^*}(x,z)$:

$$\underline{\mathbf{P}^*}(x,z) = B(xz)\left[\frac{z-1}{z}\underline{\boldsymbol{\nu}} + \underline{\mathbf{P}^*}(C(xz),z)\ \hat{\mathbf{H}}^*(z)\right], \tag{9.23}$$

where $\underline{\boldsymbol{\nu}} \triangleq \underline{\mathbf{P}^*}(0,0)\mathbf{H_1}$ is the steady-state counterpart of $\underline{\boldsymbol{\nu}_\mathbf{k}}$.

We now present a method for the determination of the unknown vector $\underline{\boldsymbol{\nu}}$. Note that $\underline{\boldsymbol{\nu}}$ contains J components $\nu_j \triangleq [\underline{\boldsymbol{\nu}}]_j$ ($j \in \{1,2,\ldots,J\}$), so that composing a system of J linearly independent equations in ν_j is sufficient to determine $\underline{\boldsymbol{\nu}}$. Therefore, we start by eliminating the recursion in (9.23) by choosing x such that the first arguments of the vector function $\underline{\mathbf{P}^*}(.,z)$ on either side of the expression become equal. In other words, we choose $x = x(z)$ with

$$\begin{aligned} x(z) &= C(zx(z)) = 1-\gamma+\gamma z x(z) \\ &= \frac{1-\gamma}{1-\gamma z} = \frac{L(z)}{z}. \end{aligned} \tag{9.24}$$

Substitution in (9.23) allows us to eliminate the recursion as

$$\underline{\mathbf{P}^*}(x(z),z)\left(\mathbf{I_J} - B(L(z))\ \hat{\mathbf{H}}^*(z)\right) = \frac{z-1}{z}B(L(z))\ \underline{\boldsymbol{\nu}}, \tag{9.25}$$

which we in general can use to find an explicit expression for $\underline{\mathbf{P}^*}(x(z),z)$ as

$$\underline{\mathbf{P}^*}(x(z),z) = \frac{(z-1)\,B(L(z))\ \underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{M}^*(z))}{z\ \det(\mathbf{M}^*(z))}, \tag{9.26}$$

where $\mathbf{M}^*(z) \triangleq \mathbf{I_J} - B(L(z))\ \hat{\mathbf{H}}^*(z)$. This technique for the determination of $\underline{\mathbf{P}^*}(x(z),z)$ can only be executed correctly if $\mathbf{M}^*(z)$ is not singular, i.e. if $1/B(L(z))$ is not an eigenvalue of $\hat{\mathbf{H}}^*(z)$. Note that every component of $\underline{\mathbf{P}^*}(x,z)$ is in fact a partial pgf and thus bounded for all arguments x and z on the closed unit disk (i.e. $|x|, |z| \leq 1$). Specifically for $x = z = 1$, we get $\underline{\mathbf{P}^*}(1,1)$ which should yield the stationary probability vector $\underline{\boldsymbol{\pi}}$. Substitution of $z = 1$ into (9.26) in order to extract some useful information concerning $\underline{\boldsymbol{\nu}}$ would however be needlessly complicated. A more desirable method is to evaluate the derivative of (9.25) for $z = 1$, from which we find that

$$\underline{\boldsymbol{\nu}} = \underline{\boldsymbol{\pi}}\ \boldsymbol{\eta} - \lambda\ \underline{\boldsymbol{\pi}} + \left.\frac{\mathrm{d}}{\mathrm{d}z}\underline{\mathbf{P}^*}(x(z),z)\right|_{z=1}(\mathbf{I_J} - \mathbf{H}). \tag{9.27}$$

Summing the components of $\underline{\boldsymbol{\nu}}$ by computing the product of (9.27) with $\underline{\mathbf{e_J}}$ then yields the first equation for determining $\underline{\boldsymbol{\nu}}$ as

$$\sum_{j=1}^{J}\nu_j = \underline{\boldsymbol{\nu}}\cdot\underline{\mathbf{e_J}} = \left(\sum_{j=1}^{J}\pi_j\eta_j\right) - \lambda. \tag{9.28}$$

For the remaining equations, we choose $z = z^*$ where $|z^*| \leq 1$, $z^* \neq 1$ and $\det(\mathbf{M}^*(z^*)) = 0$, such that the denominators of the partial pgfs in (9.26) become 0. Given the boundedness of pgfs on the closed unit disk, each of the corresponding numerators must be 0 as well, such that for each partial pgf de l'Hôpital's theorem can be applied. Expressing this condition for each pgf simultaneously, we get that

$$\underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{M}^*(z^*)) = \underline{\mathbf{0_J}}, \tag{9.29}$$

where $\underline{\mathbf{0_J}}$ is a row vector of order J with all elements equal to 0. Unfortunately, the homogeneous system of linear equations for the unknowns ν_j represented by (9.29) is linearly dependent and only yields one useful equation. In general however, the determinant of $\mathbf{M}^*(z)$ has $J-1$ zeroes $z^* \neq 1$ in the open unit disk, each zero giving rise to a distinct homogeneous system of equations similar to (9.29) and consequently to one additional equation in the unknowns ν_j, linearly independent from any other equation. This approach holds if all zeroes z^* are distinct. However, for a zero z^* of multiplicity $r > 1$, we can still obtain r unique systems of linear equations

$$\left.\frac{\mathrm{d}^i}{\mathrm{d}z^i}\underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{M}^*(z))\right|_{z=z^*} = \underline{\mathbf{0_J}}, \qquad i = 0, \ldots, r-1. \tag{9.30}$$

Together, (9.28) and the $J-1$ variants of (9.29) and/or (9.30) form a set of J independent linear equations in ν_j from which $\underline{\boldsymbol{\nu}}$ can be determined.

It may occur that there are less than $J-1$ zeroes z^* of $\det(\mathbf{M}^*(z))$ in the open unit disk, and that therefore the approach presented above fails. However, only a small modification to the approach is required to overcome this problem. From [49], it follows that $\det(z\ \mathbf{M}^*(z))$ always has exactly $J-1$ zeroes $z^\star$ in the open unit disk, if the condition

$$\left.\frac{\mathrm{d}}{\mathrm{d}z} z\ \det(\mathbf{M}^*(z))\right|_{z=1} = \left.\frac{\mathrm{d}}{\mathrm{d}z}\det(\mathbf{M}^*(z))\right|_{z=1} > 0, \tag{9.31}$$

holds. Note that this condition is in fact equivalent to the condition $\rho < 1$ for reaching steady-state, such that (9.31) always holds for stable systems. From $\det(z\ \mathbf{M}^*(z)) = z^J \det(\mathbf{M}^*(z))$ it is clear that any zero $z \neq 0$ of $\det(\mathbf{M}^*(z))$ is also a zero of $\det(z\ \mathbf{M}^*(z))$ and vice versa. If $\det(z\ \mathbf{M}^*(z))$ however has a zero in $z = 0$, this translates to a pole for $\det(\mathbf{M}^*(z))$ and less than $J-1$ zeroes z^* in the open unit disk. In order to take into account all $J-1$ zeroes of $\det(z\ \mathbf{M}^*(z))$, we therefore must modify (9.29) and (9.30) to

$$z^{\star J-1}\underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{M}^*(z^\star)) = \underline{\mathbf{0_J}}, \tag{9.32}$$

for zeroes $z^\star$ of $\det(z\ \mathbf{M}^*(z))$ of multiplicity 1 and

$$\left.\frac{\mathrm{d}^i}{\mathrm{d}z^i} z^{J-1}\underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{M}^*(z))\right|_{z=z^\star} = \underline{\mathbf{0_J}}, \qquad i = 0, \ldots, r-1, \tag{9.33}$$

for zeroes $z^\star$ of $\det(z\ \mathbf{M}^*(z))$ of multiplicity $r > 1$. All J components of $\underline{\boldsymbol{\nu}}$ can then be found from (9.28) and the $J-1$ variants of (9.32) and/or (9.33).

In practice, each of the ν_j corresponds to the probability for the system to be empty at the beginning of a steady-state slot during which the Markovian process governing the output line is in state j and the output line itself is accessible, as can be understood from the definition $\underline{\boldsymbol{\nu}} \triangleq \underline{\mathbf{P}^*}(0,0)\ \mathbf{H_1}$. The empty system probability vector $\underline{\mathbf{P}^*}(0,0)$ can therefore be found as $\underline{\mathbf{P}^*}(0,0) = \underline{\boldsymbol{\nu}}\ \mathbf{H_1}^{-1}$ and the total probability for the system to be empty at the beginning of a random steady-state slot is

$$p_0 = \sum_{j=1}^{J} P_j(0,0) = \underline{\mathbf{P}^*}(0,0) \cdot \underline{\mathbf{e_J}} = \underline{\boldsymbol{\nu}}\ \mathbf{H_1}^{-1}\ \underline{\mathbf{e_J}}. \tag{9.34}$$

Note that a necessary condition for the system to be empty at the beginning of a random slot $k+1$ (whether or not in the steady state) is that there must not have been any arrival during the preceding slot k (i.e. $a_k = 0$), independently of the state of the Markov process during either k or $k+1$. From this necessary condition, we can conclude that

$$\begin{aligned} P_{j,k+1}(x,0) &= \mathrm{E}[x^{a_k}\ \{s_k\!=j, u_{k+1}\!=0\}] = \mathrm{E}[x^{a_k}\ \{s_k\!=j, a_k\!=0, u_{k+1}\!=0\}] \\ &= P_{j,k+1}(0,0), \qquad \forall x. \end{aligned} \tag{9.35}$$

Using vector notations, we therefore have that

$$\underline{\mathbf{P}^*_\mathbf{k}}(x,0) = \underline{\mathbf{P}^*_\mathbf{k}}(0,0), \qquad \text{and} \qquad \underline{\mathbf{P}^*}(x,0) = \underline{\mathbf{P}^*}(0,0), \qquad \forall x. \tag{9.36}$$

From the above analysis of the system state, the pgf $U(z)$ of the system content u at the beginning of an arbitrary steady-state slot can be found directly as

$$\begin{aligned} U(z) \triangleq \mathrm{E}[z^u] = P(1,1,z) &= \sum_{j=1}^{J} P_j(1,z) = \underline{\mathbf{P}^*}(1,z) \cdot \underline{\mathbf{e_J}} \\ &= B(z) \left[\frac{z-1}{z} \underline{\boldsymbol{\nu}} + \underline{\mathbf{P}^*}(C(z),z)\ \hat{\mathbf{H}}^*(z) \right]. \end{aligned} \tag{9.37}$$

In order to obtain the mean system content however, it is advantageous not to start from $\underline{\mathbf{P}^*}(x,z)$, but rather from $\underline{\mathbf{P}^*}(x(z),z)$. Bearing in mind the definition of $x(z)$ in (9.24), we can calculate the first derivative of $\underline{\mathbf{P}^*}(x(z),z)$ with respect to z for $z=1$ in terms of the partial derivatives of $\underline{\mathbf{P}^*}(x,z)$ as

$$\left. \frac{\mathrm{d}}{\mathrm{d}z} \underline{\mathbf{P}^*}(x(z),z) \right|_{z=1} = (L'(1)-1)\ \frac{\partial}{\partial x} \underline{\mathbf{P}^*}(1,1) + \frac{\partial}{\partial z} \underline{\mathbf{P}^*}(1,1). \tag{9.38}$$

Summing the components of this vector then yields

$$\left. \frac{\mathrm{d}}{\mathrm{d}z} \underline{\mathbf{P}^*}(x(z),z) \right|_{z=1} \cdot \underline{\mathbf{e_J}} = \lambda\,(L'(1)-1) + U'(1), \tag{9.39}$$

from which the mean system content can be found as

$$\mathrm{E}[u] \triangleq U'(1) = \frac{\mathrm{d}}{\mathrm{d}z}\underline{\mathbf{P}^*}(x(z),z)\bigg|_{z=1} \cdot \underline{\mathbf{e_J}} - \lambda\,(L'(1)-1)\,. \tag{9.40}$$

The unknown vector on the right-hand side of (9.40) can be determined from (9.26) as

$$\frac{\mathrm{d}}{\mathrm{d}z}\underline{\mathbf{P}^*}(x(z),z)\bigg|_{z=1} = \frac{1}{M'(1)}\left[\left(\lambda - 1 - \frac{M''(1)}{2M'(1)}\right)\underline{\boldsymbol{\nu}}\ \mathrm{adj}(\mathbf{I_J}-\mathbf{H}) + \underline{\boldsymbol{\nu}}\ \frac{\mathrm{d}}{\mathrm{d}z}\mathrm{adj}(\mathbf{M}^*(z))\bigg|_{z=1}\right], \tag{9.41}$$

where we introduced the shorthand $M(z) \triangleq \det(\mathbf{M}^*(z))$.

