MASTER

On contention resolution procedures : queueing analysis and simulation

van den Broek, M.X.

Award date:
2003

TECHNISCHE UNIVERSITEIT EINDHOVEN

Department of Mathematics and Computing Science

MASTER'S THESIS

On contention resolution procedures
Queueing analysis and simulation

by

M.X. van den Broek

June, 2001

Supervisor(s): dr. ir. I.J.B.F. Adan
prof. dr. ir. O.J. Boxma

Advisor(s): prof. dr. ir. S.C. Borst
dr. S. Shankar N

Voorwoord

Dit verslag is het eindresultaat van mijn afstudeerproject voor de studie Technische Wiskunde, afstudeerrichting stochastiche besliskunde, aan de Technische Universiteit Eindhoven. Mijn afstudeerproject maakte deel uit van een groter project, het *Pelican* project, waarbij naast de TU Eindhoven ook onderzoeksinstituut Eurandom en het Philips Natuurkundig Laboratorium betrokken waren. Deze projectgroep bestond in totaal uit ongeveer 15 mensen, waarvan verschillende uit het buitenland afkomstig waren. Het hoofddoel van het project was om meer inzicht te krijgen in zogenaamde Random Access kabelnetwerken door deze te bestuderen met behulp van wachtrijtheorie.

Gedurende negen maanden heb ik gewerkt aan problemen die gerelateerd waren aan deze Random Access netwerken. Binnen de projectgroep werd gewerkt in kleinere groepjes van twee tot vier mensen. Om de twee tot drie maanden vonden gezamenlijke bijeenkomsten plaats waarbij presentaties werden gegeven door ieder van de groepjes, zodat men op de hoogte bleef van elkaars vorderingen en resultaten. Ik heb mijn afstudeerwerk verricht op de TU Eindhoven, waar ik gedurende deze negen maanden een kamer heb gedeeld met enkele AIO's.

Ik kijk met tevredenheid terug op deze afstudeerperiode van negen maanden. Ik heb met zeer veel plezier gewerkt aan de wiskundige problemen die voortkwamen uit het onderzoek en ben ook tevreden met het eindresultaat van mijn onderzoek. De projectgroep zal nog minstens een jaar blijven bestaan en ik zal er zelf ook nog wel enige maanden bij betrokken blijven, zij het alleen al door het gezamenlijk schrijven van een paper over de belangrijkste resultaten van mijn afstudeeronderzoek.

Gedurende deze negen maanden ben ik door meerdere mensen begeleid. Ik wil in alfabetische volgorde Ivo Adan, Sem Borst en Onno Boxma, allen verbonden aan de TU Eindhoven, hartelijk bedanken voor de begeleiding tijdens mijn afstuderen. Zowel het meedenken over de problemen als het doorlezen en becommentariëren van stukken van mijn verslag zijn van grote invloed geweest op het eindresultaat. Ook wil ik Sai Shankar N van Philips hartelijk danken voor zijn hulp, zowel het becommentariëren van mijn stukken tekst alsmede ook het geven van de nodige achtergrondinformatie aangaande kabelnetwerken hebben bijgedragen aan het uiteindelijke resultaat. Verder heb ik de sfeer tijdens alle onze afspraken en bijeenkomsten als zeer prettig en stimulerend ervaren. Dit geldt zeker ook voor de projectbijeenkomsten. Hiervoor wil ik iedereen die deel uitmaakt of deel heeft uitgemaakt van de Pelican projectgroep hartelijk danken.

Aangezien dit afstudeerwerk tevens een afronding is van mijn studie wil ik mijn ouders en mijn zus bedanken voor hun steun en belangstelling tijdens mijn studieperiode. Verder wil ik Ægle Hoekstra† noemen als een vriend die mij de schoonheid van de wiskunde heeft laten zien en hiermee een grote invloed heeft gehad op zowel mijn beleving van het vak als de persoon die ik nu ben.

Eindhoven, juni 2001

Mark van den Broek

Samenvatting

Hybrid Fiber Coaxial (HFC) kabelnetwerken werden oorspronkelijk ontworpen om analoge TV-signalen te versturen vanuit een centraal station naar verschillende woningen met televisie. Tegenwoordig worden deze zelfde HFC-netwerken gebruikt voor steeds meer vormen van digitale, bi-directionele communicatie tussen individuele klanten en "de rest van de wereld". Een belangrijk gegeven is dat zo'n netwerk bestaat uit één zogenaamd *upstream* kanaal en één *downstream* kanaal waarover alle heengaande respectievelijk teruggaande communicatie met het centraal station plaatsvindt. Alle klanten binnen hetzelfde netwerk maken dus gebruik van hetzelfde kanaal. Het verkeer dient dus goed geregeld te worden door het centraal station.

Wanneer een individuele klant informatie wil versturen moet de klant eerst via het upstream kanaal op een door het station opgelegd moment een verzoek indienen waarin deze vraagt om het upstream kanaal enige tijd alleen te mogen gebruiken. Wanneer de klant geluk heeft was hij de enige die op dat moment een verzoek indiende en kan de reservering direct plaatsvinden, maar in veel gevallen vragen meerdere klanten tegelijkertijd om capaciteit. Er is dan sprake van een botsing. Helaas kan het centraal station in geval van een botsing niet vaststellen wie er welke reservering wil plegen. Het enige wat het station ontvangt is ruis, waaruit slechts geconcludeerd kan worden dat er meerdere klanten tegelijkertijd een verzoek hebben ingediend. Kort daarna zal het station via het downstream kanaal aan alle klanten drie mogelijke tijdstippen aangeven waarop diegenen die betrokken waren bij de betreffende botsing wederom hun verzoek mogen versturen. De desbetreffende klanten kiezen zelf volkomen willekeurig één van deze drie aangeboden tijdstippen. Klanten die niet bij de botsing betrokken waren mogen afhankelijk van de regels wel of geen verzoek indienen op één van deze drie tijdstippen. Mogelijk ontstaan er weer botsingen op een of meerdere van deze tijdstippen. Dan wordt dezelfde procedure net zolang herhaald totdat alle verzoeken door het station zijn ontvangen. Het upstream kanaal wordt tussendoor ook gebruikt voor het daadwerkelijk verzenden van informatie, maar niet tegelijkertijd met het indienen van verzoeken. Het centraal station zal het upstream kanaal dus afwisselend moeten openstellen voor het ontvangen van verzoeken en voor het verzenden van informatie door individuele klanten.

Door het upstream kanaal op een verstandige manier te gebruiken streeft het centraal station ernaar dat het verzenden van informatie door de klanten zo snel mogelijk verloopt. Er zijn tal van vrijheden die het station kan benutten om de communicatie te optimaliseren. Allereerst heeft het controle over de tijdstippen waarop het aan klanten toestaat verzoeken in te dienen. Verder heeft het ook controle over welke klanten er op een bepaald moment wel of niet een verzoek mogen indienen. Ook het toewijzen van het upstream kanaal aan een individuele klant gebeurt door het station. Ten slotte kan het station ook de verhouding bepalen tussen de tijd die gereserveerd wordt voor het ontvangen van verzoeken en de tijd die gereserveerd wordt voor het ontvangen van verzonden informatie. In dit rapport is bestudeerd hoe dit regelproces het best kan verlopen. De nadruk ligt op het optimaliseren van het verwerken van de individuele verzoeken. Maar ook het regelproces in zijn totaliteit heeft de nodige aandacht gekregen.

In de praktijk worden momenteel twee varianten gebruikt voor het verwerken van de verzoeken. Bij de eerste variant is het zo dat na iedere botsing alléén de betrokken klanten opnieuw een verzoek mogen indienen. Alle klanten met een nieuw verzoek moeten wachten totdat deze verzoeken allemaal zijn ontvangen. De tweede variant staat toe dat op ieder tijdstip alle klanten met een nieuw verzoek dit direct mogen sturen. In dit verslag zal een derde variant worden geopperd, het *arrival-slot mechanisme*. In dit mechanisme worden regelmatig tijdstippen gereserveerd waarop uitsluitend klanten met nieuwe verzoeken deze kenbaar mogen maken. De rest van de tijdstippen is bestemd voor het oplossen van eerder plaatsgevonden botsingen. Klanten met nieuwe verzoeken worden hier geweerd.

Bij de analyse van de drie verschillende mechanismen is in eerste instantie het versturen van informatie door de individuele klanten buiten beschouwing gelaten en is er gekeken naar modellen waarbij alle inkomende verzoeken zonder onderbrekingen voor dataverkeer van individuele klanten verwerkt worden. De zo verkegen *geïsoleerde* mechanismen zijn in dit rapport geformuleerd als wachtrijsystemen en bestudeerd op hun performance, waarbij vooral gekeken is naar de capaciteit van de systemen, de gemiddelde wachttijden en de variantie in de wachttijden. Hierbij is zowel analyse als simulatie gebruikt. De nadruk heeft gelegen op de derde variant aangezien deze nieuw is en van de andere twee varianten reeds het nodige bekend was. Uiteindelijk is een vergelijking gemaakt tussen de drie mechanismen. Het arrival-slot mechanisme en enkele geopperde verfijningen hierop kwamen hierbij als beste naar voren.

Zoals eerder aangegeven moet het centraal station bij het regelen van de communicatie meer doen dan slechts het bepalen welke klanten er op bepaalde tijdstippen verzoeken mogen indienen. Het gehele regelproces is veel gecompliceerder. De eerste twee varianten voor het verwerken van verzoeken zijn weliswaar in de literatuur geanalyseerd, maar de analyse heeft zich veelal beperkt tot het *geïsoleerde* mechanisme. De vraag hoe op een verstandige manier met het mechanisme om te gaan binnen het *gehele* regelproces heeft in de literatuur nog niet veel aandacht gekregen. In dit rapport is na de analyse van de geïsoleerde mechanismen voor alle drie de mechanismen bestudeerd hoe deze optimaal gebruikt zouden kunnen worden binnen het totale regelproces. Minimalisatie van de gemiddelde totale wachttijd van een klant heeft hierbij centraal gestaan. Hierbij zijn de inzichten en resultaten die verkregen waren uit de analyses van de geïsoleerde mechanismen gebruikt. Maar waar nodig zijn ook nieuwe analyses en simulaties gedaan om tot bepaalde resultaten te komen. De gevonden resultaten maken niet alleen een uiteindelijke vergelijking tussen de verschillende mechanismen mogelijk, maar kunnen ook gebruikt worden om ieder mechanisme afzonderlijk te optimaliseren binnen het gehele regelproces.

Het rapport is afgesloten met conclusies en aanbevelingen voortkomend uit de vergelijking tussen de drie mechanismen. Voor verschillende karakteristieken van het netwerk is bekeken welk mechanisme en bijbehorende implementatie optimaal is. Ook hier kwam het arrival-slot mechanisme in veel situaties als beste naar voren.

Contents

Chapter 1

Introduction

1.1 Background

This section provides an introduction to *access networks*, which enable, globally speaking, the electronic communication between individual homes and the rest of the world. These networks are often symbolically called the "last mile" to the customer. For more detailed information, see [7] and [8].

1.1.1 Network structure

Cable TV networks, currently known as hybrid fiber coaxial (HFC) networks, were originally designed for transmitting TV signals from a central station, the so-called Head End (HE), to the homes. At the moment, these networks are used for bi-directional digital communication between individual homes and the rest of the world. In the near future, these access networks will be used for all kinds of communication.

A given access network contains a finite number of users or customers, where this number typically ranges from ten to a couple of thousand. Furthermore, there are both an upstream and a downstream channel which connect the individual users to the HE and the other way around, respectively. All users share the same up- and downstream channel. From the users' point of view, the HE can be seen as their common connection with the outer world. Furthermore, the Head End controls the traffic within the access network. The structure of an access network is schematically depicted in figure 1.1. An individual user is indicated by NT, which is an abbreviation for *Network Terminal.*

In a real access network, not every user has the same distance to the HE.

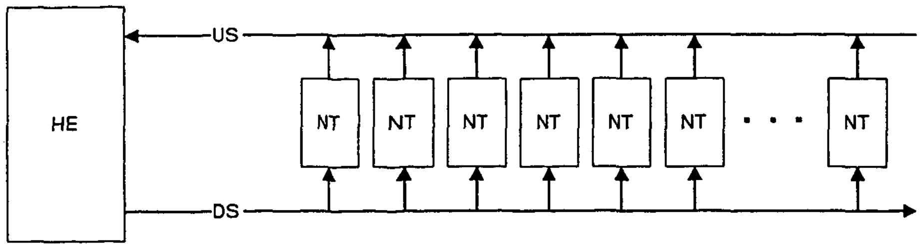

Figure 1.1: Schematic overview of an HFC access network.

Therefore, one may expect that the (propagation) delay that occurs in the communication with the HE is not the same for all users.We will return to propagation delay later. This problem is assumed to be solved by implementing a synchronization method, which makes that every user appears to have the same distance to the HE.

When a user wants to transmit data over the upstream channel (US in figure 1.1), in general the following sequential *three-step* procedure is followed:

1. At a certain moment the user transmits a *request* to the HE for using a certain amount of upstream channel time.
2. The user waits for feedback from the HE. The HE will ultimately indicate when the user's data are allowed to be transmitted over the upstream channel.
3. The user transmits the data at the time indicated by the HE. In the meantime it virtually waits in the so-called data queue.

In most access networks, the requests mentioned in the first step can only be transmitted at certain times. The HE indicates to a group of users when a request can be transmitted. When more than one user sends a request at the same time, collisions occur. To resolve these collisions, a so-called collision resolution procedure is used. To indicate how these procedures work, it is necessary to discuss the structure of the upstream channel first.

The upstream communication channel is divided into time slots of fixed length. Each slot is again divided in three mini-slots. The length of a time slot corresponds to sending a 64 byte data packet. There are typically two types of slots: data slots and contention slots.

- A data slot, also known as a *reservation slot*, is formed by concatenating three mini-slots. Such a slot is exclusively used for transmitting data of a particular user.
- A contention slot consists also of three mini-slots, but the mini-slot structure is maintained. Such a slot is used by users to transmit requests. When a collision occurs, this collision will be resolved by retransmitting the requests in a way that will be indicated shortly. The retransmitting of the requests will again take place during contention slots. There are typically two types of contention slots: ALOHA slots and tree slots, corresponding to two different contention resolution procedures. Only the tree slots will be discussed further now. For more information on the ALOHA protocol, we refer to [10], the first publication on this subject.

The slots are organized in frames, which typically comprise 18 slots. The Head End determines at the beginning of each frame the type of each slot in that upcoming frame. In figure 1.2, this structure is depicted to illustrate the frame structure.

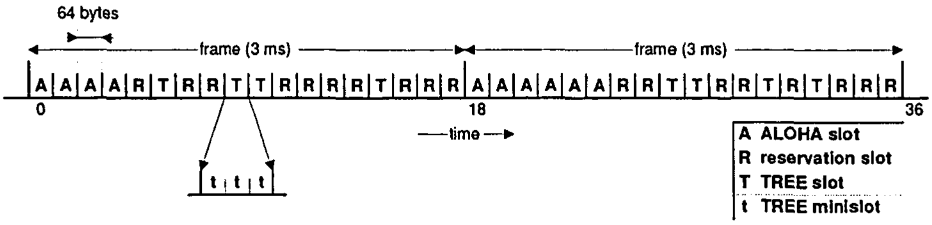

Figure 1.2: Representation of the frame structure.

Summarizing one can say that the transmission of data by an individual customer consists of two stages. First a contention stage takes place in which the individual user has to compete with other stations for exclusive data slots. Once a request has been succesfully received by the Head End, a customer enters the second stage. In this stage, a customer joins the data queue and finally occupies a certain number of data slots. This structure is schematically presented in figure 1.3.

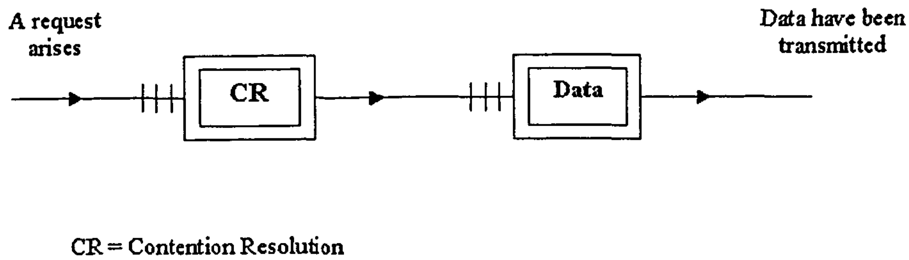

Figure 1.3: Two-stage structure of data transmission.

Figure 1.2 shows that in an access network, these two stages are run in parallel from an overall viewpoint (and not from the viewpoint of an individual customer). A certain number of slots are used for contention resolution, while the other slots are reserved for the individual transmission of data.

1.1.2 Contention resolution

In this subsection, the basics behind the most frequently used contention resolution procedures will be discussed. To explain the idea in a clear manner, it is useful to omit all data slots and think of all slots as being contention slots. The basic principle of the contention resolution is called *ternary tree* and will be explained now in more detail.

Assume that the HE allows a certain group of customers to send a request for data slots during a certain contention slot. Each customer individually flips a three-sided coin with equal probabilities for each side and determines in which of the three mini-slots it will send its request (if it wants to send a request). After this coin-flipping, every customer sends its request accordingly. There are three possible outcomes:

1. A mini-slot is empty.
2. A mini-slot contains exactly one customer.
3. A mini-slot contains more than one customer.

For each of the mini-slots, the HE can determine after the slot is over, which of these three possible outcomes occurred. In case the mini-slot contains exactly one customer, the corresponding request can be processed and the customer will receive a message from the HE, which prescribes when to use the upstream channel for the transmission of data. In case the mini-slot contains more than one customer, the HE can only conclude that there was a collision. Collisions in different mini-slots are treated separately. For all customers involved in a certain collision, a new contention slot will be generated in the near future. This means that each of these customers has to retransmit its request in that certain (mini-)slot in the future, again using the coin-flipping procedure. Which slot this will be, is indicated by the HE, by means of broadcasting a grant just before the slot will take place. Due to the fact that there will be some time in between the transmission and the retransmission, it is allowed for each customer to update its request in the sense that it requests more data slots than in the previous request.

This procedure of processing successful requests and retransmitting unsuccessful requests continues until all requests are processed. The procedure can schematically be represented by a tree as in figure 1.4. Such a tree will be called a *contention tree.*

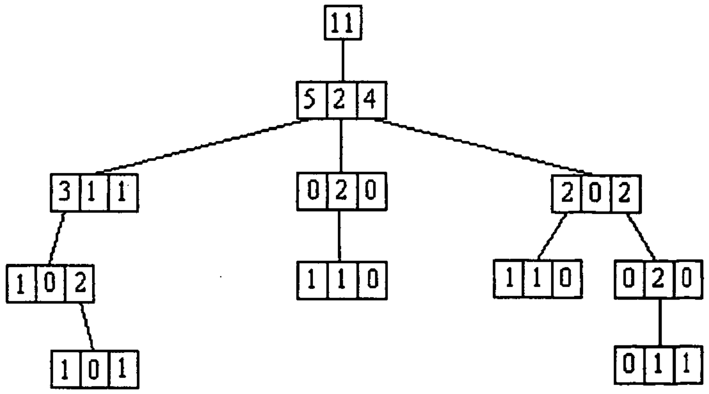

Figure 1.4: Example of a ternary contention tree.

There are several ways of processing such a tree. All nodes correspond to a contention slot and the *order* in which these slots are processed is partly free. One only has to consider the precedence relations given by the tree itself. In fact there are two commonly used *tree protocols*:

1. **Breadth-first**
   This tree protocol proceeds as follows. One starts with the root node, which is initially placed in an empty queue. This node is immediately being served, i.e. being processed. After this processing, the (possibly) formed (sub)nodes are sequentially placed in that same queue. This queue can be seen as a single-server queue that processes the node that is at the head of the queue. Once a node from the queue has been processed, the subnodes are placed at the tail of the queue and the service of the queue continues. Graphically, this results in walking through the whole tree level by level. This means that nodes that are deeper in the tree are only being processed when all the nodes at higher levels are finished.

2. **Depth-first**
   In a certain sense this tree protocol is the opposite of the breadth-first protocol. Once a node from the queue has been served, the just formed subnodes are placed at the head of the queue and thus receive priority over other nodes that are possibly already in the queue. This protocol implies that one descends directly as deep in the tree as possible.

These two protocols can be compared. In the breadth-first protocol, it takes relatively long before customers can leave the tree. The depth-first protocol generates more relatively fast departures. Though, after the first departure of a customer has taken place, the departure process resembles a relatively uniform departure process for *both* the breadth-first and depth-first protocol. The variance of the departure times in the depth-first protocol will in general be larger than that in the breadth-first protocol. Summarizing the remarks, the conclusion is that for large trees, there is not that much difference between the departure times in both protocols. The only thing is that there is for the depth-first protocol a slight advantage with respect to the mean and a sligth disadvantage with respect to the variance, compared with the breadh-first protocol. For small trees, the two protocols are nearly the same.

As an example, the tree that is depicted in figure 1.4 is analyzed using a depth-first as well as breadth-first tree protocol. For all 11 customers, the service time (departure time) is given and furthermore, the mean and variance are calculated. The results are summarized in table 1.1.

Nr.	Breadth-first	Depth-first
1	2	2
2	2	2
3	5	3
4	6	4
5	6	4
6	7	6
7	7	6
8	9	8
9	9	8
10	10	10
11	10	10
Mean	6.6364	5.7273
Variance	8.0546	8.8182

Table 1.1: Departure times of the customers in the tree of figure 1.4.

As indicated before, a customer that has transmitted a request needs feedback from the HE via the downstream channel. Either it has to retransmit its request or it is allowed to send data in the future. A customer cannot retransmit until it has received feedback concerning the last request it transmitted. Due to the involved communication, there will be some amount of time between the moment of the actual request and the moment that the customer receives the feedback. This will be called *propagation delay.* This delay requires that there is some minimal amount of time between the processing of a node and all the corresponding subnodes, if any. Consequently, contention slots can in general not be scheduled close to each other. Typically, this propagation delay takes a little less than one frame of 18 slots.

The basic idea of a ternary contention tree can be used in several ways to obtain slightly different contention resolution procedures. The analysis and finally the comparison of a couple of these procedures will be the main subject of this report. In the next section, some of these procedures are outlined. In section 1.3 an overview of the research issues and a motivation of them will be given.

1.2 Some contention resolution procedures

In this section, four different contention resolution procedures (or mechanisms) will be outlined. These procedures are all based on the idea of a ternary tree. The first two mechanisms are in fact well-known and are used in practice already. The third and fourth mechanism are closely related to each other and are new in the sense that they are not used in real access networks yet and have not been studied much either. The contents of this section is restricted to a global description of how the mechanisms work or should work in a real access network. In the next chapters, the physical mechanisms will be modelled mathematically and afterwards analyzed. Finally, the translation to a real access network will be made.

In the explanation of the contention resolution mechanisms, the transmission of individual data using data slots will be omitted. In a real access network, the main part of the slots is used for data transmission, but this is of no interest for the explanation of the contention resolution mechanism. Therefore, all mechanisms will be explained as if no data slots were in between, although in reality there are.

1.2.1 Gated mechanism

This contention resolution works as follows. Once a contention tree has been started, it is uninterruptedly being processed until termination. In the meantime, new customers may want to transmit a request, but there is no contention slot in which they are allowed to send their request because all the contention slots are dedicated to the tree completion. As soon as the tree terminates, *all* these waiting customers will enter in the first open slot. One can imagine this as if they did already arrive and have to wait (in front of a gate) for a new slot to enter the next tree.

1.2.2 Non-blocked mechanism

This contention resolution procedure is in some sense opposite to the gated mechanism. In the non-blocked mechanism, new individual users do not have to wait for the current tree to be terminated. In fact, they can enter the current tree directly when they arrive. This means that as soon as an individual user wants to send a request, it can send it in the next contention slot that occurs. So, in every slot that a tree is being processed, it may happen that there are fresh customers who also compete in that slot.

1.2.3 Static periodic arrival slot mechanism

This contention resolution mechanism has never been implemented yet. The mechanism operates as follows.

Let s be a fixed integer, with $s \geq 1$. Suppose a tree is being processed at the moment. After every s contention slots, the current tree is interrupted and one slot is dedicated to new arrivals. Thus requests that were generated during the previous period of s contention slots can be transmitted now for the first time. So every customer that "arrived" during the previous s contention slots (and the data slots in between), now enters in this so-called *arrival slot*. After this slot, some of the new customers may be lucky. They leave the system immediately. The other customers can be considered as a tree that has been processed for just one slot. This tree (if there is one) is placed in a so-called tree queue (which is an example of a so-called *distributed queue*). In the next contention slot the interrupted tree is further being processed, exactly where it was interrupted. A tree consisting of several individual customers that is placed in the tree queue will be called a *super customer*. In the gated as well as the non-blocked contention mechanism, one can think of a super customer as the currently processed tree.

Observe that the features of the mechanism strongly depend on the choice for the integer s and that in fact there is a whole class of mechanisms parameterized by s.

1.2.4 Dynamic periodic arrival slot mechanism

This mechanism has again not been implemented yet in a real access network. The mechanism resembles the previously described mechanism. In the previous mechanism, new requests were only allowed to be transmitted once per $s + 1$ contention slots. In case the system is empty, it does not seem a good idea to keep generating contention slots and wait (in the most unfavourable case) for s contention slots until generating a new arrival slot. So the periodic arrival slot mechanism will be modified to a probably superior mechanism, which will be called the dynamic periodic arrival slot mechanism. In this mechanism, an arrival slot is generated as soon as the system is empty or as soon as s contention slots have been successively generated. So in case of heavy traffic the system acts the same as the periodic arrival slot mechanism, but in case of light traffic, relatively more arrival slots are generated.

Again the features of the mechanism strongly depend on the value of s and this mechanism can also be seen as a representative of a whole class of mechanisms. It unifies in a certain sense the gated mechanism and the static arrival slot mechanism. The trade off between those two mechanisms is determined by the value of s.

1.3 Research issues

In this section it will be outlined what the (main) questions are that we will attempt to answer in this report. A motivation for investigating these questions will also be given. Furthermore, a short overview of previous related work that has already been done by others on related topics is presented.

As has been indicated already, the gated and non-blocked mechanism are known mechanisms in the sense that they have already been implemented in real access networks. Each of these mechanisms has a certain capacity, defined in terms of the maximum number of customers it can handle per time unit. Obviously, one wants this capacity to be high. Furthermore, each mechanism has its own performance characteristics in terms of the sojourn times of individual customers and particularly the mean and variance of this sojourn time. One can think of this sojourn time roughly as the time that elapses between the initial request for data slots and the ultimate assignment of exclusive data slots. Clearly, one wants this sojourn time to be small, so that customers experience short delays in their communication with the outside world.

Main question for research

In this report, a new class of contention resolution mechanisms will be introduced: the arrival slot mechanisms. In a certain sense, these mechanisms are a trade-off between the gated and non-blocked mechanism. The main research question will be whether these arrival slot mechanisms, or at least some of them, are "better" compared with the existing mechanisms. So both capacity and sojourn times are investigated. This is done first in a mathematical model. Afterwards, the obtained results will be linked to reality and finally conclusions concerning this main question will be given.

The motivation for this research question is that it is vital to maximize performance of access networks. One part of this optimization obviously includes optimization of the contention resolution part. In fact, the other parts cannot be seen in isolation from the contention resolution. Optimization of this part can be accomplished by finding a very efficient contention resolution mechanism, but also the scheduling part is important, which prescribes how contention resolution and data transmission are connected.

As indicated already, contention resolution and data transmission are run in parallel from an overall viewpoint. For instance, (on average) a fixed number c out of the 18 slots per frame is used for contention resolution, while the other slots are used for individual data traffic. A more dynamic approach would be to vary the number of contention slots per frame according to the number of requests in the data queue. So, how this actual (dynamical) scheduling of contention and data slots by the HE must be done is another interesting question for research. Studying and answering this question would be a logical follow up after the main question has been answered. At the end of this report, some attention will be paid to this second question.

In order to make a fair comparison between "old" and "new" mechanisms, some analysis will be done on the existing mechanisms: gated and non-blocked.

Furthermore, in the mathematical analysis of the mechanisms, several interesting mathematical questions arise. Sometimes these questions are further investigated, although the relevance for the actual analysis is not directly seen. Sometimes, the mathematical analysis is restricted to what is directly relevant for the primary question(s) under consideration.

We will end this section with a brief overview of related work. For a detailed overview of what has been done on random access networks up to 1985, we refer to [11]. In this special issue, there are also several articles on contention trees. The performance analysis of access networks has mainly been investigated by means of simulation, see [12].

The question how the scheduling by the HE must take place dynamically has in several contexts been investigated. It is often called *self regulation* in access networks. For more on this, see [13].

1.4 Analysis by means of queueing theory

As indicated in the previous section, the focus in this report will be on both versions of the arrival slot mechanism, but also some aspects of the other two contention resolution mechanisms are studied. All four mechanisms will be represented as mathematical queueing models. These queueing models are thoroughly studied, using both analysis and simulation.

So the four mechanisms are first translated to idealized queueing models. Of course, some aspects and characteristics are simplified in this modeling phase. The next chapter will be devoted to this modeling aspect. Not only the differences and similarities between the physical and mathematical model (for all four mechanisms) are indicated, but also the differences and similarities between the four mathematical models are discussed. The differences will be stressed and the similarities will be used several times later in this report to avoid unnecessary analysis of parts that have already been analyzed in another model.

The stability of the models will be investigated by means of an analysis. Furthermore, the mean waiting, service and sojourn times of customers will be analyzed. Also the variance of these quantities will be studied. After the analysis of the mathematical models, a simulation study will be presented. Finally, the obtained insight in those mathematical models is used to obtain results for more realistic models. Those results can eventually be used to optimize the performance of a certain access network.

Chapter 2

Queueing model descriptions

In this chapter, the four contention resolution procedures which were introduced in the previous chapter are formulated as (mathematical) queueing models. These models will be outlined and the simplifications that are made, compared with the physical system, are indicated. Furthermore, the similarities of the various models are discussed.

2.1 General outline

Super customer versus individual customers

In the physical system, there is a (finite) number of users. These users will from now on be called *individual customers*, or simply customers. In the previous chapter, the term super customer was already introduced. A super customer initially consists of a certain number of individual customers, which are grouped together. One can think of a super customer as a ternary tree. How far the tree has already evolved, determines the number of remaining individual customers in this super customer, since all customers leave individually as soon as they are "lucky".

The request process

Each individual customer sends new requests for data slots from time to time. One can think of individual data generating processes that evolve at the homes of the customers. When some data are generated at home, a customer sends a request for data slots. Once it already has a pending request that is currently involved in a contention tree, it can simply update this request to account for the extra amount of data slots needed, due to the new data that were generated. However, when the individual customer has currently no pending request, it wants to send a so-called *new* request. From that moment on, the customer waits for the first contention slot that it is allowed to compete in.

When considering all customers together, this results in an arrival process of new requests, or equivalently, new customers, since it is assumed that no customer can have more than one request at a time. A very important model assumption will be that this aggregated request process will be modelled as a *Poisson process*. This Poisson process will have a constant rate. By modelling it this way, we neglect in fact the "finiteness" of the population of users.

From now on, we will talk about the arrival of customers instead of the arrival of requests. There is a subtle difference. The moment a request arrives, does in general not exactly coincide with the moment the intention to send a request originated. There will be some latency, due to the fact that a customer has to wait for a contention slot it can compete in. The arrival of a customer is associated with the moment that the intention to send a request originates. This arrival process is consequently also a Poisson process.

The general queueing model

The queueing models that will be formulated will all be so-called single server queues. These queueing models typically have one server that serves customers, exactly one at a time. The customers in our queueing models will not be the individual customers, but just the super customers. Each super customer has a *service time*, which corresponds to the number of contention slots that are needed to complete the whole tree from the beginning until the moment that all individual customers have retransmitted their requests successfully. Observe that the service time consists only of contention slots and therefore does not correspond to the time that elapses from the beginning until the end of a tree in the physical system. We will return to this point later on. The server handles in one time slot exactly one contention slot dedicated to the contention resolution corresponding to the current super customer. Which individual customers are involved depends on at least three things: the tree protocol, the development of the contention resolution until now and the contention resolution mechanism that is used.

Contention slots only

In accordance with the slotted operation of the physical system, the time axis of the queueing models will consequently be slotted. This results in so-called discrete-time queues. A very important assumption in all models is that there are only contention slots present. The data slots that are present in the physical system are simply left out and it is assumed that the time between two consecutive (contention) slots is zero. The reason for this simplification is that it makes the analysis of the contention resolution procedures easier and more transparent. Afterwards, we return to the analysis of a more realistic model with data slots present in between contention slots.

A consequence of this simplification is that the propagation delay, which implied a certain minimal distance (in time) between two consecutive contention slots is neglected also. The propagation delay and its consequences will also be reconsidered when a model with data slots in between the contention slots is formulated and discussed.

In the next sections the mathematical models will be formulated more precisely.

2.2 Model formulation

In this section, the four contention resolution mechanisms are formulated as queueing models. As indicated before, the gated mechanism as well as the non-blocked mechanism are well-known already. However, the exact mathematical model formulation varies, due to different model assumptions and modeling slightly adjusted variants of both mechanisms. Because of the fact that we finally want to make a fair comparison between all four contention resolution procedures, we will formulate all four mechanisms in this section.

Most of the analysis that will follow in the next chapters will concern the arrival slot mechanisms. These mechanisms are also new, compared with the gated and non-blocked mechanism. In fact they can be seen as a sort of hybrid form of both existing mechanisms, unifying the advantages of both mechanisms. Therefore, the model desciptions of both arrival slot mechanisms can be seen as the most important ones.