9.5 Packet delay analysis

In order to determine the delay experienced by a random steady-state packet $\mathcal{P}$, we need not only take into account the system state at the beginning of the packet's arrival slot $\mathcal{S}$ and the number of packets that arrive simultaneously with $\mathcal{P}$, but also the output line state in all slots from $\mathcal{S}$ up to $\mathcal{P}$'s departure slot. Practically, the delay $d_{\mathcal{P}}$ of $\mathcal{P}$ is the total number of slots needed to serve and transmit all $v_{\mathcal{P}}$ packets in the queue just after slot $\mathcal{S}$, excluding any packets that have arrived during the same slot as $\mathcal{P}$ that will be transmitted later than $\mathcal{P}$. This random variable can be determined as

$$v_{\mathcal{P}} = (u_{\mathcal{S}} - r_{\mathcal{S}})^+ + \chi_{\mathcal{P}} + 1, \tag{9.42}$$

where $\chi_{\mathcal{P}}$ is the number of packets that have arrived during $\mathcal{S}$ but are to be transmitted before $\mathcal{P}$, as in Section 5.4. Given that the position of $\mathcal{P}$ is uniformly distributed over the $a_{\mathcal{S}}$ slot $\mathcal{S}$ arrivals, the pmf of $\chi_{\mathcal{P}}$ conditioned on $a_{\mathcal{S}}$ is given by

$$\mathrm{Prob}[\chi_{\mathcal{P}} = n |\, a_{\mathcal{S}} = i] = \frac{1}{i}, \qquad n \in \{0,\ldots,i-1\}\,. \tag{9.43}$$

This allows us to find the partial pgfs $V_{j,\mathcal{P}}(z)$ of $v_{\mathcal{P}}$ as

$$\begin{aligned} V_{j,\mathcal{P}}(z) &\triangleq \mathrm{E}[z^{v_{\mathcal{P}}}\,\{s_{\mathcal{S}}=j\}] = \mathrm{E}\Big[z^{(u_{\mathcal{S}}-r_{\mathcal{S}})^+ + \chi_{\mathcal{P}}+1}\,\{s_{\mathcal{S}}=j\}\Big] \\ &= z\sum_{i=1}^{\infty}\sum_{n=0}^{i-1} z^n\,\mathrm{E}\Big[z^{(u_{\mathcal{S}}-r_{\mathcal{S}})^+}\,\{s_{\mathcal{S}}=j, a_{\mathcal{S}}=i, \chi_{\mathcal{P}}=n\}\Big] \\ &= z\sum_{i=1}^{\infty}\frac{1}{i}\,\mathrm{E}\Big[z^{(u_{\mathcal{S}}-r_{\mathcal{S}})^+}\,\{s_{\mathcal{S}}=j, a_{\mathcal{S}}=i\}\Big]\sum_{n=0}^{i-1} z^n \\ &= \frac{z}{1-z}\sum_{i=1}^{\infty}\frac{1-z^i}{i}\,\mathrm{E}\Big[z^{(u_{\mathcal{S}}-r_{\mathcal{S}})^+}\,\{s_{\mathcal{S}}=j, a_{\mathcal{S}}=i\}\Big]\,. \end{aligned} \tag{9.44}$$

Similar as in previous analyses, we note that slot $\mathcal{S}$ is not a random slot, given that it features at least one arrival. Based on arguments from renewal theory (see e.g. [90]), we can relate the partial expectation on the right-hand side of (9.44) to the system state variables for a random steady-state slot k as

$$\begin{aligned}\mathrm{E}\Big[z^{(u_\mathcal{S}-r_\mathcal{S})^+}\{s_\mathcal{S}=j,a_\mathcal{S}=i\}\Big] &= \mathrm{E}\big[z^{u_{\mathcal{S}+1}-i}\{s_\mathcal{S}=j,a_\mathcal{S}=i\}\big]\\ &= z^{-i}\,\mathrm{E}[z^{u_{\mathcal{S}+1}}\{s_\mathcal{S}=j,a_\mathcal{S}=i\}]\\ &= \frac{iz^{-i}}{\lambda}\,\mathrm{E}[z^{u_{k+1}}\{s_k=j,a_k=i\}]\,. \end{aligned} \tag{9.45}$$

Substitution in (9.44) then results in

$$\begin{aligned}V_{j,\mathcal{P}}(z) &= \frac{1}{\lambda}\frac{z}{1-z}\sum_{i=1}^{\infty}\left(z^{-i}-1\right)\mathrm{E}[z^{u_{k+1}}\{s_k=j,a_k=i\}]\\ &= \frac{1}{\lambda}\frac{z}{1-z}\left(P_j(z^{-1},z)-P_j(1,z)\right), \end{aligned} \tag{9.46}$$

where $P_j(x,z)$ is the jth component of $\underline{\mathbf{P}^*}(x,z)$. Collecting all partial pgfs $V_{j,\mathcal{P}}(z)$ for different values of j, we get the partial vector generating function $\underline{\mathbf{V}^*_\mathcal{P}}(z)$ as

$$\underline{\mathbf{V}^*_\mathcal{P}}(z) = \frac{1}{\lambda}\frac{z}{1-z}\left(\underline{\mathbf{P}^*}(z^{-1},z)-\underline{\mathbf{P}^*}(1,z)\right). \tag{9.47}$$

This vector generating function $\underline{\mathbf{V}^*_\mathcal{P}}(z)$ returns a row vector of which the jth component is in fact the partial pgf $V_{j,\mathcal{P}}(z)$ of the number of packets $v_\mathcal{P}$ in the system just after slot $\mathcal{S}$, excluding the packets to be transmitted later than $\mathcal{P}$, assuming the output line state during slot $\mathcal{S}$ is j.

Next we need to determine the total number of slots needed by the system to complete the transmission of all these $v_\mathcal{P}$ packets. Note that as described before, even though the transmission times are equal to 1 slot per packet, the *effective transmission time* of a packet can be larger due to the output line interruptions. Let $s_{\mathrm{eff},k}$ be the effective transmission time of a single packet starting at slot k, we can then find the conditional joint probability

$$\mathrm{Prob}[\,s_{\mathrm{eff},k}=i,s_{k+i}=j'\,|\,s_k=j] = \left[\mathbf{H_0}^{i-1}\mathbf{H_1}\right]_{jj'}, \tag{9.48}$$

which corresponds to the individual components of the matrix generating function

$$\mathbf{S}^*_{\mathrm{eff}}(z) = z(\mathbf{I_J}-z\mathbf{H_0})^{-1}\mathbf{H_1}. \tag{9.49}$$

Assuming that during slot $\mathcal{S}$ the output line is in state j, the delay $d_\mathcal{P}$ of $\mathcal{P}$ has partial pgf $\underline{\mathbf{1_j}}(\mathbf{S}^*_{\mathrm{eff}}(z))^{v_\mathcal{P}}\underline{\mathbf{e_J}}$, where $\underline{\mathbf{1_j}}$ is a row vector with all zeroes except for the jth entry which is equal to 1. The packet delay pgf $D_{\mathcal{P}}(z)$ can then be found as

$$\begin{aligned}D_{\mathcal{P}}(z) &\triangleq \mathrm{E}\big[z^{d_{\mathcal{P}}}\big] = \sum_{j=1}^{J} \mathrm{E}\big[z^{d_{\mathcal{P}}}\, \{s_{\mathcal{S}} = j\}\big] = \sum_{j=1}^{J}\sum_{n=1}^{\infty} \mathrm{E}\big[z^{d_{\mathcal{P}}} \{s_{\mathcal{S}} = j, v_{\mathcal{P}} = n\}\big] \\ &= \sum_{j=1}^{J}\sum_{n=1}^{\infty} \underline{\mathbf{1_j}}\ (\mathbf{S}^*_{\mathrm{eff}}(z))^n\ \underline{\mathbf{e_J}}\ \mathrm{Prob}[s_{\mathcal{S}} = j, v_{\mathcal{P}} = n] \\ &= \sum_{j=1}^{J} \underline{\mathbf{1_j}}\ V_{j,\mathcal{P}}(\mathbf{S}^*_{\mathrm{eff}}(z))\ \underline{\mathbf{e_J}}. \end{aligned} \tag{9.50}$$

Note that the matrix functions $V_{j,\mathcal{P}}(\mathbf{S}^*_{\mathrm{eff}}(z))$ are well-defined if and only if the eigenvalues of the matrix $\mathbf{S}^*_{\mathrm{eff}}(z)$ are in the domain of the functions $V_{j,\mathcal{P}}(z)$. In that case, $D_{\mathcal{P}}(z)$ can be calculated using the spectral decomposition of $\mathbf{S}^*_{\mathrm{eff}}(z)$ as

$$\begin{aligned}D_{\mathcal{P}}(z) &= \sum_{j=1}^{J} \underline{\mathbf{1_j}} \left(\sum_{i=1}^{\kappa(z)} V_{j,\mathcal{P}}(\lambda_i(z)) \mathbf{S}^*_{\mathrm{eff},i}(z) \right) \underline{\mathbf{e_J}} \\ &= \sum_{j=1}^{J}\sum_{i=1}^{\kappa(z)} V_{j,\mathcal{P}}(\lambda_i(z)) \sum_{j'=1}^{J} \left[\mathbf{S}^*_{\mathrm{eff},i}(z)\right]_{jj'}, \end{aligned} \tag{9.51}$$

where the $\lambda_i(z)$ ($i \in \{1, \ldots, \kappa(z)\}$) are the distinct eigenvalues of $\mathbf{S}^*_{\mathrm{eff}}(z)$ for a particular value of z (see e.g. [89]). In case $\mathbf{S}^*_{\mathrm{eff}}(z)$ represents a diagonalizable matrix, the corresponding spectral projectors $\mathbf{S}^*_{\mathrm{eff},i}(z)$ can be obtained using Lagrange interpolation as

$$\mathbf{S}^*_{\mathrm{eff},i}(z) = \frac{\prod_{i'=1,i\neq i'}^{\kappa(z)} (\mathbf{S}^*_{\mathrm{eff}}(z) - \lambda_{i'}(z)\mathbf{I_J})}{\prod_{i'=1,i\neq i'}^{\kappa(z)} (\lambda_i(z) - \lambda_{i'}(z))}. \tag{9.52}$$

If $\mathbf{S}^*_{\mathrm{eff}}(z)$ is not diagonalizable, both (9.51) and (9.52) require a more technical treatment [89]. Essential to our analysis however, is the requirement that the functions $\lambda_i(z)$ are analytic in at least a neighborhood of $z = 1$ because we need to evaluate their derivatives in this point. In general, it is known that the eigenvalue functions may not be analytic or even continuous in the entire unit disk [48] but they are analytic in (a neighborhood of) points where all eigenvalues are distinct. We therefore assume as a sufficient condition that $\mathbf{S}^*_{\mathrm{eff}}(1)$ has $\kappa(1) = J$ distinct eigenvalues, so that ultimately, the mean packet delay $\mathrm{E}[d_{\mathcal{P}}]$ can then be found as

$$\begin{aligned}\mathrm{E}[d_{\mathcal{P}}] = D'_{\mathcal{P}}(1) = \sum_{j=1}^{J}\sum_{i=1}^{J}\Bigg(&\lambda'_i(1)V'_{j,\mathcal{P}}(\lambda_i(1))\sum_{j'=1}^{J}\left[\mathbf{S}^*_{\mathrm{eff},i}(1)\right]_{jj'}\\ &+V_{j,\mathcal{P}}(\lambda_i(1))\sum_{j'=1}^{J}\left[\mathbf{S}^*_{\mathrm{eff},i}{}'(1)\right]_{jj'}\Bigg). \qquad (9.53)\end{aligned}$$

The analysis method for the packet delay established in this section constitutes a basic step to study the train delay, as we will explain next.