2.2.1 The gated mechanism

This queueing model can be described as follows. A super customer is processed by a single server. During this processing, customers that are involved in the tree leave the system. Furthermore, individual customers arrive with a constant rate μ per slot according to a Poisson process. Without loss of generality it can be assumed that all customers that arrive during the same time slot arrive simultaneously, just before the end of a time slot. Once they have arrived, they wait in a waiting room until the gate that is placed behind this waiting room opens. The gate opens immediately after the contention slot that was necessary to finish service of the super customer that was being served. All customers that were waiting in the room pass the gate and the gate closes immediately. In case of an empty waiting room, the gate remains open until customers have passed the gate.

2.2.2 Non-blocked mechanism

This queueing model can be described without the waiting room that is present in the gated mechanism. However, it will turn out at the end, that it is more convenient to think of this mechanism with a waiting room, although actually no waiting takes place in this model at all.

Individual customers again arrive according to a Poisson process with a constant rate μ per slot and enter the waiting room simultaneously at the end of every slot. They immediately pass the waiting room and join the current super customer or tree that is being processed. What's more, they directly compete in the next contention slot dedicated to this super customer. If such an individual customer is lucky, it leaves the system immediately after this one contention slot. It is clear that if there is no currently processed super customer, it is formed by the group of new arrivals.

2.2.3 The static arrival slot mechanism

Consider the mechanism with exactly one arrival slot per $s+1$ contention slots (including the arrival slot). Due to the periodicity of the mechanism, assume without loss of generality that arrival slots are generated at $t = 0, s+1, 2s+2, \ldots$. Individual customers enter the waiting room simultaneously at the end of a time slot. The arrival process is Poisson with constant rate μ per slot. This time there is again a gate behind the waiting room. At $t = 0, s+1, \ldots$ the gate opens and lets all waiting customer pass through. After that, the gate closes immediately until a time period of $s+1$ slots has elapsed. So notice that the opening of the gate does in general not correspond to the completion of a super customer, as we saw in the gated mechanism. The moment of the gate opening does only depend on time.

Just after the gate opened and closed, an arrival slot is generated. This means that the individual customers that have just entered can exclusively compete in this slot. Service of the currently processed super customer is interrupted during this arrival slot. Lucky customers may leave the system directly after the arrival slot. But what happens to the other (unlucky) customers? These customers, which can be seen together as a super customer which has already been served for exactly one slot, are placed as a super customer (or tree) in the so-called *tree queue*. This is typically a first-come first-served queue. The super customers in this queue are waiting for the server to be served. This service clearly occurs during the contention slots, excluding the arrival slots.

After the arrival slot has passed, service of the interrupted super customer resumes. In case there was no interrupted super customer, the first super customer in the tree queue is taken into service. In case of an empty tree queue, as indicated already, the server waits s slots for the gate to open.

2.2.4 The dynamic arrival slot mechanism

This mechanism is closely related to the static arrival slot mechanism. Therefore the description of the queueing model will be based on the description in the previous subsection. The difference between both models lies only in the moments that the gate opens and consequently in the moments that an arrival slot is generated, because these two moments are coupled. The rest of the model is exactly the same.

In the static model, gate openings were completely time-driven. In this mechanism, gate openings are in addition governed by the state of the system behind the gate: the server as well as the tree queue. In the dynamic model, the gate opens when either one of the following two conditions is satisfied:

1. There have been s successive contention slots dedicated to the service of super customers that were originally in the tree queue. Or equivalently: there was no arrival slot generated during the previous s contention slots.

2. The server is idle since there is no super customer to be served.

The idea behind this is that waiting for several slots, while in fact the waiting room may already be filled with new customers seems not a good idea.

2.3 Sojourn times of individual customers

In this section, the sojourn times of individual customers in all four models are discussed. In all models, the sojourn times consist of several components. Some components are the same for all models, others are exclusively connected to one model. The "structure" of all sojourn times is discussed, so that the analysis and final comparison of the models can take place in a structured and clear way.

A first decomposition of the sojourn time is made by defining the waiting time and the service time of an individual customer. Together they form exactly the sojourn time. The definitions will be general, so that they are valid for all mechanisms.

Consider an arbitrary individual customer. Its service time is defined as the number of contention slots that are used to finish the contention resolution of the corresponding super customer. This number of slots is counted from the moment that the individual customer is part of the super customer.

The waiting time is the remaining part of the sojourn time. This waiting time will again be split up into two components. The first component corresponds to the waiting time that an individual customer experiences in the waiting room. This time corresponds to the number of slots that elapse between the desire to request and the first request itself. The second part of the waiting time is defined as the remaining part of the waiting time and corresponds to the waiting time that an individual customer experiences in the tree queue.

Now, the various parts of the individual sojourn time are discussed for all four mechanisms. For the static and dynamic arrival slot mechanism the structure of the several parts is the same, only the stochastical behaviour differs.

First component of the waiting time (waiting room time)
The first component of the waiting time corresponds to the (waiting) time that elapses between the moment a request arises at the home of the customer and the first contention slot in which it can be transmitted:

1. In the gated model this can take a very long time. In fact, there is no upper bound.

2. In the non-blocked model this takes zero slots. An important remark is that in a real situation, this waiting time is in general not zero, due to the data slots (which occupy time) in between two contention slots.

3. In the static arrival-slot mechanism this takes at most s slots.

4. In the dynamic arrival-slot mechanism this takes at most s slots. Compared to the static model, this component will on average be smaller, at least in a stable queueing system.

Second component of the waiting time (tree queue time)
Second, a customer has to wait for the tree he participates in to be processed. This waiting takes place in the virtual tree queue. For the gated as well as the non-blocked mechanism, the tree-queue does not exist. Therefore, for the second component of this waiting time, the following holds:

1. In the gated mechanism this waiting time is zero.

2. In the non-blocked mechanism this time is also zero.

3. In the static arrival slot mechanism this waiting time can be zero if the customer is lucky, but in general this time is positive. It is a matter of definition whether the arrival slot in which the tree is formed is included in the waiting time. In our definition, the arrival slot is seen as a part of the service time and not as a part of the waiting time.

4. The situation for the dynamic mechanism is equal to that of the (static) periodic arrival slot mechanism.

Service time (tree time)
Summarizing, the service time can be seen as the total time that a customer "stays" in the tree that it is involved in, only counting the slots that are dedicated to the tree. So interrupting arrival slots are not counted as service time. Observe that this depends in general on the tree protocol. We can make the following observations:

1. In the gated mechanism this service time is on average not relatively short or long, compared with the other mechanisms. It clearly depends mainly on the initial number of customers in a tree and in general this varies considerably over time.

2. In the non-blocked mechanism the average service time is relatively long compared with the others. Furthermore, there will be a large variation, due to the fact that there will be many short service times (the lucky customers) as well as many long service times.

3. In the static arrival slot mechanism the variation in service times will be relatively small. This is due to the fact that the number of initial customers in a tree does not vary that much. The average service time is dependent on the value of s. An important remark is that this service time is in general split up over two periods. It starts with the arrival slot. Then the service is interrupted and continued when the super customer leaves the tree queue.

4. The situation for the dynamic mechanism is similar to that of the (static) periodic arrival slot mechanism, with as only difference that the number of initial customers is smaller when the system is empty. As a result, the service time will also be slightly smaller in that case.

Summarizing, we can make a schematic picture, representing the structure of the sojourn times for all four mechanisms. We also tried to give an indication of the relative average length of all parts, but this clearly depends on so many things, that it is only a rough indication. The picture is shown in figure 2.1.

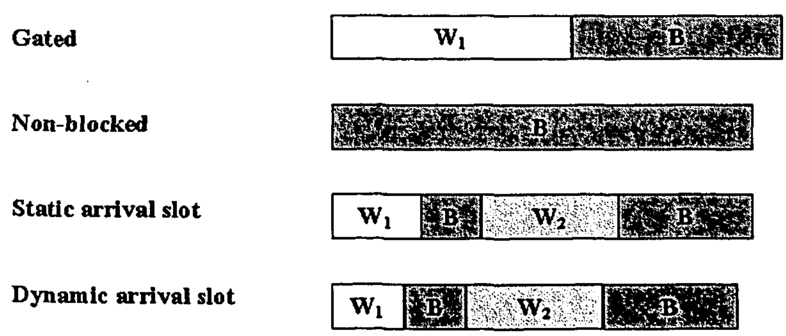

Figure 2.1: Structure of the sojourn times for all mechanisms.

Chapter 3

The static arrival slot mechanism

3.1 Introduction

In this chapter, we make a start with the analysis of the static arrival slot mechanism. This analysis continues in the next chapters.

The waiting time, service time and sojourn time of customers will be discussed. Furthermore, the capacity of the system is determined. In this chapter, it will be outlined what exactly will be analyzed and how, and in which order the analysis will proceed in the next chapters.

For the ease of presentation, we will in fact not directly analyze the static arrival slot mechanism. First a slightly different model will be studied. This fairly general model, which will be called the *periodic* $Geo|G|1$*-queue*, will first be formulated (and explained) in the next section. Subsequently it will be indicated how this more general model is related to the static arrival slot model and how it will be used to analyze the static arrival slot mechanism.

3.2 Periodic $Geo|G|1$-queue

Consider a discrete-time (slotted) queue with discrete service times and 1 server which serves the queue according to a first-come first-served discipline. Define a *frame* to be exactly s consecutive slots, $s \in \mathbb{N}$. An important remark is that this definition of frame is not the same as the earlier mentioned "frame" in the context of an access network. Anyway, both definitions are related to each other indirectly. An arrival of a (super)customer can only occur just before the first slot of every frame: with (constant) probability α there is an arrival just before the first slot of a frame and with probability $1-\alpha$ there is no arrival. In case of an arrival, the (super)customer just enters the queue and service continues normally during this first slot of every frame. The service time distribution of a super customer is allowed to be general.

3.3 A special case of the periodic $Geo|G|1$-queue

In this section, a special case of the periodic $Geo|G|1$-queue will be formulated. This special case is strongly related to the static arrival slot mechanism.

Consider the static arrival slot mechanism. Now think of this mechanism, when omitting the arrival slots. This has the consequence that arrival slots do not occupy time anymore. Lucky individual customers that leave the system directly after their first arrival slot are not seen anymore. Only super customers that are placed in the tree queue remain. The model that is formed this way can be seen as a special case of the periodic $Geo|G|1$-queue. As will turn out, every s slots, with a constant probability a super customer is placed in the tree-queue and with the complementary probability no super customer enters the queue. So the super customers act as the customers in the periodic $Geo|G|1$-queue. As indicated before, the tree queue was also a first-come first-served queue with 1 server. The service time distribution deserves some extra attention. Therefore, the next section is devoted to this distribution.

3.4 The service time distribution

In this section we focus again on the arrival slot mechanism. The general service time distribution in the $Geo|G|1$-queue will now be chosen such that it correctly fits that of an arbitrary super customer that is placed in the tree queue in the static arrival slot mechanism.

Now the service time distribution of interest will be outlined. Sometimes, we will repeat things that are already mentioned in the description of the contention tree, but this is done to keep things clear.

Consider a discrete time (slotted) queue with a frame structure. Individual customers can only arrive in the first slot of every frame which has a length of $s+1$ slots. The underlying arrival process is a Poisson process with rate μ per slot. So the effective arrival process of individual customers is a Poisson process with rate $\lambda(s)$, where $\lambda(s)$ is given by

$$\lambda(s) \quad = \quad (s+1)\mu.$$

As indicated before, each slot is divided into three mini-slots of equal length. When more than one customer arrives in a mini-slot, all these customers are grouped together. When exactly one customer arrives in a mini-slot, the customer leaves the system immediately. After each slot, the groups of customers belonging to the three mini-slots of that particular slot are grouped together as a so-called *super customer.* Obviously, in that case a super customer consists of at least 2 individual customers. When a super customer has been formed it is placed in the queue. Two examples of super customers are [3,4,2] and [0,3,2] (see figure 3.1).

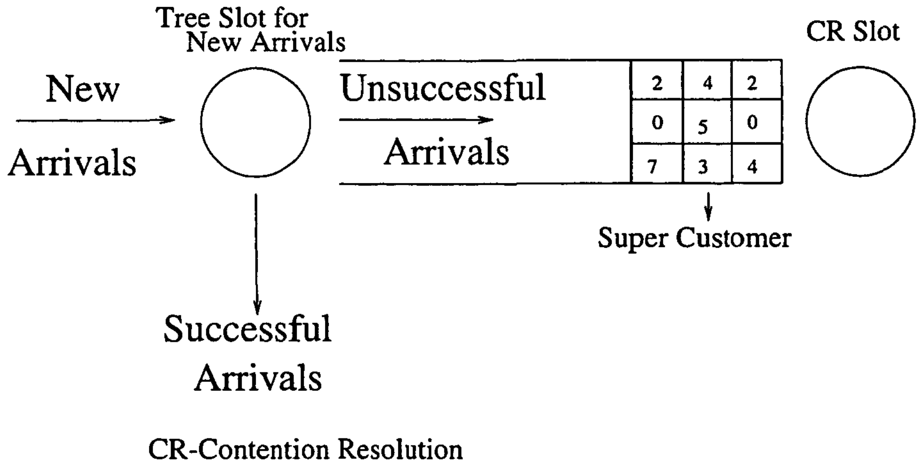

Figure 3.1: Representation of the queueing system of interest

Each super customer corresponds to the root of a so-called ternary contention tree. The service time of a super customer is just the time it takes (measured in slots) to complete this contention tree. This time is counted from the moment that the tree is placed in the so called tree queue and so the arrival slot is *not* counted as service time. The service time distribution of an arbitrary super customer in this queue is exactly the distribution that will be looked at more closely.

It is clear that with some probability α, a super customer is formed. When the first slot of every frame is left out, a special case of the $Geo|G|1$-queue arises.

It is clear that the service time distribution will depend on the actual value of $\lambda(s)$. Hence the random variable with the just described distribution will be denoted by $B_{\lambda(s)}$ and the corresponding generating function will be denoted by $B_{\lambda(s)}(z)$. For completeness $W_{\lambda(s)}(z)$ is defined as the generating function of the waiting time in this case. The probability that a super customer is placed in the queue is called the arrival probability. This probability depends on the value of $\lambda(s)$ and therefore it is denoted by $\alpha_{\lambda(s)}$.

3.5 Overview of the analysis approach

So the approach for the mathematical analysis of the static arrival slot mechanism, is to analyze at first a sligthly simpler queueing system than the arrival slot mechanism. Simpler in the sense that the arrival slots in between are left out, what implies that service is not interrupted anymore and this makes waiting and service time analysis more straightforward. So, stressing the difference between both models again, the special case of the $Geo|G|1$-queue, which will be the actual name for the model, is nearly the same as the static arrival slot mechanism, but the only difference is that arrivals do not occur in an arrival slot, but occur *instantaneously* and therefore do not occupy a time slot.

In the end, when the analysis is done for the $Geo|G|1$-queue, some chapters will be devoted to a general approach for *translating* the results of the analysis of this queueing system to results for the original static arrival slot mechanism. This translation will obviously take into account that service is in fact interrupted every s slots, resulting in larger waiting and service times than found in the $Geo|G|1$-queue. Furthermore, it will take into account the presence of lucky customers, which are invisible in the $Geo|G|1$-queue, since they are only visible in the arrival slots, which are left out in the $Geo|G|1$-queue.

After the translation approach has been outlined, the translation will also actually be executed following the developed procedure, which will eventually lead to some numerical results. Now, a more detailed description of the contents of the first part of this report, which is devoted to the analysis of the static arrival slot mechanism, follows. The contents of each of the chapters will be roughly indicated.

Waiting time analysis
In the next chapter, Chapter 4, the $Geo|G|1$-queue will be analyzed first in the most general way, using a generating function approach. The focus will be on the waiting time that a (super) customer experiences.

In the following chapter, Chapter 5, we will return to the special case of the $Geo|G|1$-queue, which is related to the static arrival slot mechanism. The results of Chapter 4 will be used in the analysis. To use these results, some work must be done on finding the distribution of the service time in this special case of the $Geo|G|1$-queue, and more specifically on finding the generating function of this distribution, which is not trivial. This gives ultimately results for the second component of the waiting time.

Service time analysis
In Chapter 6, the service time of individual customers in the special case of the $Geo|G|1$-queue is considered. The depth-first case is analyzed. An expression for the generating function of the service time distribution will be derived. This will eventually lead to expressions for the mean and variance of the service time, which are useful from a numerical point of view.

Translation
In Chapter 7, the method for exact translation from the periodic $Geo|G|1$-queue to the static arrival slot mechanism will be outlined in general. This method again makes use of generating functions. Furthermore, some approximation methods will be introduced. Afterwards, the translation method is applied to obtain expressions for both mean and variance of the waiting and sojourn times of individual customers.

Capacity analysis
In Chapter 8, a capacity analysis is done for the static arrival slot mechanisms. Furthermore, this analysis is applied to assess the capacity for several mechanisms.

Numerical results
Chapter 9 is devoted to the presentation of numerical results on the waiting

and service time, obtained via the analysis of the previous chapters. These results are validated by means of simulation. The focus will be on the arrival slot mechanism with $s = 2$, which is both illustrative and numerically tractable. Furthermore, this mechanism has the highest capacity as will turn out later.

Simulation study
After the analytic part of this report, a simulation study follows. This simulation is discussed in Chapter 14. In this chapter the four contention resolution procedures are compared on a couple of performance measures as capacity, waiting time, service time and sojourn time.

Link to a real access network
After these chapters, which concentrate (for the sake of simplicity) only on the contention resolution, in Chapter 15 the transmission of data by means of data slots is discussed again. Among other things, we study in this chapter in which way the (positive) results concerning the arrival slot mechanism(s), obtained in the previous chapters, can be used in a more realistic context.

Chapter 4

Waiting time analysis for the periodic $Geo|G|1$-queue

In this chapter, we will analyze the periodic $Geo|G|1$-queue. This queueing model has already been outlined in Section 3.2. More specifically, we will analyze the waiting time of (super)customer. A generating function approach will be used.

Several related queueing models have already been investigated by others. As far as we have found, this model has not been analyzed yet. The *periodic* aspect of the arrival process of this queueing model is different from all other queueing models that have already been studied. For an overview of several related models, we refer to [14] and [15]. In [15], Takagi starts with the analysis of several $Geo|G|1$- and Batch Arrival $Geo|G|1$ queueing models. He distinguishes for instance between *late arrival* and textitearly arrival models. In late arrival models, customers arrive at the end of a slot, where in early arrival models customers arrive at the beginning of a slot. Besides waiting time and queue length, the input and output process of those queueing models are studied.

4.1 Notation

In this section, some notation will be introduced.

Let the random variable B denote the service time of the arrived customer and:

$$b_k = P[B = k], \qquad k = 1, 2, 3, \ldots$$

Define $B(z)$ as the generating function of the service time distribution:

$$B(z) = \sum_{k=1}^{\infty} b_k z^k, \qquad |z| \leq 1.$$

Let W be a random variable which has as distribution the steady-state distribution of the total amount of work just before the first slot of a frame in the periodic $Geo|G|1$-queue with:

$$w_k = P[W = k], \qquad k = 0, 1, 2, ...$$

Finally, define the generating function $W(z)$ corresponding to W as follows:

$$W(z) = \sum_{k=0}^{\infty} w_k z^k, \qquad |z| \leq 1.$$

4.2 Analysis of the model

In this section, an expression for the generating function of the random variable W is derived. Afterwards, it will be indicated how the, yet unknown, stationary probabilities in this expression can be found.

An important remark is that W is not defined as being a waiting time. But because of the arrival process of (super)customers, which has geometrically distributed interarrival times, the waiting time that an arbitrary (super)customer experiences is the same random variable as the total amount of work just before the first slot of a frame. This will not be proven here. For this, we refer to [16], where the BASTA (Bernoulli Arrivals See Time Averages) property is discussed. So in the following chapter W is often treated (and this is justified) as a waiting time variable.

4.2.1 An expression for $W(z)$

Consider the discrete Markov chain defined by the total amount of work at the queue just before the first slot of every frame. This total amount of work is defined as the total time it takes to complete service of all the customers that are currently in the system. A possible new arrival is not counted yet.

The stationary distribution satisfies the balance equations:

$$\begin{aligned}
w_0 &= \alpha\sum_{n=0}^{s}\sum_{k=1}^{s-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s} w_n \\
&= \alpha\sum_{n=0}^{0} b_{s+n}w_{-n} + \alpha\sum_{n=1}^{s-1} b_{s-n}w_n + (1-\alpha)w_s + \alpha\sum_{n=0}^{s-1}\sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s-1} w_n, \\
w_k &= \alpha\sum_{n=0}^{k} b_{s+n}w_{k-n} + \alpha\sum_{n=1}^{s-1} b_{s-n}w_{k+n} + (1-\alpha)w_{k+s}, \qquad k = 1,2,3,...
\end{aligned}$$

The balance equation for w_0 is rewritten so that the right-hand side is of nearly the same form as the right-hand side of the general balance equation for w_k for $k = 1,2,\ldots$. The general balance equation can be explained as follows. Consider all possible transitions from a state j to the current state $k \geq 1$. When the system is currently in state $k \geq 1$, the previous frame took s slots and so the total amount of work just before the beginning of that previous frame was at most $k+s$. This implies that $0 \leq j \leq k+s$. The total amount of work that arrived at the beginning of the previous frame plus j, the total amount of work that was already present, minus s must be equal to k to ensure a transition from j to k for $k \geq 1$. This explains the balance equation.

Multiplying both sides of each equation with z^k and summing all the equations for $k = 0,1,2,...$ leads to the following equation:

$$\begin{aligned}
W(z) &= \alpha\sum_{k=0}^{\infty}\sum_{n=0}^{k} b_{s+n}w_{k-n}z^k + \alpha\sum_{k=0}^{\infty}\sum_{n=1}^{s-1} b_{s-n}w_{k+n}z^k + (1-\alpha)\sum_{k=0}^{\infty} w_{k+s}z^k + \\
&\quad \alpha\sum_{n=0}^{s-1}\sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s-1} w_n \\
&= z^{-s}\alpha\sum_{n=0}^{\infty} b_{s+n}z^{n+s}\sum_{k=n}^{\infty} w_{k-n}z^{k-n} + z^{-s}\alpha\sum_{n=1}^{s-1} b_{s-n}z^{s-n}\sum_{k=0}^{\infty} w_{k+n}z^{k+n} + \\
&\quad z^{-s}(1-\alpha)\left(W(z) - \sum_{k=0}^{s-1} w_k z^k\right) + \alpha\sum_{n=0}^{s-1}\sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s-1} w_n \\
&= W(z)z^{-s}\alpha\left(B(z) - \sum_{n=0}^{s-1} b_n z^n\right) + W(z)z^{-s}\alpha\sum_{n=1}^{s-1} b_{s-n}z^{s-n} -
\end{aligned}$$

$$
\begin{aligned}
& z^{-s}\alpha\sum_{n=1}^{s-1} b_{s-n}z^{s-n}\sum_{k=0}^{n-1} w_k z^k + z^{-s}(1-\alpha)\left(W(z)-\sum_{k=0}^{s-1} w_k z^k\right) + \\
& \alpha\sum_{n=0}^{s-1}\sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s-1} w_n \\
= \;& W(z)z^{-s}\alpha B(z) - z^{-s}\alpha\sum_{k=0}^{s-2} w_k z^k \sum_{n=k+1}^{s-1} b_{s-n}z^{s-n} + \\
& z^{-s}(1-\alpha)\left(W(z)-\sum_{k=0}^{s-1} w_k z^k\right) + \alpha\sum_{n=0}^{s-1}\sum_{k=1}^{s-1-n} b_k w_n + (1-\alpha)\sum_{n=0}^{s-1} w_n.
\end{aligned}
$$

This yields the following expression for $W(z)$:

$$
\frac{-\alpha\sum_{k=0}^{s-2}\sum_{n=k+1}^{s-1} b_{s-n}w_k z^{k+s-n} - (1-\alpha)\sum_{k=0}^{s-1} w_k z^k + z^s\left(\sum_{k=0}^{s-1} w_k\left(\left(\sum_{n=1}^{s-1-k}\alpha b_n\right)+(1-\alpha)\right)\right)}{z^s - \alpha B(z) - (1-\alpha)}.
$$

After some algebraic manipulation, this expression can be written in a slightly different form:

$$
W(z) = \frac{\alpha\sum_{k=0}^{s-1}\sum_{n=1}^{s-k-1} b_n w_k (z^s - z^{k+n}) + (1-\alpha)\sum_{k=0}^{s-1} w_k (z^s - z^k)}{z^s - \alpha B(z) - (1-\alpha)}.
$$

In this expression one can see that for $z = 1$ the numerator equals zero. Therefore the numerator can be factorized leading to the following expression for $W(z)$:

$$
W(z) = \frac{(z-1)\left(\alpha\sum_{k=0}^{s-1}\sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} z^i + (1-\alpha)\sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} z^i\right)}{z^s - \alpha B(z) - (1-\alpha)}.
$$

In this expression there are still s unknowns: $w_0, w_1, \ldots, w_{s-1}$. In the next subsection, it will be explained how these yet unknown probabilities can be found, but first an analytic expression for $\mathbb{E}W = W'(1)$ will be derived.

$$\begin{aligned}
\mathbb{E}W &= W'(1) \\
&= \lim_{\delta\to 0} \frac{W(1+\delta) - W(1)}{\delta} \\
&= \lim_{\delta\to 0} \frac{W(1+\delta) - 1}{\delta} \\
&= \lim_{\delta\to 0} \frac{\dfrac{\delta(\alpha \sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} (1+\delta)^i + (1-\alpha) \sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} (1+\delta)^i)}{(1+\delta)^s - \alpha B(1+\delta) - (1-\alpha)} - 1}{\delta} \\
&= \ldots \\
&= \frac{\alpha(\sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} i) + (1-\alpha)(\sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} i) - (\frac{s(s-1)}{2} - \alpha \frac{B''(1)}{2})}{s - \alpha B'(1)}.
\end{aligned}$$

4.2.2 Determination of $w_0, w_1, \ldots, w_{s-1}$

We know that $W(z)$ is well-defined for z with $|z| \leq 1$. It will be shown that the denominator of the final expression for $W(z)$ has s zeros $z_1, z_2, \ldots, z_s$ that lie on or within the (complex) unit circle. For these zeros $W(z)$ is well-defined and consequently also the numerator has to be zero for these $z_1, z_2, \ldots, z_s$. This gives s equations. It can easily be shown that for $z = 1$ both the numerator and denominator always equal zero, independent of the values of the w_k's and the b_k's. So in fact there are at most $s-1$ equations that give information about the w_k's. But there is also a normalization condition: $W(1) = 1$. When we assume that there are indeed $s-1$ different zeros which lead to $s-1$ different equations, we do have, together with the boundary condition, enough equations to solve for $w_0, w_1, \ldots, w_{s-1}$. It might happen that there are less than $s-1$ different zeros, but in that case some zeros have a multiplicity that is larger than 1. In this case the numerator must also have that zero with the same multiplicity, because $W(z)$ exists and is uniquely determined by the balance equations. This gives us the right number of equations.

By using an indirect argument, it can be shown that these equations are also sufficient to solve for the unknown probabilities. Assume that the obtained set of equations is dependent. This implies that, according to the set of equations, there will be more than one solution. This contradicts the uniqueness of $W(z)$ and the fact that all information including all boundary conditions are used.

As a result, it will be possible to solve for the unknown w_k's with the obtained equations.

Now it will be shown that the following equation has s zeros $z_1, z_2, \ldots, z_s$ that lie on or within the (complex) unit circle. Define:

$$\begin{aligned} f(z) &= -(\alpha B(z) + (1-\alpha)), \\ g(z) &= z^s. \end{aligned}$$

Because $B(z)$ converges for all $|z| \leq 1$, it follows that the convergence radius of $B(z)$, denoted by ρ, satisfies $\rho > 1$. Say $\rho := 1 + \beta, \qquad \beta > 0$.
Define:

$$C \; : \; \{z \in \mathbb{C} \mid |z| = 1 + \varepsilon\}, \qquad 0 < \varepsilon < \beta.$$

The stability condition for the queueing system gives us:

$$\alpha \mathbb{E} B = \alpha B'(1) < s.$$

Because

$$\left(\frac{d}{dz}(\alpha B(z) + 1 - \alpha)\right)_{z=1} = \alpha B'(1) < s = \left(\frac{d}{dz}(z^s)\right)_{z=1}$$

and

$$\alpha B(1) + 1 - \alpha = 1^s$$

it follows that for all z on C, provided that ε is chosen small enough, the following inequality holds:

$$\alpha B(|z|) + 1 - \alpha < |z|^s.$$

So for all z on C the following inequality holds:

$$|f(z)| \leq \alpha B(|z|) + 1 - \alpha < |z^s| = |g(z)|.$$

This means that $|f(z)| < |g(z)|$ for all z on C. Because both $f(z)$ and $g(z)$ are analytic on and within C, Rouché's theorem, see [4], tells us that $g(z)$ and $g(z) + f(z)$ have the same number of zeros (according to their multiplicity) within C. $g(z)$ has exactly one zero within C which has multiplicity s and therefore it follows that $g(z) + f(z)$ also has s zeros within C. Because we can take ε arbitrarily close to zero, it must be the case that $g(z) + f(z)$ has s zeros on or within the unit circle.

We need to find the s zeros numerically. From a numerical point of view, it is more convenient to solve s equations with exactly one root, than to solve 1 equation which has exactly s roots. Therefore, a method will be presented to find these s zeros in a numerically efficient way. This method is in general only valid for $\alpha < \frac{1}{2}$.

To find these s zeros numerically, we can use the following. Consider

$$\phi(z) = \sqrt[s]{\alpha B(z) + 1 - \alpha}.$$

We are interested in finding the roots of the equations

$$z = \sigma_i \phi(z), \quad i = 1, \ldots, s,$$

with σ_i satisfying

$$\sigma_i^s = 1.$$

We will prove that for each σ_i, $\sigma_i\phi(z)$ is a contraction for all z on the complex unit disc. Because of the fact that we are interested in solving the equations $z = \sigma_i\phi(z)$, we can use successive substitution to find the root of each equation exponentially fast. For more details on this approach, see [9]. So for each i the iteration scheme becomes

$$z^{(n+1)} = \sigma_i\phi(z^{(n)}), \quad n = 0, 1, \ldots,$$

with $z^{(0)} = 0$.

The only thing left to prove is that $\sigma_i\phi(z)$ is indeed a contraction on the disc $|z| \leq 1$. For $|z| \leq 1$ we have that

$$|\sigma_i\phi(z)| \leq |\phi(|z|)| \leq \phi(1) = 1.$$

Therefore, $\sigma_i\phi(z)$ maps the unit disc into itself. We can write the following for z_1, z_2 on the unit disc:

$$\begin{aligned} |\sigma_i\phi(z_1) - \sigma_i\phi(z_2)| &= \left| \int_{t=0}^{t=1} \phi'(z_2 + t(z_1 - z_2))(z_1 - z_2)dt \right| \\ &\leq |z_1 - z_2| \max_{0 \leq t \leq 1} |\phi'(z_2 + t(z_1 - z_2))| . \end{aligned}$$

Here we use that $\alpha < \frac{1}{2}$, so that $\phi(z)$ is analytic on the unit disc. Furthermore we have that for all $|z| \leq 1$:

$$\begin{aligned} |\phi'(z)| &\leq \phi'(|z|) \leq \phi'(1) \\ &= \frac{1}{s}\alpha B'(1) \leq 1, \end{aligned}$$

where the last inequality sign follows from the stability condition. Consequently, we obtain the following:

$$|\sigma_i\phi(z_1) - \sigma_i\phi(z_2)| \leq \phi'(1)\,|z_1 - z_2|\,.$$

This means that for each σ_i, $\sigma_i\phi(z)$ is indeed a contraction on the complex unit disc.

Chapter 5

A special case of the periodic $Geo|G|1$-queue

5.1 Introduction

In the previous chapter the periodic $Geo|G|1$-queue was analyzed. The generating function of the waiting time in this model, denoted by $W(z) = \sum_{k=0}^{\infty} w_k z^k$, was derived. In this chapter, we focus again on the arrival slot mechanism. The general service time distribution in the $Geo|G|1$-queue will now be chosen such that it is closely related to the static arrival slot mechanism.

For completeness, the resulting expression for $W(z)$ in the general case was found to be the following:

$$W(z) = \frac{(z-1)\left(\alpha \sum_{k=0}^{s-1} \sum_{n=1}^{s-k-1} b_n w_k \sum_{i=k+n}^{s-1} z^i + (1-\alpha)\sum_{k=0}^{s-1} w_k \sum_{i=k}^{s-1} z^i\right)}{z^s - \alpha B(z) - (1-\alpha)}.$$

In this equation, there are still some unknown w_k's. These can be found if the zeros of the denominator that lie on or within the complex unit circle can be found.

In the previous chapter the distribution of the service time of an arrived customer was supposed to be general. In this chapter we consider one special distribution for these service times, which is closely related to the arrival slot mechanism. A numerical method for finding the zeros on or within the unit circle for this special choice of $B(z)$ will be presented. This will be done by finding an accurate approximation for $B(z)$ with $|z| \leq 1$. Furthermore, it is shown how one can determine the probabilities b_k, $k = 0, \ldots, s-1$ in the numerator of the expression for $W(z)$.

5.2 Derivation of a functional equation

In this section a 2-dimensional functional equation will be derived. This equation will be investigated in later sections in this chapter.

Let $\tilde{B}(n)$ be a random variable, which has as distribution the tree length distribution of a tree which starts with n customers, $n \geq 2$. One can get a sort of recursive formula for the distribution of the $\tilde{B}(n)$'s. The actual number of slots to complete a tree can be seen as 1+number of slots to complete the 3 trees that have been formed in that initial slot. So, for all $n \geq 2$, the following recursion holds:

$$P(\tilde{B}(n) = k) = \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1 + n_2 + n_3 = n}} \xi(n_1, n_2, n_3) P(\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) = k - 1),$$
$$k = 1, 2, \ldots,$$

with:

$$\begin{aligned} \tilde{B}(0) &= \tilde{B}(1) = 0, \\ P(\tilde{B}(n) = 0) &= 0, \qquad n = 2, 3, \ldots, \\ \xi(n_1, n_2, n_3) &= \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n. \end{aligned}$$

Multiplying both sides of the recursive equation with z^k and substituting the expression for $\xi(n_1, n_2, n_3)$ yields:

$$P(\tilde{B}(n) = k) z^k = z \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n P(\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) = k - 1) z^{k-1}.$$

Summing this equation over $k = 1, 2, \ldots$ and introducing the following generating function:

$$\tilde{B}_n(z) = \sum_{k=0}^{\infty} P(\tilde{B}(n) = k) z^k, \quad |z| \leq 1,$$

finally yields:

$$\tilde{B}_n(z) = z \sum_{k=1}^{\infty} \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n P(\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) = k-1) z^{k-1},$$

$$3^n \frac{\tilde{B}_n(z)}{n!} = z \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1+n_2+n_3=n}} \frac{\tilde{B}_{n_1}(z)}{n_1!} \frac{\tilde{B}_{n_2}(z)}{n_2!} \frac{\tilde{B}_{n_2}(z)}{n_2!}.$$

Introducing $f_n(z) = \dfrac{\tilde{B}_n(z)}{n!}$ for $n \geq 0$ the last equation becomes:

$$3^n f_n(z) = z \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1+n_2+n_3=n}} f_{n_1}(z) f_{n_2}(z) f_{n_3}(z), \qquad n \geq 2.$$

This equation can be used to obtain closed-form expressions for the $f_n(z)$'s. In Appendix B closed-form expressions for $f_n(z)$ for $n = 1, \ldots, 6$ are given. In the next chapter, we are interested in finding $\mathbb{E}B(n)$ for small n. This can be done by using that $\mathbb{E}B(n) = n! f_n'(1)$. A sligthly more direct approach is to obtain a recursion for the $\mathbb{E}B(n)$'s by differentiating the above recursive equation for $\tilde{B}_n(z)$ with respect to z and substituting $z = 1$. Using that $\tilde{B}_n'(1) = \mathbb{E}B(n)$, one obtains a recursion for $\mathbb{E}B(n)$, $n \geq 2$. For completeness, values for $\mathbb{E}B(n)$ are also given in Appendix B.

Multiplying both sides of the recursive equation by x^n we get:

$$(3x)^n f_n(z) = z \sum_{\substack{n_1, n_2, n_3 \geq 0 \\ n_1+n_2+n_3=n}} f_{n_1}(z) f_{n_2}(z) f_{n_3}(z) x^n, \qquad n \geq 2.$$

Summing these equations over $n \geq 2$ and introducing the following generating function:

$$F(x,z) := \sum_{n=0}^{\infty} f_n(z)x^n,$$

which is defined for all $z \leq 1$ and all x, we get the following result:

$$F(3x,z) - 3xf_1(z) - f_0(z) = z\left(F^3(x,z) - f_0^3(z) - 3xf_0^2(z)f_1(z)\right).$$

Furthermore, we know:

$$\begin{aligned} f_0(z) &= \frac{\tilde{B}_0(z)}{0!} = 1, \\ f_1(z) &= \frac{\tilde{B}_1(z)}{1!} = 1. \end{aligned}$$

Substituting this in the previous equation, we finally get the following functional equation:

$$\begin{aligned} F(3x,z) - 3x - 1 &= z(F^3(x,z) - 1 - 3x), \\ F(3x,z) &= zF^3(x,z) + (1+3x)(1-z). \end{aligned}$$

This equation will be thoroughly investigated in sections 5.6, 5.7 and 5.8. After these sections, some numerical results for the mean waiting time in the arrival slot mechanism with $s = 2$ are presented for a range of arrival rates.

5.3 A relation between $F(x,z)$ and $B_{\lambda(s)}(z)$

Remember that $B_{\lambda(s)}$ denotes the random variable that represents the service time distribution of an arbitrary super customer. Now we return to the arrival slot mechanism. Define $\pi_{\lambda(s)}(n)$ as the probability that an arbitrary batch of customers that arrives has size n, i.e. consists of n customers, given that a super customer is formed, $n \geq 2$. Then $B_{\lambda(s)}$ is distributed as follows:

$$B_{\lambda(s)} = \begin{cases} \tilde{B}(2) & \text{with probability } \pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3), \\ \tilde{B}(3) & \text{with probability } \frac{1}{7}\pi_{\lambda(s)}(3), \\ \tilde{B}(n) - 1 & \text{with probability } \pi_{\lambda(s)}(n). \qquad n \geq 4. \end{cases}$$

$\pi_{\lambda(s)}(3)$ is the probability that an arbitrary batch of customers that arrives has size 3. An arbitrary batch of customers with size 3 has 27 possibilities to distribute itself over three mini-slots, where each distribution or arrangement has equal probability to occur:

- 6 of these arrangements result in no super customer, because all 3 customers are lucky.
- 3 of these arrangements result in a super customer of size 3.
- The remaining 18 arrangements lead to a super customer of size 2.

This implies that given that a super customer is formed and the original batch has size 3, with probability $\frac{3}{27-6} = \frac{1}{7}$, this results in a super customer of size 3 and with probability $\frac{6}{7}$, this results in a super customer of size 2. This explains the distribution of $B_{\lambda(s)}$ as given above.

So:

$$\begin{aligned} P(B_{\lambda(s)} = 0) &= 0, \\ P(B_{\lambda(s)} = k) &= \sum_{n=4}^{\infty} \pi_{\lambda(s)}(n) P(\tilde{B}(n) = k+1) + (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3)) P(\tilde{B}(2) = k) \\ &\quad + \frac{1}{7}\pi_{\lambda(s)}(3) P(\tilde{B}(3) = k), \qquad k = 1, 2, \ldots \end{aligned}$$

Expressions for the $\pi_{\lambda(s)}(n)$'s are as follows:

$$\begin{aligned}
\pi_{\lambda(s)}(2) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} (\frac{(\lambda(s))^2}{2!} - 3(\frac{\lambda(s)}{3})^2) \\
&= \frac{1}{6}\alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} (\lambda(s))^2, \\
\pi_{\lambda(s)}(3) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} (\frac{\lambda(s)^3}{3!} - (\frac{\lambda(s)}{3})^3) = \frac{7}{54}\alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \lambda(s)^3, \\
\pi_{\lambda(s)}(n) &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \frac{(\lambda(s))^n}{n!}, \qquad n \geq 4,
\end{aligned}$$

with arrival probability: $\alpha_{\lambda(s)} = 1 - e^{-\lambda(s)}(1 + \frac{\lambda(s)}{3})^3.$

Now $B_{\lambda(s)}(z)$ will be expressed in terms of the function $F(x, z)$.

$$\begin{aligned}
B_{\lambda(s)}(z) &= \sum_{k=1}^{\infty} P(B_{\lambda(s)} = k) z^k \\
&= \sum_{k=1}^{\infty}\sum_{n=4}^{\infty} \pi_{\lambda(s)}(n) P(\tilde{B}(n) = k+1) z^k + \sum_{k=1}^{\infty} (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3)) P(\tilde{B}(2) = k) z^k \\
&\quad + \sum_{k=1}^{\infty} \frac{1}{7}\pi_{\lambda(s)}(3) P(\tilde{B}(3) = k) z^k \\
&= \sum_{n=4}^{\infty} \frac{1}{z}\pi_{\lambda(s)}(n)(\tilde{B}_n(z) - 0) + (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3))\tilde{B}_2(z) \\
&\quad + \frac{1}{7}\pi_{\lambda(s)}(3)\tilde{B}_3(z).
\end{aligned}$$

For compactness, $\lambda(s)$ will be replaced by λ in the remaining formulas of this section. Plugging in the expressions for the $\pi_{\lambda(s)}(n)$'s we get the following:

$$\begin{aligned}
B_\lambda(z) &= \alpha_\lambda^{-1} e^{-\lambda} \left(\frac{1}{z} \left(\sum_{n=4}^{\infty} \lambda^n f_n(z) \right) + (\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3) f_2(z) + \frac{1}{9}\lambda^3 f_3(z) \right) \\
&= \alpha_\lambda^{-1} e^{-\lambda} (\frac{1}{z} \left(F(\lambda, z) - f_0(z) - \lambda f_1(z) - (\lambda)^2 f_2(z) - (\lambda)^3 f_3(z) \right) + \\
&\quad + (\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3) f_2(z) + \frac{1}{9}\lambda^3 f_3(z)).
\end{aligned}$$

One can obtain closed-form expressions for $f_n(z)$ as indicated in the previous section using:

$$
\begin{aligned}
3^n f_n(z) &= z \sum_{\substack{n_1,n_2,n_3 \geq 0 \\ n_1+n_2+n_3=n}} f_{n_1}(z) f_{n_2}(z) f_{n_3}(z), \qquad n \geq 2, \\
f_0(z) &= \frac{\tilde{B}_0(z)}{0!} = 1, \\
f_1(z) &= \frac{\tilde{B}_1(z)}{1!} = 1.
\end{aligned}
$$

So, when it is possible to determine $F(\lambda(s), z_0)$, we have found a way of determining $B_{\lambda(s)}(z_0)$ for arbitrary z_0 with $|z_0| \leq 1$. In Section 5.6 it will be explained how $B_{\lambda(s)}(z_0)$ can be approximated.

5.4 Determination of $\mathbb{E}B_{\lambda(s)}$

In the previous section we found a relation between $F(x,z)$ and $B_{\lambda(s)}(z)$. In this section $\mathbb{E}B_{\lambda(s)}$ will be expressed in terms of an infinite sum. This will be done by using the 2-dimensional functional equation, presented in Section 5.2. The final result will be used in this chapter (when determining $\mathbb{E}W_{\lambda(s)}$) as well as in the next chapter, which gives a service time analysis. Furthermore, the method that is used in this section will be used several times in chapter 6.

We obviously have the following, where λ is shorthand for $\lambda(s)$:

$$\begin{aligned}
\mathbb{E}B_\lambda &= B'_\lambda(1) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial}{\partial z}F(\lambda,1) - f'_0(1) - \lambda f'_1(1) - \lambda^2 f'_2(1) - \lambda^3 f'_3(1)\right) \\
&\quad -\alpha_\lambda^{-1}e^{-\lambda}\left(F(\lambda,1) - f_0(1) - \lambda f_1(1) - \lambda^2 f_2(1) - \lambda^3 f_3(1)\right) \\
&\quad +\alpha_\lambda^{-1}e^{-\lambda}\left((\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3)f'_2(1) + \frac{1}{9}\lambda^3 f'_3(1))\right) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial}{\partial z}F(\lambda,1) - \frac{3}{4}\lambda^2 - \frac{3}{8}\lambda^3\right) \\
&\quad -\alpha_\lambda^{-1}e^{-\lambda}\left(e^\lambda - 1 - \lambda - \frac{1}{2}\lambda^2 - \frac{1}{6}\lambda^3\right) \\
&\quad +\alpha_\lambda^{-1}e^{-\lambda}\left(\frac{1}{4}\lambda^2 + \frac{1}{6}\lambda^3 + \frac{1}{24}\lambda^3\right) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial}{\partial z}F(\lambda,1) - (e^\lambda - 1 - \lambda)\right).
\end{aligned}$$

As it will turn out, $\frac{\partial}{\partial z}F(\lambda,1)$ can be represented by an infinite sum, which is easily computed numerically. This makes that $\mathbb{E}B_\lambda$ can be evaluated numerically. We now indicate how this can be done.

Define the generating function $\tilde{F}(x)$:

$$\tilde{F}(x) := \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{B}(n)}{n!}x^n, \qquad x \in \mathbb{R}.$$

This $\tilde{F}(x)$ is related to $F(x,z)$ as follows:

$$
\begin{aligned}
\tilde{F}(x) &= \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{B}(n)}{n!} x^n \\
&= \sum_{n=2}^{\infty} \frac{\tilde{B}'_n(1)}{n!} x^n \\
&= \frac{\partial F}{\partial z}(x,1) - \frac{\tilde{B}'_0(1)}{0!} x^0 - \frac{\tilde{B}'_1(1)}{1!} x^1 \\
&= \frac{\partial F}{\partial z}(x,1).
\end{aligned}
$$

So, if we take the (partial) derivative with respect to z on both sides of the (2-dimensional) functional equation and substitute $z = 1$, this gives the following 1-dimensional functional equation in $\tilde{F}(x)$:

$$
\begin{aligned}
\tilde{F}(3x) &= F^3(x,1) + 1 \cdot 3F^2(x,1)\tilde{F}(x) - 1 - 3x \\
&= e^{3x} - 3x - 1 + 3e^{2x}\tilde{F}(x).
\end{aligned}
$$

This functional equation can be solved by iteration. In the remaining part of this report, functional equations like the one above will often occur. Therefore, in Appendix A, the more general functional equation

$$
\tilde{H}(3x) = \tilde{f}(x) + \tilde{g}(x)\tilde{H}(x)
$$

will be discussed.

$$
\begin{aligned}
\tilde{F}(3x) &= e^{3x} - 3x - 1 + 3e^{2x}\tilde{F}(x) \\
&= e^{3x} - 3x - 1 + 3e^{2x}\left(e^{x} - x - 1 + 3e^{\frac{2}{3}x}\tilde{F}(\frac{1}{3}x)\right) \\
&= \ldots
\end{aligned}
$$

Continuing in this way yields the following:

$$
\tilde{F}(3x) = \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})}\left(e^{\frac{3x}{3^i}} - \frac{3x}{3^i} - 1\right) +
$$

$$
\begin{aligned}
&\lim_{n\to\infty}\left[\prod_{i=0}^{n} 3e^{\frac{2x}{3^i}}\tilde{F}(\frac{x}{3^n})\right] \\
&= \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})}\left(e^{\frac{3x}{3^i}} - \frac{3x}{3^i} - 1\right) + \\
&\lim_{n\to\infty}\left[3^{n+1}e^{3x(1-\frac{1}{3^{n+1}})}\tilde{F}(\frac{x}{3^n})\right].
\end{aligned}
$$

The question arises if this expression converges. We will now show that this expression indeed converges.

First, $\tilde{F}(\frac{x}{3^n})$ is investigated. Because of the fact that $\tilde{F}(0) = \tilde{F}'(0) = 0$, the following holds:

$$
\tilde{F}(\frac{x}{3^n}) = \frac{x^2}{3^{2n}}\frac{\tilde{F}''(0)}{2} + \mathcal{O}\left(\frac{x^3}{3^{3n}}\right).
$$

This implies that the limit, that appears in the expression for $\tilde{F}(3x)$ converges.

For given x, $\left(e^{\frac{3x}{3^i}} - \frac{3x}{3^i} - 1\right)$ is of order $\mathcal{O}\left(\frac{1}{3^{2i}}\right)$ as can be verified by Taylor expansion of $e^{\frac{3x}{3^i}}$. This implies that the summand is of order $\mathcal{O}\left(\frac{1}{3^i}\right)$, which means that the sum converges geometrically fast as $i \to \infty$.

This last summation formula will be used later in this chapter to determine $\mathbb{E}B_\lambda$ for which the final expression becomes:

$$
\begin{aligned}
\mathbb{E}B_\lambda &= B'_\lambda(1) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\tilde{F}(\lambda) - (e^\lambda - 1 - \lambda)\right).
\end{aligned}
$$

5.5 Determination of $\mathbb{E}B^2_{\lambda(s)}$

In the previous section we presented a method to determine $B'_{\lambda(s)}(1) = \mathbb{E}B_{\lambda(s)}$. In this section the focus is on determining an expression for $B''_{\lambda(s)}(1)$, because this quantity is needed for the evaluation of $\mathbb{E}W_{\lambda(s)}$. We start again from the 2-dimensional functional equation presented in Section 5.2.

We obviously have the following, where λ is shorthand for $\lambda(s)$:

$$\begin{aligned}
B''_\lambda(1) &= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial^2}{\partial z^2}F(\lambda,1) - f''_0(1) - \lambda f''_1(1) - \lambda^2 f''_2(1) - \lambda^3 f''_3(1)\right) \\
&\quad -2\alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial}{\partial z}F(\lambda,1) - f'_0(1) - \lambda f'_1(1) - \lambda^2 f'_2(1) - \lambda^3 f'_3(1)\right) \\
&\quad 2\alpha_\lambda^{-1}e^{-\lambda}\left(F(\lambda,1) - f_0(1) - \lambda f_1(1) - \lambda^2 f_2(1) - \lambda^3 f_3(1)\right) \\
&\quad +\alpha_\lambda^{-1}e^{-\lambda}\left((\frac{1}{3}\lambda^2 + \frac{2}{9}\lambda^3)f''_2(1) + \frac{1}{9}\lambda^3 f''_3(1)\right) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial^2}{\partial z^2}F(\lambda,1) - \frac{3}{4}\lambda^2 - \frac{21}{32}\lambda^3\right) \\
&\quad -2\alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial}{\partial z}F(\lambda,1) - \frac{3}{4}\lambda^2 - \frac{3}{8}\lambda^3\right) \\
&\quad +2\alpha_\lambda^{-1}e^{-\lambda}\left(F(\lambda,1) - 1 - \lambda - \frac{1}{2}\lambda^2 - \frac{1}{6}\lambda^3\right) \\
&\quad +\alpha_\lambda^{-1}e^{-\lambda}\left((\frac{1}{3}\lambda^2 + \frac{1}{6}\lambda^3) + \frac{7}{96}\lambda^3\right) \\
&= \alpha_\lambda^{-1}e^{-\lambda}\left(\frac{\partial^2}{\partial z^2}F(\lambda,1) - 2\frac{\partial}{\partial z}F(\lambda,1) + 2(e^\lambda - 1 - \lambda) + \frac{1}{12}\lambda^2\right).
\end{aligned}$$

As we have already seen, $\frac{\partial}{\partial z}F(\lambda,1)$ can be represented by an infinite sum. Thus we are now interested in finding an expression for $\frac{\partial^2}{\partial z^2}F(\lambda,1)$ which can easily be computed. We will show how this can be done.

Define the following function $\tilde{C}(x)$:

$$\tilde{C}(x) := \frac{\partial^2}{\partial z^2}F(x,1).$$

We start by taking the second (partial) derivative with respect to z on both sides of the (2-dimensional) functional equation and substitute $z = 1$, this gives the following 1-dimensional functional equation in $\tilde{C}(x)$:

$$\begin{aligned}\tilde{C}(3x) &= 0 + 2 \cdot 1 \cdot 3F^2(x,1)\tilde{F}(x) + 1 \cdot \Big(6F(x,1)\tilde{F}^2(x) + 3F^2(x,1)\tilde{C}(x)\Big) \\ &= 6e^{2x}\tilde{F}(x) + 6e^{x}\tilde{F}^2(x) + 3e^{2x}\tilde{C}(x).\end{aligned}$$

This functional equation for $\tilde{C}(3x)$ can be solved by iteration. See appendix A for details. The resulting expression is obtained as follows:

$$\begin{aligned}\tilde{C}(3x) &= 6e^{2x}\tilde{F}(x) + 6e^{x}\tilde{F}^2(x) + 3e^{2x}\Big(6e^{\frac{2x}{3}}\tilde{F}(\frac{x}{3}) + 6e^{\frac{x}{3}}\tilde{F}^2(\frac{x}{3}) + 3e^{\frac{2x}{3}}\tilde{C}(\frac{x}{3})\Big) \\ &= \dots \\ &= \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})}\Big(6e^{\frac{2x}{3^i}}\tilde{F}(\frac{x}{3^i}) + 6e^{\frac{x}{3^i}}\tilde{F}^2(\frac{x}{3^i})\Big) + \\ &\quad \lim_{n\to\infty}\left[\prod_{i=0}^{n} 3e^{\frac{2x}{3^i}}\tilde{C}(\frac{x}{3^n})\right] \\ &= \sum_{i=0}^{\infty} 3^i e^{3x(1-\frac{1}{3^i})}\Big(6e^{\frac{2x}{3^i}}\tilde{F}(\frac{x}{3^i}) + 6e^{\frac{x}{3^i}}\tilde{F}^2(\frac{x}{3^i})\Big) + \\ &\quad \lim_{n\to\infty}\left[3^{n+1}e^{3x(1-\frac{1}{3^{n+1}})}\tilde{C}(\frac{x}{3^n})\right].\end{aligned}$$

One can verify that the expression above converges. So this gives a method to evaluate $\tilde{C}(3x)$ because of the fact that $\tilde{F}(x)$ also can be evaluated accurately. This last formula will be used later in this chapter to determine $B''_\lambda(1)$ for which the final expression becomes:

$$B''_\lambda(1) = \alpha_\lambda^{-1}e^{-\lambda}\Big(\tilde{C}(\lambda) - 2\tilde{F}(\lambda) + 2(e^\lambda - 1 - \lambda) + \frac{1}{12}\lambda^2\Big).$$

5.6 Approximation of $B_{\lambda(s)}(z_0)$ using an iterative method

Because of the direct relation between $B_{\lambda(s)}(z_0)$ and $F(3\lambda(s), z_0)$, an approximation of $F(3\lambda(s), z_0)$ will directly lead to an approximation of $B_{\lambda(s)}(z_0)$. In this section we will focus on approximating the (complex) value of $F(x, z)$ for $x > 0$ and z with $|z| \leq 1$.

Let's return to the functional equation for $F(\cdot, \cdot)$:

$$F(x,z) \;=\; zF^3(\frac{x}{3}, z) + (1+x)(1-z).$$

In this equation, we can replace $F(\frac{x}{3}, z)$ by $zF^3(\frac{x}{9}, z) + (1+\frac{x}{3})(1-z)$. Then we obtain:

$$F(x,z) \;=\; z\left(zF^3(\frac{x}{9}, z) + (1+\frac{x}{3})(1-z)\right)^3 + (1+x)(1-z).$$

In this equation, we can again replace $F(\frac{x}{9}, z)$ by $zF^3(\frac{x}{27}, z) + (1+\frac{x}{9})(1-z)$. In fact, we can keep iterating in this manner. What expression for $F(x, z)$ do we have after n iterations? First define:

$$\alpha_n(x,z) \;:=\; (1+\frac{x}{3^n})(1-z) \qquad n = 0, 1, \ldots$$

The expression for $F(x, z)$ after n iterations then becomes:

$$F(x,z) \;=\; z\left(\ldots z\left(z\left(zF^3(\frac{x}{3^{n+1}}, z) + \alpha_n(x,z)\right)^3 + \alpha_{n-1}(x,z)\right)^3 + \ldots\right) + \alpha_0(x,z).$$

Obviously $\lim_{n\to\infty} \frac{x}{3^{n+1}} = 0$. So:

$$\lim_{n\to\infty} F(\frac{x}{3^{n+1}}, z) \;=\; F(0,z) = f_0(z) = 1.$$

One possible way to approximate $F(x_0, z_0)$ is to perform n iterations and then simply replace $F(\frac{x_0}{3^n+1}, z_0)$ by 1. This seems a sensible approach. The hope is of course that this approximation converges to $F(x_0, z_0)$ as $n \to \infty$. But it turns out that this is not the case. To see this, look at the case in which $z_0 = 1$. The following holds:

$$\begin{aligned} F(x_0, 1) &= \sum_{n=0}^{\infty} f_n(1) x_0^n \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} x_0^n \\ &= e^{x_0}. \end{aligned}$$

Because of the fact that $\alpha_n(x_0, 1) = 0$ for all x_0 the approximation that will be found after n iterations equals 1. This is obviously not equal to e^{x_0}. Because of the fact that the approximation formula after n iterations is a continuous function of z, the approximation will not converge for other z than $z = 1$ either.

So, replacing $F(\frac{x_0}{3^n+1}, z_0)$ by $F(0, z_0)$ does not give the desired results. Another approach could be to replace $F(\frac{x_0}{3^n+1}, z_0)$ by $F(\frac{x_0}{3^n+1}, 1) = e^{\frac{x_0}{3^n+1}}$. This gives at least for $z_0 = 1$ an exact approximation.
Furthermore, $\lim_{n \to \infty} e^{\frac{x_0}{3^n+1}} = 1 = F(0, z_0)$ for all z_0. The hope is that this approximation, which will be denoted by $F_n(x_0, z_0)$, converges to $F(x_0, z_0)$ as $n \to \infty$. The next section will be devoted to the proof of this convergence.

To illustrate the fact that both approximations lead to different numerical approximations, a plot of the two approximations of $B_{\lambda(s)}(z)$ is given in figure 5.1, for $0 \leq z \leq 1$ in the case that $s = 3$ and $\mu = 0.75$. As one can see, for z not too large, both approximations lead to the same values. But when z approaches 1 both graphs diverge.

In the previous figure two approximations of $B_{\lambda(s)}(z)$ with $s = 3$ and $\mu = 0.75$ are plotted for real z between 0 and 1. In figure 5.2, the convergent approximation for $B_{\lambda(s)}(z)$ with $s = 3$ and $\mu = 0.75$ is plotted for all z in the complex unit circle. This is done by plotting the (complex) image of several circles in the complex plane (all having 0 as origin) under the function $B_{\lambda(s)}(\cdot)$. The complex circles have images that look like circles that are slightly compressed on one side. The different lines in the plot that are all starting from the origin

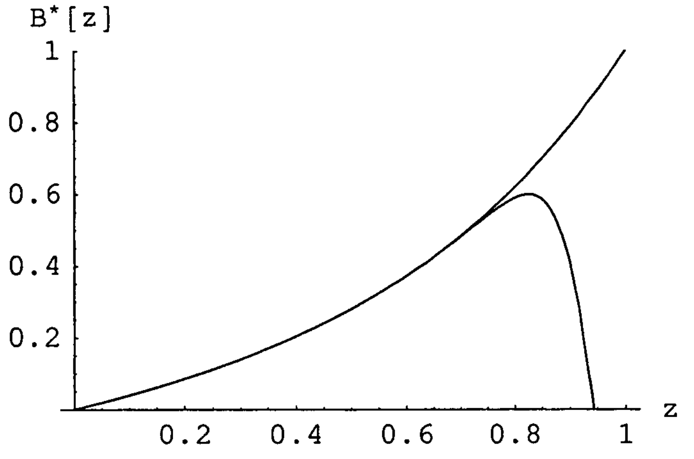

Figure 5.1: Two different approximations of $B_{\lambda(s)}(z)$ with $s = 3$ and $\mu = 0.75$ for $0 \leq z \leq 1$.

are the images of complex numbers with the same argument.

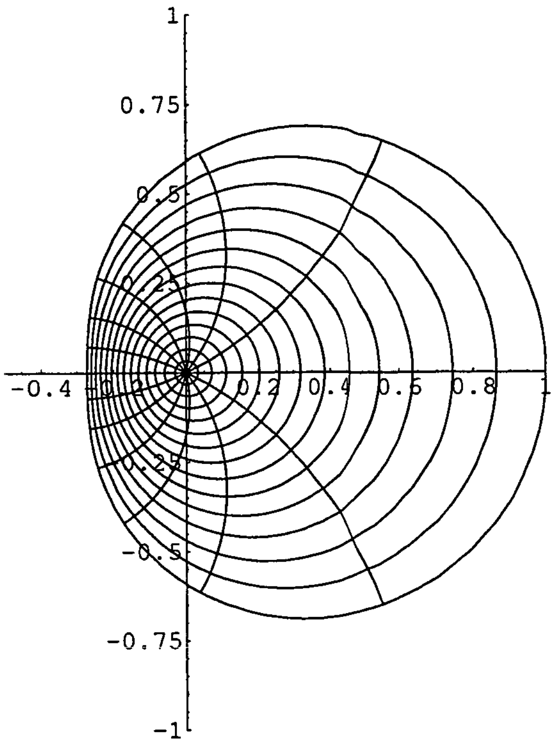

Figure 5.2: Plot of $B_{\lambda(s)}(z)$ with $s = 3$ and $\mu = 0.75$ for $0 \leq |z| \leq 1$.

5.7 Convergence of $F_n(x, z)$ to $F(x, z)$

5.7.1 Intermezzo

Consider two complex numbers c_1 and c_2 that lie within a distance r from each other, with $r \leq |c_1|$. What can be said about the (maximum) distance between c_1^3 and c_2^3?

Because $|c_1 - c_2| \leq r$ we can write:

$$c_2 = c_1 + r^* e^{i\varphi^*}, \qquad r^* \leq r,$$

$$\begin{aligned} |c_1^3 - c_2^3| &= |c_1^3 - (c_1 + r^* e^{i\varphi^*})^3| \\ &= |3c_1^2 r^* e^{i\varphi^*} + 3c_1 (r^*)^2 e^{2i\varphi^*} + (r^*)^3 e^{3i\varphi^*}| \\ &\leq |3c_1^2 r^*| + |3c_1 (r^*)^2| + |(r^*)^3| \\ &\leq 7r^* |c_1|^2, \qquad \text{because } r^* \leq |c_1|. \end{aligned}$$

We are going to use this result later.

5.7.2 Proof of convergence

Recall that $F_n(x, z)$ is the approximation for $F(x, z)$ that is obtained when after n iterations $F(\frac{x}{3^{n+1}}, z)$ has been replaced by $F(\frac{x}{3^{n+1}}, 1)$. In this section it will be proven that $F_n(x, z)$ converges to $F(x, z)$ as $n \to \infty$ for all (complex) z with $|z| \leq 1$. Formally:

$$\forall x > 0, \forall z \in \mathbb{C}, |z| \leq 1, \quad |F_n(x, z) - F(x, z)| \to 0 \quad \text{as } n \to \infty.$$

From the definition of F we get the following:

$$\begin{aligned}
F(\frac{x}{3^n}, z) &= 1 + \frac{x}{3^n} + \sum_{k=2}^{\infty} f_k(z) \left(\frac{x}{3^n}\right)^k, \\
\left|F(\frac{x}{3^n}, z)\right| &= \left|1 + \frac{x}{3^n} + \sum_{k=2}^{\infty} f_k(z) \left(\frac{x}{3^n}\right)^k\right| \\
&\leq 1 + \frac{x}{3^n} + \sum_{k=2}^{\infty} |f_k(z)| \left(\frac{x}{3^n}\right)^k \\
&\leq 1 + \frac{x}{3^n} + \sum_{k=2}^{\infty} \frac{1}{k!} \left(\frac{x}{3^n}\right)^k \\
&= e^{\frac{x}{3^n}}.
\end{aligned}$$

So the approximation of $F(\frac{x}{3^n}, z)$ by $e^{\frac{x}{3^n}}$ is in fact an overestimation with respect to the absolute value of both complex numbers. It is important to know how far these two complex numbers can lie apart in the complex plane:

$$\begin{aligned}
\left|e^{\frac{x}{3^n}} - F(\frac{x}{3^n}, z)\right| &= \left|e^{\frac{x}{3^n}} - (1 + \frac{x}{3^n}) + (1 + \frac{x}{3^n}) - F(\frac{x}{3^n}, z)\right| \\
&\leq \left|e^{\frac{x}{3^n}} - (1 + \frac{x}{3^n})\right| + \left|(1 + \frac{x}{3^n}) - F(\frac{x}{3^n}, z)\right| \\
&\leq \frac{1}{2}\left(\frac{x}{3^n}\right)^2 + \mathcal{O}\left(\left(\frac{x}{3^n}\right)^3\right) + \frac{1}{2}\left(\frac{x}{3^n}\right)^2 + \mathcal{O}\left(\left(\frac{x}{3^n}\right)^3\right)
\end{aligned}$$

$$= \left(\frac{x}{3^n}\right)^2 + \mathcal{O}\left(\left(\frac{x}{3^n}\right)^3\right).$$

Now we are going to use the result obtained in the previous section to show that $F_n(x,z)$ lies close enough to $F(x,z)$. We know that the complex numbers $F(\frac{x}{3^n},z)$ and $e^{\frac{x}{3^n}}$ are very close to each other in the complex plane, at least for large n. When we work one iteration back with both the approximation and the exact formula, we can show that these two complex numbers are still close to each other. Working back one iteration from iteration n means raising the complex number to the power 3, then multiplying it with z and adding $\alpha_n(x,z)$. When we proceed in that manner until we are back at $F_n(x,z)$ and $F(x,z)$, we have an expression for the maximum distance between those two numbers. When we take the limit for $n \to \infty$, the hope is that this maximum distance goes to zero. So let's try.

Define the following:

$$r_{n+1} = \left|e^{\frac{x}{3^{n+1}}} - F(\frac{x}{3^{n+1}},z)\right| = \left(\frac{x}{3^{n+1}}\right)^2 + \mathcal{O}\left(\left(\frac{x}{3^{n+1}}\right)^3\right).$$

Working back one iteration yields:

$$\begin{aligned}
\left|\left(zF^3(\frac{x}{3^{n+1}},z) + \alpha_n(x,z)\right) - \left(z\left(e^{\frac{x}{3^{n+1}}}\right)^3 + \alpha_n(x,z)\right)\right| &= |z|\left|F^3(\frac{x}{3^{n+1}},z) - e^{\frac{x}{3^n}}\right| \\
&\leq 7r_{n+1}\left|F(\frac{x}{3^{n+1}},z)\right|^2 \\
&\leq 7r_{n+1}\left(e^{\frac{x}{3^{n+1}}}\right)^2.
\end{aligned}$$

This is only true if $r_{n+1} \leq e^{\frac{x}{3^{n+1}}}$, which holds when n is sufficiently large. Proceeding in the same way eventually yields the following:

$$\begin{aligned}
\left|\left(zF^3(\frac{x}{3},z) + \alpha_0(x,z)\right) - F_n(x,z)\right| &= |F(x,z) - F_n(x,z)| \\
&\leq 7^n r_{n+1}\prod_{i=1}^{n} e^{\frac{x}{3^i}} \\
&= r_{n+1}\prod_{i=1}^{n}\left(7e^{\frac{x}{3^i}}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\left(\frac{x}{3^{n+1}}\right)^2 + \mathcal{O}\left(\left(\frac{x}{3^{n+1}}\right)^3\right)\right)\prod_{i=1}^{n}\left(7e^{\frac{x}{3^i}}\right) \\
&= \frac{x^2}{9}\prod_{i=1}^{n}\left(\frac{7}{9}e^{\frac{x}{3^i}}\right) + \mathcal{O}\left(\frac{x^3}{3^{n+3}}\right)\prod_{i=1}^{n}\left(\frac{7}{9}e^{\frac{x}{3^i}}\right).
\end{aligned}
$$

The first $\leq$-sign is explained by successively working back the n iterations and using that $F(\frac{x}{3^n}, z) \leq e^{\frac{x}{3^n}}$.