9.6 Train delay analysis

Similar to the session delay in the previous chapter, the delay $d_{\mathcal{M}}$ of a random steady-state train $\mathcal{M}$ is defined as the integer number of slots between the end of the arrival slot $\mathcal{S}$ of the first packet $\mathcal{P}_0$ of the train, until the end of the slot in which the final packet $\bar{\mathcal{P}} = \mathcal{P}_{\ell(\mathcal{M})-1}$ of $\mathcal{M}$ departs from the system. Note that this in fact corresponds to the delay of the train's final packet, augmented with the number of slots between the arrival of the train's first and last packet, i.e.

$$d_{\mathcal{M}} = \ell(\mathcal{M}) - 1 + d_{\bar{\mathcal{P}}}. \qquad (9.54)$$

Similar to the delay $d_{\mathcal{P}}$ of a random packet, the delay $d_{\bar{\mathcal{P}}}$ of the train's final packet $\bar{\mathcal{P}}$ can be determined as the total effective transmission time of all $v_{\bar{\mathcal{P}}}$ packets in the system just after $\bar{\mathcal{P}}$'s arrival slot $\bar{\mathcal{S}}$, excluding the packets that arrived during $\bar{\mathcal{S}}$ but have to be transmitted later than $\bar{\mathcal{P}}$. This number $v_{\bar{\mathcal{P}}}$ can be determined from the system state distribution at the beginning of slot $\bar{\mathcal{S}}$ and the arrival process during $\bar{\mathcal{S}}$ as

$$v_{\bar{\mathcal{P}}} = (u_{\bar{\mathcal{S}}} - r_{\bar{\mathcal{S}}})^+ + \chi_{\bar{\mathcal{P}}} + 1, \qquad (9.55)$$

with $\chi_{\bar{\mathcal{P}}}$ being the number of packets that have arrived during $\bar{\mathcal{S}}$ but are to be transmitted before $\bar{\mathcal{P}}$. The partial pgfs $V_{j,\bar{\mathcal{P}}}$ of $v_{\bar{\mathcal{P}}}$ can then be found similarly as before, yielding

$$\begin{aligned}
V_{j,\bar{\mathcal{P}}} &\triangleq \mathrm{E}[z^{v_{\bar{\mathcal{P}}}}\{s_{\bar{\mathcal{S}}}=j\}] = \mathrm{E}\Big[z^{(u_{\bar{\mathcal{S}}}-r_{\bar{\mathcal{S}}})^{+}+\chi_{\bar{\mathcal{P}}}+1}\{s_{\bar{\mathcal{S}}}=j\}\Big] \\
&= \frac{z}{1-z}\sum_{i=1}^{\infty}\frac{1-z^{i}}{i}\,\mathrm{E}\Big[z^{(u_{\bar{\mathcal{S}}}-r_{\bar{\mathcal{S}}})^{+}}\{s_{\bar{\mathcal{S}}}=j,a_{\bar{\mathcal{S}}}=i\}\Big] \\
&= \frac{z}{1-z}\sum_{i=1}^{\infty}\frac{1-z^{i}}{i}\,\mathrm{E}\big[z^{u_{\bar{\mathcal{S}}+1}-i}\{s_{\bar{\mathcal{S}}}=j,a_{\bar{\mathcal{S}}}=i\}\big] \\
&= \frac{z}{1-z}\sum_{i=1}^{\infty}\frac{z^{-i}-1}{i}\,\mathrm{E}[z^{u_{\bar{\mathcal{S}}+1}}\{s_{\bar{\mathcal{S}}}=j,a_{\bar{\mathcal{S}}}=i\}] \\
&= \frac{z}{1-z}\int_{1}^{1/z}\mathrm{E}\big[\alpha^{a_{\bar{\mathcal{S}}}-1}z^{u_{\bar{\mathcal{S}}+1}}\{s_{\bar{\mathcal{S}}}=j\}\big]\,\mathrm{d}\alpha \\
&= \frac{z}{1-z}\int_{1}^{1/z}\frac{1}{\alpha}P_{j,\bar{\mathcal{S}}+1}(\alpha,z)\mathrm{d}\alpha. \qquad (9.56)
\end{aligned}$$

In order to obtain the delay $d_{\bar{\mathcal{P}}}$ of $\mathcal{M}$'s final packet $\bar{\mathcal{P}}$, we first need to determine the system state distribution just after slot $\bar{\mathcal{S}}$, which can be found from the system state at the beginning of $\bar{\mathcal{S}}$ and the number of arrivals during slot $\bar{\mathcal{S}}$. As before, the number of arrivals $a_{\bar{\mathcal{S}}}$ during the train's final slot $\bar{\mathcal{S}}$ has a distribution that is not identical to the distribution of the number of arrivals during an arbitrary steady-state slot. Due to the correlated nature of the packet arrival process, the system state distribution at the beginning of $\bar{\mathcal{S}}$ in general will also be different from the steady-state system state distribution at the beginning of a random slot.

Given the fact that individual trains start independently, the first packet $\mathcal{P}_0$ of a steady-state train does however perceive the system in an arbitrary state. In other words, the distribution of the system state at the beginning of the train's first slot $\mathcal{S}$ is identical to the system state distribution at the beginning of a random steady-state slot as studied in Section 9.4. Conversely, the distribution of the number of new trains $b_{\mathcal{S}}$ started during slot $\mathcal{S}$ is not identical to the distribution of the number of new trains started in an arbitrary slot, as we know that at least one new train must start during $\mathcal{S}$. It can be seen that the number $b_{\mathcal{S}}$ of new trains in slot $\mathcal{S}$ is proportional to $b_{\mathcal{S}}$, such that we find the pmf

$$\mathrm{Prob}[b_{\mathcal{S}}=\beta] = \frac{\beta}{B'(1)}\mathrm{Prob}[b_k=\beta]\,, \qquad \beta\geq 1, \qquad (9.57)$$

similar as in (5.25), (8.19) and (8.37). From (9.57), the pgf $B_{\mathcal{S}}(z)$ of $b_{\mathcal{S}}$ can be expressed in terms of its arbitrary steady-slot slot counterpart $B(z)$ as (see e.g. [90])

$$B_{\mathcal{S}}(z) \triangleq \mathrm{E}\big[z^{b_{\mathcal{S}}}\big] = \frac{z}{B'(1)}\sum_{\beta=1}^{\infty}\mathrm{Prob}[b_k]\,\frac{\mathrm{d}}{\mathrm{d}z}z^{\beta} = z\frac{B'(z)}{B'(1)}. \qquad (9.58)$$

These observations allow us to find the partial system state pgfs $P_{j,\mathcal{S}+1}(x,z)$ at the beginning of slot $\mathcal{S}+1$ as

$$\begin{aligned}
P_{j,\mathcal{S}+1}(x,z) &\triangleq \mathrm{E}[x^{a_\mathcal{S}} z^{u_{\mathcal{S}+1}} \{s_\mathcal{S}=j\}] = \mathrm{E}\Big[(xz)^{a_\mathcal{S}} z^{(u_\mathcal{S}-r_\mathcal{S})^+} \{s_\mathcal{S}=j\}\Big] \\
&= xz\frac{B'(xz)}{B'(1)}\,\mathrm{E}\Big[(C(xz))^{a_{\mathcal{S}-1}} z^{(u_\mathcal{S}-r_\mathcal{S})^+} \{s_\mathcal{S}=j\}\Big] \\
&= \frac{xz}{B(xz)}\frac{B'(xz)}{B'(1)}P_j(x,z) \\
&= xz\frac{B'(xz)}{B'(1)}\left[\frac{z-1}{z}\underline{\boldsymbol{\nu}} + \underline{\mathbf{P}^*}(C(xz),z)\ \hat{\mathbf{H}}^*(z)\right]_j,
\end{aligned} \tag{9.59}$$

or combined as a vector generating function

$$\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{1}}}(x,z) = xz\frac{B'(xz)}{B'(1)}\left[\frac{z-1}{z}\underline{\boldsymbol{\nu}} + \underline{\mathbf{P}^*}(C(xz),z)\ \hat{\mathbf{H}}^*(z)\right]. \tag{9.60}$$

For a subsequent packet $\mathcal{P}_k$ ($k \in \{1,\ldots,\ell(\mathcal{M})-1\}$) of $\mathcal{M}$ arriving during slot $\mathcal{S}+k$, we must take into account the correlated nature of the packet arrival process. In practice this means that we know that $\mathcal{P}_k$ is a continuation of one of the $a_{\mathcal{S}+k-1}$ trains active in slot $\mathcal{S}+k-1$ and that the system cannot be empty at the beginning of slot $\mathcal{S}+k$. From these considerations, we can find the partial system state pgfs $P_{j,\mathcal{S}+k+1}(x,z)$ at the beginning of slot $\mathcal{S}+k+1$ as

$$\begin{aligned}
P_{j,\mathcal{S}+k+1}(x,z) &\triangleq \mathrm{E}[x^{a_{\mathcal{S}+k}} z^{u_{\mathcal{S}+k+1}} \{s_{\mathcal{S}+k}=j\}] \\
&= \mathrm{E}\big[(xz)^{a_{\mathcal{S}+k}} z^{u_{\mathcal{S}+k}-r_{\mathcal{S}+k}} \{s_{\mathcal{S}+k}=j\}\big] \\
&= xz\frac{B(xz)}{C(xz)}\hat{H}_j(z)\,\mathrm{E}[(C(xz))^{a_{\mathcal{S}+k-1}} z^{u_{\mathcal{S}+k}} \{s_{\mathcal{S}+k}=j\}] \\
&= xz\frac{B(xz)}{C(xz)}\left[\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{k}}}(C(xz),z)\ \hat{\mathbf{H}}^*(z)\right]_j,
\end{aligned} \tag{9.61}$$

which yields the vector generating function

$$\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{k}+\mathbf{1}}}(x,z) = xz\frac{B(xz)}{C(xz)}\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{k}}}(C(xz),z)\ \hat{\mathbf{H}}^*(z). \tag{9.62}$$

By iteration of (9.62) and substitution of (9.60) we can then relate the pgf $\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{k}+\mathbf{1}}}(x,z)$ to its steady-state slot counterpart $\underline{\mathbf{P}^*}(x,z)$ as

$$\begin{aligned}
\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{k}+\mathbf{1}}}(x,z) &= \frac{xz^k}{G_k(x,z)}\left(\prod_{i=0}^{k-1} B(zG_i(x,z))\right)\underline{\mathbf{P}^*_{\mathcal{S}+\mathbf{1}}}(G_k(x,z),z)\ \hat{\mathbf{H}}^*(z)^k \\
&= xz^{k+1}\frac{B'(zG_k(x,z))}{B'(1)}\left(\prod_{i=0}^{k-1} B(zG_i(x,z))\right) \\
&\qquad \left[\frac{z-1}{z}\underline{\boldsymbol{\nu}} + \underline{\mathbf{P}^*}(G_{k+1}(xz),z)\ \hat{\mathbf{H}}^*(z)\right]\ \hat{\mathbf{H}}^*(z)^k,
\end{aligned} \tag{9.63}$$

where $G_k(x,z)$ is defined iteratively as

$$G_0(x,z) = x, \tag{9.64}$$

$$G_k(x,z) = C(zG_{k-1}(x,z)), \qquad k > 0. \tag{9.65}$$

Note that (9.63) in fact yields (9.60) when substituting $k = 0$.

Substitution of $k = \ell(\mathcal{M})-1$ into (9.63) then gives the vector generating function $\underline{\mathbf{P}^*_{\mathcal{S}+\ell(\mathcal{M})}}(x,z) = \underline{\mathbf{P}^*_{\bar{\mathcal{S}}+\mathbf{1}}}(x,z)$ of the system state just after the arrival slot $\bar{\mathcal{S}} = \mathcal{S}+\ell(\mathcal{M})-1$ of $\mathcal{M}$'s final packet. The individual components of $\underline{\mathbf{P}^*_{\bar{\mathcal{S}}+\mathbf{1}}}(x,z)$ can then be substituted into (9.56) to produce a closed-form expression for the partial pgfs $V_{j,\bar{\mathcal{P}}}$ of $v_{\bar{\mathcal{P}}}$. From $V_{j,\bar{\mathcal{P}}}$, we can then obtain the pgf $D_{\bar{\mathcal{P}}}(z)$ of the train's final packet $\bar{\mathcal{P}}$ as

$$D_{\bar{\mathcal{P}}}(z) = \sum_{j=1}^{J} \underline{\mathbf{1_j}}\; V_{j,\bar{\mathcal{P}}}(\mathbf{S}^*_{\text{eff}}(z))\; \underline{\mathbf{e_J}}. \tag{9.66}$$

Finally, the pgf $D_{\mathcal{M}}(z)$ of the delay of a random train can be determined as

$$D_{\mathcal{M}}(z) = \sum_{i=1}^{\infty} \mathrm{E}\left[z^{i-1+d_{\bar{\mathcal{P}}}}\,\{\ell(\mathcal{M}) = i\}\right]. \tag{9.67}$$

The mean train delay $\mathrm{E}[d_{\mathcal{M}}]$ can then be found by substituting $z = 1$ into the first derivative of (9.67).

9.7 Special cases

9.7.1 Blocking states

The method for finding the row vector $\underline{\boldsymbol{\nu}}$ was based on the assumption that $\det(\mathbf{M}^*(z))$ has $J-1$ roots $z^* \neq 1$ on the unit disk, which is a reasonable assumption in the majority of realistic cases. One realistic scenario where this assumption fails, is when for one or more states the output line is always interrupted. We refer to such states, if any, as *blocking states*.

For example, let state j_b be such a blocking state, i.e. $\eta_{j_b} = 0$ and therefore the j_bth column of $\mathbf{H_1}$ will contain only 0s. Assuming all other $J-1$ states are non-blocking states, the determinant of $\mathbf{M}^*(z)$ will in general have $J-2$ roots $z^* \neq 1$ in the open unit disk. When combining (9.28) with the corresponding $J-2$ variants of (9.29) and/or (9.30), we are still one equation short in order to obtain $\underline{\boldsymbol{\nu}}$. However, an additional equation can be found directly from the definition $\underline{\boldsymbol{\nu}} \triangleq \underline{\mathbf{P}^*}(0,0)\ \mathbf{H_1}$. Considering only the j_bth component of $\underline{\boldsymbol{\nu}}$, this yields

$$\nu_{j_b} = \sum_{j=1}^{J} P_j(0,0)[\mathbf{H_1}]_{jj_b} = 0. \tag{9.68}$$

Together with the $J-1$ equations supplied by the default method, we now can calculate $\underline{\boldsymbol{\nu}}$.

In case there are $b>1$ blocking states, each of these blocking states will reduce the number of roots with 1, leaving $J-b-1$ roots $z^* \neq 1$ to be found. In turn, we can apply (9.68) for each of these blocking states.