Because of the fact that $\lim_{n\to\infty} e^{\frac{x}{3^n}} = 1$ we have that:

$$
\lim_{n\to\infty}\left(\frac{x^2}{9}\prod_{i=1}^{n}\left(\frac{7}{9}e^{\frac{x}{3^i}}\right) + \mathcal{O}\left(\frac{x^3}{3^{n+3}}\right)\prod_{i=1}^{n}\left(\frac{7}{9}e^{\frac{x}{3^i}}\right)\right) = 0.
$$

This means that $F_n(x,z)$ indeed converges to $F(x,z)$ as $n \to \infty$. The only thing that still has to be proven is that during every step backwards the maximum distance between the two complex numbers does not exceed the used upper bound. More precisely, the following must hold:

$$
\forall i \in \{0,\ldots,n\} \qquad : \ [r_{n+1}\prod_{j=i+1}^{n}\left(7e^{\frac{x}{3^j}}\right) \leq e^{\frac{x}{3^i}}].
$$

Now, we are going to prove this last statement.

$$
\begin{aligned}
r_{n+1}\prod_{j=i+1}^{n}\left(7e^{\frac{x}{3^j}}\right) &= r_{n+1}7^{n-i}e^{x\frac{1}{3^i}\frac{3}{2}\left(1-\left(\frac{1}{3}\right)^{n-i}\right)} \\
&\leq r_{n+1}7^{n-i}e^{x\frac{1}{3^i}\frac{3}{2}} \\
&= r_{n+1}7^{n-i}e^{\frac{1}{2}\frac{x}{3^i}}\cdot e^{\frac{x}{3^i}} \\
&= \left(x^2\left(\frac{7}{9}\right)^n + x^3\mathcal{O}\left(\left(\frac{7}{27}\right)^n\right)\right)7^{-i}e^{\frac{1}{2}\frac{x}{3^i}}\cdot e^{\frac{x}{3^i}} \\
&\leq \left(x^2\left(\frac{7}{9}\right)^n + x^3\mathcal{O}\left(\left(\frac{7}{27}\right)^n\right)\right)e^{\frac{1}{2}\frac{x}{3^i}}\cdot e^{\frac{x}{3^i}}
\end{aligned}
$$

$$
\begin{aligned}
&\leq \left(x^2\left(\frac{7}{9}\right)^n + x^3\mathcal{O}\left(\left(\frac{7}{27}\right)^n\right)\right)e^{\frac{1}{2}x}\cdot e^{\frac{x}{3^i}} \\
&\leq e^{\frac{x}{3^i}}, \qquad \text{as } n \text{ sufficiently large.}
\end{aligned}
$$

5.8 Approximation of the function $B_{\lambda(s)}(z)$ through Taylor expansion

In Section 5.6 it was shown how one can get a very accurate numerical approximation for the value of $B_{\lambda(s)}(z_0)$ for all z_0 with $|z_0| \leq 1$. It is also possible to obtain a closed-form approximation for the function $B_{\lambda(s)}(z)$. This can be done by first using a Taylor expansion to approximate the function $F(x_0, z)$. When this approximation is obtained, this function can be used to approximate $B_{\lambda(s)}(z)$ by using the direct relationship between both functions that has been derived in section 5.3.

We start with the following Taylor expansion for $F(x_0, z)$ around zero:

$$F(x_0, z) \;=\; F(x_0, 0) + z\frac{\partial}{\partial z}F(x_0, 0) + \frac{z^2}{2}\frac{\partial^2}{\partial z^2}F(x_0, 0) + \mathcal{O}(z^3), \qquad |z| \leq 1.$$

$F(x_0, 0)$ can be determined by using the functional equation for F. This leads to $F(x_0, 0) = 1 + x_0$. When this is known, $\frac{\partial}{\partial z}F(x_0, 0)$ can be determined by differentiating both sides of the functional equation for F with respect to z. Higher (partial) derivatives can be determined in the same way. Proceeding in this way we find for the first two derivatives :

$$\begin{aligned}
\frac{\partial}{\partial z}F(x_0, 0) &= \frac{x^2}{3} + \frac{x^3}{27}, \\
\frac{\partial^2}{\partial z^2}F(x_0, 0) &= \frac{2\,x^2}{9} + \frac{38\,x^3}{243} + \frac{22\,x^4}{729} + \frac{2\,x^5}{2187}.
\end{aligned}$$

5.9 Numerical solution method for particular functional equations

In this section, a method for solving 2-dimensional functional equations will be outlined, which uses power series substitution. For the moment, we only consider the already presented functional equation for the function $F(x,z)$. In the future, this method can possibly be generalized.

The functional equation that will be studied is the following:

$$F(3x,z) = zF^3(x,z) + (1+3x)(1-z).$$

The boundary conditions are not specified yet. Assume that a solution exists. This solution can be represented by means of a power series in both x and z:

$$F(x,z) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{i,j}x^i z^j.$$

This formal solution can be plugged into the functional equation to obtain a recursion for the coefficients $a_{i,j}$ with $i,j = 0,1,\ldots$. This results in the following:

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{i,j}3^i x^i z^j = z\left(\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{i,j}x^i z^j\right)^3 + (1+3x)(1-z).$$

Furthermore, the following holds:

$$\begin{aligned}\left(\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{i,j}x^i z^j\right)^3 &= \sum_{i_1=0}^{\infty}\sum_{j_1=0}^{\infty}\sum_{i_2=0}^{\infty}\sum_{j_2=0}^{\infty}\sum_{i_3=0}^{\infty}\sum_{j_3=0}^{\infty} a_{i_1,j_1}a_{i_2,j_2}a_{i_3,j_3}x^{i_1+i_2+i_3}z^{j_1+j_2+j_3} \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\left(\sum_{\substack{i_1,i_2,i_3\geq 0\\ i_1+i_2+i_3=i}}\ \sum_{\substack{j_1,j_2,j_3\geq 0\\ j_1+j_2+j_3=j}} a_{i_1,j_1}a_{i_2,j_2}a_{i_3,j_3}\right)x^i z^j.\end{aligned}$$

Equating the coefficients of $x^i z^j$ on both sides of the functional equation yields the following:

$$
\begin{aligned}
a_{0,0} &= 1, \\
a_{0,1} &= a_{0,0}^3 - 1 = 0, \\
3a_{1,0} &= 3, \\
3a_{1,1} &= 3a_{1,0}a_{0,0}^2 - 3 = 0, \\
3^i a_{i,j} &= \left(\sum_{\substack{i_1,i_2,i_3 \geq 0 \\ i_1+i_2+i_3=i}} \sum_{\substack{j_1,j_2,j_3 \geq 0 \\ j_1+j_2+j_3=j-1}} a_{i_1,j_1} a_{i_2,j_2} a_{i_3,j_3} \right), \\
i,j &\geq 1.
\end{aligned}
$$

This recursion can be solved. This gives a way to determine in principle all coefficients. As it turns out, these coefficients are uniquely determined. Therefore, we can conclude that in this case, no boundary conditions are necessary to obtain a unique solution of the indicated power series form.

When this recursion is implemented (in Mathematica for instance), we can use it to determine the first $n \times m$ coefficients to obtain an estimate, denoted by $F_{m,n}(x,z)$, of the solution for the functional equation:

$$
\begin{aligned}
F_{m,n}(x,z) &:= \sum_{i=0}^{m} \sum_{j=0}^{n} a_{i,j} x^i z^j, \\
n, m \in \mathbb{N}.
\end{aligned}
$$

Such an estimate can be used to generate a plot, to obtain insight in the behaviour of the function. For $m = 7, n = 7$, this plot is given in figure 5.3.
Note that this estimate can also be used to obtain an estimate for the generating function $B_\lambda(z)$, using the relation described in Section 5.3.

The method that has been outlined in this section raises several questions. For which functional equations will this method work? What solutions are found? We were especially interested in obtaining insights in the behaviour of

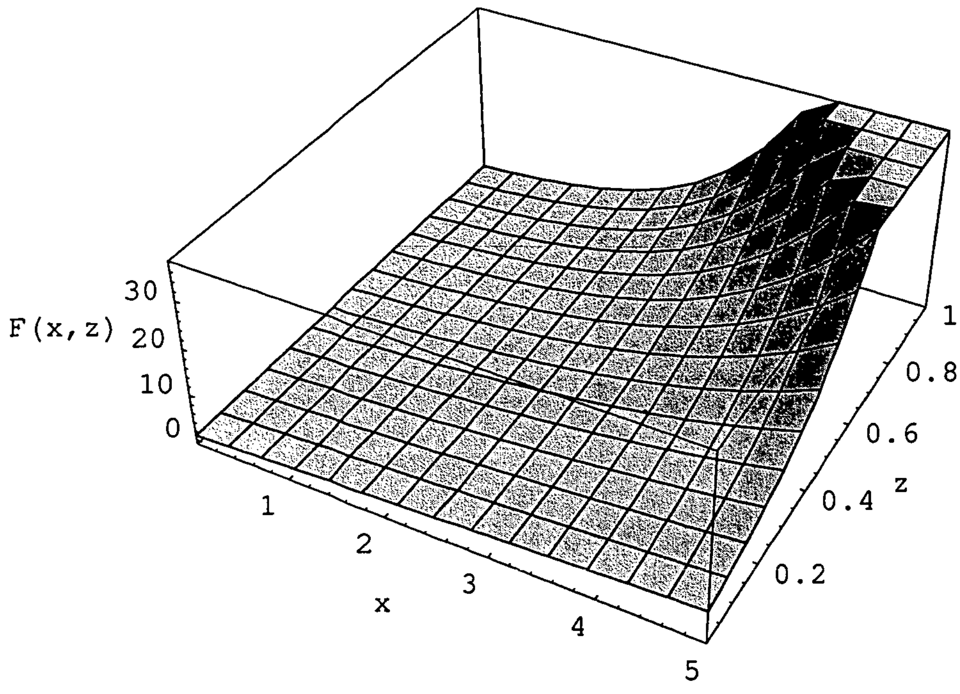

Figure 5.3: Plot of $F_{m,n}(x, z)$ for $0 \le z \le 1$ and $0 \le x \le 5$.

the solution of one particular functional equation. It was known beforehand that this solution of interest could be represented by means of the proposed power series, since the solution was a generating function that was more or less constructed as a two-dimensional power series. However, it is worth investigating the open questions on this topic in future.

Chapter 6

Service time analysis for depth-first trees

6.1 Overview

So far we have concentrated on the waiting time of an arbitrary super customer. In this chapter the focus will be on the service time of an arbitrary *individual* customer in the static arrival slot mechanism. This service time is defined as the part of the processing time of the tree (measured in slots) that an individual customer is still in the tree. This individual service time is clearly dependent on the way a tree is processed. Breadth-first or depth-first will surely not lead to the same service times for individual customers. For the moment only the depth-first case will be analyzed.

In sections 6.2 and 6.3 of this chapter, the focus will be on the *mean* individual service time for which a related functional equation will be derived and solved.

The generating function of the service time distribution in this model is needed in order to finally translate the results to the arrival slot mechanism. Therefore an expression for this generating function (and its first derivative) as well as a method to evaluate this function will be presented in section 6.4.

In the next section, section 6.5, the variance of the individual service time is considered. The generating function of the service time will be used to obtain an expression for the variance.

In the last section, section 6.6, the correlation between the tree length and the (aggregated) service time is considered.

6.2 A functional equation for the mean service time

Define $\tilde{S}(n)$ as the random variable which represents the total internal delay of all n customers in a tree, when it is processed in depth-first order, $n \geq 2$. Recall that $\tilde{B}(n)$ is defined as the random variable that represents the total length of a tree with n customers in it, $n \geq 2$, and $\tilde{B}(0) = \tilde{B}(1) = 0$. The $\tilde{B}_n(z)$'s are the corresponding generating functions.

We can derive a recursion for the $\tilde{S}(n)$'s. This recursion can be obtained as follows. A tree which has initially n customers in it, splits after the first slot with a certain probability in three subtrees with n_1, n_2 and n_3 customers in it. When we consider the total aggregated service time after this first slot, it can be observed that all n customers are already delayed for 1 slot. Furthermore, those subtrees with more than one customer in it contribute also to the total internal delay $\tilde{S}(n)$. The customers in the first subtree do not have to wait for any other subtree, but the second and third subtree (if they exist) do have to wait for the first and first and second subtree to be finished respectively. This takes exactly $\tilde{B}(n_1)$ and $\tilde{B}(n_2)$ slots respectively. Using this insight, we obtain a recursion. The sum notations have to be interpreted as follows: $\sum\limits_{\substack{n_1, n_2 \neq 1, n_3 = 1 \\ n_1+n_2+n_3=n}}$ is a sum over all non-negative integers n_1, n_2 and n_3 satisfying $n_1 + n_2 + n_3 = n$ and additionally $n_2 \neq 1$ and $n_3 = 1$. The other sum notations have to be read analogously. The resulting recursion is the following:

$$
\begin{aligned}
P(\tilde{S}(n) = k) &= \sum_{\substack{n_1, n_2 \neq 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \cdot \\
&\qquad P(\tilde{S}(n_1) + \tilde{S}(n_2) + \tilde{S}(n_3) + (n_2 + n_3)\tilde{B}(n_1) + n_3\tilde{B}(n_2) + n = k) \\
&+ \sum_{\substack{n_1, n_2 = 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \cdot P(\tilde{S}(n_1) + \tilde{S}(n_3) + n_3\tilde{B}(n_1) + n = k) \\
&+ \sum_{\substack{n_1, n_2 \neq 1, n_3 = 1 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \cdot P(\tilde{S}(n_1) + \tilde{S}(n_2) + n_2\tilde{B}(n_1) + n = k)
\end{aligned}
$$

$$+ \sum_{\substack{n_1, n_2=1, n_3=1 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \cdot P(\tilde{S}(n_1) + \tilde{S}(n_2) + \tilde{S}(n_3) + n = k),$$

$$n \geq 2, \qquad k = 1, 2, \ldots$$

with:

$$\begin{aligned} \tilde{S}(0) &= \tilde{S}(1) = 0, \\ P(\tilde{S}(n) = 0) &= 0, \qquad n = 2, 3, \ldots, \\ \xi(n_1, n_2, n_3) &= \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n. \end{aligned}$$

Define the following:

$$\tilde{S}_n(z) := \sum_{k=0}^{\infty} P(\tilde{S}(n) = k) z^k.$$

When we write the recursion in terms of generating functions we get the following:

$$\tilde{S}_0(z) = \tilde{S}_1(z) = 1 \quad ,$$

$$\begin{aligned} \tilde{S}_n(z) = & \sum_{k=0}^{\infty} \sum_{\substack{n_1, n_2 \neq 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n z^k \cdot \\ & P(\tilde{S}(n_1) + \tilde{S}(n_2) + \tilde{S}(n_3) + (n_2 + n_3)\tilde{B}(n_1) + n_3 \tilde{B}(n_2) + n = k) \\ & + \sum_{k=0}^{\infty} \sum_{\substack{n_1, n_2 = 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n z^k \cdot \\ & P(\tilde{S}(n_1) + \tilde{S}(n_3) + n_3 \tilde{B}(n_1) + n = k) \\ & + \sum_{k=0}^{\infty} \sum_{\substack{n_1, n_2 \neq 1, n_3 = 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n z^k \cdot P(\tilde{S}(n_1) + \tilde{S}(n_2) + n_2 \tilde{B}(n_1) + n = k) \end{aligned}$$

$$+\sum_{k=0}^{\infty} \sum_{\substack{n_1,n_2=1,n_3=1\\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n z^k \cdot P(\tilde{S}(n_1)+n=k),$$

$n \geq 2.$

Now the summations are interchanged, both sides are differentiated with respect to z and $z = 1$ is substituted. Furthermore, using that $\tilde{S}'_n(1) = \mathbb{E}\tilde{S}(n)$ and $\tilde{B}'_n(1) = \mathbb{E}\tilde{B}(n)$ finally yields the following:

$$\begin{aligned}
\mathbb{E}\tilde{S}(n) \quad = \quad & \sum_{\substack{n_1,n_2\neq 1,n_3\neq 1\\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \cdot \\
& \left(\mathbb{E}\tilde{S}(n_1)+\mathbb{E}\tilde{S}(n_2)+\mathbb{E}\tilde{S}(n_3)+(n_2+n_3)\mathbb{E}\tilde{B}(n_1)+n_3\mathbb{E}\tilde{B}(n_2)\right) \\
& + \sum_{\substack{n_1,n_2=1,n_3\neq 1\\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1)+\mathbb{E}\tilde{S}(n_3)+n_3\mathbb{E}\tilde{B}(n_1)\right) \\
& + \sum_{\substack{n_1,n_2\neq 1,n_3=1\\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1)+\mathbb{E}\tilde{S}(n_2)+n_2\mathbb{E}\tilde{B}(n_1)\right) \\
& + \sum_{\substack{n_1,n_2=1,n_3=1\\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1)\right)+n,
\end{aligned}$$

$n \geq 2.$

After some manipulation and simplification the last equation can be written in a slightly different way:

$$\begin{aligned}
\frac{3^n}{n!}\mathbb{E}\tilde{S}(n) \quad = \quad & \sum_{\substack{n_1,n_2,n_3\\ n_1+n_2+n_3=n}} \left(\frac{\mathbb{E}\tilde{S}(n_1)}{n_1!}(\frac{1}{n_2!}\frac{1}{n_3!})+\frac{\mathbb{E}\tilde{S}(n_2)}{n_2!}(\frac{1}{n_1!}\frac{1}{n_3!})+\frac{\mathbb{E}\tilde{S}(n_3)}{n_3!}(\frac{1}{n_1!}\frac{1}{n_2!})\right)+ \\
& n\frac{3^n}{n!}+3\sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \left(\frac{1}{n_1!}\frac{1}{n_2!}\frac{1}{n_3!}\left(n_2\mathbb{E}\tilde{B}(n_1)\right)\right),
\end{aligned}$$

$n \geq 2$.

Define the function $Q(x) := \sum_{n=2}^{\infty} \frac{\mathbb{E}\tilde{S}(n)}{n!} x^n$. After multiplying both sides of the last equation with x^n and summing these equations over n, we get the following result:

$$Q(3x) \;=\; 3e^{2x}Q(x) + \sum_{n=2}^{\infty} n\frac{3^n}{n!}x^n + 3\sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!} n_2 \mathbb{E}\tilde{B}_{n_1}.$$

In this last equation, some expressions can still be further evaluated:

$$\begin{aligned}
\sum_{n=2}^{\infty} n\frac{(3x)^n}{n!} &= 3x\sum_{n=2}^{\infty} n\frac{(3x)^{n-1}}{n!} \\
&= x\left(\sum_{n=2}^{\infty}\frac{(3x)^n}{n!}\right)' \\
&= x\left(e^{3x}-1-3x\right)' \\
&= 3x\left(e^{3x}-1\right),
\end{aligned}$$

and

$$\begin{aligned}
\sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2\geq 2,n_3\\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!} n_2 \mathbb{E}\tilde{B}(n_1) &= \sum_{n_1=0}^{\infty}\frac{\mathbb{E}\tilde{B}(n_1)}{n_1!}x^{n_1}\sum_{n_2=2}^{\infty} n_2\frac{x^{n_2}}{n_2!}\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!} \\
&= \tilde{F}(x)\cdot x\left(e^x-1\right)\cdot e^x \\
&= xe^x\left(e^x-1\right)\tilde{F}(x).
\end{aligned}$$

Here $\tilde{F}(x)$ is the generating function as defined in Section 5.4. Substituting the results from above, we obtain the following final equation:

$$Q(3x) \;=\; 3e^{2x}Q(x) + 3x\left(e^{3x}-1\right) + 3xe^x\left(e^x-1\right)\tilde{F}(x).$$

This equation can be solved by iteration. This procedure has already been described at the end of Section 5.2, where $\tilde{F}(x)$ is represented as an infinite sum. Proceeding in the same way yields:

$$
\begin{aligned}
Q(3x) &= 3x\left(e^{3x}-1\right)+3xe^{x}\left(e^{x}-1\right)\tilde{F}(x)+3e^{2x}\left(x\left(e^{x}-1\right)+xe^{\frac{1}{3}x}\left(e^{\frac{1}{3}x}-1\right)\tilde{F}(\frac{x}{3})\right) \\
&= \dots \\
&= \sum_{j=0}^{\infty} 3^j e^{3x(1-\frac{1}{3^j})}\frac{3x}{3^j}\left(e^{\frac{3x}{3^j}}-1\right)+\sum_{j=0}^{\infty} 3^j e^{3x(1-\frac{1}{3^j})}\frac{3x}{3^j}e^{\frac{x}{3^j}}\left(e^{\frac{x}{3^j}}-1\right)\tilde{F}(\frac{x}{3^j}) \\
&= \sum_{j=0}^{\infty} 3xe^{3x(1-\frac{1}{3^j})}\left(\left(e^{\frac{3x}{3^j}}-1\right)+e^{\frac{x}{3^j}}\left(e^{\frac{x}{3^j}}-1\right)\left(\sum_{i=0}^{\infty} 3^i e^{\frac{x}{3^j}(1-\frac{1}{3^i})}\left(e^{\frac{x}{3^{i+j}}}-\frac{x}{3^{i+j}}-1\right)\right)\right) \\
&= \sum_{j=0}^{\infty} 3xe^{3x(1-\frac{1}{3^j})}\left(e^{\frac{3x}{3^j}}-1\right)+\sum_{j=0}^{\infty}\sum_{i=0}^{\infty} 3^{i+1}xe^{x(3-\frac{1}{3^j}-\frac{1}{3^{i+j}})}\left(e^{\frac{x}{3^j}}-1\right)\left(e^{\frac{x}{3^{i+j}}}-\frac{x}{3^{i+j}}-1\right).
\end{aligned}
$$

6.3 Determination of the mean individual service time

In this section, the results of the previous section are related to the mean individual service time in the queueing model that is considered.

Consider the case in which the arrival process of individual customers is a Poisson process with rate μ (per slot). Let S_{static} denote the random variable that represents the total internal delay of an arbitrary batch of customers that arrives after a frame of t slots. This t is only important in determining the total (Poisson) arrival rate that occurs in an arrival slot, when the rate per slot is given by μ. For the expectation of S_{static}, denoted by $\mathbb{E}S_{static}$, the following holds:

$$
\begin{aligned}
\mathbb{E}S_{static} &= e^{-\mu t}\cdot 0+\mu t e^{-\mu t}\cdot 1+\sum_{n=2}^{\infty} e^{-\mu t}\frac{(\mu t)^n}{n!}S_n'(1) \\
&= e^{-\mu t}\left(\mu t+Q\left(\mu t\right)\right).
\end{aligned}
$$

The mean service time of an arbitrary individual customer, denoted by $\mathbb{E}S_{static,ind}$, can be obtained by dividing the mean total internal delay of an arbitrary batch by the mean number of customers that arrives in a batch. So the mean service time of an arbitrary individual customer satisfies the following relation:

$$\begin{aligned}\mathbb{E}S_{static,ind} &= \frac{\mathbb{E}S_{static}}{\mu t}\\ &= e^{-\mu t}\frac{\mu t + Q(\mu t)}{\mu t}.\end{aligned}$$

So the final conclusion is that the mean individual service time can be obtained numerically by computing an infinite sum (which converges very fast).

As an illustration, the mean individual service time as a function of μ is plotted for three different frame lengths. As one can see in figure 6.1, the mean service time is approximately linear in μ for all three cases.

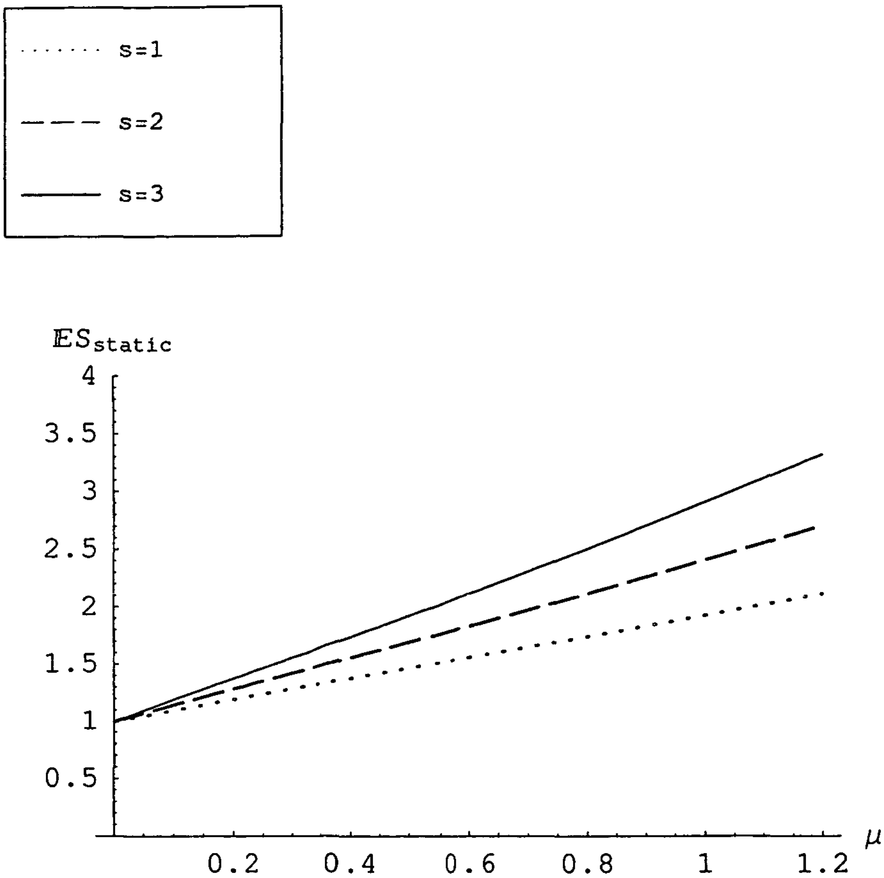

Figure 6.1: Mean individual service time as a function of μ

6.4 The generating function of the individual service time distribution

This section is divided into two subsections. The first subsection is devoted to the derivation of a 2-dimensional functional equation with $H(x,z)$ as unknown function. This functional equation relates the tree length distribution and the number of customers at the beginning of a tree. The second subsection relates the generating function of the tree length distribution of an arbitrary tree (assuming a Poisson arrival process) to $H(x,z)$.

6.4.1 Derivation of a 2-dimensional functional equation

In this subsection a 2-dimensional functional equation will be derived, analogously as in Section 5.2. With the use of this equation, the generating function of the individual service time distribution can be evaluated. This will be shown in the next subsection. Recall that the tree protocol is assumed to be depth-first.

Let $\tilde{D}(n)$ be a random variable, which has as distribution the individual service time distribution of an arbitrary customer in a tree with n customers. One can obtain a recursive formula for the $\tilde{D}(n)$'s. We adopt the same notation as in Section 5.2.

A tree with initially n customers in it, splits with a certain probability in three subtrees with n_1, n_2 and n_3 customers in it. An arbitrary customer faces a probability of $\frac{n_i}{n}$ to enter the i^{th} subtree. As we have seen before, the customers in the second and third subtree have to wait for other subtrees to be finished. This is of course only the case when a subtree is indeed formed, i.e. consists of at least 2 customers.

Putting everything together, the following recursion holds:

$$\begin{aligned} P(\tilde{D}(n)=k) &= \sum_{\substack{n_1,n_2,n_3\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_1}{n}P(\tilde{D}(n_1)=k-1) \\ &+ \sum_{\substack{n_1,n_2\neq 1,n_3\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_2}{n}P(\tilde{D}(n_2)+\tilde{B}(n_1)=k-1) \\ &+ \sum_{\substack{n_1,n_2,n_3\neq 1\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_3}{n}P(\tilde{D}(n_3)+\tilde{B}(n_1)+\tilde{B}(n_2)=k-1) \\ &+ \sum_{\substack{n_1,n_2=1,n_3\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_2}{n}P(\tilde{D}(n_2)=k-1) \\ &+ \sum_{\substack{n_1,n_2,n_3=1\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_3}{n}P(\tilde{D}(n_3)=k-1), \end{aligned}$$

$k=1,2,\ldots,$

with:

$$\begin{aligned}
\tilde{D}(0) &= \tilde{D}(1) = 0, \\
P(\tilde{D}(n) = 0) &= 0, \qquad n = 2, 3, \dots, \\
\xi(n_1, n_2, n_3) &= \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n .
\end{aligned}$$

Multiplying both sides of the recursive equation with z^k and substituting the expression for $\xi(n_1, n_2, n_3)$ yields:

$$\begin{aligned}
P(\tilde{D}(n) = k) z^k &= z \sum_{\substack{n_1, n_2, n_3 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_1}{n} P(\tilde{D}(n_1) = k - 1) z^{k-1} \\
&+ z \sum_{\substack{n_1, n_2 \neq 1, n_3 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_2}{n} P(\tilde{D}(n_2) + \tilde{B}(n_1) = k - 1) z^{k-1} \\
&+ z \sum_{\substack{n_1, n_2, n_3 \neq 1 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_3}{n} P(\tilde{D}(n_3) + \tilde{B}(n_1) + \tilde{B}(n_2) = k - 1) z^{k-1} \\
&+ 2z \sum_{\substack{n_1, n_2 = 1, n_3 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_2}{n} I_{\{k=1\}} z^{k-1}.
\end{aligned}$$

Here, $I_{\{k=1\}}$ is the indicator function. Summing this equation over $k = 1, 2, \dots$ and introducing the following generating function:

$$\tilde{D}_n(z) = \sum_{k=0}^{\infty} P(\tilde{D}(n) = k) z^k, \qquad |z| \leq 1,$$

finally yields:

$$\begin{aligned}
\tilde{D}_n(z) &= z \sum_{k=1}^{\infty} \sum_{\substack{n_1, n_2, n_3 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_1}{n} P(\tilde{D}(n_1) = k - 1) z^{k-1} \\
&+ z \sum_{k=1}^{\infty} \sum_{\substack{n_1, n_2 \neq 1, n_3 \\ n_1 + n_2 + n_3 = n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \frac{n_2}{n} P(\tilde{D}(n_2) + \tilde{B}(n_1) = k - 1) z^{k-1}
\end{aligned}$$

$$+z\sum_{k=1}^{\infty}\sum_{\substack{n_1,n_2,n_3\neq 1\\ n_1+n_2+n_3=n}}\frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n\frac{n_3}{n}P(\tilde{D}(n_3)+\tilde{B}(n_1)+\tilde{B}(n_2)=k-1)z^{k-1}$$

$$+2z\sum_{\substack{n_1,n_2=1,n_3\\ n_1+n_2+n_3=n}}\frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n\frac{n_2}{n}I_{\{k=1\}}z^{k-1},$$

$$\begin{aligned}3^n\frac{n\tilde{D}_n(z)}{n!} &= z\sum_{\substack{n_1,n_2,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_2!}\frac{1}{n_3!}\frac{n_1\tilde{D}_{n_1}(z)}{n_1!}\\ &+z\sum_{\substack{n_1,n_2\neq 1,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_3!}\frac{n_2\tilde{D}_{n_2}(z)}{n_2!}\frac{\tilde{B}_{n_1}(z)}{n_1!}\\ &+z\sum_{\substack{n_1,n_2,n_3\neq 1\\ n_1+n_2+n_3=n}}\frac{n_3\tilde{D}_{n_3}(z)}{n_3!}\frac{\tilde{B}_{n_1}(z)}{n_1!}\frac{\tilde{B}_{n_2}(z)}{n_2!}\\ &+2z\sum_{\substack{n_1,n_2=1,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_1!}\frac{1}{n_3!}\frac{n_2}{n_2!}.\end{aligned}$$

Introducing $h_n(z) = \frac{n\tilde{D}_n(z)}{n!}$ for $n \geq 0$ and multiplying each side of the equation with x^n the last equation becomes:

$$\begin{aligned}(3x)^n h_n(z) &= z\sum_{\substack{n_1,n_2,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_2!}\frac{1}{n_3!}h_{n_1}(z)x^n\\ &+z\sum_{\substack{n_1,n_2\neq 1,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_3!}h_{n_2}(z)f_{n_1}(z)x^n\\ &+z\sum_{\substack{n_1,n_2,n_3\neq 1\\ n_1+n_2+n_3=n}}h_{n_3}(z)f_{n_1}(z)f_{n_2}(z)x^n\\ &+2z\sum_{\substack{n_1,n_2=1,n_3\\ n_1+n_2+n_3=n}}\frac{1}{n_1!}\frac{1}{n_3!}x^n.\end{aligned}$$

Summing these equations over $n \geq 2$ and introducing the following generating function:

$$H(x,z) := \sum_{n=0}^{\infty}h_n(z)x^n, \quad x\in\mathbb{R}, |z|\leq 1,$$

we get the following result:

$$
\begin{aligned}
H(3x,z) - 3xh_1(z) - h_0(z) &= z\left(e^{2x}H(x,z) - h_0(z) - 2xh_0(z) - xh_1(z)\right) + \\
& z\left(e^{x}\left(H(x,z) - xh_1(z)\right)F(x,z) - h_0(z)f_0(z) - \right. \\
& \left. xh_0(z)f_0(z) - xh_0(z)f_1(z)\right) \\
& + z\left(\left(H(x,z) - xh_1(z)\right)F^2(x,z) - h_0(z)f_0^2(z) - \right. \\
& \left. 2xh_0(z)f_0(z)f_1(z)\right) + 2zxh_1(z)\left(e^{2x} - 1\right).
\end{aligned}
$$

Furthermore, we know:

$$
\begin{aligned}
h_0(z) &= \frac{0 \cdot \tilde{D}_0(z)}{0!} = 0, \\
h_1(z) &= \frac{1 \cdot \tilde{D}_1(z)}{1!} = 1.
\end{aligned}
$$

Substituting this in the previous equation we finally get the following functional equation:

$$
\begin{aligned}
H(3x,z) - 3x &= z\left(e^{2x}H(x,z) - x\right) + z\left(e^{x}\left(H(x,z) - x\right)F(x,z)\right) + \\
& z\left(\left(H(x,z) - x\right)F^2(x,z)\right) + 2zx\left(e^{2x} - 1\right) \\
&= z\left(\left(H(x,z) - x\right)\left(e^{2x} + e^{x}F(x,z) + F^2(x,z)\right) + 3x\left(e^{2x} - 1\right)\right).
\end{aligned}
$$

This equation can be manipulated and written in a different way:

$$
\begin{aligned}
H(3x,z) &= z\left(e^{2x} + e^{x}F(x,z) + F^2(x,z)\right)H(x,z) + \\
& x\left(-ze^{x}F(x,z) - zF^2(x,z) + 2ze^{2x} + 3(1-z)\right).
\end{aligned}
$$

When it is assumed that $F(x,z)$ is a known function, this equation can be used to approximate $H(x_0, z_0)$ numerically for given x_0 and $|z_0| \leq 1$.

In Section 5.2, we have seen a 2-dimensional functional equation for $F(x,z)$, in which $F^3(x,z)$ appeared. $F(x,z)$ was approximated by iteration. To ensure that the approximation converged to $F(x,z)$ (as the number of iterations goes to infinity), some non-trivial things had to be done. Simply truncating after a number of iterations turned out not to be sufficient for convergence. Although in the equation for $H(3x,z)$ no powers appear, the equation exhibits a similar behaviour. This will now be discussed briefly.

The resulting infinite-sum representation for $H(x,z)$ is found to be the following (where an empty product equals 1):

$$\begin{aligned} H(3x,z) &= \sum_{i=0}^{\infty} \left(\prod_{j=1}^{i} z \left(e^{\frac{2x}{3^{j-1}}} + e^{\frac{x}{3^{j-1}}} F(\frac{x}{3^{j-1}}, z) + F^2(\frac{x}{3^{j-1}}, z) \right) \right) \times \\ & \left(\frac{x}{3^i} \left(-z e^{\frac{x}{3^i}} F(\frac{x}{3^i}, z) - z F^2(\frac{x}{3^i}, z) + 2z e^{\frac{2x}{3^i}} + 3(1-z) \right) \right). \end{aligned}$$

First observe that $H(0,z) = h_0(z) = 0$. Therefore, $\lim_{n\to\infty} H(\frac{x}{3^n}, z) = 0$. So truncating the infinite sum in order to obtain a numerical approximation is at first sight justified. Considering the iteration approach as described in section 5.2, this truncation is equivalent to substitution of $H(\frac{x}{3^n}, z) = 0$ after n iterations. As will turn out, the numerical approximation for $H(3x,z)$ that is obtained in this way does not converge to $H(3x,z)$ as $n \to \infty$.

However, when after n iterations the "correct" value is plugged in (instead of plugging in zero), the series of approximations converges as $n \to \infty$. In this functional equation, the "correct" value that has to be plugged in after n iterations turns out to be $\frac{x}{3^n} e^{\frac{x}{3^n}}$. This value is found by requiring that the approximation that is eventually obtained is exact for $z = 1$.

As an illustration, in figure 6.2 the graph with both the divergent and the convergent approximation method for $H(x,z)$ is presented. To keep a two-dimensional plot, x is chosen to be fixed and equal to 1.

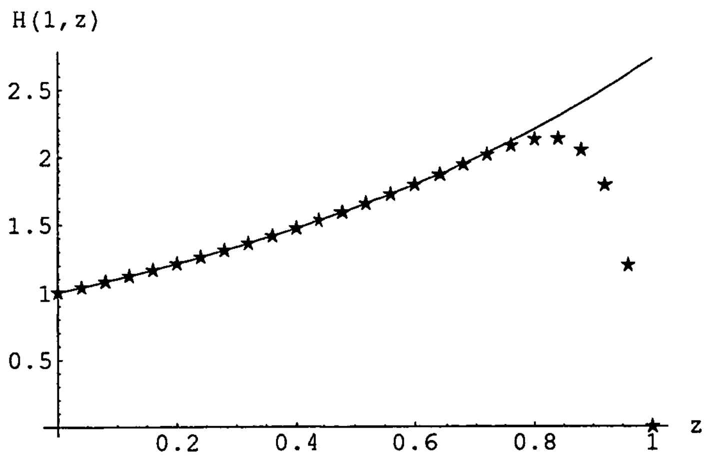

Figure 6.2: Plot of approximations for $H(1, z)$ as a function of z

6.4.2 Relation between $H(x, z)$ and the service time GF

In this subsection, $H(x, z)$ will be related directly to the generating function of the individual service time distribution.

Define $\tilde{D}_{\lambda(s)}$ as the random variable that represents the service time of an arbitrary individual customer (in the model without the arrival slots, the special case of the periodic $Geo|G|1$-queue). Clearly, this random variable depends on $\lambda(s)$, hence a subscript is added. Furthermore, define the corresponding generating function to be:

$$\tilde{D}_{\lambda(s)}(z) = \sum_{k=0}^{\infty} P(\tilde{D}_{\lambda(s)} = k) z^k.$$

From now on, $\lambda(s)$ will be abbreviated to λ. We proceed by deriving an expression for $P(\tilde{D}_\lambda = k)$. We consider an arbitrary customer. Let us say that a customer is of *type* n if it belongs to a tree with initially n customers in it. We condition on the type of customer. The probability $\hat{\pi}(n)$ that a customer is of type n is given by:

$$\hat{\pi}(n) = \frac{n e^{-\lambda} \frac{\lambda^n}{n!}}{\sum_{n=1}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!}}.$$

Using this, we obtain the following:

$$P(\tilde{D}_\lambda = k) = \frac{\sum_{n=1}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!} P\left(\tilde{D}_\lambda(n) = k\right)}{\sum_{n=1}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!}}.$$

Putting everything together, we obtain the following expression for the generating function:

$$\begin{aligned}
\tilde{D}_\lambda(z) &= \sum_{k=0}^{\infty} P(\tilde{D}_\lambda = k) z^k \\
&= \frac{\sum_{k=0}^{\infty} \left(e^{-\lambda} \lambda z + \sum_{n=2}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!} P(\tilde{D}_\lambda(n) = k) z^k \right)}{\sum_{n=1}^{\infty} n e^{-\lambda} \frac{\lambda^n}{n!}} \\
&= e^{-\lambda} \frac{H(\lambda, z) - \lambda + \lambda z}{\lambda}.
\end{aligned}$$

6.5 Determination of the variance of the service time

6.5.1 Introduction

In this section, the variance of the individual service time is determined. In notation, $\lambda(s)$ will again be shortened to λ. The variance of $\tilde{D}_\lambda$, denoted by $Var\left(\tilde{D}_\lambda\right)$, is given by:

$$\begin{aligned}
Var\left(\tilde{D}_\lambda\right) &= \mathbb{E}\left[\tilde{D}_\lambda\right]^2 - \left[\mathbb{E}\tilde{D}_\lambda\right]^2 \\
&= \tilde{D}_\lambda''(1) + \tilde{D}_\lambda'(1) - \left(\tilde{D}_\lambda'(1)\right)^2.
\end{aligned}$$

The aim is to evaluate both $\tilde{D}_\lambda'(1)$ and $\tilde{D}_\lambda''(1)$. Recall that the relation between $\tilde{D}_\lambda(z)$ and $H(\lambda, z)$ was given by:

$$\tilde{D}_\lambda(z) = e^{-\lambda}\frac{H(\lambda, z) - \lambda + \lambda z}{\lambda}.$$

Therefore, the focus is on evaluating $\frac{\partial H}{\partial z}(\lambda, 1)$ as well as $\frac{\partial^2 H}{\partial z^2}(\lambda, 1)$. Because of the fact that in chapter 7, it is required to evaluate $\frac{\partial H}{\partial z}(x, z)$ for general x and z, it will be outlined how this general case can be handled. It will turn out that also the evaluation of both $\frac{\partial F}{\partial z}(x, z)$ and $\frac{\partial^2 F}{\partial z^2}(x, z)$ is necessary. The next four subsections are devoted to the evaluation of these functions.

6.5.2 Evaluation of $\frac{\partial F}{\partial z}(x, z)$

In this subsection the focus is on evaluating $\frac{\partial F}{\partial z}(x, z)$ for arbitrary x and arbitrary z with $|z| \leq 1$. Therefore, we return to the functional equation for $F(x, z)$:

$$F(3x, z) = zF^3(x, z) + (1 + 3x)(1 - z).$$

Taking the first derivative with respect to z on both sides of the equation yields the following:

$$\frac{\partial F}{\partial z}(3x, z) = F^3(x, z) + 3zF^2(x, z)\frac{\partial F}{\partial z}(x, z) - (1 + 3x).$$

When we regard $F(x, z)$ as a known function, which it is in fact, due to the approximation method discussed in chapter 5, this equation can be used to evaluate $\frac{\partial F}{\partial z}(x, z)$.

Iterating finally yields the following expression for $\frac{\partial F}{\partial z}(3x, z)$:

$$\frac{\partial F}{\partial z}(3x, z) = \sum_{k=0}^{\infty}\left(\prod_{j=1}^{k} 3zF^2(\frac{3x}{3^j}, z)\right)\left(F^3(\frac{x}{3^k}, z) - (1 + \frac{3x}{3^k})\right).$$

6.5.3 Evaluation of $\frac{\partial^2 F}{\partial z^2}(x,z)$

In this subsection we consider the evaluation of $\frac{\partial^2 F}{\partial z^2}(x,z)$ for arbitrary x and arbitrary z with $|z| \leq 1$.

Taking the second derivative with respect to z on both sides of the functional equation for $F(x,z)$ yields the following:

$$\frac{\partial^2 F}{\partial z^2}(3x,z) = 6F^2(x,z)\frac{\partial F}{\partial z}(x,z) + 6zF(x,z)\left(\frac{\partial F}{\partial z}(x,z)\right)^2 + 3zF^2(x,z)\frac{\partial^2 F}{\partial z^2}(x,z).$$

When we regard $F(x,z)$ and $\frac{\partial F}{\partial z}(x,z)$ as known functions, which they are in fact, due to the earlier discussed approximation methods, this equation can be used to evaluate $\frac{\partial^2 F}{\partial z^2}(x,z)$ accurately.

Iterating finally yields the following expression for $\frac{\partial^2 F}{\partial z^2}(3x,z)$:

$$\begin{aligned}\frac{\partial^2 F}{\partial z^2}(3x,z) &= \sum_{k=0}^{\infty}\left(\prod_{j=1}^{k} 3zF^2(\frac{3x}{3^j},z)\right)\cdot \\ &\left(6F^2(\frac{x}{3^k},z)\frac{\partial F}{\partial z}(\frac{3x}{3^k},z) + 6zF(\frac{x}{3^k},z)\left(\frac{\partial F}{\partial z}(\frac{3x}{3^k},z)\right)^2\right).\end{aligned}$$

6.5.4 Evaluation of $\frac{\partial H}{\partial z}(x,z)$

Using the results of the previous two subsections, we focus in this subsection on the evaluation of $H(x,z)$ for arbitrary x and arbitrary z with $|z| \leq 1$.

We return to the 2-dimensional functional equation for $H(3x,z)$:

$$\begin{aligned}H(3x,z) &= z\left(e^{2x} + e^x F(x,z) + F^2(x,z)\right)H(x,z) + \\ &\quad x\left(-ze^x F(x,z) - zF^2(x,z) + 2ze^{2x} + 3(1-z)\right).\end{aligned}$$

Now we take on both sides the first derivative with respect to z. This yields the following:

$$
\begin{aligned}
\frac{\partial H}{\partial z}(3x,z) &= (H(x,z)-x)\left(e^{2x}+e^{x}F(x,z)+F^{2}(x,z)\right)+3x\left(e^{2x}-1\right)+\\
&\quad z\frac{\partial H}{\partial z}(x,z)\left(e^{2x}+e^{x}F(x,z)+F^{2}(x,z)\right)+\\
&\quad z\,(H(x,z)-x)\left(e^{x}\frac{\partial F}{\partial z}(x,z)+2F(x,z)\frac{\partial F}{\partial z}(x,z)\right).
\end{aligned}
$$

When we assume that both $F(x,z)$ and $\frac{\partial F}{\partial z}(x,z)$ and also $H(x,z)$ are known functions, then this equation can be used to evaluate $\frac{\partial H}{\partial z}(x,z)$.

Iterating finally yields the following:

$$
\begin{aligned}
\frac{\partial H}{\partial z}(3x,z) &= \sum_{i=0}^{\infty}\left(\prod_{j=1}^{i} z\left(e^{\frac{2x}{3^{j-1}}}+e^{\frac{x}{3^{j-1}}}F(\frac{x}{3^{j-1}},z)+F^{2}(\frac{x}{3^{j-1}},z)\right)\right)\cdot\\
&\quad \left(\left(H(\frac{x}{3^{i}},z)-\frac{x}{3^{i}}\right)\left(e^{\frac{2x}{3^{i}}}+e^{\frac{x}{3^{i}}}F(\frac{x}{3^{i}},z)+F^{2}(\frac{x}{3^{i}},z)+\right.\right.\\
&\quad \left.\left. z\left(e^{\frac{x}{3^{i}}}\frac{\partial F}{\partial z}(\frac{x}{3^{i}},z)+2F(\frac{x}{3^{i}},z)\frac{\partial F}{\partial z}(\frac{x}{3^{i}},z)\right)\right)+3\frac{x}{3^{i}}\left(e^{\frac{2x}{3^{i}}}-1\right)\right).
\end{aligned}
$$

6.5.5 Evaluation of $\frac{\partial^2 H}{\partial z^2}(x,z)$

Finally, the evaluation of $\frac{\partial^2 H}{\partial z^2}(x,z)$ is considered. The functional equation for $H(3x,z)$ is used again.

Taking the second derivative with respect to z on both sides of the functional equation yields the following:

$$
\begin{aligned}
\frac{\partial^2 H}{\partial z^2}(3x,z) &= 2\left(e^{2x}+e^{x}F(x,z)+F(x,z)^2+z\left(e^{x}\frac{\partial F}{\partial z}(x,z)+2F(x,z)\frac{\partial F}{\partial z}(x,z)\right)\right)\cdot\\
&\quad \frac{\partial H}{\partial z}(x,z)-e^{x}x\left(2\frac{\partial F}{\partial z}(x,z)+z\frac{\partial^2 F}{\partial z^2}(x,z)\right)-\\
&\quad x\left(4F(x,z)\frac{\partial F}{\partial z}(x,z)+z\left(2\left(\frac{\partial F}{\partial z}(x,z)\right)^2+2F(x,z)\frac{\partial^2 F}{\partial z^2}(x,z)\right)\right)+
\end{aligned}
$$

$$
\begin{aligned}
&H(x,z)\left(2\left(e^{x}\frac{\partial F}{\partial z}(x,z)+2\,F(x,z)\,\frac{\partial F}{\partial z}(x,z)\right)+\right.\\
&z\left.\left(2\left(\frac{\partial F}{\partial z}(x,z)\right)^{2}+e^{x}\,\frac{\partial^2 F}{\partial z^2}(x,z)+2\,F(x,z)\,\frac{\partial^2 F}{\partial z^2}(x,z)\right)\right)+\\
&z\left(e^{2x}+e^{x}\,F(x,z)+F(x,z)^2\right)\frac{\partial^2 H}{\partial z^2}(x,z).
\end{aligned}
$$

Again, iteration yields an infinite-sum representation for $\frac{\partial^2 H}{\partial z^2}(3x,z)$, which will not be presented here.

As an illustration, the variance of the individual service time as a function of μ is plotted for three different frame lengths. The resulting graph is depicted in figure 6.3. Furthermore, we can mention that these analytically obtained results on the variance of the individual service time closely agree with the simulation results that are found. For more on these simulation results, see Chapter 14.

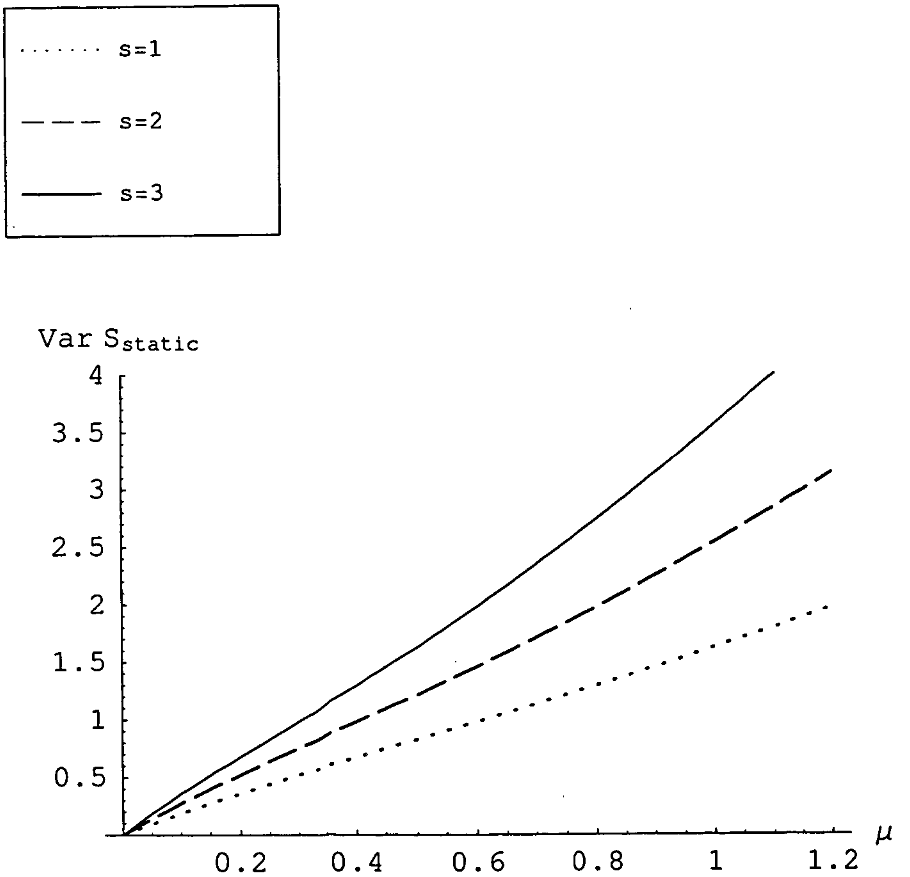

Figure 6.3: Variance of the individual service time as a function of μ.

6.6 Covariance between $\tilde{B}$ and $\tilde{S}$

In this section a functional equation will be derived from which the covariance between the tree length and the aggregated service time can be obtained. In the derivation of the functional equation, the results presented in the previous sections of this chapter are used.

The following generating function is considered:

$$\tilde{M}(x) \quad := \quad \sum_{n=0}^{\infty} \frac{\mathbb{E}\left[\tilde{B}(n)\tilde{S}(n)\right]}{n!} x^n.$$

There is clearly a dependence between the two variables $\tilde{B}(n)$ and $\tilde{S}(n)$ as long as they correspond to the same tree.

Furthermore, define the following generating function:

$$R(x) \quad := \quad \sum_{n=0}^{\infty} \frac{\mathbb{E}\tilde{B}^2(n)}{n!} x^n.$$

A remark is that $R(x)$ is related to $\tilde{C}(x)$ (which was introduced in Section 5.5) and $\tilde{F}(x)$ as follows:

$$R(x) \quad = \quad \tilde{C}(x) + \tilde{F}(x).$$

So $R(x)$ can be represented directly as an infinite sum, due to the given representation of $\tilde{C}(x)$ and $\tilde{F}(x)$ in sections 5.4 and 5.5.

Now consider the joint distribution of $\tilde{B}(n)$ and $\tilde{S}(n)$, $n \geq 2$. The following recursion holds, where $\xi(n_1, n_2, n_3)$ is defined as earlier:

$$P(\tilde{B}(n) = m, \tilde{S}(n) = k) \quad = \sum_{\substack{n_1, n_2 \neq 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \cdot$$

$$P\left[\tilde{B}(n_1) + \tilde{B}(n_2) + \tilde{B}(n_3) + 1 = m, \tilde{S}(n_1) + \tilde{S}(n_2)\right.$$

$$+\tilde{S}(n_3) + (n_2+n_3)\tilde{B}(n_1) + n_3\tilde{B}(n_2) + n = k\Big] +$$

$$\sum_{\substack{n_1,n_2=1,n_3\neq 1\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3) P\Big[\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1=m,$$

$$\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n_3\tilde{B}(n_1)+n_3\tilde{B}(n_2)+n=k\Big] +$$

$$\sum_{\substack{n_1,n_2\neq 1,n_3=1\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3) P\Big[\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1=m,$$

$$\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n_2\tilde{B}(n_1)+n=k\Big] +$$

$$\sum_{\substack{n_1,n_2=1,n_3=1\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3) P\Big[\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1=m,$$

$$\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n=k\Big],$$

$$n\geq 2, \qquad k=1,2,\ldots, \qquad m=1,2,\ldots$$

When we multiply both sides of the equation with $k\cdot m\cdot 3^n\cdot\dfrac{x^n}{n!}$ and sum over $n=2,3,\ldots$, $k=1,2,\ldots$ and $m=1,2,\ldots$, we obtain the following:

$$\sum_{n=2}^{\infty}\frac{\mathbb{E}\left[\tilde{B}(n)\tilde{S}(n)\right]}{n!}(3x)^n = \sum_{n=2}^{\infty}\sum_{\substack{n_1,n_2\neq 1,n_3\neq 1\\ n_1+n_2+n_3=n}}\frac{x^n}{n_1!n_2!n_3!}\mathbb{E}\Big[\Big(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1\Big)\cdot$$

$$\Big(\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+(n_2+n_3)\tilde{B}(n_1)+n_3\tilde{B}(n_2)+n\Big)\Big] +$$

$$\sum_{n=2}^{\infty}\sum_{\substack{n_1,n_2=1,n_3\neq 1\\ n_1+n_2+n_3=n}}\frac{x^n}{n_1!n_2!n_3!}\mathbb{E}\Big[\Big(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1\Big)\cdot$$

$$\Big(\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n_3\tilde{B}(n_1)+n_3\tilde{B}(n_2)+n\Big)\Big] +$$

$$\sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2\neq 1,n_3=1\\ n_1+n_2+n_3=n}} \frac{x^n}{n_1!n_2!n_{3!}} \mathbb{E}\left[\left(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1\right)\cdot\right.$$
$$\left.\left(\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n_2\tilde{B}(n_1)+n\right)\right]+$$
$$\sum_{n=2}^{\infty} \sum_{\substack{n_1,n_2=1,n_3=1\\ n_1+n_2+n_3=n}} \frac{x^n}{n_1!n_2!n_{3!}} \mathbb{E}\left[\left(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)+1\right)\cdot\right.$$
$$\left.\left(\tilde{S}(n_1)+\tilde{S}(n_2)+\tilde{S}(n_3)+n\right)\right].$$

Expanding the expectation terms appearing on the right-hand side of the previous equation, yields after some manipulation the following:

$$\begin{aligned}
\tilde{M}(3x) &= 2\,e^x\,(-1+e^x)\,(e^x-x)\,x+e^x\,(e^x-x)^2\,x+2\,e^x\,x^2+e^x\,x^3+\\
&3\,e^x\,(-1+e^x)\,x\,\tilde{F}(x)+2\,e^x\,(e^x-x)\,x\,\tilde{F}(x)+4\,(-1+e^x)\,(e^x-x)\,x\,\tilde{F}(x)+\\
&2\,x^2\,\tilde{F}(x)+3\,(-1+e^x)\,x\,\tilde{F}(x)^2+(e^x-x)\,x\,\tilde{F}(x)^2+2\,e^x\,(e^x-x)\,\tilde{M}(x)+\\
&(e^x-x)^2\,\tilde{M}(x)+x^2\,\tilde{M}(x)+2\,e^x\,(e^x-x)\,Q(x)+(e^x-x)^2\,Q(x)+\\
&x^2\,Q(x)+6\,(e^x-x)\,\tilde{F}(x)\,Q(x)+e^x\,(-1+e^x)\,x\,R(x)+\\
&2\,(-1+e^x)\,(e^x-x)\,x\,R(x)+2\,e^x\,(e^x-x)\,x\,\tilde{F}'(x)+(e^x-x)^2\,x\,\tilde{F}'(x)+\\
&x^3\,\tilde{F}'(x)+e^x\,x\,\tilde{F}(x)\,\tilde{F}'(x)+(e^x-x)\,x\,\tilde{F}(x)\,\tilde{F}'(x)+\\
&2\,\{e^x\,(e^x-x)\,x+e^x\,(-1+e^x)\,x^2+e^x\,(e^x-x)\,x^2+(e^x-x)\,x\,\tilde{F}(x)+\\
&2\,(-1+e^x)\,x^2\,\tilde{F}(x)+(e^x-x)\,x^2\,\tilde{F}(x)+e^x\,x\,\tilde{M}(x)+(e^x-x)\,x\,\tilde{M}(x)+\\
&e^x\,x\,Q(x)+(e^x-x)\,x\,Q(x)+2\,x\,\tilde{F}(x)\,Q(x)+(-1+e^x)\,x^2\,R(x)+e^x\,x^2\,\tilde{F}'(x)\}\\
&= 3\,e^{3x}\,x-x\,(3-4\,e^x+x)\,\tilde{F}(x)^2+3\,e^{2x}\,\tilde{M}(x)+3\,e^{2x}\,Q(x)-
\end{aligned}$$

$$R(x)\left(3\,e^{x}\,x-3\,e^{2\,x}\,x\right)+\tilde{F}'(x)\left(3\,e^{2\,x}\,x-2\,e^{x}\,x^{2}+2\,x^{3}\right)+$$

$$\tilde{F}(x)\left(\left(6\,e^{x}-2\,x\right)\,Q(x)-x\left(5\,e^{x}-9\,e^{2\,x}+2\,x^{2}+\left(-2\,e^{x}+x\right)\,\tilde{F}'(x)\right)\right).$$

Since $\tilde{F}(x)$, $\tilde{F}'(x)$, $Q(x)$ and $R(x)$ are known functions (in the sense that they can be represented by infinite sums and approximated by truncating these sums), this last functional equation can be used to represent $\tilde{M}(3x)$ by means of an infinite sum as well. This can be done analogously as in several cases before.

For the covariance between $\tilde{B}$ and $\tilde{S}$, denoted by $Cov(\tilde{B},\tilde{S})$, the following holds:

$$\begin{aligned}
Cov(\tilde{B},\tilde{S}) &= \mathbb{E}\tilde{B}\tilde{S}-\mathbb{E}\tilde{B}\mathbb{E}\tilde{S}\\
&= e^{-\lambda}\lambda+\sum_{n=2}^{\infty}\mathbb{E}\tilde{B}(n)\tilde{S}(n)e^{-\lambda}\frac{\lambda^{n}}{n!}-\\
&\quad\left(e^{-\lambda}(1+\lambda)+\sum_{n=2}^{\infty}\mathbb{E}\tilde{B}(n)e^{-\lambda}\frac{\lambda^{n}}{n!}\right)\cdot\left(e^{-\lambda}\lambda+\sum_{n=2}^{\infty}\mathbb{E}\tilde{S}(n)e^{-\lambda}\frac{\lambda^{n}}{n!}\right)\\
&= e^{-\lambda}\left(\tilde{M}(\lambda)+\lambda\right)-e^{-2\lambda}\left(\tilde{F}(\lambda)+1+\lambda\right)\left(Q(\lambda)+\lambda\right).
\end{aligned}$$

This shows that the covariance can be determined straightforwardly when the three functions can be evaluated accurately.

Chapter 7

Translation to the static arrival slot mechanism

In the first part of this chapter the focus will be on a general approach to relate the waiting time, service time and sojourn time results that were found in the mechanism "without an arrival slot" to the static arrival slot mechanism. This approach will use the generating functions of waiting, service and sojourn time distributions. Furthermore, an approximation method will be presented that can be used instead of the exact generating function approach.

In the second part of this chapter, the focus is on applying the exact method as well as the approximation method to the waiting and sojourn time of *unlucky* customers (i.e. customers that are part of a super customer). The methods are applied to find expressions for the first two moments.

After that, in the third part of this chapter the results will be used to obtain expressions for the first moment as well as for the variance of the waiting and sojourn time of *individual* customers, incorporating the fact that there are both lucky customers which leave the system in the first arrival slot and unlucky customers which experience additional waiting time in the tree-queue.

7.1 Exact translation approach

In this section, an exact method will be developed that translates generating functions of the waiting time, service time and sojourn time distribution in the model "without an arrival slot" to generating functions of these distributions in the static (as well as the dynamic) arrival slot mechanism. In the remaining part of this section, the approach will be illustrated using the second part of the waiting time. At the end of the chapter the method will eventually be used for translating the sojourn time distributions.

As said before, the translation method will be outlined using the second component of the waiting time as an example. The translation method is based on the derivation of a relation between the two generating functions of the waiting time, so in both the model with and the model without the actual arrival slots. So the problem is to "add" the arrival slots, which contribute to the waiting time. Frames consist of t slots and $t+1$ slots respectively.

First some additional notation will be introduced.

Let H_λ be a random variable which has as distribution the steady-state distribution of the waiting time in the static arrival slot mechanism. Clearly, H depends on $\lambda(t)$ and therefore a subscript λ is added. Define:

$$h_k = P[H_\lambda = k], \qquad k = 0, 1, 2, \ldots$$

Furthermore, define the generating function corresponding to the distribution of H_λ to be:

$$H_\lambda(z) = \sum_{k=0}^{\infty} h_k z^k, \qquad |z| \leq 1.$$

Consider the discrete Markov chain defined by the amount of work at the queue just before the first slot of every frame. This total amount of work is defined as the total number of contention slots it takes to complete service of all customers that are currently in the system. So arrival slots are not counted. This Markov chain is in both models the same. Because of the fact that in the arrival slot model every first slot of a frame is dedicated to new arrivals, the time it takes (measured in slots) to complete the service of all customers that are currently in the system will be longer than in the other model in which the waiting time is equal to the total amount of work in the system.

There is a relation between the total amount of work k in the arrival slot system and the time n it takes to complete service of all these jobs. This relation can be obtained in a straightforward manner. Because of the fact that this relation depends on the actual value of t, an index t is added to k

and n:

$$n_t(k) = (k(t+1)) \text{ div } t.$$

Using this relation, we are going to derive a relation between $H_\lambda(z)$ and $W_\lambda(z)$. First, some notation will be introduced. For compactness, the subscript λ is omitted:

$$W_i(z) = \sum_{k=0}^{\infty} w_{kt+i} z^{kt+i}, \qquad i = 0, 1, \ldots, t-1.$$

More specifically, a relation between $H_\lambda(z)$ and the $W_i(z)$'s will be derived:

$$\begin{aligned}
H_\lambda(z) &= \sum_{n=0}^{\infty} h_n z^n \\
&= \sum_{k=0}^{\infty} h_{n_t(k)} z^{n_t(k)} \\
&= \sum_{k=0}^{\infty} w_k z^{(k(t+1))\text{div } t} \\
&= \sum_{i=0}^{t-1} \sum_{k=0}^{\infty} w_{kt+i} z^{((kt+i)(t+1))\text{div } t} \\
&= \sum_{i=0}^{t-1} \sum_{k=0}^{\infty} w_{kt+i} z^{k(t+1)+i} \\
&= \sum_{i=0}^{t-1} z^{\frac{-i}{t}} \sum_{k=0}^{\infty} w_{kt+i} (z^{\frac{t+1}{t}})^{kt+i} \\
&= \sum_{i=0}^{t-1} z^{\frac{-i}{t}} W_i(z^{\frac{t+1}{t}}).
\end{aligned}$$

It is assumed that $W_\lambda(z)$ is a known function. In our case this is not exactly true, but it will be indicated later how this approach can be used, even when $W_\lambda(z)$ is not explicitly known. Now it will be shown how the $W_i(z)$'s can be obtained from $W_\lambda(z)$. It turns out that only a system of t linear equations has to be solved. There is one linear equation relating $W_\lambda(z)$ and the $W_i(z)$'s that is obvious:

$$W_\lambda(z) = \sum_{i=0}^{t-1} W_i(z).$$

But there are more linear equations. To see this, define the following:

$$\beta_t = e^{\frac{2}{t}\pi i}.$$

Obviously we have that:

$$\begin{aligned} |\beta_t| &= 1, \\ (\beta_t)^t &= 1. \end{aligned}$$

To obtain t linear equations we successively substitute $x_0 = (\beta_t)^0 z$, $x_1 = (\beta_t)^1 z$, ..., $x_{t-1} = (\beta_t)^{t-1} z$, in $W_\lambda(z) = \sum_{i=0}^{t-1} W_i(z)$. This leads to the following set of equations:

$$\begin{aligned} W_\lambda((\beta_t)^0 z) &= \sum_{i=0}^{t-1} W_i((\beta_t)^0 z), \\ W_\lambda((\beta_t)^1 z) &= \sum_{i=0}^{t-1} W_i((\beta_t)^1 z), \\ \ldots &= \ldots \\ W_\lambda((\beta_t)^{t-1} z) &= \sum_{i=0}^{t-1} W_i((\beta_t)^{t-1} z). \end{aligned}$$

These equations can be simplified. Look at the $(j+1)^{th}$ equation:

$$W_\lambda((\beta_t)^j z) = \sum_{i=0}^{t-1} W_i((\beta_t)^j z)$$

$$
\begin{aligned}
&= \sum_{i=0}^{t-1}\sum_{k=0}^{\infty} w_{kt+i}((\beta_t)^j z)^{kt+i} \\
&= \sum_{i=0}^{t-1}\sum_{k=0}^{\infty} w_{kt+i}((\beta_t)^j)^{kt+i}(z)^{kt+i} \\
&= \sum_{i=0}^{t-1}\sum_{k=0}^{\infty} w_{kt+i}(\beta_t)^{ij}(z)^{kt+i} \\
&= \sum_{i=0}^{t-1}(\beta_t)^{ij} W_i(z).
\end{aligned}
$$

We have used that $(\beta_t)^{kt} = 1$. So when $W((\beta_t)^j z)$ is known for $j = 0, 1, \ldots, t-1$ this leads to a set of t linear equations with exactly t unknown functions. It can be shown that this set of equations is always linearly independent, so that the unknown functions can be found by solving the equations. To see that this set of equations is independent, observe the following.