9.7.2 $J=1$: Geometric output line interruptions

If the Markovian process governing the output line interruptions has only one single state, the system boils down to a system with Bernoulli output line interruptions, as discussed in Chapter 8. During an arbitrary slot, the output line is then accessible with a fixed probability η and interrupted with probability $1-\eta$.

In this case, the system state pgf breaks down to

$$P(x,z) = B(xz)\left[\frac{z-1}{z}\eta p_0 + \hat{H}(z)P(C(xz),z)\right], \tag{9.69}$$

where $\hat{H}(z) \triangleq 1-\eta+\eta/z$ and with the empty system probability $p_0 = P(0,0) = 1-\lambda/\eta$. For $x = x(z) = L(z)/z$, we get the recursion-free expression

$$P(x(z),z) = \frac{B(L(z))\,(z-1)}{z\left(1-\hat{H}(z)B(L(z))\right)}\eta p_0, \tag{9.70}$$

which after derivation to z and evaluation for $z=1$ allows us to find the mean system content $\mathrm{E}[u]$ as

$$\mathrm{E}[u] = \frac{1}{(\eta-\lambda)}\left[\lambda\,(1-\lambda) + \frac{\gamma B'(1)}{(1-\gamma)^2}\,(1-\eta+\lambda) + \frac{B''(1)}{2(1-\gamma)^2}\right]. \tag{9.71}$$

For the analysis of the packet delay, we notice that the effective transmission times have a shifted geometric distribution, with pgf

$$S_{\text{eff}}(z) = \frac{\eta z}{1-(1-\eta)\,z}. \tag{9.72}$$

The pgf $V_{\mathcal{P}}(z)$ of the number of packets in the system just after the arrival slot of a randomly selected packet $\mathcal{P}$, excluding any packets to be transmitted after $\mathcal{P}$ can be found from (9.46) as

$$\begin{aligned} V_{\mathcal{P}}(z) &= \frac{1}{\lambda}\frac{z}{z-1}\left(P(z^{-1},z) - U(z)\right) \\ &= \frac{1}{\lambda}\frac{z}{z-1}\left(\frac{z-1}{z}\eta p_0 + \hat{H}(z)U(z) - U(z)\right) \\ &= \frac{\eta}{\lambda}\left(U(z)-p_0\right), \end{aligned} \tag{9.73}$$

where $U(z) = P(1, z)$ is the pgf of the system content. The pgf $D_{\mathcal{P}}(z)$ of the packet delay becomes

$$D_{\mathcal{P}}(z) = \frac{\eta}{\lambda}\left(B(S_{\text{eff}}(z))\left[\frac{z-1}{z}p_0 + \frac{1}{z}P(C(S_{\text{eff}}(z)), S_{\text{eff}}(z))\right] - p_0\right), \tag{9.74}$$

due to the specific nature of $S_{\text{eff}}(z)$ for the $J = 1$ case, as given in (9.72). More specifically, we have that

$$\frac{S_{\text{eff}}(z) - 1}{S_{\text{eff}}(z)} = \frac{z-1}{\eta z}, \qquad \text{and} \qquad \hat{H}(S_{\text{eff}}(z)) = \frac{1}{z}. \tag{9.75}$$

Similarly, the pgf of the train delay can be found from (9.67), where

$$D_{\bar{\mathcal{P}}}(z) = V_{\bar{\mathcal{P}}}(S_{\text{eff}}(z)), \tag{9.76}$$

$$V_{\bar{\mathcal{P}}}(z) = \frac{z}{1-z}\int_1^{1/z} \frac{1}{\alpha} P_{\bar{\mathcal{S}}+1}(\alpha, z)\mathrm{d}\alpha, \tag{9.77}$$

and

$$P_{\bar{\mathcal{S}}+1}(x, z) = xz^{\ell(\mathcal{M})}\frac{B'(zG_{\ell(\mathcal{M})-1}(x,z))}{B'(1)}\left(\prod_{i=0}^{\ell(\mathcal{M})-2} B(zG_i(x,z))\right)$$
$$\left[\frac{z-1}{z}\eta p_0 + P(G_{\ell(\mathcal{M})}(x,z), z)\ H(z^{-1})\right]\ H(z^{-1})^{\ell(\mathcal{M})-1}. \tag{9.78}$$

9.7.3 $J = 2$: Two-state Markovian process

For the case where the Markovian process governing the output line interruptions has $J = 2$ states, most results can be determined explicitly as tractable expressions. We will present these explicit formulations here, for the analysis up to the mean system content. For the delays, even this case becomes quite tedious, owing to the spectral decomposition of the matrix generating function of the effective transmission times.

For $J = 2$, only two parameters σ_1 and σ_2 are needed to fully describe the state transitions in the Markovian process with the transition probability matrix

$$\mathbf{H} = \begin{bmatrix} \sigma_1 & \sigma_{2|1} \\ \sigma_{1|2} & \sigma_2 \end{bmatrix} = \begin{bmatrix} \sigma_1 & 1-\sigma_1 \\ 1-\sigma_2 & \sigma_2 \end{bmatrix}. \tag{9.79}$$

This matrix has two eigenvalues, namely $\lambda_1 = 1$ and $\lambda_2 = \sigma_1 + \sigma_2 - 1 = \phi(1)$, with associated left eigenvectors

$$\underline{\boldsymbol{\pi}} = \frac{1}{2-\sigma_1-\sigma_2}\begin{bmatrix} 1-\sigma_2 \\ 1-\sigma_1 \end{bmatrix}^T \qquad \text{and} \qquad \begin{bmatrix} 1 \\ -1 \end{bmatrix}^T, \tag{9.80}$$

respectively.

Note that $\phi(1)$ is in fact the steady-state lag-1 correlation coefficient between the output line state in a random slot k and the output line state in two subsequent slots, defined in (9.8) as

$$\phi(1) \triangleq \lim_{k\to\infty} \rho_{s_k,s_{k-1}} = \lim_{k\to\infty} \frac{\mathrm{E}[s_k s_{k-1}] - \mathrm{E}[s_k]\,\mathrm{E}[s_{k-1}]}{\sqrt{\mathrm{Var}[s_k]\,\mathrm{Var}[s_{k-1}]}}. \tag{9.81}$$

If $\phi(1) \approx -1$ this can be interpreted as a high probability that the Markovian process does not remain in the same state in two consecutive slots. When $\phi(1) \approx 0$, the Markovian process moves rather arbitrarily through its state space. If $\phi(1) \approx 1$, one can expect the Markovian process to remain in a state for numerous slots before making a transition to another state.

Knowing the stationary probability vector $\underline{\boldsymbol{\pi}}$, we can calculate the system load ρ as

$$\rho = \frac{\lambda}{\underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_2}}} = \frac{\lambda\,(2-\sigma_1-\sigma_2)}{\eta_1\,(1-\sigma_2)+\eta_2\,(1-\sigma_1)}. \tag{9.82}$$

In order to calculate $\underline{\boldsymbol{\nu}}$, we find the first equation from (9.28) as

$$\nu_1+\nu_2 = \eta_1\pi_1+\eta_2\pi_2-\lambda. \tag{9.83}$$

Next, we determine the matrix function $\mathbf{M}^*(z) \triangleq \mathbf{I_J} - B(L(z))\ \hat{\mathbf{H}}^*(z)$ as

$$\mathbf{M}^*(z) = \begin{bmatrix} 1-\sigma_1 B(L(z))\hat{H}_1(z) & -(1-\sigma_1)\,B(L(z))\hat{H}_2(z) \\ -(1-\sigma_2)\,B(L(z))\hat{H}_1(z) & 1-\sigma_2 B(L(z))\hat{H}_2(z) \end{bmatrix}, \tag{9.84}$$

such that z^* can be found as the unique zero of $\det(\mathbf{M}^*(z))$ on the unit disk different from 1, i.e.

$$\phi(1)B(L(z^*))^2\hat{H}_1(z^*)\hat{H}_2(z^*) - B(L(z^*))\left(\sigma_1\hat{H}_1(z^*)+\sigma_2\hat{H}_2(z^*)\right)+1=0. \tag{9.85}$$

Application of (9.29) then yields a system of equivalent equations given by

$$\begin{cases} \nu_1 - \left(\nu_1\sigma_2\hat{H}_2(z^*)+\nu_2\,(\sigma_2-1)\,\hat{H}_1(z^*)\right)B(L(z^*))=0, \\ \nu_2 - \left(\nu_2\sigma_1\hat{H}_1(z^*)+\nu_1\,(\sigma_1-1)\,\hat{H}_2(z^*)\right)B(L(z^*))=0. \end{cases} \tag{9.86}$$

From (9.83) and (9.86) we can then find the individual components of $\underline{\boldsymbol{\nu}}$ as

$$\nu_1 = \frac{(1-\sigma_2)\,(\eta_1\pi_1+\eta_2\pi_2-\lambda)}{\phi(1)\left(B(L(z^*))\hat{H}_2(z^*)-1\right)},\quad \nu_2 = \frac{(1-\sigma_1)\,(\eta_1\pi_1+\eta_2\pi_2-\lambda)}{\phi(1)\left(B(L(z^*))\hat{H}_1(z^*)-1\right)}, \tag{9.87}$$

such that the system state vector $\underline{\mathbf{P}^*}(0,0)$ corresponding to an empty system follows as

$$\underline{\mathbf{P}^*}(0,0) = \frac{1}{\phi(1)\eta_1\eta_2}\left[\eta_2\nu_1\sigma_2-\eta_1\nu_2\,(1-\sigma_2)\,,\eta_1\nu_2\sigma_1-\eta_2\nu_1\,(1-\sigma_1)\right]. \tag{9.88}$$

Note that the components of $\underline{\boldsymbol{\nu}}$, and by consequence the components of $\underline{\mathbf{P}^*}(0,0)$ as well, are symmetrical. In other words, the expression for ν_2 is identical to its ν_1 counterpart, after replacing the index 1 by 2 and vice versa. This illustrates the fact that the order of the states j ($j \in \{1,2,\ldots,J\}$) bears no real significance, even for $J > 2$.

Finally, we determine the mean system content (9.40). Therefore, we first determine the first and second derivatives of $M(z) \triangleq \det(\mathbf{M}{*}(z))$ for $z = 1$ as

$$M'(1) = -\lambda\,(\sigma_1+\sigma_2) + \eta_1\sigma_1 + \eta_2\sigma_2 + \phi(1)\,[2\lambda - (\eta_1+\eta_2)]\,, \tag{9.89}$$

and

$$\begin{aligned} M''(1) = {} & -\lambda'\,(\sigma_1+\sigma_2) + 2\,(\lambda-1)\,(\eta_1\sigma_1+\eta_2\sigma_2) \\ & + 2\phi(1)\left[\lambda' + \lambda^2 - (2\lambda-1)\,(\eta_1+\eta_2) + \eta_1\eta_2\right], \end{aligned} \tag{9.90}$$

where we introduced $\lambda' \triangleq L''(1)B'(1) + L'(1)^2B''(1)$. Note that (9.82) and (9.89) allow us to verify that indeed the equilibrium condition $\rho < 1$ is equivalent to $M'(1) > 0$. In order to obtain $\mathrm{d}\underline{\mathbf{P}^*}(x(z),z)/\mathrm{d}z$ for $z = 1$, we note that, specifically for the case $J = 2$, we have that

$$\operatorname{adj}(\mathbf{M}^*(z)) = \mathbf{I_2} - B(L(z))\ \operatorname{adj}\!\left(\hat{\mathbf{H}}^*(z)\right), \tag{9.91}$$

$$\operatorname{adj}(\mathbf{H}^*(z)) = \operatorname{adj}(\mathbf{H_0}) + z\ \operatorname{adj}(\mathbf{H_1}), \tag{9.92}$$

such that

$$\begin{aligned} \left.\frac{\mathrm{d}}{\mathrm{d}z}\ \underline{\boldsymbol{\nu}}\ \operatorname{adj}(\mathbf{M}^*(z))\ \underline{\mathbf{e_2}}\right|_{z=1} &= \underline{\boldsymbol{\nu}}\,(\operatorname{adj}(\mathbf{H_1}) - \lambda\ \operatorname{adj}(\mathbf{H}))\,\underline{\mathbf{e_2}} \\ &= \phi(1)\,(\eta_1\nu_2 + \eta_2\nu_1 - \lambda\,(\nu_1+\nu_2))\,. \end{aligned} \tag{9.93}$$

The mean system content then follows as

$$\begin{aligned} \mathrm{E}[u] = {} & \frac{1}{M'(1)}\left(\lambda - 1 - \frac{M''(1)}{2M'(1)}\right)(1-\phi(1))\,(\nu_1+\nu_2) \\ & + \frac{\phi(1)}{M'(1)}\,(\eta_1\nu_2 + \eta_2\nu_1 - \lambda\,(\nu_1+\nu_2)) - \frac{\gamma\lambda}{1-\gamma}. \end{aligned} \tag{9.94}$$

9.8 Numerical examples

To finish our study of the system with geometric train arrivals and Markovian output line interruptions, we present some numerical examples to illustrate the effect of different system parameters on the buffer performance. Note that obtaining an analytical result for the mean packet delay as in (9.53), or similarly the mean train delay from (9.67), has not yet succeeded.

Therefore we will limit ourselves to investigating the mean packet delay as obtained using Little's result.

First, we look at the effect of both the incidence and train length of the arrival process on the mean packet delay. In Figure 9.2, the mean packet delay $\mathrm{E}[d_{\mathcal{P}}]$ is plotted as a function of the system load, with Poisson distributed numbers of new trains per slot and for various sizes J of the Markovian state space. For each value of J, the entries of the transition matrix $\mathbf{H}$ are given by:

$$[\mathbf{H}]_{ij} = \begin{cases} p_i, & i = j, \\ q^k, & |i-j| = k > 0, \end{cases} \tag{9.95}$$

where the transition probability is fixed at $q = 0.15$ and the probabilities $p_i = 1 - \sum_{k=1}^{J-i} q^k - \sum_{k=1}^{i-1} q^k$ ensure the rows of $\mathbf{H}$ are normalized. For each state j the probability for the output line to be open is given by $\eta_j = 1 - (j-1)/(J-1)$, such that the states are ordered according to the linearly decreasing probability for the output line to be accessible. In this configuration, for each $J > 1$, the output line is always open when the Markovian process is in state 1 and state J is always a blocking state. For the case of geometric output line interruptions ($J = 1$), the Markovian process is defined slightly different, with $\mathbf{H} = [1]$ by definition and $\boldsymbol{\eta} = [0.5]$. In this configuration, the stationary probability vector is given by $\underline{\boldsymbol{\pi}} = \underline{\mathbf{e_J}}^T/J$ and the accessibility rate of the output line is $\underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}} = 0.5$. In order to let the load $\rho = \lambda/\big(\underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}}\big)$ move through the interval $]0,1[$, we then either fix the mean train length at $L'(1) = 4$ while the incidence varies (solid lines) or we fix the mean train arrival rate at $B'(1) = 0.04$ for a variable mean train length (dotted lines). The gray vertical line marks the point $\rho = 0.32$ for which both the mean train arrival rate $B'(1)$ and the mean train length $L'(1)$ assume their default values. Similar as for the session-based arrival process described in Chapter 8, we see that the curves obtained for varying mean train lengths pass the gray line at a steeper angle than the curves corresponding to varying train incidence rates. This illustrates that the impact of the mean train length on the system's performance is clearly more pronounced than the impact of the incidence rate. This can be understood by the fact that the sole contribution to the time-correlated nature of the train arrival process comes from the fact that trains can generate multiple packets over consecutive slots. From Figure 9.2, we can also draw some conclusions about the effect of the size of the state space of this particular Markovian process on the system's performance. Most notably, we see that the curves for the case of a single output line state ($J = 1$) are positioned considerably lower than the other curves, illustrating the fact that the case $J = 1$ corresponds to geometric output line interruptions, which in fact do not cause any correlation effects. Note that, due to the construction method for $\mathbf{H}$, not only the granularity of the Markovian process increases when the state space increases, but also the correlation coefficient

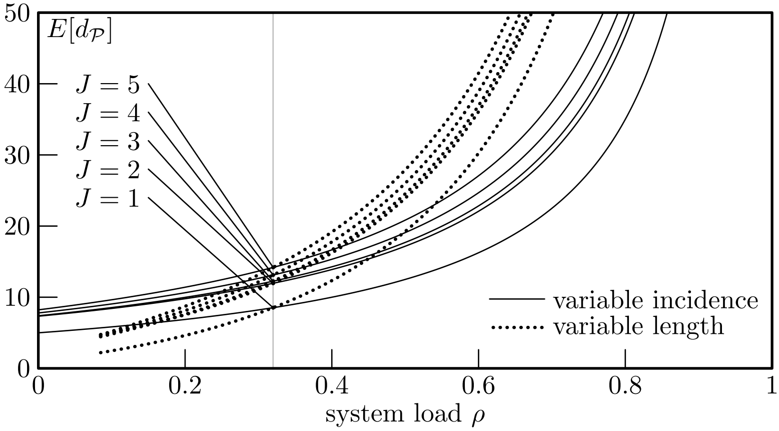

Figure 9.2: The mean packet delay as a function of the system load $\rho = \lambda/\sigma$ for various sizes J of the state space of the Markovian process.

$\phi(1)$ between the output line state in two consecutive slots. For clarity, this correlation coefficient for each of the Markovian processes is given in Table 9.1. Intuitively, we can understand this by considering the mean number of slots needed for the Markovian process to move from the blocking state J to the interruption-free state 1. Due to the structure of $\mathbf{H}$ for different values of J, this transition will generally require more slots for higher values of J, such that the lag-1 correlation coefficient $\phi(1)$, and by consequence also the mean packet delay, increases for $J = 2 \rightarrow 5$.

In Figure 9.3, we reconsider the system studied in Figure 9.2, except that we now only consider a $J = 3$-state Markovian process governing the output line, but for various values of the transition probability q to direct neighbors. As could be expected from previous comments, we again see the same difference in impact on the mean packet delay of variations in the train incidence rate and variations in the mean train length as before. Therefore we shift our attention to the impact of the transition probability q to direct neighbors on the correlation coefficient $\phi(1)$ between the output line state in two consecutive slots and, by consequence, on the mean packet delay. In Table 9.2, where the lag-1 correlation coefficient for each configuration is

J	1	2	3	4	5
$\phi(1)$	-	0.7	0.805	0.867925	0.906115

Table 9.1: The correlation coefficient between the output line states in two consecutive slots for the different Markovian processes in Figure 9.2.

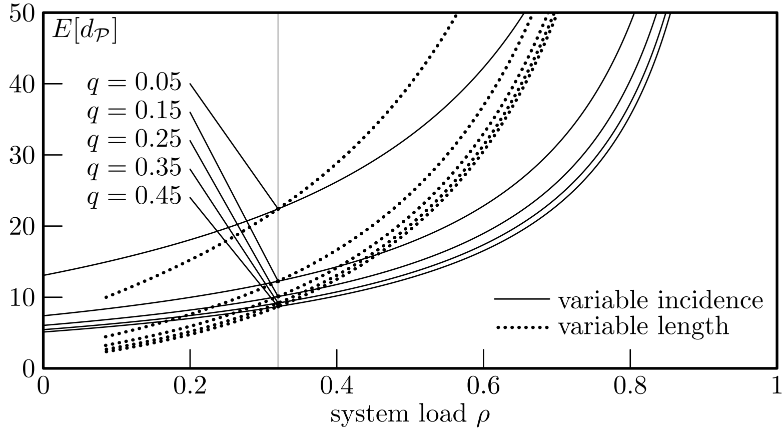

Figure 9.3: The mean packet delay as a function of the system load $\rho = \lambda/\sigma$ for various values of the transition probability q to direct neighbors.

given, we see that while q increases from 0.05 to 0.45, the lag-1 correlation coefficient decreases. Clearly, due to the structure of **H**, a very small value of q corresponds to a Markovian process where state transitions occur rarely at best, leading to a high lag-1 correlation coefficient and thus a high mean packet delay. Conversely, when q is rather high, it will be very unlikely for the Markovian process to remain in a certain state during consecutive slots, such that the lag-1 correlation coefficient will be rather small. From Figure 9.3 and Table 9.2, it is clear that the effect of the lag-1 correlation coefficient is by no means proportional. More specifically, if the lag-1 correlation coefficient $\phi(1)$ is small, a small increase $\Delta\phi(1)$ in the lag-1 correlation coefficient has only a small effect on the mean packet delay. If however the lag-1 correlation coefficient is already significantly high, the same small increase $\Delta\phi(1)$ can have a devastating effect on the mean packet delay.

Next, we investigate the impact of the correlation coefficient $\phi(1)$ between the output line state in two consecutive slots further. Figure 9.4 shows the mean packet delay on a logarithmic scale as a function of the lag-1 correlation coefficient $\phi(1)$. We consider a J = 3-state Markovian

q	0.05	0.15	0.25	0.35	0.45
$\phi(1)$	0.945	0.805	0.625	0.405	0.145

Table 9.2: The correlation coefficient between the output line states in two consecutive slots for the different values of the transition probability q to direct neighbors in Figure 9.3.

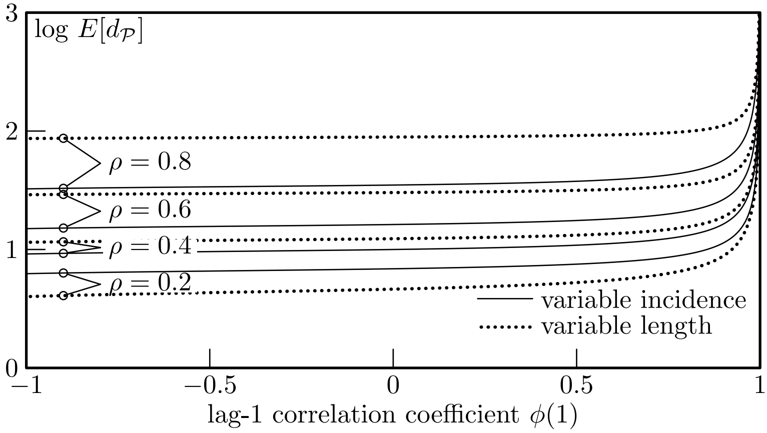

Figure 9.4: The mean packet delay as a function of the correlation coefficient $\phi(1)$ between the output line states in two consecutive slots.

process governing the output line, with $\boldsymbol{\eta} = \mathrm{diag}(1, 0.5, 0)$, structured as in the previous examples and the transition matrix

$$\mathbf{H} = \begin{bmatrix} p^2 & 2p\,(1-p) & (1-p)^2 \\ \frac{1}{2}\left(1-p^2\right) & p^2 & \frac{1}{2}\left(1-p^2\right) \\ (1-p)^2 & 2p\,(1-p) & p^2 \end{bmatrix}, \tag{9.96}$$

such that the stationary probability vector is given by

$$\underline{\boldsymbol{\pi}} = \frac{1}{3p+1}\left[\tfrac{1}{2}\,(p+1) \quad 2p \quad \tfrac{1}{2}\,(p+1)\right], \tag{9.97}$$

and the accessibility rate of the output line is $\underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \overline{\mathbf{e_J}} = 0.5$. Note that the lag-1 correlation coefficient can be found from (9.96) and (9.97) as $\phi(1) = 2p - 1$. By letting the probability p cover the range $]0, 1[$, the lag-1 correlation coefficient $\phi(1)$ then covers the complete range $]-1, 1[$. Similar as in previous figures, Figure 9.4 shows multiple pairs of curves, each pair now corresponding to different values of the system load ρ. The default settings for the train incidence distribution and the mean train length are identical to those in the previous figures. In other words, the solid line curves correspond to $L'(1) = 4$ and a Poisson incidence distribution with variable mean, whereas for the dotted line curves the mean number of new trains per slot is fixed at $B'(1) = 0.4$. The variable parameters of the arrival processes are then chosen such that the equation $L'(1)B'(1) = \rho\sigma$ holds, where $\sigma \triangleq \underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \overline{\mathbf{e_J}}$ is the output line accessibility rate. Note that with these settings for the arrival processes, a system load of $\rho = 0.2$ is smaller than the load $\rho_{\mathrm{default}} = 0.32$ that would be obtained when both

$B'(1)$ and $L'(1)$ assume their default values (i.e. the vertical gray line in the previous figures). Therefore, the dotted line curve for $\rho = 0.2$ is positioned lower than its solid line counterpart, whereas for the other values of the system load ρ, the dotted line curves are positioned higher than their solid line counterparts. We also see our previous findings concerning the impact of the correlation coefficient confirmed in Figure 9.4. For negative, small and even moderate positive values of the lag-1 correlation coefficient $\phi(1)$, it seems that the mean packet delay is hardly influenced by the correlation coefficient between the output line state in two consecutive slots. For high values of $\phi(1)$ however, it becomes clear that the lag-1 correlation coefficient can in fact have a dramatic effect on the mean packet delay.