The set of equations has a special structure, which is known as a *Van der Monde* structure. A set of n equations in n unknowns given by $y_1, \ldots, y_n$ has a Van der Monde structure if the set of equations can be written in the following way:

$$
\underline{b} = A\underline{y},
$$

with $\underline{b} \in \mathbb{C}^n$ and A a $n \times n$-matrix with coefficients in $\mathbb{C}$. This coefficient matrix A must have the following structure:

$$
A = \begin{pmatrix} a_1^0 & a_2^0 & \ldots & a_n^0 \\ a_1^1 & a_2^1 & \ldots & a_n^1 \\ \ldots & \ldots & \ldots & \ldots \\ a_1^{n-1} & a_2^{n-1} & \ldots & a_n^{n-1} \end{pmatrix}.
$$

If $\underline{b}$ has this same structure, the Van der Monde set of equations has an explicit solution in its coefficients $a_1, \ldots, a_n$ and $\underline{b}_1, \ldots, \underline{b}_n$, which makes it at least numerically easier to obtain the unknowns. However, $\underline{b}$ does not have this structure, so the numerical advantages cannot be fully exploited. But the fact that A does have this structure, implies directly that the determinant of A is unequal to zero. Hence, the set of equations is independent.

When the $W_i(z)$'s are obtained they can be plugged into the equation for $H_\lambda(z)$.

Two remarks can be made:

1. An important remark is that this procedure can also be used to obtain derivatives. When for example one needs to find the $W_i'(z)$'s, this can be done by taking the first derivative on both sides of the t equations. Another set of t linearly independent equations that can be solved is obtained when $W_\lambda(z)$ (and as a result $W_\lambda'(z)$) is explicitly known. This set of equations has again a Van der Monde structure.

2. Another remark is that when $W_\lambda(z)$ (or $W_\lambda^{(n)}(z)$) is not explicitly known as a function of z, but it is possible to evaluate $W_\lambda(z)$ (or $W_\lambda^{(n)}(z)$) for $|z| \leq 1$, the described procedure can be used to evaluate the $W_i(z)$'s (or $W_i^{(n)}(z)$'s) for $|z| \leq 1$. The n^{th} derivative of $W_\lambda(z)$ with respect to z is denoted by $W_\lambda^{(n)}(z)$.

7.2 An approximation method

In the previous section it was indicated how $H_\lambda(z)$ can be obtained exactly from the generating function of the waiting time. It is of course also possible to approximate $H_\lambda(z)$. In this section it will be shown how this can be done. Again, the waiting time is used to illustrate the approach, which can also be used for service and sojourn time approximations.

The difficulty in the exact analysis comes from the fact that there is a non-linear relation between n_t and k_t. Most of the times an increase of 1 for k_t will lead to an increase of 1 for n_t, but sometimes it will lead to an increase of 2, because of an arrival slot that is given priority. But an increase of t for k_t will always lead to an increase of $t+1$ for n_t. The idea is to approximate the non-linear relation $n_t(k)$ by a linear relation, denoted by $\tilde{n}_t(k)$. Expressions for $n_t(k)$ as well as for $\tilde{n}_t(k)$ are the following:

$$
\begin{aligned}
n_t(k) &= (k(t+1)) \text{ div } t, \\
\tilde{n}_t(k) &= \begin{cases} 0 & \text{if } k = 0 \\ k\frac{t+1}{t} & \text{if } k > 0 \end{cases} .
\end{aligned}
$$

In this way, there is a linear relationship between the total amount of work and the time it takes to complete this total amount. The interpretation is that the arrival slot that will be used at the beginning of a frame is equally divided over three consecutive contention slots (excluding the arrival slot), removing the jump that actually takes place each t units of work. So in fact, the total time it takes to complete an amount k of work in the system is overestimated if $k \bmod t \neq 0$ and exactly determined if $k \bmod t = 0$. How this approximation of $n_t(k)$ can be used to approximate $H_\lambda(z)$ will be shown now:

$$
\begin{aligned}
H_\lambda(z) &= \sum_{n=0}^{\infty} h_n z^n \\
&= \sum_{k=0}^{\infty} h_{n_t(k)} z^{n_t(k)} \\
&= \sum_{k=0}^{\infty} w_k z^{n_t(k)}
\end{aligned}
$$

$$
\begin{aligned}
&\approx \sum_{k=0}^{\infty} w_k z^{\tilde{n}_t(k)} \\
&= w_0 + \sum_{k=1}^{\infty} w_k z^{k\frac{t+1}{t}} \\
&= w_0 + \sum_{k=1}^{\infty} w_k z^{k\frac{t+1}{t}} \\
&= w_0 + \sum_{k=1}^{\infty} w_k (z^{\frac{t+1}{t}})^k \\
&= W_\lambda(z^{\frac{t+1}{t}}) \equiv \tilde{H}_1(z).
\end{aligned}
$$

So when $W_\lambda(z)$ is known, this leads to an approximation of $H_\lambda(z)$. How good this approximation is depends on the actual distribution of W_λ. When w_0 is relatively large, the approximation will be good because of the fact that $n_t(0) = \tilde{n}_t(0)$. Since $w_0, \ldots, w_{t-1}$ have to be calculated when determining $W_\lambda(z)$, these probabilities can be used to improve upon the approximation by making $\tilde{n}_t(k)$ exact for $k = 0, \ldots, t-1$. So:

$$
\begin{aligned}
n_t(k) &= (k(t+1)) \text{ div } t, \\
\tilde{n}_t(k) &= \begin{cases} (k(t+1)) \text{ div } t & \text{if } 0 \le k \le t-1 \\ k\frac{t+1}{t} & \text{if } k > t-1 \end{cases} .
\end{aligned}
$$

Proceeding in the same way as before this leads to the following approximation:

$$
\begin{aligned}
H_\lambda(z) &\approx \sum_{k=0}^{\infty} w_k z^{\tilde{n}_t(k)} \\
&= \sum_{k=0}^{t-1} w_k z^{(k(t+1)) \text{ div } t} + \sum_{k=t}^{\infty} w_k z^{k\frac{t+1}{t}} \\
&= \sum_{k=0}^{t-1} w_k z^{(k(t+1)) \text{ div } t} + \sum_{k=t}^{\infty} w_k (z^{\frac{t+1}{t}})^k
\end{aligned}
$$

$$= \sum_{k=0}^{t-1} w_k z^{(k(t+1)) \text{ div } t} + W_\lambda(z^{\frac{t+1}{t}}) - \sum_{k=0}^{t-1} w_k (z^{\frac{t+1}{t}})^k \equiv \tilde{H}_2(z).$$

Of course one can still further improve upon the approximation by calculating more probabilities than just $w_0, \ldots, w_{t-1}$ and incorporating these in the approximation.

Another remark is that the used approximations are in fact upper bounds on the waiting time. The same approach can be followed in deriving lower bounds. This comes down to "dividing" the arrival slot at the beginning of a frame over the contention slots of the previous frame. This will not explicitly be done now. The upper bound approximation is sufficient for the moment.

7.3 Exact translation to the arrival slot mechanism

In this section, the translation method will be applied to find an expression for the mean value of the second component of the waiting time which is equal to the mean waiting time of a super customer. This expression can be used later, to determine the mean waiting time of an individual customer. Furthermore, the mean *remaining* (i.e., starting just after the arrival slot) sojourn time of an *unlucky* individual customer in the arrival slot mechanism with $t+1$ slots per frame is analyzed. The procedure can be extended to find higher moments of these variables as well. The second moment will also be analyzed.

7.3.1 Mean waiting time of a super customer

The mean waiting time of an individual customer consists of two components. The first component will be analyzed in Section 7.5. Now, the focus will be on the second component, which is equal to the waiting time of a super customer.

The exact method to find the first moment of H_λ uses the obtained exact expression for $H_\lambda(z)$. Determination of $H'_\lambda(1)$ will then give an exact result. The expression for $H'_\lambda(1)$ is as follows:

$$\begin{aligned} H'_\lambda(1) &= \left(\sum_{i=0}^{t-1} z^{\frac{-i}{t}} W_i(z^{\frac{t+1}{t}})\right)'_{z=1} \\ &= \left(\sum_{i=0}^{t-1} \frac{-i}{t} z^{\frac{-i-t}{t}} W_i(z^{\frac{t+1}{t}}) + \sum_{i=0}^{t-1} \frac{t+1}{t} z^{\frac{1}{t}} z^{\frac{-i}{t}} W'_i(z^{\frac{t+1}{t}})\right)_{z=1} \\ &= \sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1). \end{aligned}$$

In this expression for $H'_\lambda(1)$ terms appear like $W_i(1)$ and $W'_i(1)$ for $i = 0, \ldots, t-1$. Because of the fact that $W_\lambda(z)$ can be evaluated for arbitrary $|z| \leq 1$, the $W_i(1)$'s can be found. For finding the $W'_i(1)$'s, it is necessary that $W'_\lambda(z)$ can be evaluated for arbitrary $|z| \leq 1$. One can obtain an exact expression for $W'_\lambda(z)$ in a straightforward manner. $B'_\lambda(z)$ will appear in this expression as the only unknown function of z. As we have seen, $B_\lambda(z)$ is directly related to $F(\lambda, z)$. This relation shows that when one is able to evaluate $\frac{\partial F}{\partial z}(\lambda, z)$ accurately, it is straightforward to evaluate $B'_\lambda(z)$ accurately. In the previous chapter, it has been indicated how $\frac{\partial F}{\partial z}(\lambda, z)$ can be evaluated.

7.3.2 Variance of the waiting time of a super customer

In this subsection, the variance of the waiting time of a super customer is considered. This is an important quantity for determining the variance of the individual waiting time, later in this chapter.

For the variance of the waiting time of a super customer the following holds:

$$
\begin{aligned}
Var(H_\lambda) &= H''_\lambda(1) + H'_\lambda(1) - (H'_\lambda(1))^2 \\
&= \left.\left(\sum_{i=0}^{t-1} z^{\frac{-i}{t}} W_i(z^{\frac{t+1}{t}})\right)''\right|_{z=1} + \sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1) - \\
&\quad \left(\sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1)\right)^2 \\
&= \sum_{i=0}^{t-1} \frac{i(i+t)}{t^2} W_i(1) + \sum_{i=0}^{t-1} \frac{-i(t+1)}{t^2} W'_i(1) + \\
&\quad \sum_{i=0}^{t-1} \frac{(t+1)}{t^2} \left(W'_i(1) + (t+1) W''_i(1)\right) + \sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1) - \\
&\quad \left(\sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1)\right)^2 \\
&= \sum_{i=0}^{t-1} \frac{i^2}{t^2} W_i(1) + \sum_{i=0}^{t-1} \frac{(t+1-i)(t+1)}{t^2} W'_i(1) + \sum_{i=0}^{t-1} \frac{(t+1)^2}{t^2} W''_i(1) - \\
&\quad \left(\sum_{i=0}^{t-1} \frac{-i}{t} W_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t} W'_i(1)\right)^2 .
\end{aligned}
$$

In this expression for the variance of H_λ, terms appear like $W_i(1)$, $W'_i(1)$ and $W''_i(1)$ for $i = 0, \ldots, t-1$. In the previous section, it has already been shown how the first two can be found. For determination of the $W_i''(1)$'s, it is necessary that $W_\lambda''(z)$ can be evaluated for arbitrary $|z| \leq 1$. In the expression for $W_\lambda''(z)$, $B_\lambda''(z)$ will show up as the only yet unknown (in the sense that it cannot be evaluated) function of z Because of the relation between $B_\lambda(z)$ and $F(\lambda, z)$, it holds that when one is able to evaluate $\frac{\partial^2 F}{\partial z^2}(\lambda, z)$ accurately, $B_\lambda''(z)$ can be evaluated accurately. In the previous chapter, it has been indicated how $\frac{\partial^2 F}{\partial z^2}(\lambda, z)$ can be evaluated. This makes that the variance can be found numerically.

7.3.3 The mean remaining individual sojourn time

The second component of the waiting time always starts immediately after an arrival slot. For the (remaining) service time this is different. Service may start with different probabilities in each of the t remaining slots of a frame. This makes that the translation approach is not directly applicable to the service time. But for an individual customer who is present in the tree queue, service (re)starts immediately after the super customer he belongs to leaves the tree queue.

So when we consider the second component of the waiting time together with the *remaining* service time, this always starts just after an arrival slot. The sojourn time of an individual customer that is part of a super customer, a so-called *unlucky* customer, is exactly this quantity plus the first component of the waiting time, which has already been analyzed. The sojourn time of a *lucky* individual customer equals the first component of the waiting time plus exactly one (arrival) slot. In this subsection, we focus on the non-trivial case: the (remaining) sojourn time of an unlucky customer.

Let $\tilde{T}$ be the random variable that represents this remaining sojourn time of an individual customer. Clearly, $\tilde{T}$ depends on $\lambda(s)$ (which will again be abbreviated to λ in this section), and therefore a subscript λ is added. The following holds:

$$\tilde{T}_\lambda = W_\lambda + \hat{D}_\lambda,$$

where $\hat{D}_\lambda$ is the variable that represents the remaining service time of an individual customer. There is a relation between $\hat{D}_\lambda$ and $\tilde{D}_\lambda$:

$$\tilde{D}_\lambda = \begin{cases} 1 & \text{with probability } p_1 \\ 1 + \hat{D}_\lambda & \text{with probability } 1 - p_1 \end{cases},$$

$$\text{with } p_1 = P(\tilde{D}_\lambda = 1) = \tilde{D}'(0).$$

This implies a relation between the generating function of $\hat{D}_\lambda$, denoted by $\hat{D}_\lambda(z)$, and $\tilde{D}_\lambda(z)$. This relation can be found as follows:

$$\begin{aligned} \tilde{D}_\lambda(z) &= \sum_{k=0}^{\infty} P(\tilde{D}_\lambda = k) z^k \\ &= p_1 \sum_{k=0}^{\infty} P(1 = k) z^k + (1 - p_1) P(1 + \hat{D}_\lambda = k) z^k \\ &= p_1 z + (1 - p_1) z \hat{D}_\lambda(z), \end{aligned}$$

or equivalently:

$$\hat{D}_\lambda(z) = \frac{\tilde{D}_\lambda(z) - p_1 z}{(1 - p_1) z}.$$

For the generating function corresponding to $\tilde{T}$, denoted by $\tilde{T}_\lambda(z)$, the following holds, due to the independence of W_λ and $\hat{D}_\lambda$:

$$\tilde{T}_\lambda(z) = W_\lambda(z) \hat{D}_\lambda(z).$$

Define $\tilde{U}_\lambda$ as the random variable that represents the remaining sojourn time in the static arrival slot mechanism. $\tilde{U}_\lambda(z)$ is the corresponding generating function. Translating the generating function $\tilde{T}_\lambda(z)$ to the generating function $\tilde{U}_\lambda(z)$ finally yields the following:

$$\tilde{U}_\lambda(z) = \sum_{i=0}^{t-1} z^{\frac{-i}{t}} T_i(z^{\frac{i+1}{t}}).$$

For the expectation of $\tilde{U}_\lambda$, denoted by $\mathbb{E}\tilde{U}_\lambda$, the following holds:

$$\mathbb{E}\tilde{U}_\lambda = \sum_{i=0}^{t-1} \frac{-i}{t}\tilde{T}_i(1) + \sum_{i=0}^{t-1} \frac{t+1}{t}\tilde{T}_i'(1).$$

$\tilde{T}_\lambda(z)$ can be evaluated for arbitrary $|z| \leq 1$. This is true because $W_\lambda(z)$ and $\hat{D}_\lambda(z)$ can be evaluated, the latter because of the fact that $\tilde{D}_\lambda(z)$ can be evaluated. This makes that the $\tilde{T}_i(1)$'s can be found.

The $\tilde{T}_i'(1)$'s can be found if the derivative of $\tilde{T}_\lambda(z)$ can be evaluated for all z with $|z| \leq 1$. For the derivative of $\tilde{T}_\lambda(z)$ the following holds:

$$\tilde{T}_\lambda'(z) = W_\lambda'(z)\hat{D}_\lambda(z) + W_\lambda(z)\hat{D}_\lambda'(z).$$

If we want that we can evaluate $\tilde{T}_\lambda'(z)$ for all z with $|z| \leq 1$, it is needed that $W_\lambda'(z)$ and $\tilde{D}_\lambda'(z)$ can be evaluated. In the previous subsection it was shown that $W_\lambda'(z)$ can be evaluated accurately. So the only thing we still need is that $\hat{D}_\lambda'(z)$ can be evaluated for all z with $|z| \leq 1$. For $\hat{D}_\lambda'(z)$ the following holds:

$$\begin{aligned}\hat{D}_\lambda'(z) &= \frac{\left(\tilde{D}_\lambda'(z) - p_1\right)(1-p_1)z - (1-p_1)\left(\tilde{D}_\lambda(z) - p_1 z\right)}{(1-p_1)^2 z^2} \\ &= \frac{\tilde{D}_\lambda'(z)z - \tilde{D}_\lambda(z)}{(1-p_1)z^2}.\end{aligned}$$

So the only thing we still need is that $\tilde{D}_\lambda'(z)$ can be evaluated. Because of the relation

$$\tilde{D}_\lambda(z) = e^{-\lambda}\frac{H(\lambda, z) - \lambda + \lambda z}{\lambda},$$

we must have that $\frac{\partial H}{\partial z}(\lambda, z)$ can be evaluated. How this can be done has been outlined in the previous chapter.

7.4 Illustration of the approximation method

In this section, the approximation method discussed in Section 7.2 is used to obtain approximations for the moments of the second component of the waiting time as well as for the remaining sojourn time. Furthermore, an alternative method to approximate the mean second component of the waiting time is discussed. For now, only the first moment will be analyzed.

7.4.1 Waiting time moments

Standard approach

To find an approximation for the first moment of the waiting time (of a super customer), the approximation method for $H_\lambda(z)$ given in the previous section can be used. When $\tilde{H}_1(z)$ is used this leads to the following approximation for $\mathbb{E}H_\lambda$:

$$\begin{aligned}
\mathbb{E}H_\lambda &\approx \tilde{H}_1'(z)\,\big|_{z=1} \\
&= \left(W_\lambda(z^{\frac{t+1}{t}})\right)'_{z=1} \\
&= \left(\frac{t+1}{t} z^{\frac{1}{t}} W_\lambda'(z^{\frac{t+1}{t}})\right)_{z=1} \\
&= \frac{t+1}{t} W_\lambda'(1).
\end{aligned}$$

When the slightly better approximation $\tilde{H}_2(z)$ will be used, this leads to the following approximation for $\mathbb{E}H_\lambda$:

$$\begin{aligned}
\mathbb{E}H_\lambda &\approx \tilde{H}_2'(z)\,\big|_{z=1} \\
&= \left(w_1 z + W_\lambda(z^{\frac{t+1}{t}}) - w_1 z^{\frac{t+1}{t}}\right)'_{z=1} \\
&= \left(w_1\left(1 - \frac{t+1}{t} z^{\frac{1}{t}}\right) + \frac{t+1}{t} z^{\frac{1}{t}} W_\lambda'(z^{\frac{t+1}{t}})\right)_{z=1} \\
&= \frac{-1}{t} w_1 + \frac{t+1}{t} W_\lambda'(1).
\end{aligned}$$

This approximation approach can also be used for approximating higher moments by taking higher derivatives of $\tilde{H}_\lambda(z)$ and evaluating these derivatives in $z = 1$.

Alternative approximation method for $\mathbb{E}H_\lambda$

Now, an alternative approximation method for $\mathbb{E}H_\lambda$ will be presented. This method to find $\mathbb{E}H_\lambda$ uses Little's law. In an arbitrary frame, a customer arrives with a service time equal to B_λ, a certain number of slots. Look at the customer as a super customer consisting of B_λ mini-customers, all having a service time equal to 1. On average, $\frac{\alpha \mathbb{E}B_\lambda}{t+1}$ of these mini-customers arrive per slot. Now Little's law comes in. When all mini-customers have to pay their total sojourn time in guilders, there are two different ways to collect the money. Each mini-customer can pay directly his total amount when he arrives at the queue or the mini-customer can pay one guilder for every slot that the mini-customer is in the system. Per slot a certain amount of money is collected. On average, the same amount of money is collected per slot.

Define $\mathbb{E}S_\lambda$ as the average sojourn time of a mini-customer and $\mathbb{E}L_\lambda$ as the average amount of work (measured in slots). This leads to the following equation, based on Little's law:

$$\frac{\alpha \mathbb{E}B_\lambda}{t+1}\mathbb{E}S_\lambda = \mathbb{E}L_\lambda.$$

To determine $\mathbb{E}L_\lambda$, $W_\lambda(z)$ will be used. $W_\lambda(z)$ is the generating function of the total amount of work just before an arbitrary arrival slot.

Define $\tilde{W}_i(z)$ as the generating function of the total amount of work just before the i-th slot of a frame, for $i = 1, \ldots, t+1$. The subscript λ is omitted for conciseness. Clearly we have that $\tilde{W}_1(z) = W_\lambda(z)$. In general $\tilde{W}_j(z)$ satisfies the following relation:

$$\begin{aligned}\tilde{W}_j(z) &= \sum_{k=j-1}^{\infty} w_k z^{k+1-j} \\ &= \frac{1}{z^{j-1}}\left(W_\lambda(z) - \sum_{k=0}^{j-2} w_k z^k\right).\end{aligned}$$

From these relations, $\mathbb{E}L_\lambda$ can be determined as follows:

$$\mathbb{E}L_\lambda = \frac{1}{t+1}\sum_{j=1}^{t+1}\tilde{W}_j'(1).$$

When $\mathbb{E}L_\lambda$ is known together with $\mathbb{E}B_\lambda$, α and t, $\mathbb{E}S_\lambda$ can be calculated from Little's Law. Remember that we were interested in an expression for $\mathbb{E}H_\lambda$, the mean waiting time of a super customer in the static arrival slot mechanism. So the only thing that remains is to obtain an expression for $\mathbb{E}S_\lambda$ from which $\mathbb{E}H_\lambda$ can be obtained.

The sojourn time of a mini-customer consists of two parts: the waiting time and the service time. The waiting time of mini-customers belonging to the same super customer will be defined as the time it takes (in slots) until the first mini-customer of the super customer will be served. The service time of a mini-customer is then the remaining part of the sojourn time. The mean service time of a mini-customer without the interrupting arrival slots, which will be denoted by $\mathbb{E}\tilde{B}_{pure}$, can be obtained. A super customer of size n consists of n mini-customers all having a mean service time (without arrival slots) of $\frac{1}{2}(n+1)$. So the mean service time equals

$$\begin{aligned}\mathbb{E}\tilde{B}_{pure} &= \frac{\sum_{k=0}^{\infty} b_k k \frac{1}{2}(k+1)}{\sum_{k=0}^{\infty} b_k k} \\ &= \frac{\mathbb{E}B_\lambda^2 + \mathbb{E}B_\lambda}{2\mathbb{E}B_\lambda}.\end{aligned}$$

From this expression the mean service time including the interrupting arrival slots of an individual mini-customer, denoted by $\mathbb{E}\tilde{B}_{total}$, can be estimated as follows:

$$\begin{aligned}\mathbb{E}\tilde{B}_{total} &\approx \frac{(t+1)}{t}\mathbb{E}\tilde{B}_{pure} \\ &= \frac{(t+1)}{t}\frac{\mathbb{E}B_\lambda^2 + \mathbb{E}B_\lambda}{2\mathbb{E}B_\lambda}.\end{aligned}$$

This eventually gives the following approximation formula for $\mathbb{E}H_\lambda$:

$$\begin{aligned}\mathbb{E}H_\lambda &\approx \frac{t+1}{\alpha\mathbb{E}B_\lambda}\frac{1}{t+1}\sum_{j=1}^{t+1}\tilde{W}_j'(1) - \frac{(t+1)}{t}\frac{\mathbb{E}B_\lambda^2+\mathbb{E}B_\lambda}{2\mathbb{E}B_\lambda} \\ &= \frac{1}{\alpha\mathbb{E}B_\lambda}\sum_{j=1}^{t+1}\tilde{W}_j'(1) - \frac{(t+1)}{t}\frac{\mathbb{E}B_\lambda^2+\mathbb{E}B_\lambda}{2\mathbb{E}B_\lambda}.\end{aligned}$$

7.4.2 The mean remaining sojourn time

In this subsection, the approximation method is applied to find an expression for the mean remaining individual sojourn time (of an unlucky customer), denoted by $\mathbb{E}\tilde{U}_\lambda$. Following the same approach as for the waiting time, we obtain two approximations for the generating function $\tilde{U}_\lambda(z)$:

$$\tilde{U}_\lambda(z) \approx \tilde{T}_\lambda(z^{\frac{t+1}{t}}),$$

and the slightly better one:

$$\tilde{U}_\lambda(z) \approx \tilde{t}_1 z + W_\lambda(z^{\frac{t+1}{t}}) - \tilde{t}_1 z^{\frac{t+1}{t}},$$

where $\tilde{t}_1$ is the probability that $\tilde{T}$ equals 1. Therefore, for $\tilde{t}_1$ the following holds:

$$\tilde{t}_1 = w_0 P(\hat{D}_\lambda = 1) = \hat{D}_\lambda'(0).$$

Using these two approximations for the generating function, this leads to the following two approximations for the mean remaining service time:

$$\begin{aligned}\mathbb{E}\tilde{U}_\lambda &\approx \left(\tilde{T}_\lambda(z^{\frac{t+1}{t}})\right)'_{z=1} \\ &= \frac{t+1}{t}\tilde{T}_\lambda'(1),\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}\tilde{U}_\lambda &\approx \left(\tilde{t}_1 z + \tilde{T}_\lambda(z^{\frac{t+1}{t}}) - \tilde{t}_1 z^{\frac{t+1}{t}}\right)'_{z=1} \\
&= \tilde{t}_1\left(1 - \frac{t+1}{t}\right) + \frac{t+1}{t}\tilde{T}'_\lambda(1) \\
&= w_0 \hat{D}'_\lambda(0)\left(1 - \frac{t+1}{t}\right) + \frac{t+1}{t}\tilde{T}'_\lambda(1).
\end{aligned}$$

7.5 First component of the individual waiting time

In the previous sections the second component of the waiting time, as well as the remaining sojourn time were discussed. The analysis of the first component of the waiting time was postponed. We will return to this now.

Assume again an arrival process that is Poisson with a constant rate μ per slot. A frame consists of $s+1$ slots, so including an arrival slot. Due to the Poisson arrival process, we have that for each individual customer his waiting time until the next arrival slot is uniformly distributed over $\{0, 1, \ldots, s\}$. Define $V_{\lambda(s)}$ as the random variable that represents the first component of the waiting time in the static arrival slot mechanism. For the generating function of the distribution of $V_{\lambda(s)}$, denoted by $V_{\lambda(s)}(z)$, the following holds:

$$\begin{aligned}
V_{\lambda(s)}(z) &= \sum_{k=0}^{s} P(V_{\lambda(s)} = k) z^k \\
&= \sum_{k=0}^{s} \frac{1}{s+1} z^k \\
&= \frac{1 - z^{s+1}}{1 - z}\frac{1}{s+1}.
\end{aligned}$$

Using this generating function, or directly, we can find expressions for the first two moments of $V_{\lambda(s)}$.

The expectation of $V_{\lambda(s)}$, denoted by $\mathbb{E}V_{\lambda(s)}$, is given by:

$$
\begin{aligned}
\mathbb{E}V_{\lambda(s)} &= \sum_{k=0}^{s} P(V_{\lambda(s)} = k)k \\
&= \frac{1}{s+1}\frac{s(s+1)}{2} \\
&= \frac{s}{2}.
\end{aligned}
$$

The second moment of $V_{\lambda(s)}$, denoted by $\mathbb{E}V_{\lambda(s)}^2$, is given by:

$$
\begin{aligned}
\mathbb{E}V_{\lambda(s)}^2 &= \sum_{k=0}^{s} P(V_{\lambda(s)} = k)k^2 \\
&= \frac{1}{s+1}\frac{s(s+1)(2s+1)}{6} \\
&= \frac{s(2s+1)}{6}.
\end{aligned}
$$

For the variance of $V_{\lambda(s)}$, denoted by $Var(V_{\lambda(s)})$, consequently the following holds:

$$
\begin{aligned}
Var(V_{\lambda(s)}) &= \mathbb{E}V_{\lambda(s)}^2 - \left(\mathbb{E}V_{\lambda(s)}\right)^2 \\
&= \frac{s(2s+1)}{6} - \frac{s^2}{4} \\
&= \frac{s(s+2)}{12}.
\end{aligned}
$$

7.6 The mean total waiting time of an individual customer

In this section, the mean total waiting time of an individual customer in the static arrival slot mechanism will be discussed. In this queueing model the (mean) waiting time of an individual customer consists of two parts. First, a customer may have to wait for an arrival slot to enter the system. After that, a customer may have to wait in the super customer (or tree) queue.

The fixed number of slots in a frame will be denoted by $s+1$. $H_{ind,st}$ denotes the total waiting time of an individual customer in the static mechanism. The two parts of the waiting time will be denoted by $H_{ind,st,1}$ and $H_{ind,st,2}$ respectively. Note that in the previous section $H_{ind,st,1}$ was denoted by $V_{\lambda(s)}$. As we have seen in the previous section:

$$\mathbb{E}H_{ind,st,1} = \frac{1}{2}s.$$

When an individual customer is part of a batch of n customers with $n \geq 1$, it has a probability $\hat{p}_n$ that its second part of the waiting time equals zero. With probability $1-\hat{p}_n$ its second part consists of the total amount of work in the system, just before the arrival of the super customer. For $\hat{p}_n$ the following holds:

$$\begin{aligned}\hat{p}_n &= P(\text{none of the } n-1 \text{ other customers arrive in same mini-slot}) \\ &= \left(\frac{2}{3}\right)^{n-1}, \qquad n \geq 1.\end{aligned}$$

So the second part of the mean waiting time of an individual customer that arrives in a batch of n customers equals $(1-\hat{p}_n)\mathbb{E}H_{\lambda(s)}$. The probability $\hat{q}_n$ that a batch of n customers is formed, equals:

$$\hat{q}_n = e^{-\lambda(s)}\frac{\lambda(s)^n}{n!}, \qquad n \geq 1.$$

Using the last two results, we can obtain an expression for the second part of the mean waiting time of an arbitrary individual customer as follows:

$$\begin{aligned}\mathbb{E}H_{ind,st,2} &= \frac{\sum_{n=1}^{\infty}\hat{q}_n n(1-\hat{p}_n)\mathbb{E}H_{\lambda(s)}}{\sum_{n=1}^{\infty}\hat{q}_n n} \\ &= \frac{e^{-\lambda(s)}\sum_{n=1}^{\infty}\frac{\lambda(s)^n}{n!}n(1-\left(\frac{2}{3}\right)^{n-1})\mathbb{E}H_{\lambda(s)}}{e^{-\lambda(s)}\sum_{n=1}^{\infty}\frac{\lambda(s)^n}{n!}n}\end{aligned}$$

$$
\begin{aligned}
&= \frac{e^{-\lambda(s)}\mathbb{E}H_{\lambda(s)}\lambda(s)\left(e^{\lambda(s)} - e^{\frac{2}{3}\lambda(s)}\right)}{e^{-\lambda(s)}\lambda(s)e^{\lambda(s)}} \\
&= \frac{\lambda(s)\mathbb{E}H_{\lambda(s)}\left(1 - e^{-\frac{1}{3}\lambda(s)}\right)}{\lambda(s)} \\
&= \mathbb{E}H_{\lambda(s)}\left(1 - e^{-\frac{1}{3}\lambda(s)}\right).
\end{aligned}
$$

The term $1 - e^{-\frac{1}{3}\lambda(s)}$ can be interpreted as the probability that an arbitrary mini-slot will be used by at least one customer. Because of the Poisson arrivals, an arbitrary customer will arrive exactly with this probability in a mini-slot with at least one other customer. The mean waiting time equals in that case $\mathbb{E}H_{\lambda(s)}$ and zero otherwise. This gives another way to derive the formula.

Finally, an expression for the mean total waiting time of an individual customer is as follows:

$$
\begin{aligned}
\mathbb{E}H_{ind,st} &= \mathbb{E}H_{ind,st,1} + \mathbb{E}H_{ind,st,2} \\
&= \frac{1}{2}s + \mathbb{E}H_{\lambda(s)}\left(1 - e^{-\frac{1}{3}\lambda(s)}\right).
\end{aligned}
$$

7.7 The mean individual sojourn time

In this section, an expression for the mean individual sojourn time will be derived. In the derivation, we will distinguish between lucky and unlucky customers, analogously as in the previous section.

Define $U_{ind,st}$ as the individual sojourn time in the static arrival slot mechanism. Remember that $\hat{U}_\lambda$ was defined as the remaining part of the individual sojourn time in the static arrival slot mechanism.

Considering the expression for $\mathbb{E}H_{ind,st}$ at the end of the previous section, we immediately obtain the following for the expectation of $U_{ind,st}$, denoted by $\mathbb{E}U_{ind,st}$:

$$
\mathbb{E}U_{ind,st} = \frac{1}{2}s + e^{-\frac{1}{3}\lambda(s)} \cdot 1 + \left(1 - e^{-\frac{1}{3}\lambda(s)}\right)\left(1 + \mathbb{E}\tilde{U}_\lambda\right).
$$

7.8 The variance of the individual waiting time

In this section, an expression for the variance of the individual waiting time $H_{ind,st}$ will be derived. The mechanism under consideration is the static arrival slot mechanism with $s+1$ slots per frame.

As indicated before, the waiting time consists of two components, denoted by $H_{ind,st,1}$ and $H_{ind,st,2}$, which are independent. Therefore the following holds:

$$Var(H_{ind,st}) = Var(H_{ind,st,1}) + Var(H_{ind,st,2}).$$

In Section 7.5 it was shown that $Var(H_{ind,st,1}) = \frac{s(s+2)}{12}$. We now focus on the variance of the second component. As we have seen already in the previous sections, with probability $1 - e^{-\frac{1}{3}\lambda(s)}$ the second component of the waiting time equals $H_{\lambda(s)}$ and with probability $e^{-\frac{1}{3}\lambda(s)}$ the second component equals zero. In subsection 7.3.2, an expression for the variance of $H_{\lambda(s)}$ has been given. So, eventually we find for the variance of the individual waiting time:

$$Var(H_{ind,st}) = \frac{s(s+2)}{12} + \left(1 - e^{-\frac{1}{3}\lambda(s)}\right)^2 Var(H_{\lambda(s)}).$$

7.9 The variance of the individual sojourn time

In principle, an expression for the variance of the individual sojourn time in the static arrival slot mechanism with $s+1$ slots per frame can be derived. We will restrict ourselves to a description of how this expression can be obtained.

Again, we distinguish between lucky and unlucky customers. Lucky customers experience a first component of the waiting time plus an additional service time of exactly one slot (which does not contribute to the variance). Unlucky customers experience a first component of the waiting time as well and furthermore, an unlucky customer experiences the so-called "remaining individual sojourn" time, denoted by $\tilde{U}$.

For both lucky and unlucky customers, the two components of the sojourn time are independent. Therefore the variance of the sojourn time is the sum of the two variances of the two corresponding components. For the variance of $\tilde{U}$, no expression has been derived yet. This can be done straightforwardly, analagously to the derivation of the expression for $\mathbb{E}\tilde{U}$ in subsection 7.3.3. For the other components, the variance is explicitly known.

With probability $1 - e^{-\frac{1}{3}\lambda(s)}$ a customer is unlucky and with the complementary probability it is lucky. This leads to the final expression for the variance of the individual sojourn time, denoted by $Var(S_{ind,st})$:

$$Var(S_{ind,st}) = \frac{s(s+2)}{12} + \left(1 - e^{-\frac{1}{3}\lambda(s)}\right)^2 Var(\tilde{U}_{\lambda(s)}).$$

Chapter 8

Capacity analysis of the arrival slot mechanism

8.1 Stability condition of the arrival slot mechanism

In this section the capacity of the queueing system with $s+1$ slots per frame with the first being an arrival slot is determined numerically. This means that the minimal rate μ_{max} per slot of the individual arrival process is determined for which the system is not stable. This is of course equivalent with determining the rate $\lambda(s)_{max}$ because $\lambda(s)_{max} = (s+1)\mu_{max}$.

To determine the capacity of the system, it is relevant to see how the rate $\lambda(s)$ and the arrival probability α are related. So far, we have only looked at ternary contention trees. The relation between $\lambda(s)$ and α can be easily obtained for q-ary contention trees.

$$
\begin{aligned}
1-\alpha &= P(\text{no super customer arrives}) \\
&= P(\text{no collisions in all of the } q \text{ mini slots }) \\
&= \sum_{k=0}^{q} P(\text{no collisions}|\#\text{ individual arrivals} = k)P(\#\text{ individual arrivals} = k) \\
&= \sum_{k=0}^{q} \left(\frac{\binom{q}{k}k!}{q^k}\right) e^{-\lambda(s)}\frac{(\lambda(s))^k}{k!} \\
&= \sum_{k=0}^{q} e^{-\lambda(s)}\binom{q}{k}(\frac{\lambda(s)}{q})^k
\end{aligned}
$$

$$= e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q.$$

This expression can also be derived in another way. No super customer is formed if in every mini-slot either zero or one customer arrives. The probability that in a certain mini-slot either zero or one customer arrives equals $e^{\frac{\lambda(s)}{q}} + e^{\frac{\lambda(s)}{q}}\frac{\lambda(s)}{q}$. Taking this expression to the power q yields eventually the same expression as above.

So for $\alpha_{\lambda(s)}$ we eventually get the following:

$$\alpha_{\lambda(s)} = 1 - e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q.$$

Now that this relation is obtained, we return to the stability condition:

$$\alpha\mathbb{E}B = \alpha B'(1) < s, \qquad ,$$

or in this case

$$\begin{aligned} \alpha_{\lambda(s)}\mathbb{E}B_{\lambda(s)} &= \alpha_{\lambda(s)}B'_{\lambda(s)}(1) \\ &= \left(1 - e^{-\lambda(s)}\left(1+\frac{\lambda(s)}{q}\right)^q\right)B'_{\lambda(s)}(1) < s. \end{aligned}$$

So now we can make a plot of $\alpha_{\lambda(s)}\mathbb{E}B_{\lambda(s)}$ against the arrival rate. That is, if we can numerically get a good approximation of $\mathbb{E}B_{\lambda(s)}$. In Section 5.4, a method for determining $\mathbb{E}B_{\lambda(s)}$ was given. In Section 8.2, some graphs will be presented.

8.2 Illustrative graphs concerning capacity

The numerical scheme to compute $\mathbb{E}B_{\lambda(s)}$ is implemented in Mathematica 4.0. In fact, $\alpha_{\lambda(s)}\mathbb{E}B_{\lambda(s)}$ is plotted as a function of $\frac{\mu}{3}$. The value of $\frac{\mu}{3}$ at which this graph intersects with the horizontal line with constant value s determines μ_{max}. To illustrate this approach, the cases in which $s = 1, \ldots, 4$

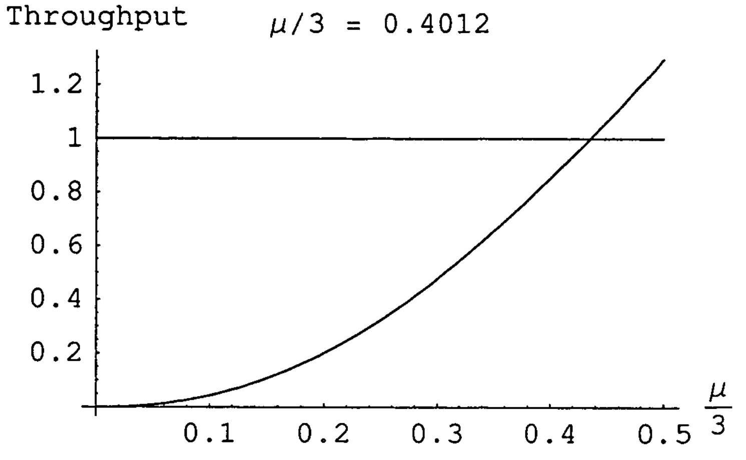

Figure 8.1: Capacity determination in case $s = 1$.

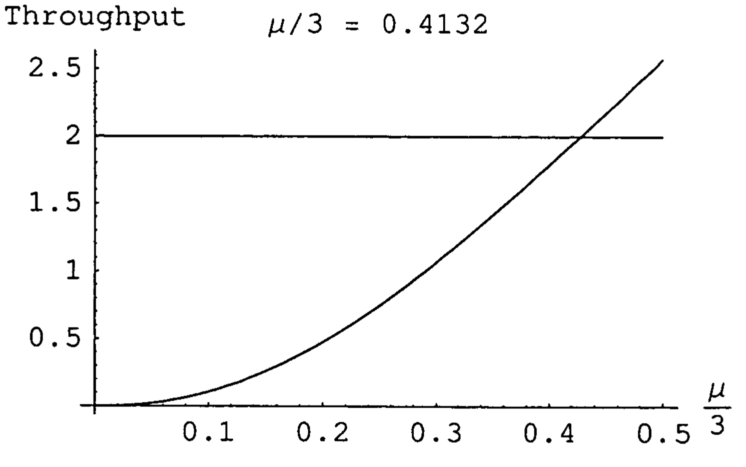

Figure 8.2: Capacity determination in case $s = 2$.

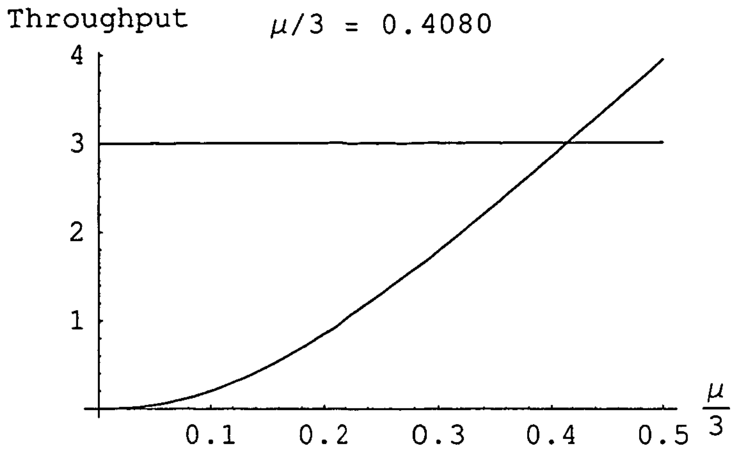

Figure 8.3: Capacity determination in case $s = 3$.

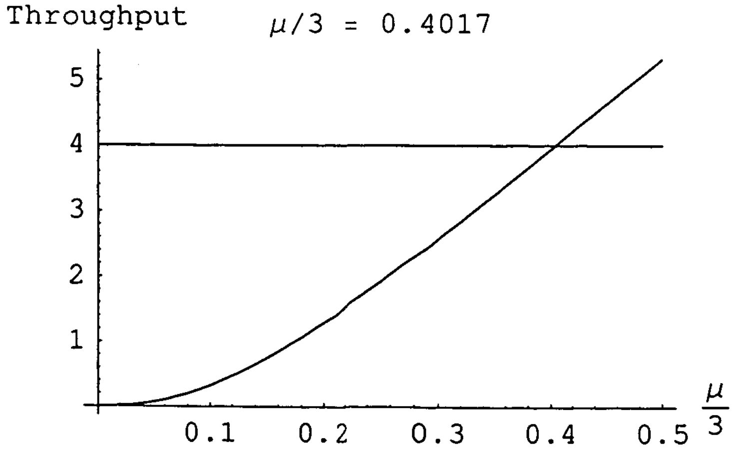

Figure 8.4: Capacity determination in case $s = 4$.

will be shown.

As one can see from the figures, the system with $s = 2$ outperforms the other systems. So $s = 2$ seems to be an optimal choice as far as capacity is considered. It is better than both $s = 1$ and $s = 3$.

An interesting question is whether there is a non-integer s for which the capacity is slightly higher than for $s = 2$. This will be the subject of the next section.

8.3 Capacity in case of a non-integer s

This section discusses capacity results in case s is not restricted to a positive integer value. A mechanism with a non-integer s seems at first sight perhaps practically irrelevant, due to the fact that it does not represent a "real" mechanism. But it gives at least an indication of the optimality of the system with $s = 2$. Furthermore, as will turn out later in Chapter 15, there is practical relevance for mechanisms with non-integer values of s in the sense that they can indeed be interpreted as real mechanisms.

As far as the formulas are concerned, there is no reason to restrict ourselves to integer values of s. Therefore, we will present a plot, given in figure 8.5, with the capacity given as a function of s. This plot is obtained as follows.

In Section 8.1, the stability condition for an arrival slot mechanism was presented. This condition was the following:

$$\left(1 - e^{-\lambda(s)}\left(1 + \frac{\lambda(s)}{q}\right)^q\right) B'_{\lambda(s)}(1) \quad < \quad s.$$

If we replace the inequality sign by an equality sign, we obtain an equation. If we fix s to some real value and subsequently solve this equation numerically for $\lambda(s)$, or equivalently because s is fixed, for μ, we obtain μ_{max} for that certain value of s. If we do this for a range of real values for s, we can present the results in a plot.

As one can see, the function reaches an optimum for $s \approx 1.8$. This is a rather positive result for the moment, because 1.8 is close to 2 and the optimal capacity is close to the capacity of the system with $s = 2$. Furthermore, one

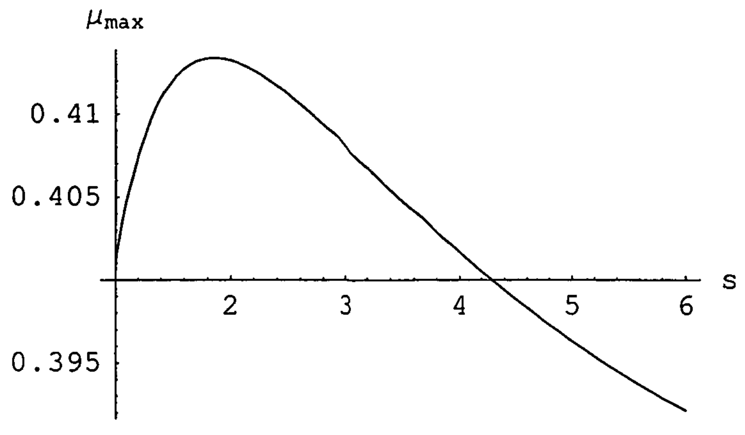

Figure 8.5: Capacity as a function of s

can see that the function is decreasing after the optimum is reached.

It is the case that the capacity of the gated mechanism is a horizontal asymptote of this graph. We will not prove this formally, but restrict ourselves to give some intuition to support this observation. Recall that the capacity of the static arrival slot mechanism is exactly the same as the capacity of the dynamic arrival slot mechanism, of course for the same value of s. This is true, due to the fact that both mechanisms behave identically when the system is busy. So, if both systems tend to "explode", they behave exactly the same. This implies that both capacities must be equal.

Assume that s is very large. When we compare the gated mechanism and the dynamic arrival slot mechanism (with large s), it can be observed that both mechanisms act identically as long as a tree finishes within s slots. It is even the case that for every finite tree - and in a stable system, every tree is finite - we can find an s such that the tree length is smaller than s. This argument shows informally that the capacity of the gated mechanism must equal the capacity of the dynamic arrival slot mechanism as s tends to infinity. Combining the arguments implies that the capacity of the gated mechanism is indeed a horizontal asymptote of the graph.

Chapter 9

Numerical results for the static arrival slot mechanism

In this chapter, numerical results are obtained, using the analytical procedures of the previous chapters. It will be illustrated how all the calculations can be done, to find eventually useful results (from a practical point of view).

The main part of the numerical results concerns the static arrival slot mechanism with $s = 2$. Besides these results, also numerical results are presented for the first two moments of the service time of a super customer, denoted by $\mathbb{E}B_\lambda$ and $\mathbb{E}B_\lambda^2$, as a function of λ.

First the results about service time of a super customer are presented and compared with simulation results. After that, waiting time results are presented, both for individual customers and for super customers. These results are again compared with simulation results. Subsequently, sojourn time results of individual customers are given. These results are eventually compared with simulation results as well.

Chapter 14 is dedicated to a simulation study of several contention resolution mechanism. The static arrival slot mechanism is one of these mechanism. So, for more details on the simulation approach that has been followed, see Chapter 14.

9.1 Numerical results on $\mathbb{E}B_\lambda$ and $\mathbb{E}B_\lambda^2$

In this section two graphs are presented. The first graph shows $\mathbb{E}B_\lambda$ as a function of μ. The second graph shows $\mathbb{E}B_\lambda^2$ as a function of μ. In both graphs, the simulation results are plotted as stars. As one can see, the stars are almost on the curve, which demonstrates the agreement between analysis and simulation.

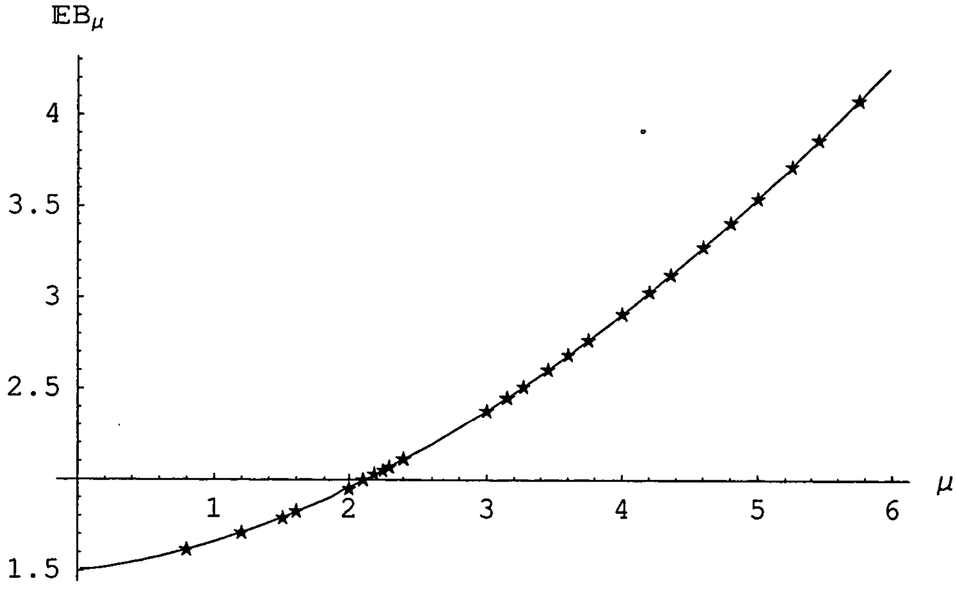

Figure 9.1: Plot of $\mathbb{E}B_\lambda$ against μ

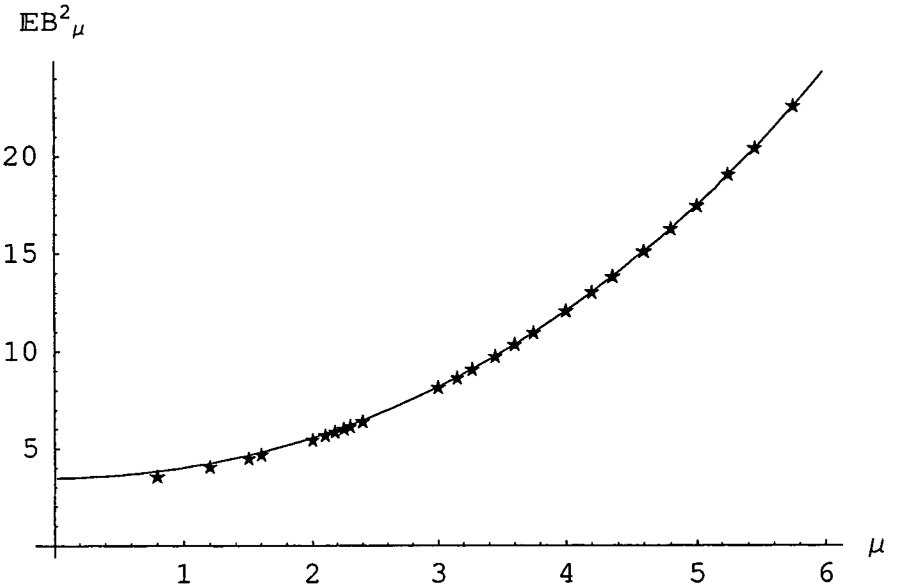

Figure 9.2: Plot of $\mathbb{E}B_\lambda^2$ against μ

9.2 Numerical results on the waiting time

9.2.1 Determination of $b_0, \ldots, b_{s-1}$

In Chapter 5 we found a method to approximate $B_{\lambda(s)}(z)$ for z on or within the complex unit circle. This method can be used to determine numerically the zeros of the denominator of the given expression for $W(z)$ with $B(z) = B_{\lambda(s)}(z)$. This can be done with Mathematica for instance, using the successive substitution approach that has been outlined in Section 4.2.2. Now these s zeros are found, we have to solve a set of s equations to get the unknown probabilities in the numerator of the expression for $W(z)$. In this numerator appear $b_0, \ldots, b_{s-1}$. This are model parameters which are directly related to the service time distribution. In order to solve the set of s equations, the values of these probabilities must be known.

In this special case, the service time distribution is chosen to be the distribution of $B_{\lambda(s)}$. For this choice for $B(z)$, or equivalently the service time distribution, the probabilities $b_0, \ldots, b_{s-1}$ will be determined:

$$b_{\lambda(s),k} \quad = \quad P(B_{\lambda(s)} = k), \qquad k = 1, 1, \ldots$$

We are interested in $b_{\lambda(s),k}, \quad k = 1, \ldots, s-1$.

In Section 5.3, an expression for these probabilities is presented. We can slightly simplify this formula to obtain the following:

$$\begin{aligned} b_{\lambda(s),k} &= (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3))P(\tilde{B}(2) = k) \\ &\quad + \frac{1}{7}\pi_{\lambda(s)}(3)P(\tilde{B}(3) = k) + \sum_{n=4}^{2(k+1)+1} \pi_{\lambda(s)}(n)P(\tilde{B}(n) = k+1), \\ &\quad k = 1, 2, \ldots \end{aligned}$$

The infinite sum is replaced by a finite sum. To see why this simplification is allowed, consider the following.

A ternary tree with $2n+1$ leaves has exactly n nodes. This can be shown by an induction argument. So it follows that when the contention resolution procedure starts with $2n+1$ customers, there will be at least n contention slots required. Or considering it the other way around: if a tree has initially more than $2n+1$ customers, there will be more than n slots needed to finish the tree. This justifies the truncation of the infinite sum.

So when determining $b_{\lambda(s),k}$, we only have to find a limited number of probabilities of the form: $P(\tilde{B}(n) = k)$. These probabilities for a certain k can for instance be found by inverting the obtained expressions (see Appendix B) for the $f_n(z)$'s. Proceeding in this way, we find for example:

$$\begin{aligned} b_{\lambda(s),0} &= 0, \\ b_{\lambda(s),1} &= (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3))P(\tilde{B}(2) = 1) + \frac{1}{7}\pi_{\lambda(s)}(3)P(\tilde{B}(3) = 1) + \\ &\quad \pi_{\lambda(s)}(4)P(\tilde{B}(4) = 2) + \pi_{\lambda(s)}(5)P(\tilde{B}(5) = 2) \\ &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{1}{6}\frac{2}{3}(\lambda(s))^2 + \frac{1}{9}\frac{2}{3}(\lambda(s))^3 + \frac{1}{54}\frac{2}{9}(\lambda(s))^3 + \frac{1}{4!}\frac{88}{243}(\lambda(s))^4 + \frac{1}{5!}\frac{40}{729}(\lambda(s))^5 \right) \\ &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{1}{9}(\lambda(s))^2 + \frac{19}{243}(\lambda(s))^3 + \frac{11}{729}(\lambda(s))^4 + \frac{1}{2187}(\lambda(s))^5 \right). \end{aligned}$$

9.2.2 Calculation of $\mathbb{E}H_{\lambda(2)}$

In this subsection, the focus will be on the calculation of the mean waiting time of a super customer (tree queue waiting time) in the arrival slot mechanism with $s = 2$. In the next subsection, numerical results for this mean waiting time for a range of values for μ will be presented and compared with simulation results.

So consider the mechanism with $s = 2$. In Section 4.2 an analytical expression for $\mathbb{E}W_{\lambda(s)}$ has been derived. When we plug in $s = 2$ and set $\alpha := \alpha_{\lambda(2)}$ this expression becomes:

$$\begin{aligned}\mathbb{E}W_{\lambda(2)} &= W'_{\lambda(2)}(1)\\ &= \frac{\alpha b_1 w_0 + (1-\alpha)(w_0 + w_1) - 1 - \alpha\frac{B''_{\lambda(2)}(1)}{2})}{2 - \alpha B'_{\lambda(2)}(1)}.\end{aligned}$$

Assume that the individual arrival rate μ is known.

- $\lambda(2)$ can be computed as $(2+1)\mu = 3\mu$.

- α, the arrival probability, can be computed as $1 - e^{-\lambda(2)}(1 + \frac{\lambda(2)}{3})^3$.

- Using the iterative method to evaluate $B_{\lambda(2)}(z)$, the zeros of the equation $z^2 - \alpha B_{\lambda(2)}(z) - (1-\alpha)$ can be found. These zeros can be used to obtain w_0 and w_1.

- $B'_{\lambda(2)}(1)$ and $B''_{\lambda(2)}(1)$ can be determined using the expressions that are derived in sections 5.4 and 5.5.

- b_1 can be computed using the expression in the previous subsection.

This illustrates that $\mathbb{E}W_{\lambda(2)}$ can be obtained in a straightforward manner. This result can be used directly to obtain two approximations for $\mathbb{E}H_{\lambda(2)}$.

The approximations are based on results from Section 7.2. The first approximation is the following:

$$\mathbb{E}H_{\lambda(2)} \approx \frac{3}{2}W'_{\lambda(2)}(1),$$

and a sligtly better one:

$$\begin{aligned}\mathbb{E}H_{\lambda(2)} &\approx w_1 + \frac{3}{2}(W'_{\lambda(2)}(1) - w_1) \\ &= -\frac{1}{2}w_1 + \frac{3}{2}W'_{\lambda(2)}(1).\end{aligned}$$

An exact method to obtain $\mathbb{E}H_{\lambda(2)}$ has also been described. For this method, the generating function $W_{\lambda(2)}(z)$ as well as its derivative (with respect to z) must be evaluated in $z = 1$ and $z = -1$. For $z = 1$ this results in 1 and $\mathbb{E}W_{\lambda(2)}$, respectively. For $z = -1$ it is slightly more difficult. How this can be done has already been outlined in section 7.3. The expression for $\mathbb{E}H_{\lambda(2)}$ that has already been given is the following:

$$\mathbb{E}H_{\lambda(2)} = -\frac{1}{2}W_1(1) + \frac{3}{2}W'_0(1) + \frac{3}{2}W'_1(1).$$

9.2.3 Numerical results for $\mathbb{E}H_{\lambda(s)}$

In this subsection some graphs will be presented. First, $\mathbb{E}W_{\lambda(2)}$ is calculated for a range of values for μ. The results are presented in 9.3. After that, the (best) approximation for $\mathbb{E}H_{\lambda(2)}$ is presented, together with simulation results. This makes a comparison possible. Finally, the results of the exact method are presented, again together with the simulation results to make a comparison.

In figure 9.3, $\mathbb{E}W_{\lambda(2)}$ is plotted as a function of the arrival rate μ per slot. The values of μ that will be used later in this subsection are indicated with a star. As a very rough approximation, the mean waiting time in the arrival slot mechanism with $s = 2$ is $\frac{3}{2}$ times $\mathbb{E}W_{\lambda(2)}$.

In figure 9.4, the obtained simulation results for $\mathbb{E}H_{\lambda(2)}$ as a function of μ

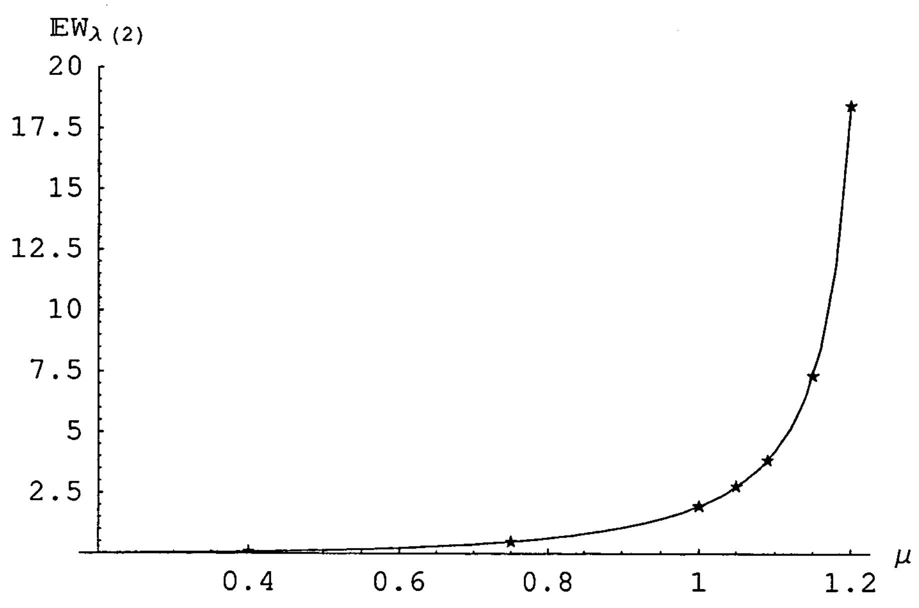

Figure 9.3: Plot of $\mathbb{E}W_{\lambda(2)}$ as a function of μ

are plotted as stars. These values can almost be considered as the true values, due to extensive simulation. These results can be compared with the approximation results found when using the approximation based on w_0 and w_1. As one can see, the relative difference decreases very fast as μ increases, leading to good approximations.

In figure 9.5, the obtained simulation results for $\mathbb{E}H_{\lambda(2)}$ as well as the results found when using the "exact" method to find $\mathbb{E}H_{\lambda(2)}$ from $W_{\lambda(2)}(z)$ are shown. As one can see, the stars are almost on the curve.

9.2.4 Numerical results on mean total waiting time of an individual customer

In this subsection, the focus will be on the mean waiting time of an individual customer. A graph will be presented in which this mean waiting time of an individual customer is plotted as a function of the arrival rate μ. The calculations are based on the expression at the end of Section 7.6:

$$\mathbb{E}H_{ind,st} = \mathbb{E}H_{ind,st,1} + \mathbb{E}H_{ind,st,2}$$

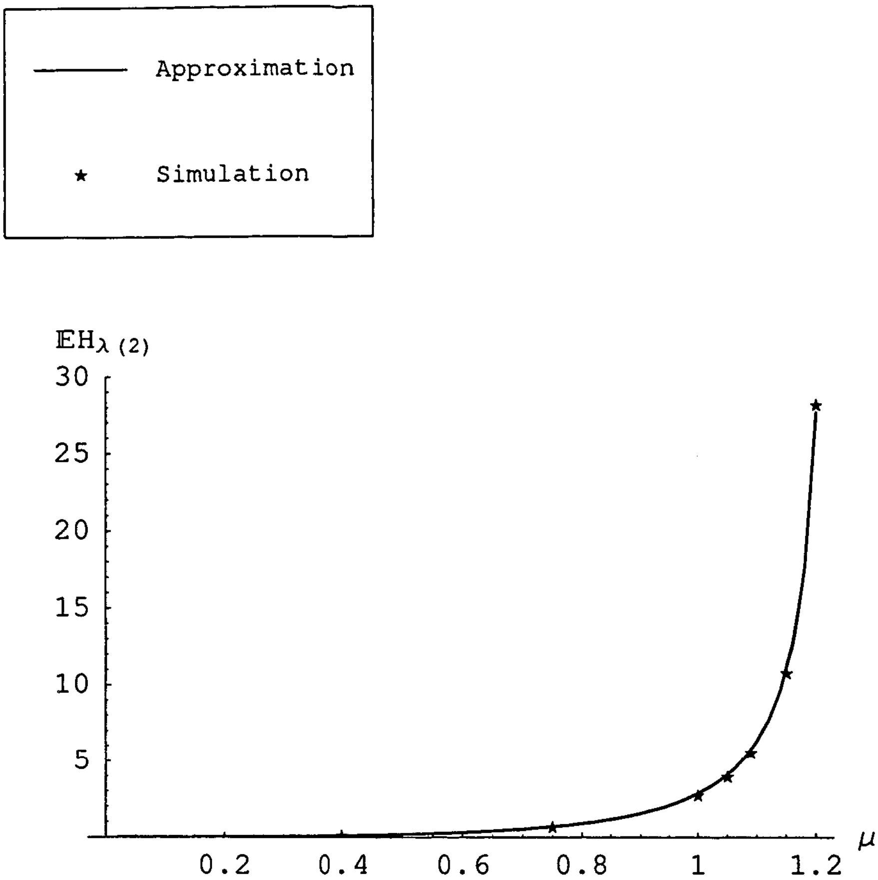

Figure 9.4: Plot of approximation (based on w_0 and w_1) for $\mathbb{E}H_{\lambda(2)}$ as a function of μ

$$= \frac{1}{2}s + \mathbb{E}H_{\lambda(s)}\left(1 - e^{-\frac{1}{3}\lambda(s)}\right).$$

Therefore, the numerical results of the previous subsection concerning $\mathbb{E}H_{\lambda(2)}$ (based on the exact translation approach) will be used as well.

The graph is presented in figure 9.6. The simulation results are indicated with stars. There is almost no difference between analysis and simulation.