Finally, we present the effect of the output line accessibility rate $\sigma \triangleq \underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}}$ on the mean packet delay in Figure 9.5. Note that this accessibility rate is determined by both the transition probabilities via the stationary probability vector $\underline{\boldsymbol{\pi}}$ and the probabilities for the output line to be accessible in each state of the Markovian process. In order to separate both contributors, we consider a $J = 3$-state Markovian process governing the output line, controlled by two parameters $x, y \in \,]0, 2[$, with the transition matrix given by

$$\mathbf{H} = \begin{bmatrix} p_1(x)^2 & 2p_1(x)\left(1 - p_1(x)\right) & \left(1 - p_1(x)\right)^2 \\ p_2(x)^2 & 2p_2(x)\left(1 - p_2(x)\right) & \left(1 - p_2(x)\right)^2 \\ p_3(x)^2 & 2p_3(x)\left(1 - p_3(x)\right) & \left(1 - p_3(x)\right)^2 \end{bmatrix}, \tag{9.98}$$

where the probabilities $p_j(x)$ $(j \in \{1, 2, 3\})$ are defined as

$$p_j(x) = \begin{cases} \frac{1}{4}x\left(4 - j\right), & 0 < x \leq 1, \\ 1 - \frac{1}{4}\left(2 - x\right)j, & 1 \leq x < 2. \end{cases} \tag{9.99}$$

The output line accessibility matrix is given by

$$\boldsymbol{\eta} = \operatorname{diag}\left(\min(1, y), y/2, \max(0, y - 1)\right). \tag{9.100}$$

With this configuration, the parameter x allows us to control the transition probabilities such that for $x \to 0$, the Markovian process is biased to state 3 with the worst accessibility rate, whereas for $x \to 2$ the state 1 with the least interruptions is favored. Similarly, the parameter y allows us to control the overall accessibility of the output line. Now we can fix the transition parameter at $x = 1$ while the accessibility parameter y covers the range $]0, 2[$ in order to study the effect of the accessibility matrix $\boldsymbol{\eta}$ on the mean packet delay (solid line). Conversely, we can also isolate the effect of the stationary probability vector $\underline{\boldsymbol{\pi}}$ on the mean packet delay by fixing the accessibility parameter at $y = 1$ while x covers $]0, 2[$ (dotted line). Note that the relation between (x, y) and σ is not straightforward, even when one of the parameters x or y is set to its default value, the values for σ on the horizontal axis are therefore also calculated values, just like the values of $\log\ \mathrm{E}[d_{\mathcal{P}}]$ on the

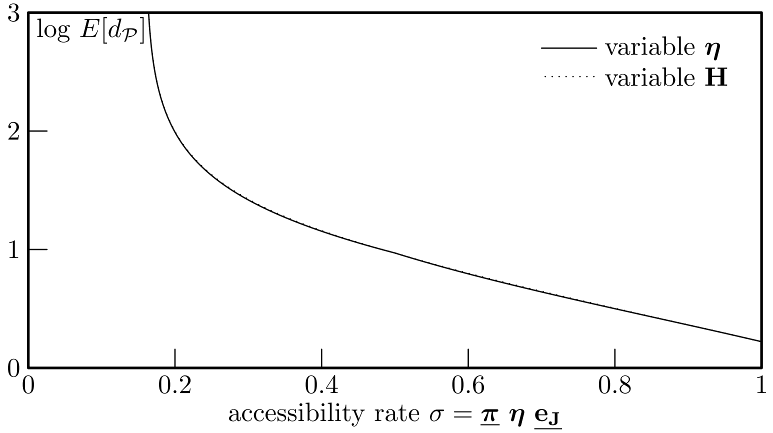

Figure 9.5: The mean packet delay as a function of the output line accessibility rate $\sigma \triangleq \underline{\boldsymbol{\pi}}\ \boldsymbol{\eta}\ \underline{\mathbf{e_J}}$.

vertical axis. The train arrival process is identical to the process described for the previous figures, with the default parameter settings, i.e. the train incidence distribution is a Poisson distribution with mean $B'(1) = 0.04$ and the mean train length is $L'(1) = 4$, such that the mean packet arrival rate is $\lambda = 0.16$. When looking at Figure 9.5, it is virtually impossible to tell both curves apart. We have repeated the experiment for other values of J and the default settings for x and y but every time, the alternative curves seemed to overlap with the curves displayed here. Therefore it is a reasonable assumption that the total effect of the accessibility rate σ on the mean packet delay overshadows the individual contributions of $\mathbf{H}$ and $\boldsymbol{\eta}$.

Epilogue

In this dissertation, we have presented and analyzed various discrete-time queueing models, each with their specific qualities and challenges. We now revisit these models and summarize the main contributions of these analyses. Furthermore, we also present some of the possibilities and challenges that were left untouched in this work.

E.1 Main contributions

In Chapter 6, we have considered a scenario where a network buffer is fed by two distinct classes of information streams of which one should be prioritized. We have analyzed a reservation-based priority scheduling discipline and we have compared its characteristics with the two extremes of FIFO scheduling and Absolute Priority. We have established closed-form expressions for the pgfs of the steady-state system state and the packet delay for both individual classes. Our results show that the Reservation discipline effectively solves the problem of starvation of low-priority packets under a heavy high-priority load, which is known to occur under Absolute Priority. This comes at the cost of a reduction of the high-priority delay performance, which can amount to a considerable pullback, especially under heavy load conditions or a large relative high-priority load. By simulation, we have shown that by inserting additional reserved positions, the delay differentiation offered by the Reservation discipline can be made adjustable.

Chapter 7 was set in an operations research context, such as production processes, where activation of the service unit comes at a considerable cost. In order to reduce the operational cost of the queueing system, service to customers is postponed until a certain number of customers have accumulated. Preventing excessive delays, the arrival of the first customer initiates a timer which activates the service unit after a specified amount of time, regardless of the number of additional customers in the queue. This double threshold policy, referred to as the NT-policy, induces a three-phase cyclic pattern in the system's functioning, such that at first the system is empty, after the first arrival it is accumulating more customers while the service unit remains idle and finally, the service unit is activated and is serving customers. We have presented two different scenarios for the customer service times and correspondingly showed two distinct methods for the performance analysis. In each case, this analysis resulted in closed-form expressions for the distributions of the sojourn times, the system content and the customer delay, conditioned on each of the phases. We have illustrated the effects of the system parameters and have compared the NT-policy with policies that employ only one of the two thresholds. Additionally, we also presented an approximation technique with reasonably accurate results that can drastically reduce the computational power needed to acquire results for large values of the thresholds.

In Chapter 8, we were concerned with an accurate and realistic model for arrival streams generated by file servers and media streaming servers in modern packet-based communication networks. Due to fragmentation of large data volumes, such traffic sources tend to generate sequences of packets which together form a whole, rather than individual packets that occur independently from each other. We have modelled these traffic sources using session-based arrivals and applied them to a queueing system with single-slot service times and geometric output line interruptions. With this traffic model, packets occur as individual parts of stochastic structures called sessions, which are characterized by three distributions describing the session arrival rate, the length in slots of the sessions and the number of packets generated per session per slot. Existing research studied the system content and the packet delay, we on the other hand have extended this research by analyzing the session delay.

Finally, in Chapter 9, we went a step further in modelling a realistic buffer in a file server or media streaming server in contemporary packet-based communication networks. In this scenario, we considered not only a correlated train-based arrival process, but also correlated interruptions of the queueing system's output line. Similar to session-based arrivals, train-based arrival processes produce sequences of packets referred to as trains, which are characterized by an incidence and length distribution, but trains produce exactly one packet per slot. The accessibility of the output line is described by a Bernoulli random variable that depends on the state of an arbitrary-sized Markovian process. We have presented an analytical technique for the determination of the system content distribution and the distributions of the packet and train delays. We illustrated the impact of the different system parameters on the packet delay, with a specific interest in the effects of the correlation in both the arrival process and the interruption process.

E.2 Future work

As opposed to the time and energy one can spend on the research of even well-contained topics, the research possibilities of those topics are limitless. The research presented in this dissertation is no different. In this section, we therefore point out some topics that were left untouched in our analysis but that may be worth investigating.

As mentioned in Chapter 6, the insertion of additional reserved positions in a queueing system with reservation-based scheduling introduces adjustability of the delay differentiation between the two traffic classes. Even though we presented simulated results for such a queueing system with an arbitrary number N of reservations, an analytical study was not performed for *iid* service times. In order to successfully perform such an analysis, a $(N+2)$-dimensional state description will be required, since one will need to keep track of the positions of all N reservations. As such, it is expected that the analysis will be quite complicated. Additionally, it can be interesting to investigate how a correlated arrival process affects the delay performance of the system and the delay differentiation obtained by reservation-based scheduling. Another interesting extension would be to consider class-specific service time distributions.

Our analysis of the NT-policy in Chapter 7 considered individual customers arriving with geometric interarrival times only. Alternative arrival processes may yield different results, especially if the arrival process allows for multiple arrivals during one slot. In that case, it might happen that the N-threshold is met during the same slot as the arrival of the first customer, such that the accumulating phase should be skipped. The underlying goal of the NT-policy is to reduce a certain cost induced by the initialization of the service unit. A detailed cost analysis of such a system might therefore be another interesting extension to the research presented here.

In Chapters 8 and 9, we constructed realistic and accurate models for the output buffers of file servers and media streaming servers. Once the packets have moved further in the network, the correlation between individual packets in a single session or packet train becomes less clear, as these packets will be heavily dispersed. As such the paradigms of sessions and trains as defined in this dissertation are not fit for the analysis of a mid-network buffer. In order to tackle this problem, it could be interesting to study how packet trains and sessions evolve as they pass through one or more queueing systems.

Bibliography

[1] Joseph Abate and Ward Whitt. Numerical inversion of probability generating functions. *Operations Research Letters*, 12(4):245–251, 1992. ISSN 0167-6377. doi: 10.1016/0167-6377(92)90050-D.

[2] Joseph Abate and Ward Whitt. Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing*, 7(1): 36–43, Winter 1995. doi: 10.1287/ijoc.7.1.36.

[3] Attahiru Sule Alfa and Wei Li. Optimal (N,T)-policy for $M/G/1$ system with cost structures. *Performance Evaluation*, 42(4):265–277, November 2000. ISSN 0166-5316. doi: 10.1016/S0166-5316(00)00015-8.

[4] Mustafa Mehmet Ali and Xinxin Song. A performance analysis of a discrete-time priority queueing system with correlated arrivals. *Performance Evaluation*, 57(3):307–339, July 2004. doi: 10.1016/j.peva.2004.01.001.

[5] Chris Blondia and Olga Casals. Statistical multiplexing of VBR sources: A matrix-analytic approach. *Performance Evaluation*, 16(1–3):5–20, November 1992. ISSN 0166-5316. doi: 10.1016/0166-5316(92)90064-N.

[6] Walter Böhm and Sri Gopal Mohanty. On discrete-time Markovian N-policy queues involving batches. *Sankhya: The Indian Journal of Statistics, Series A*, 56(1):144–163, February 1994. ISSN 0581572X.

[7] Onno J. Boxma and Wim P. Groenendijk. Waiting times in discrete-time cyclic-service systems. *IEEE Transactions on Communications*, 36 (2):164–170, 1988. ISSN 0090-6778. doi: 10.1109/26.2746. URL `http://oai.cwi.nl/oai/asset/1727/1727A.pdf`.

[8] Herwig Bruneel. Buffers with stochastic output interruptions. *Electronics Letters*, 19(18):735–737, September 1983. ISSN 0013-5194. doi: 10.1049/el:19830501.

[9] Herwig Bruneel. On the behavior of buffers with random server interruptions. *Performance Evaluation*, 3(3):165–175, August 1983. ISSN 0166-5316. doi: 10.1016/0166-5316(83)90001-9.

[10] Herwig Bruneel. A general model for the behaviour of infinite buffer with periodic service-opportunities. *European Journal of Operational Research*, 16(1):98–106, 1984. ISSN 0377-2217. doi: http://dx.doi.org/10.1016/0377-2217(84)90317-5.

[11] Herwig Bruneel. Performance of discrete-time queueing systems. *Computers and Operations Research*, 20(3):303–320, April 1993. ISSN 0305-0548. doi: 10.1016/0305-0548(93)90006-5.

[12] Herwig Bruneel. Calculation of message delays and message waiting times in switching elements with slow access lines. *IEEE Transactions on Communications*, 42(2-4):255–259, 1994. ISSN 0090-6778. doi: 10.1109/TCOMM.1994.577026.

[13] Herwig Bruneel and Byung G. Kim. *Discrete-time models for communication systems including ATM*. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1993. ISBN 9780792392927.

[14] Herwig Bruneel, Bart Steyaert, Emmanuel Desmet, and Guido H. Petit. Analytic derivation of tail probabilities for queue lengths and waiting times in atm multiserver queues. *European Journal of Operational Research*, 76(3):563–572, 1994. ISSN 0377-2217. doi: 10.1016/0377-2217(94)90287-9.

[15] Wojciech Burakowski and Halina Tarasiuk. On new strategy for prioritising the selected flow in queuing system. In *Proceedings of the COST 257 11th Management Committee Meeting*, COST-257 TD(00)03, Barcelona, Spain, January 2000.

[16] Carna Botnet. Internet Census 2012 - Port scanning /0 using insecure embedded devices, March 2013. URL `http://internetcensus2012.bitbucket.org/paper.html`.

[17] Bong Dae Choi, Doo Il Choi, Yoonju Lee, and Dan Keun Sung. Priority queueing system with fixed-length packet-train arrivals. *IEE Proceedings-Communications*, 145(5):331–336, October 1998. ISSN 1350-2425. doi: 10.1049/ip-com:19982288.

[18] Doo Il Choi, Tae-Sung Kim, and Sangmin Lee. Analysis of a queueing system with a general service scheduling function, with applications to telecommunication network traffic control. *European Journal of Operational Research*, 178(2):463–471, April 2007. doi: 10.1016/j.ejor.2005.12.036.

[19] Andrzej Chydzinski. Time to reach buffer capacity in a BMAP queue. *Stochastic Models*, 23(2):195–209, 2007. doi: 10.1080/15326340701300746.

[20] Israel Cidon, Asad Khamisy, and Moshe Sidi. Delay, jitter and threshold crossing in ATM systems with dispersed messages. *Performance Evaluation*, 29(2):85–104, March 1997. ISSN 0166-5316. doi: 10.1016/S0166-5316(96)00006-5.

[21] John N. Daigle. Message delays at packet-switching nodes serving multiple classes. *IEEE Transactions on Communications*, 38(4):447–455, April 1990. ISSN 0090-6778. doi: 10.1109/26.52655.

[22] Sofian De Clercq, Koen De Turck, Bart Steyaert, and Herwig Bruneel. Frame-bound priority scheduling in discrete-time queueing systems. *Journal of Industrial and Management Optimization*, 7(3):767–788, August 2011. doi: 10.3934/jimo.2011.7.767.

[23] Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Statistical multiplexing of correlated variable-length packet trains: an analytic performance study. *Journal of the Operational Research Society*, 52(3):318–327, March 2001. ISSN 0160-5682. doi: 10.1057/palgrave.jors.2601102.

[24] Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Mean value and tail distribution of the message delay in statistical multiplexers with correlated train arrivals. *Performance Evaluation*, 48(1-4):103–129, May 2002. ISSN 0166-5316. doi: 10.1016/S0166-5316(02)00033-0.

[25] Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Delay differentiation by reserving space in queue. *Electronics Letters*, 41(9):564–565, April 2005. ISSN 0013-5194. doi: 10.1049/el:20050116.

[26] Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Place reservation: delay analysis of a novel scheduling mechanism. *Computers & Operations Research*, 35(8):2447–2462, August 2008. ISSN 0305-0548. doi: 10.1016/j.cor.2006.12.003.

[27] Stijn De Vuyst, Sabine Wittevrongel, Dieter Fiems, and Herwig Bruneel. Controlling the delay trade-off between packet flows using multiple reserved places. *Performance Evaluation*, 65(6–7):484–511, June 2008. ISSN 0166-5316. doi: 10.1016/j.peva.2007.12.008.

[28] Stijn De Vuyst, Krzysztof Tworus, Sabine Wittevrongel, and Herwig Bruneel. Analysis of stop-and-wait ARQ for a wireless channel. *4OR*, 7(1):61–78, March 2009. ISSN 1619-4500. doi: 10.1007/s10288-008-0072-x.

[29] Stijn De Vuyst, Sabine Wittevrongel, Carl Sys, and Herwig Bruneel. Method and device for scheduling data traffic. *WO 2014/032960 A1*, PCT application PCT/EPP2013/066874, March 2014.

[30] M. Dowell and P. Jarrat. A modified regula falsi method for computing the root of an equation. *BIT Numerical Mathematics*, 11(2):168–174, 1971. ISSN 0006-3835. doi: 10.1007/BF01934364.

[31] Khaled M. Faud Elsayed and Harry G. Perros. The superposition of discrete-time Markov renewal processes with an application to statistical multiplexing of bursty traffic sources. *Applied Mathematics and Computation*, 115(1):43–62, 2000. ISSN 0096-3003. doi: 10.1016/S0096-3003(99)00134-4.

[32] Agner Krarup Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik*, 20(B):33–39, 1909.

[33] Agner Krarup Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektrotkeknikeren*, 13:5–13, 1917.

[34] Bart Feyaerts and Sabine Wittevrongel. Performance analysis of a priority queue with place reservation and general transmission times. In *Lecture Notes in Computer Science*, volume 5261, pages 197–211, Palma de Mallorca, Spain, 2008. Springer-Verlag, Berlin. ISBN 978-3-540-87411-9.

[35] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Delay analysis of a discrete-time $GI - GI - 1$ queue with reservation-based priority scheduling. *European Journal of Operational Research*, page To be accepted.

[36] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Analysis of a discrete-time priority queue with place reservations and geometric service times. In *Proceedings of the Sixth Conference on Design, Analysis, and Simulation of Distributed Systems, DASD 2008*, Edinburgh, Scotland, United Kingdom, June 2008.

[37] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Analysis of a discrete-time queueing system with an NT-policy. In *Proceedings of the 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010)*, volume 6148 of *Lecture Notes in Computer Science*, pages 29–43, Cardiff, United Kingdom, June 2010. ISBN 3-642-13567-6, 978-3-642-13567-5.

[38] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Session delay in file server output buffers with general session lengths. In *Proceedings of the 2010 IEEE International Conference on Communications (ICC)*, pages 1–5, Cape Town, South-Africa, May 2010. doi: 10.1109/ICC.2010.5502624.

[39] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Modelling data traffic performance in file servers : session-based arrivals. In *Proceedings of the 24th Belgian Conference on Operations Research (ORBEL 24)*, pages 154–155, Liége, Belgium, January 2010.

[40] Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel, and Sabine Wittevrongel. Analysis of discrete-time buffers with heterogeneous session-based arrivals and general session lengths. *Computers & Operations Research*, 39(12):2905–2914, December 2012. ISSN 0305-0548. doi: 10.1016/j.cor.2011.11.023.

[41] Bart Feyaerts, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Analysis of a discrete-time NT-policy queue with general service times. In *Proceedings of the 7th International Conference on Queueing Theory and Network Applications (QTNA 2012)*, Kyoto, Japan, August 2012.

[42] Bart Feyaerts, Sabine Wittevrongel, Stijn De Vuyst, and Herwig Bruneel. Discrete-time queues with train arrivals and Markovian server interruptions. In *Proceedings of the 8th International Conference on Queueing Theory and Network Applications (QTNA 2013)*, pages 83–89, Taichung, Taiwan, July/August 2013.

[43] Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel, and Sabine Wittevrongel. The impact of the NT-policy on the behaviour of a discrete-time queue with general service times. *Journal of Industrial and Management Optimization*, 10(1):131–149, January 2014. doi: 10.3934/jimo.2014.10.131.

[44] Bart Feyaerts, Stijn De Vuyst, Herwig Bruneel, and Sabine Wittevrongel. Performance analysis of buffers with train arrivals and correlated output interruptions. *Journal of Industrial and Management Optimization*, page Accepted for publication, 2015.

[45] Dieter Fiems and Herwig Bruneel. A note on the discretization of Little's result. *Operations Research Letters*, 30(1):17–18, February 2002. ISSN 0167-6377. doi: 10.1016/S0167-6377(01)00112-2.

[46] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, January 2009. ISBN 978-0-521-89806-5.

[47] Rod Fretwell and Demetres Kouvatsos. LRD and SRD traffic: review of results and open issues for the batch renewal process. *Performance Evaluation*, 48(14):267–284, May 2002. ISSN 0166-5316. doi: 10.1016/S0166-5316(02)00041-X.

[48] H.R. Gail, Hantler S.L., Konheim A.G., and B.A. Taylor. An analysis of a class of telecommunications models. *Performance Evaluation*, 21(1–2): 151–161, November 1994. ISSN 0166-5316. doi: 10.1016/0166-5316(94) 90032-9.

[49] H.R. Gail, Hantler S.L., , and B.A. Taylor. Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains. *Advances in Applied Probability*, 28 (1):114–165, March 1996. ISSN 00018678. doi: 10.2307/1427915.

[50] Denos C. Gazis. The origins of traffic theory. *Operations Research*, 50 (1):69–77, January/February 2002. ISSN 0030-364X. doi: 10.1287/opre. 50.1.69.17776.

[51] Nicolas D. Georganas. Buffer behavior with Poisson arrival and bulk geometric service. *IEEE Transactions on Communications*, 24(8):938–940, August 1976. ISSN 0090-6778. doi: 10.1109/TCOM.1976.1093372.

[52] Manish K. Govil and Michael Cathal Fu. Queueing theory in manufacturing: A survey. *Journal of Manufacturing Systems*, 18(3):214–240, 1999. ISSN 0278-6125. doi: 10.1016/S0278-6125(99)80033-8.

[53] Linda Green. Queueing analysis in healthcare. In Randolph W. Hall, editor, *Patient Flow: Reducing Delay in Healthcare Delivery*, volume 91 of *International Series in Operations Research & Management Science*, pages 281–307. Springer US, 2006. ISBN 978-0-387-33635-0. doi: 10. 1007/978-0-387-33636-7_10.

[54] Matthias Grossglauser and Jean-Chrysostome Bolot. On the relevance of long-range dependence in network traffic. *IEEE/ACM Transactions on Networking*, 7(5):629–640, October 1999. ISSN 1063-6692. doi: 10. 1109/90.803379.

[55] Roch Guérin and Vinod Peris. Quality-of-service in packet networks: basic mechanisms and directions. *Computer Networks*, 31(3):169–189, February 1999. doi: 10.1016/S0169-7552(98)00261-X.

[56] Refael Hassin, Justo Puerto, and Francisco R. Fernández. The use of relative priorities in optimizing the performance of a queueing system. *European Journal of Operational Research*, 193(2):476–483, March 2009. doi: 10.1016/j.ejor.2007.11.058.

[57] Moshe Haviv and Jan van der Wal. Mean sojourn times for phase-type discriminatory processor sharing systems. *European Journal of Operational Research*, 189(2):375–386, September 2008. doi: 10.1016/j. ejor.2007.05.051.

[58] Thomas S. Heines. Buffer behavior in computer communication systems. *IEEE Transactions on Computers*, C-28(8):573–576, August 1979. ISSN 0018-9340. doi: 10.1109/TC.1979.1675413.

[59] Alfredo G. Hernández-Díaz and Pilar Moreno. Analysis and optimal control of a discrete-time queueing system under the (m, N)-policy. In *Proceedings of the 1st International Conference on Performance Evaluation Methodolgies and Tools*, Valuetools '06, Pisa, Italy, 2006. ISBN 1-59593-504-5. doi: 10.1145/1190095.1190115.

[60] Daniel P. Heyman. The T-policy for the $M/G/1$ queue. *Management Science*, 23(7):775–778, March 1977. ISSN 0025-1909. doi: 10.1287/mnsc.23.7.775.

[61] Laurence Hoflack, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. Modeling web server traffic with session-based arrival streams. In *Proceedings of the 15th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2008)*, volume 5055 of *Lecture Notes in Computer Science*, pages 47–60, Nicosia, Cyprus, June 2008. ISBN 978-3-540-68980-5. doi: 10.1007/978-3-540-68982-9_4.

[62] Laurence Hoflack, Stijn De Vuyst, Sabine Wittevrongel, and Herwig Bruneel. System content and packet delay in discrete-time queues with session-based arrivals. In *Proceedings of the Fifth International Conference on Information Technology: New Generations (ITNG '08)*, pages 1053–1058, Las Vegas, USA, April 2008. ISBN 978-0-7695-3099-4. doi: 10.1109/ITNG.2008.151.

[63] Jiunn Hsu. Buffer behavior with Poisson arrival and geometric output processes. *IEEE Transactions on Communications*, 22(12):1940–1941, December 1974. ISSN 0090-6778. doi: 10.1109/TCOM.1974.1092142.

[64] Xiaolong Jin and Geyong Min. Performance analysis of priority scheduling mechanisms under heterogeneous network traffic. *Journal of Computer and System Sciences*, 73(8):1207–1220, December 2007. doi: 10.1016/j.jcss.2007.02.008.

[65] Ahmed E. Kamal and Samyukta Sankaran. A combined delay and throughput proportional scheduling scheme for differentiated services. *Computer Communications*, 29(10):1754–1771, June 2006. doi: 10.1016/j.comcom.2005.10.012.

[66] Faouzi Kamoun. Performance evaluation of a queuing system with correlated packet-trains and server interruption. *Telecommunication Systems*, 41(4):267–277, August 2009. ISSN 1018-4864. doi: 10.1007/s11235-009-9160-2.

[67] Edward P.C. Kao and Sandra D. Wilson. Analysis of nonpreemptive priority queues with multiple servers and two priority classes. *European Journal of Operational Research*, 118(1):181–193, October 1999. doi: 10.1016/S0377-2217(98)00280-X.

[68] Jau-Chuan Ke. Optimal NT policies for $M/G/1$ system with a startup and unreliable server. *Computers & Industrial Engineering*, 50(3):248–262, July 2006. ISSN 0360-8352. doi: 10.1016/j.cie.2006.04.004.

[69] Jau-Chuan Ke, Hsin-I Huang, and Yunn-Kuang Chu. Batch arrival queue with N-policy and at most J vacations. *Applied Mathematical Modelling*, 34(2):451–466, February 2010. ISSN 0307-904X. doi: 10.1016/j.apm.2009.06.003.

[70] David George Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *The Annals of Mathematical Statistics*, 24(3):338–354, September 1953. doi: 10.1214/aoms/1177728975.

[71] Aleksandr Khintchin. Mathematische Theorie der stazionären Reihe. *Matematicheskii Sbornik*, 39:73–84, 1932.

[72] Bara Kim and Khosrow Sohraby. Tail behavior of the queue size and waiting time in a queue with discrete autoregressive arrivals. *Advances in Applied Probability*, 38(4):1116–1131, December 2006. ISSN 00018678. doi: 10.1239/aap/1165414594.

[73] Kilhwan Kim and Kyung C. Chae. Discrete-time queues with discretionary priorities. *European Journal of Operational Research*, 200(2): 473–485, January 2010. doi: 10.1016/j.ejor.2008.12.035.

[74] Demetres D. Kouvatsos and International Federation for Information Processing. *Performance modelling and evaluation of ATM networks*. Ifip International Federation for Information Processing. Chapman & Hall, 1995. ISBN 9780412711404.

[75] Guy Latouche and Tetsuya Takine. Markov-renewal fluid queues. *Journal of Applied Probability*, 41(3):746–757, September 2004. ISSN 00219002. doi: 10.1239/jap/1091543423.

[76] Ho Woo Lee and Won Joo Seo. The performance of the $M/G/1$ queue under the dyadic Min(N, D)-policy and its cost optimization. *Performance Evaluation*, 65(10):742–758, October 2008. ISSN 0166-5316. doi: 10.1016/j.peva.2008.04.006.

[77] Soon Seok Lee, Ho Woo Lee, Seung Hyun Yoon, and Kyung Chul Chae. Batch arrival queue with N-policy and single vacation. *Computers & Operations Research*, 22(2):173–189, February 1995. ISSN 0305-0548. doi: 10.1016/0305-0548(94)E0015-Y.

[78] San-qi Li and Chia-lin Hwang. Queue response to input correlation functions: discrete spectral analysis. *IEEE/ACM Transactions on Networking*, 1(5):522–533, October 1993. ISSN 1063-6692. doi: 10.1109/90.251911.

[79] Rosa Elvira Lillo. Ergodicity and analysis of the process describing the system state in polling systems with two queues. *European Journal of Operational Research*, 167(1):144–162, November 2005. doi: 10.1016/j.ejor.2003.10.055.

[80] Youngho Lim and John E. Kobza. Analysis of a delay-dependent priority discipline in an integrated multiclass traffic fast packet switch. *IEEE Transactions on Communications*, 38(5):659–685, May 1990. ISSN 0090-6778. doi: 10.1109/26.54979.

[81] John D. C. Little. A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9(3):383–387, May/June 1961. doi: 10.1287/opre.9.3.383.

[82] John D. C. Little. Little's law as viewed on its 50th anniversary. *Operations Research*, 59(3):536–549, May/June 2011. ISSN 0030-364X. doi: 10.1287/opre.1110.0940.

[83] Miron Livny, Benjamin Melamed, and Athanassios K. Tsiolis. The impact of autocorrelation on queuing systems. *Management Science*, 39 (3):322–339, March 1993. ISSN 00251909. doi: 10.1287/mnsc.39.3.322.

[84] David M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7(1):1–46, 1991. doi: 10.1080/15326349108807174.

[85] David M. Lucantoni, Kathleen S. Meier-Hellstern, and Marcel F. Neuts. A single-server queue with server vacations and a class of non-renewal arrival processes. *Advances in Applied Probability*, 22(3):676–705, September 1990. ISSN 00018678. doi: 10.2307/1427464.

[86] Tom Maertens, Joris Walraevens, and Herwig Bruneel. A modified HOL priority scheduling discipline: performance analysis. *European Journal of Operational Research*, 180(3):1168–1185, August 2007. doi: 10.1016/j.ejor.2006.05.004.

[87] Gianluca Mazzini, Riccardo Rovatti, and Gianluca Setti. A closed form solution of bernoullian two-classes priority queue. *IEEE Communications Letters*, 9(3):264–266, March 2005. doi: 10.1109/LCOMM.2005.03020.

[88] Torben Meisling. Discrete-time queuing theory. *Operations Research*, 6 (1):96–105, Januari/February 1958. ISSN 0030364X. doi: 10.1287/opre.6.1.96.

[89] Carl Dean Meyer. *Matrix Analysis and Applied Linear Algebra.* Society for Industrial and Applied Mathematics, 2000. ISBN 9780898714548.

[90] Israel Mitrani. *Modelling of computer and communication systems.* Number 24 in Cambridge Computer Science Texts. Cambridge University Press, Cambridge, UK, 1987. ISBN 9780521314220.

[91] Pilar Moreno. A discrete-time single-server queue with a modified N-policy. *International Journal of Systems Science*, 38(6):483–492, June 2007. ISSN 0020-7721. doi: 10.1080/00207720701353405.

[92] Sokol Ndreca and Benedetto Scoppola. Discrete time $GI/Geom/1$ queueing system with priority. *European Journal of Operational Research*, 189(3):1403–1408, September 2008. doi: 10.1016/j.ejor.2007.02.056.

[93] Marcel F. Neuts. The single server queue in discrete time-numerical analysis i. *Naval Research Logistics Quarterly*, 20(2):297–304, June 1973. ISSN 1931-9193. doi: 10.1002/nav.3800200210.

[94] Marcel F. Neuts. *Structured Stochastic Matrices of $M/G/1$ Type and Their Applications*, volume 5 of *Probability : Pure and Applied a Series of Textbooks and Reference Books.* Marcel Dekker, New York, 1989. ISBN 9780824782832.

[95] Marcel F. Neuts. Probabilistic modelling requires a certain imagination, 1990.

[96] Marcel F. Neuts. Models based on the Markovian arrival process. *IEICE Transactions on Communications*, E75B(12):1255–1265, December 1992. ISSN 0916-8516.

[97] Ilkka Norros. A storage model with self-similar input. *Queueing Systems*, 16(3-4):387–396, 1994. ISSN 0257-0130. doi: 10.1007/BF01158964.

[98] R. Kannapiran Palvannan and Kiok Liang Teow. Queueing for healthcare. *Journal of Medical Systems*, 36(2):541–547, April 2012. ISSN 0148-5598. doi: 10.1007/s10916-010-9499-7.

[99] Chrissoleon T. Papadopolous and C. Heavey. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research*, 92(1):1–27, July 1996. ISSN 0377-2217. doi: 10.1016/0377-2217(95)00378-9.

[100] Kihong Park and Walter Willinger. *Self-similar network traffic and performance evaluation.* John Wiley & Sons, Inc., New York, 2000. ISBN 9780471206446. doi: 10.1002/047120644X.

[101] Félix Pollaczek. Über eine Aufgabe der Wahrscheinlichkeitstheorie. I. *Mathematische Zeitschrift*, 32:64–100, 1930. ISSN 0025-5874. doi: 10.1007/BF01194620.

[102] Sujit Kumar Samanta, Umesh Chandra Gupta, and Rajendra Kumar Sharma. Analyzing discrete-time $D-BMAP/G/1/N$ queue with single and multiple vacations. *European Journal of Operational Research*, 182 (1):321–339, October 2007. ISSN 0377-2217. doi: 10.1016/j.ejor.2006.09.031.

[103] Chuck Semeria. Supporting differentiated service classes: queue scheduling disciplines. *Juniper Networks white paper*, 2001.

[104] Bart Steyaert, Herwig Bruneel, and Yijun Xiong. An efficient solution technique for discrete-time queues fed by heterogeneous traffic. *International Journal of Communication Systems*, 10(2):73–86, March 1997. ISSN 1099-1131. doi: 10.1002/(SICI)1099-1131(199703)10:2⟨73::AID-DAC325⟩3.0.CO;2-T.

[105] Hideaki Takagi. *Queueing Analysis: Discrete-time systems*. Queueing Analysis: A Foundation of Performance Evaluation. North-Holland, 1993. ISBN 9780444816115.

[106] Tetsuya Takine, Bhaskar Sengupta, and Toshiharu Hasegawa. An analysis of a discrete-time queue for broadband ISDN with priorities among traffic classes. *IEEE Transactions on Communications*, 42(24): 1837–1845, February/March/April 1994. doi: 10.1109/TCOMM.1994.582893.

[107] Junhua Tang, Gang Feng, Chee-Kheong Siew, and Liren Zhang. Providing differentiated services over shared wireless downlink through buffer management. *IEEE Transactions On Vehicular Technology*, 57 (1):548–555, January 2008. doi: 10.1109/TVT.2007.905246.

[108] Chen-Khong Tham, Qi Yao, and Yuming Jiang. A multi-class probabilistic priority scheduling discipline for differentiated services networks. *Computer Communications*, 25(17):1487–1496, November 2002. doi: 10.1016/S0140-3664(02)00035-X.

[109] Sophia Tsakiridou and Ioannis Stavrakakis. Mean delay analysis of a statistical multiplexer with batch arrival processesa generalization to Viterbi's formula. *Performance Evaluation*, 25(1):1–15, March 1996. ISSN 0166-5316. doi: 10.1016/0166-5316(94)00036-0.

[110] Nico Vandaele, Tom Van Woensel, and Aviel Verbruggen. A queueing based traffic flow model. *Transportation Research Part D: Transport and Environment*, 5(2):121–135, March 2000. ISSN 1361-9209. doi: 10.1016/S1361-9209(99)00028-0.

[111] Vladimir Mironovich Vishnevskii and Olga Valerevna Semenova. Mathematical methods to study the polling systems. *Automation and Remote Control*, 67(2):173–220, February 2006. ISSN 0005-1179. doi: 10.1134/S0005117906020019.

[112] Joris Walraevens, Sabine Wittevrongel, and Herwig Bruneel. A discrete-time priority queue with train arrivals. *Stochastic Models*, 23 (3):489–512, July 2007. doi: 10.1080/15326340701471158.

[113] Joris Walraevens, Bart Steyaert, and Herwig Bruneel. Analysis of a discrete-time preemptive resume priority buffer. *European Journal of Operational Research*, 186(1):182–201, April 2008. doi: 10.1016/j.ejor.2007.01.028.

[114] Joris Walraevens, Johan S.H. van Leeuwaarden, and Onno J. Boxma. Power series approximations for two-class generalized processor sharing systems. *Queueing Systems*, 66(2):107–130, October 2010. ISSN 0257-0130. doi: 10.1007/s11134-010-9188-8.

[115] Joris Walraevens, Dieter Fiems, and Herwig Bruneel. Performance analysis of priority queueing systems in discrete time. In *Network Performance Engineering*, volume 5233 of *Lecture Notes in Computer Science*, pages 203–232, 2011. ISBN 978-3-642-02741-3. doi: 10.1007/978-3-642-02742-0_10.

[116] Joris Walraevens, Tom Maertens, and Herwig Bruneel. A semi-preemptive priority scheduling discipline: performance analysis. *European Journal of Operational Research*, 224(2):324–332, January 2013. doi: 10.1016/j.ejor.2012.08.008.

[117] Kuo-Hsiung Wang, Tsung-Yin Wang, and Wen Lea Pearn. Optimal control of the N-policy $M/G/1$ queueing system with server breakdowns and general startup times. *Applied Mathematical Modelling*, 31(10): 2199–2212, October 2007. ISSN 0307-904X. doi: 10.1016/j.apm.2006.08.016.

[118] Tsung-Yin Wang, Kuo-Hsiung Wang, and Wen Lea Pearn. Optimization of the T policy $M/G/1$ queue with server breakdowns and general startup times. *Journal of Computational and Applied Mathematics*, 228 (1):270–278, June 2009. ISSN 0377-0427. doi: 10.1016/j.cam.2008.09.021.

[119] Jianbin Wei, Cheng-Zhong Xu, Xiaobo Zhou, and Qing Li. A robust packet scheduling algorithm for proportional delay differentiation services. *Computer Communications*, 29(18):3679–3690, November 2006. doi: 10.1016/j.comcom.2006.06.009.

[120] Sabine Wittevrongel and Herwig Bruneel. Correlation effects in ATM queues due to data format conversions. *Performance Evaluation*, 32(1): 35–56, February 1998. ISSN 0166-5316. doi: 10.1016/S0166-5316(97) 00015-1.

[121] Sabine Wittevrongel, Stijn De Vuyst, and Herwig Bruneel. Analysis of discrete-time buffers with general session-based arrivals. In *Proceedings of the 16th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2009)*, volume 5513 of *Lecture Notes in Computer Science*, pages 189–203, Madrid, Spain, June 2009. ISBN 978-3-642-02204-3. doi: 10.1007/978-3-642-02205-0_14.

[122] Ronald W. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2):223–231, March/April 1982. doi: 10.1287/opre.30.2.223.

[123] Chin-Chi Wu, Hsien-Ming Wu, and Woei Lin. High-performance packet scheduling to provide relative differentiation in future high-speed networks. *Computer Communications*, 31(10):1865–1876, June 2008. doi: 10.1016/j.comcom.2007.12.016.

[124] Yijun Xiong and Herwig Bruneel. Buffer behaviour of statistical multiplexers with correlated train arrivals. *International Journal of Electronics and Communications*, 51(3):178–186, May 1997. ISSN 0001-1096. doi: 10.1016/0166-5316(93)90010-R.

[125] Micha Yadin and Pinhas Naor. Queueing systems with a removable service station. *Operational Research Quarterly*, 14(4):393–405, December 1963. ISSN 14732858. doi: 10.2307/3006802.

[126] Wen-Hui Zhou and Ai-Hu Wang. Discrete-time queue with Bernoulli bursty source arrival and generally distributed service times. *Applied Mathematical Modelling*, 32(11):2233–2240, November 2008. ISSN 0307-904X. doi: 10.1016/j.apm.2007.07.014.