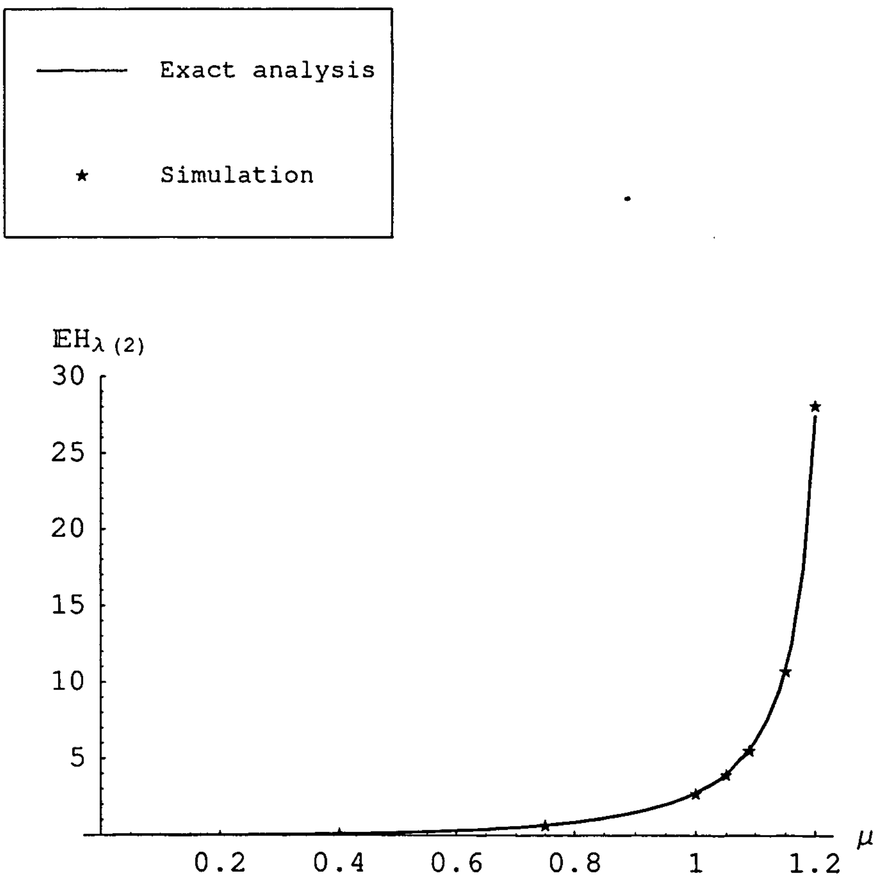

Figure 9.5: Plot of simulation results and exact calculations for $\mathbb{E}H_{\lambda(2)}$ as a function of μ

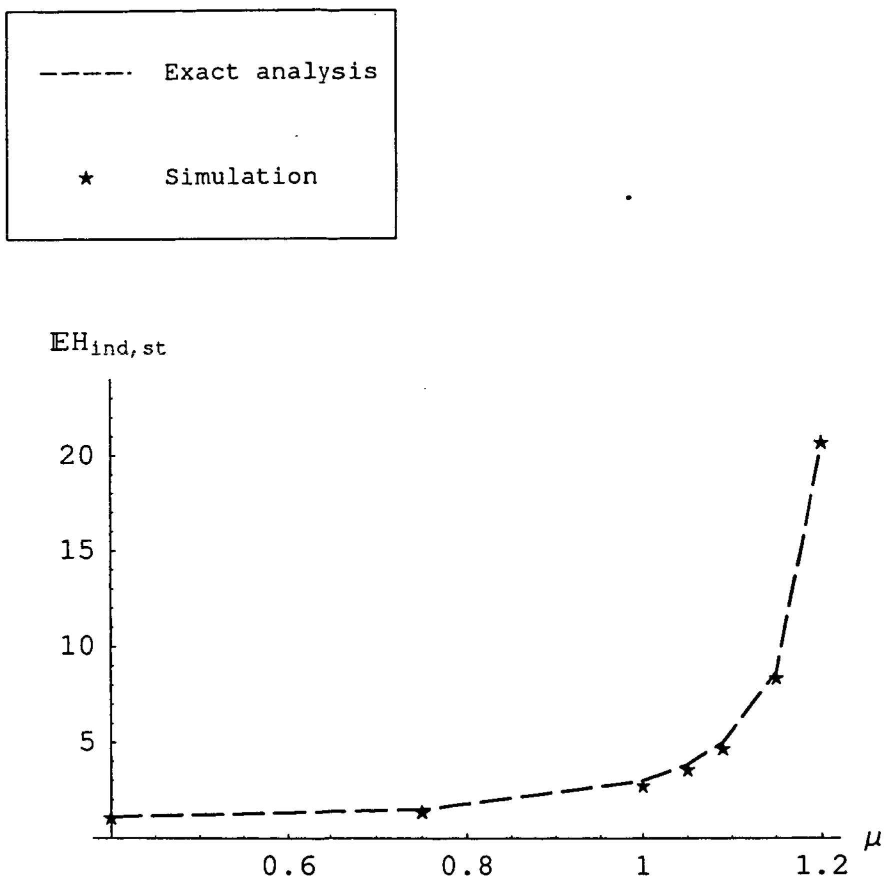

Figure 9.6: Plot of analytical and simulation results for $\mathbb{E}H_{ind,st}$ as a function of μ

9.3 Numerical results on the service and sojourn time

This section contains results for the sojourn time of an individual customer in the mechanism with $s = 2$. The results are obtained using the translation methods presented in Chapter 7.

Two graphs will be presented. The first graph, figure 9.7, contains the simulation results (indicated with stars) together with the results found when using the exact translation approach. The second graph, given in figure 9.8 contains the simulation results compared with the approximation method results. Both graphs show a close agreement between analysis and simulation.

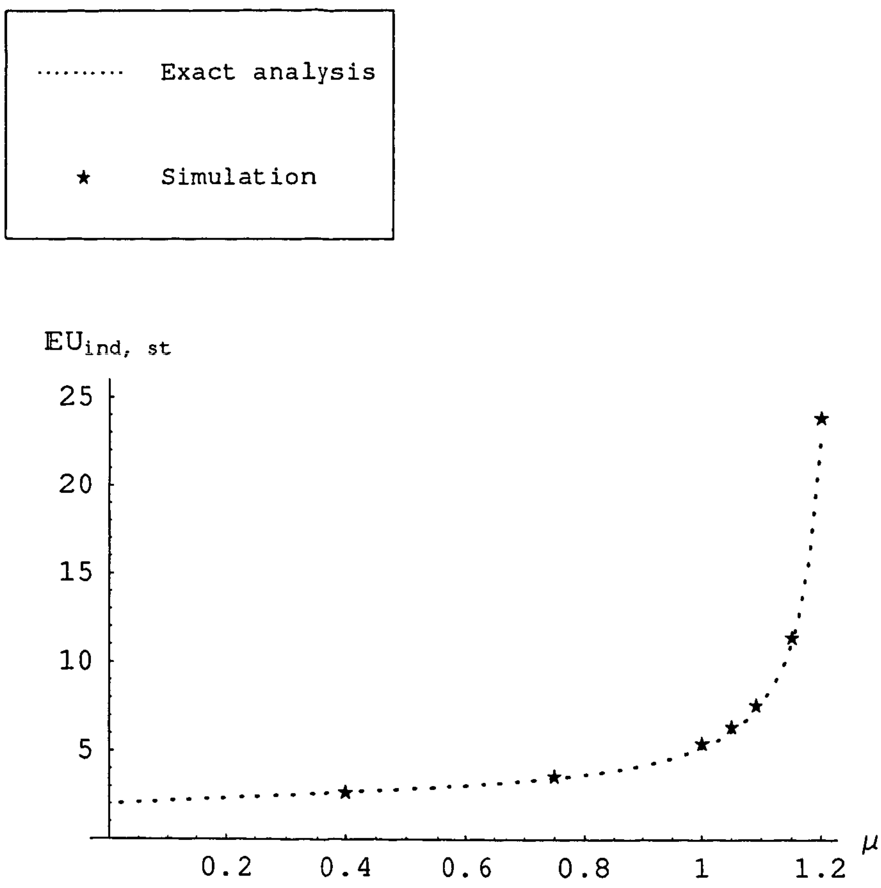

Figure 9.7: Plot of both exact analytical and simulation results for $\mathbb{E}U_{ind,st}$ as a function of μ

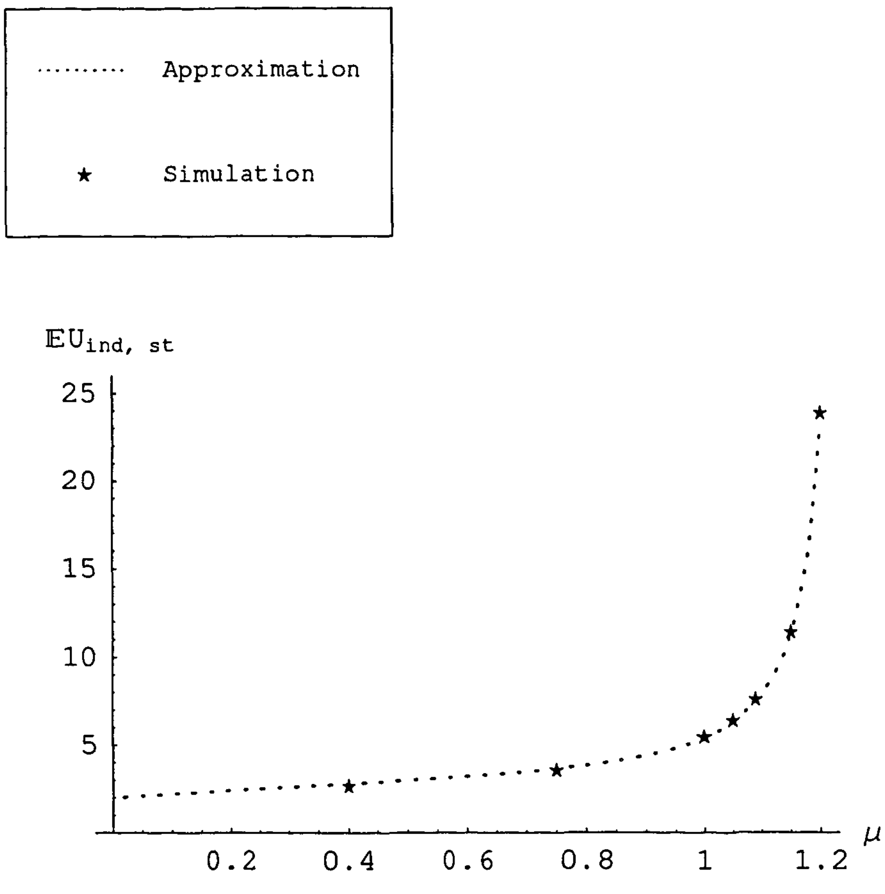

Figure 9.8: Plot of both approximation and simulation results for $\mathbb{E}U_{ind,st}$ as a function of μ

Chapter 10

The dynamic arrival slot mechanism

10.1 Introduction

So far, the focus has been on the static arrival slot mechanism. In this chapter, we make a beginning with the analysis of the dynamic arrival slot mechanism. This analysis continues in the next chapters. For a description of the dynamic arrival slot mechanism, we refer to section 2.2. The analysis approach, which will be outlined shortly, will be roughly the same as the approach for the static mechanism.

The waiting time, service time and sojourn time of individual customers in the dynamic mechanism will also be discussed. The capacity of the system does not have to be determined anymore, because the capacity of the dynamic mechanism turns out to be the same as the capacity of the static mechanism. This is true because both mechanisms only differ when the system is empty. As the arrival rate approximates the capacity of the system, the probability that the system is empty goes to zero.

In the remaining part of this chapter, we will outline in more detail what will be analyzed and in which order the analysis will proceed in the next chapters.

Just as for the analysis of the static mechanism, we will, again for ease of presentation, not directly analyze the dynamic arrival slot mechanism itself. First a slightly different model will be studied. This more general model, which will be called the *dynamic periodic* $Geo|G|1$*-queue*, will first be formulated (and explained) in the next section. Subsequently, we will indicate how this queueing model is related to the dynamic arrival slot model. Furthermore, we indicate how it will be used to analyze the dynamic arrival slot mechanism.

10.2 Dynamic periodic $Geo|G|1$-queue

Consider a discrete-time (slotted) queue with discrete service times and 1 server which serves the queue according to a first-come first-served discipline. An arrival of a (super)customer can, analogously as in the (static) periodic $Geo|G|1$-queue, only occur just before the first slot of every frame: with a probability α there is an arrival just before the first slot of a frame and with probability $1-\alpha$ there is not. In case of an arrival, the (super)customer just enters the queue and service continues normally during this first slot of every frame. The service time distribution of a super customer is allowed to be general.

So far, there is no difference in comparison with the static periodic $Geo|G|1$-queue. The differences will be outlined now.

Recall that a *frame* was defined as exactly s consecutive slots, $s \in \mathbb{N}$. In this model we redefine a frame as at most s consecutive slots, $s \in \mathbb{N}$. When s slots are used in a certain frame, a new frame is started. Another trigger point for starting a new frame is when the system empties in the sense that no customers are present anymore. So, the length of a frame varies between 0 and s. A frame of length zero may look strange at first sight, but it corresponds to the situation of an empty system and no arrival.

As indicated already, an arrival can only occur just before the first slot of every frame. In the dynamic periodic $Geo|G|1$-queue, the arrival probability of a customer may depend on the length of the previous frame. In case of an arrival, the customer just enters the queue and service continues normally during this first slot of every frame. The service time distribution of a customer may also depend on the length of the previous frame, i.e. the frame that ended just before the arrival took place.

The frame length that varies over time is one of the main differences of the dynamic model when compared with the static model. The dependence between frame length and both arrival probability and service time distribution is another main difference between these two models.

10.3 A special case of the dynamic periodic $Geo|G|1$-queue

In this section, a special case of the dynamic periodic $Geo|G|1$-queue will be formulated. Special in the sense that a particular choice is made for the service time distribution, which is allowed to be general. This special case is closely related to the dynamic arrival slot mechanism. This is the reason that this special case is formulated and subsequently analyzed.

Consider the dynamic arrival slot mechanism. Analogously as in the static case, think of this mechanism when omitting the arrival slots. Arrival slots are left out and so in fact the time axis is compressed in a certain way. Individual customers that leave the system directly after their first arrival slot, the so-called *lucky* customers, are not seen anymore in the model. It may happen in the dynamic arrival slot mechanism, that there are several consecutive arrival slots. All these slots are not seen in the $Geo|G|1$-model. So only super customers that are placed in the tree queue remain. The model that is formed this way can be regarded as a special case of the dynamic periodic $Geo|G|1$-queue. This will be explained now.

Consider a frame of c slots that has just ended:

- At the beginning of the next frame, with a certain probability (that is constant but dependent on c), a super customer is placed in the tree-queue and with one minus that probability, no super customer arrives. So the super customers act as the customers in the dynamic periodic $Geo|G|1$-queue. As indicated in Section 2.2, the tree queue was also a first-come first-served queue with 1 server.

- The service time distribution of the super customer that may arrive will not be described in detail anymore, because this distribution is just the same as the distribution of a super customer in the static model with (a constant) c slots per frame. Therefore, we refer to Section 3.4 for a detailed description of the service time distribution, where s can simply be replaced by c. Finally, observe that the distribution indeed depends on c, the length of the previous frame.

10.4 Overview of the analysis approach

Similarly as in the analysis of the static arrival slot mechanism, the approach for the mathematical analysis of the dynamic arrival slot mechanism is to study at first a sligthly more tractable queueing system than the arrival slot mechanism. It is more tractable, due to the fact that the arrival slots in between are left out, what means that service is not interrupted anymore and this makes waiting and service time analysis more straightforward.

So, the main difference between this special case of the dynamic periodic $Geo|G|1$-queue and the dynamic arrival slot mechanism, is that arrivals do not occur in an arrival slot, but occur *instantaneously* and therefore do not occupy a time slot.

In the end, when the analysis is done for the (special case of the) dynamic periodic $Geo|G|1$-queue, the results of the analysis of this queueing system will be translated to results for the original dynamic arrival slot mechanism. The translation approach that was outlined in Chapter 7 is used. Recall that this exact translation takes into account that service is interrupted at the end of every frame, resulting in larger waiting and service times than found in the dynamic periodic $Geo|G|1$-queue. Furthermore, it takes into account the presence of lucky customers, which are invisible in the dynamic $Geo|G|1$-queue, since they are only visible in the arrival slots, which are left out in this queueing model.

The translation from dynamic periodic $Geo|G|1$-queue to dynamic arrival slot mechanism will be illustrated, leading to some numerical results.

We finish this section with a description of the contents of the second part of this report, which contains the analysis of the dynamic arrival slot mechanism. The contents of each chapter will be roughly indicated.

Waiting time analysis
In the next chapter, Chapter 11, the waiting time is completely analyzed. First, the waiting time in the dynamic periodic $Geo|G|1$-queue will be analyzed using again a generating function approach. Subsequently, the results of this model are applied to the special case of the dynamic $Geo|G|1$-queue. This gives eventually results for the second component of the waiting time in this model. Finally, the results will be translated following the approach that has been outlined in Chapter 7.

Because of the fact that the first component of the waiting time in the dynamic arrival slot mechanism is not so trivial as in the static arrival slot mechanism, a section is devoted to the analysis of this component. Both the mean and the variance of this waiting time component will be analyzed.

Service time analysis

The service time analysis is nearly the same as for the static mechanism and will therefore not be done again. It is true that in the dynamic model, the service time depends on the frame length, but given the length of the frame, the analysis is identical to that of the static mechanism.

Simulation study

As indicated already, after the analytic part of this report, a simulation study follows. This simulation is discussed in Chapter 14. In this chapter the four contention resolution procedures are compared on a couple of performance measures such as capacity, waiting time, service time and sojourn time.

Chapter 11

Waiting time analysis

11.1 Waiting time analysis for the dynamic periodic $Geo|G|1$-queue

In this section, the waiting time in the dynamic periodic $Geo|G|1$-queue is analyzed. For a detailed description of this queueing model, we refer to Section 2.2. The analysis uses a generating function approach.

11.1.1 Notation

The following (additional) notation will be used throughout this section.

Let α_c be the arrival probability of a customer when the length of the previous frame is equal to c, $c = 0, \ldots, s$. Let the random variable $\tilde{B}_c$ denote the service time of an arriving customer after a frame of length c for $c = 0, \ldots, s$ and:

$$\tilde{b}_{k,c} = Pr[\tilde{B}_c = k], \qquad c = 0, \ldots, s, \quad k = 0, 1, 2, \ldots$$

Define $\tilde{B}_c(z)$ as the generating function corresponding to the random variable $\tilde{B}_c$:

$$\tilde{B}_c(z) = \sum_{k=1}^{\infty} \tilde{b}_{k,c} z^k, \qquad |z| \leq 1, \quad c = 0, \ldots, s.$$

Let $\tilde{P}$ be a random variable with distribution the steady state distribution of the total amount of work, just before the first slot of an arbitrary frame in the dynamic periodic $Geo|G|1$ queueing model. Furthermore, define:

$$p_{0,c} = P[\tilde{P} = 0 \text{ and previous frame was of length } c\ ], \qquad c = 0, ..., s,$$

$$p_k = P[\tilde{P} = k], \qquad k = 0, 1, \ldots$$

Finally, define the generating function $\tilde{P}(z)$ corresponding to the distribution of $\tilde{P}$:

$$\tilde{P}(z) = \sum_{k=0}^{\infty} p_k z^k, \qquad |z| \leq 1.$$

11.1.2 Analysis of the model

In this subsection, an expression for the generating function of the random variable $\tilde{P}$ is derived. Analogously as in the static case, this $\tilde{P}$ can be treated as a waiting time variable, due to the "memoryless property" of the arrival process of (super)customers.

As mentioned before, both the arrival probability and the service time distribution depend on the length of the previous frame. When we consider the total amount of work just before the first slot of a new frame (*not* including the amount of work from a possible new arrival), it holds that if this total amount of work is unequal to zero, then the length of the previous frame is equal to s. In case the total amount of work equals zero, the length of the previous frame can vary between 0 and s slots. So, consider the Markov chain defined by the total amount of work at the queue and the length of the previous frame (in case the total amount equals zero). For completeness, the total amount of work is defined as the total time (i.e., the number of contention slots) it takes to complete service of all the customers that are currently in the system. A state will be defined by (P_n, k_n), with P_n the total amount of work after the n^{th} frame and k_n the length of the n^{th} frame. Because of the fact that k_n always equals s if $P_n > 0$, states with $P_n > 0$ will be denoted by P_n only.

The stationary distribution satisfies the balance equations, which will be given below:

$$
\begin{aligned}
p_{0,0} &= \sum_{c=0}^{s}(1-\alpha_c)p_{0,c}, \\
p_{0,1} &= \sum_{c=0}^{s}\alpha_c p_{0,c}\tilde{b}_{1,c} + (1-\alpha_s)p_1, \\
\ldots \\
p_{0,s} &= \sum_{c=0}^{s}\alpha_c p_{0,c}\tilde{b}_{s,c} + \alpha_s\sum_{n=1}^{s-1}\tilde{b}_{s-n,s}p_n + (1-\alpha_s)p_s, \\
p_k &= \sum_{c=0}^{s}\alpha_c p_{0,c}\tilde{b}_{k+s,c} + \alpha_s\sum_{n=0}^{k-1}p_{k-n}\tilde{b}_{s+n,s} + \alpha_s\sum_{n=1}^{s-1}p_{k+n}\tilde{b}_{s-n,s} \\
&\quad +(1-\alpha_s)p_{k+s}, \qquad k=1,2,\ldots
\end{aligned}
$$

First, define p_0 to be the following:

$$
p_0 := \sum_{c=0}^{s}p_{0,c}.
$$

When we add up the first $s+1$ equations and plug in the definition of p_0, we obtain for p_0:

$$
p_0 = \sum_{c=0}^{s}\alpha_c p_{0,c}\tilde{b}_{0+s,c} + \alpha_s\sum_{n=1}^{s-1}p_{0+n}\tilde{b}_{s-n,s} + (1-\alpha_s)p_{0+s} + f(s),
$$

with $f(s) := \sum_{c=0}^{s-1}p_{0,c}$.

So, now we have balance equations for p_k with $k = 0, 1, \ldots$. Multiplying both sides of each equation with z^k and summing all equations for $k = 0, 1, 2, \ldots$ leads to the following equation:

$$
\tilde{P}(z) = \sum_{k=0}^{\infty}\sum_{c=0}^{s}\alpha_c\tilde{b}_{s+k,c}p_{0,c}z^k + \alpha_s\sum_{k=0}^{\infty}\sum_{n=1}^{s-1}\tilde{b}_{s-n,s}p_{k+n}z^k + \alpha_s\sum_{k=1}^{\infty}\sum_{n=0}^{k-1}\tilde{b}_{s+n,s}p_{k-n}z^k +
$$

$$
\begin{aligned}
& (1-\alpha_s)\sum_{k=0}^{\infty} p_{k+s} z^k + f(s) \\
= \; & z^{-s}\sum_{c=0}^{s}\alpha_c p_{0,c}\left(\tilde{B}_c(z) - \sum_{i=0}^{s-1}\tilde{b}_{i,c} z^i\right) + \alpha_s z^{-s}\sum_{n=1}^{s-1}\tilde{b}_{s-n,s} z^{s-n}\sum_{k=0}^{\infty} p_{k+n} z^{k+n} + \\
& \alpha_s z^{-s}\sum_{n=0}^{\infty}\tilde{b}_{s+n,s} z^{s+n}\sum_{k=n+1}^{\infty} p_{k-n} z^{k-n} + z^{-s}(1-\alpha_s)\sum_{k=0}^{\infty} p_{k+s} z^{k+s} + f(s) \\
= \; & z^{-s}\sum_{c=0}^{s}\alpha_c p_{0,c}\left(\tilde{B}_c(z) - \sum_{i=0}^{s-1}\tilde{b}_{i,c} z^i\right) + \alpha_s z^{-s}\tilde{P}(z)\sum_{n=1}^{s-1}\tilde{b}_{n,s} z^n - \\
& \alpha_s z^{-s}\sum_{n=1}^{s-1}\tilde{b}_{n,s} z^n\sum_{i=0}^{s-n-1} p_i z^i + \alpha_s z^{-s}\left(\tilde{B}_s(z) - \sum_{i=1}^{s-1}\tilde{b}_{i,s} z^i\right)\left(\tilde{P}(z) - p_0\right) + \\
& z^{-s}(1-\alpha_s)\left(\tilde{P}(z) - \sum_{k=0}^{s-1} p_k z^k\right) + f(s) \\
= \; & z^{-s}\sum_{c=0}^{s}\alpha_c p_{0,c}\left(\tilde{B}_c(z) - \sum_{i=0}^{s-1}\tilde{b}_{i,c} z^i\right) - \alpha_s z^{-s}\sum_{n=1}^{s-1}\tilde{b}_{n,s} z^n\sum_{i=0}^{s-n-1} p_i z^i + \\
& \alpha_s z^{-s}\tilde{B}_s(z)\tilde{P}(z) - \alpha_s z^{-s} p_0\left(\tilde{B}_s(z) - \sum_{i=1}^{s-1}\tilde{b}_{i,s} z^i\right) + \\
& z^{-s}(1-\alpha_s)\left(\tilde{P}(z) - \sum_{k=0}^{s-1} p_k z^k\right) + f(s).
\end{aligned}
$$

From this we obtain the following expression for $\tilde{P}(z)$:

$$
\begin{aligned}
\tilde{P}(z) &= \frac{\sum_{c=0}^{s}\alpha_c p_{0,c}\hat{B}_c(z) - \alpha_s\sum_{n=1}^{s-1}\tilde{b}_{n,s}\sum_{i=0}^{s-n-1} p_i z^{n+i} - \alpha_s p_0\hat{B}_s(z) - (1-\alpha_s)\sum_{k=0}^{s-1} p_k z^k + z^s f(s)}{z^s - \alpha_s\tilde{B}_s(z) - (1-\alpha_s)} \\
&= \frac{\sum_{c=0}^{s} p_{0,c}\left(\alpha_c\hat{B}_c(z) - \alpha_s\hat{B}_s(z)\right) - \alpha_s\sum_{n=1}^{s-1}\tilde{b}_{n,s}\sum_{i=0}^{s-n-1} p_i z^{n+i} - (1-\alpha_s)\sum_{k=0}^{s-1} p_k z^k + z^s f(s)}{z^s - \alpha_s\tilde{B}_s(z) - (1-\alpha_s)},
\end{aligned}
$$

with $\hat{B}_c(z) = \tilde{B}_c(z) - \sum_{i=0}^{s-1}\tilde{b}_{i,c} z^i$.

This expression is somewhat more difficult than the expression obtained for $W(z)$ in the static arrival slot mechanism. In the expression for $P(z)$, there are $2s+1$ unknown stationary probabilities: $p_{0,0},\ldots,p_{0,s}$ and $p_0,\ldots,p_{s-1}$. It has already been proven in Section 4.2.2 that the denominator of $\tilde{P}(z)$ has s zeros on or within the complex unit circle. For these s zeros the numerator has to be zero as well. One of these zeros is always equal to 1. There is also the normalization condition: $\tilde{P}(1)=1$. In addition, we have the definition equation $p_0 := \sum_{c=0}^{s} p_{0,c}$ and the s balance equations for $p_{0,0},\ldots,p_{0,s-1}$. Here, we do not consider the equation for $p_{0,s}$. This equation has been replaced by the balance equation for p_0. Furthermore, since the equilibrium equations are dependent, we can omit exactly one of the balance equations for $p_{0,0},p_{0,1},\ldots,p_{0,s-1}$. Hence, this results in a total of $2s+1$ equations to solve for $2s+1$ unknown probabilities.

When compared with the analogon in the static model, it is rather difficult to derive an expression for $\mathbb{E}\tilde{P}=\tilde{P}'(1)$. Therefore, an analytical expression will not be given. This does not give more insights. $\mathbb{E}\tilde{P}$ can be approximated numerically, using the expression for the generating function $\tilde{P}(z)$ to approximate the derivative in $z=1$.

11.2 A special case of the dynamic periodic *Geo*|*G*|1-queue

In this section, a special case of the dynamic periodic *Geo*|*G*|1-queue which is related to the dynamic arrival slot mechanism will be discussed. Special refers to the particular choice of the service time distribution, which is allowed to be general in the *Geo*|*G*|1 queueing model.

In Section 11.1.2, an expression was given for the generating function of the total amount of work, denoted by $\tilde{P}(z)$. In this expression for $\tilde{P}(z)$ a couple of model parameters appear. We will outline how these parameters must be chosen to fit this special case of the *Geo*|*G*|1 queueing model. Assume that the individual arrival rate per slot, denoted by μ, is known.

In the expression for $\tilde{P}(z)$, the probabilities $\tilde{b}_{1,0},\tilde{b}_{2,0},\ldots,\tilde{b}_{s-1,s}$ appear. In the balance equations for $p_{0,1},\ldots,p_{0,s}$ the probabilities $\tilde{b}_{s,1},\ldots,\tilde{b}_{s,s}$ appear. Clearly, these probabilities depend on the arrival rate μ, but this is not indicated by means of a subscript for ease of notation. To obtain expressions for the probabilities in this special case, the results derived in subsection 9.2.1

are used. The relations are as follows:

$$\begin{aligned}
\tilde{b}_{1,0} &= \ldots = \tilde{b}_{s,0} = 0, \\
\tilde{b}_{1,1} &= b_{\lambda(1),1}, \\
&\ldots \\
\tilde{b}_{1,s} &= b_{\lambda(s),1}, \\
\ldots \\
\tilde{b}_{s,1} &= b_{\lambda(1),s}, \\
&\ldots \\
\tilde{b}_{s,s} &= b_{\lambda(s),s}.
\end{aligned}$$

It has already been outlined in subsection 9.2.1 how these probabilities can be determined.

Other parameters, besides the $b_{k,c}$'s, that appear in the expression for $\tilde{P}(z)$ or in the balance equations are the α_c's. These α_c's, for $c = 0, \ldots, s$, defined as the arrival probabilities, can be computed as follows:

$$\alpha_c = 1 - e^{-\lambda(c)} \left(1 + \frac{\lambda(c)}{3}\right)^3 .$$

This is directly based on the expression for the arrival probability in the static model, which was derived in Chapter 8.

The $\hat{B}_c(z)$'s in this special case, for $c = 0, \ldots, s$, can be found (at least numerically) by determining the $\tilde{B}_c(z)$'s in this special case of the $Geo|G|1$-queue. The $\tilde{B}_c(z)$'s can be found by using the results from Chapter 5. These generating functions of the service time of a super customer satisfy the following relation:

$$\tilde{B}_c(z) = B_{\lambda(c)}(z).$$

Using the iterative method to evaluate $B_{\lambda(s)}(z)$ that was presented in Chapter 5, the zeros of the equation $z^s - \alpha_s B_{\lambda(s)}(z) - (1 - \alpha_s)$ can be found. In fact, this is exactly the same equation that has to be solved numerically and therefore, these zeros are the same as in the static case.

For completeness, we mention that the parameters $\lambda(1), \ldots, \lambda(s)$ can be computed as $2\mu, \ldots, (1 + s)\mu$, respectively.

When the generating function $\tilde{P}(z)$ has been determined, it can be used to make the translation from the dynamic periodic $Geo|G|1$ queueing model to the dynamic arrival slot mechanism. Either exact translation methods or approximation methods can be used. All of these methods have already been outlined in Chapter 7. Whether a static or a dynamic $Geo|G|1$ queueing model is considered, there is no difference in using these translation methods. Therefore, we now only refer to Chapter 7.

Finally, one can obtain the first two moments of the total amount of work after the first slot of an arbitrary frame in the dynamic arrival slot mechanism. This total amount of work is related to both the waiting time of a super customer and the second component of the waiting time of an individual customer, the so-called tree queue waiting time. This component is needed in the analysis of the *total individual* waiting time. The next section is devoted to this subject. We will assume in the following section that results on the tree queue waiting time are available.

11.3 Analysis of the individual waiting time

This section focuses on the total waiting time of an individual customer in the dynamic arrival slot mechanism. For a definition and the decomposition of this waiting time into two components, we refer to Section 2.3. The first component of the waiting time (waiting room time) is analyzed first. Subsequently, results for the total waiting time are derived, using the results from the previous section on the tree queue waiting time.

In the beginning of this section, the generating function of the distribution of the first component of the waiting time (in the dynamic arrival slot mechanism) is studied. We also derive expressions for the expectation and variance of this first component of the waiting time.

Finally, we derive an expression for the mean total individual waiting time (so both components put together).

11.3.1 The first component of the individual waiting time

In this subsection, the generating function of the first component of the waiting time in the dynamic arrival slot mechanism will be derived. Furthermore, the first two moments of the first component of the mean waiting time of an individual customer are determined.

Assume again an arrival process that is Poisson with a constant rate μ per slot. A frame consists of at most $s+1$ slots, so including an arrival slot.

Recall that in the dynamic periodic $Geo|G|1$-queue, the stationary probability that the length of an arbitrary frame equals c, $c \in \{0, \ldots, s-1\}$ is denoted by $p_{0,c}$. Consequently, the probability that the length of an arbitrary frame in this dynamic periodic $Geo|G|1$ queueing model equals s is equal to:

$$P[\text{Arbitrary frame has length } s] = 1 - \sum_{c=0}^{s-1} p_{0,c}.$$

A frame of length c in the $Geo|G|1$ queueing model corresponds to a frame of length $c+1$ in the dynamic arrival slot mechanism. Now, define $g_{0,c}$ as the stationary probability that the length of an arbitrary frame in the dynamic arrival slot mechanism equals c for $c = 1, \ldots, s+1$. The following relations hold:

$$\begin{aligned} g_{0,c} &= p_{0,c-1}, \qquad c = 1, \ldots, s, \\ g_{0,s+1} &= 1 - \sum_{c=1}^{s} p_{0,c-1} \\ &= 1 - \sum_{c=1}^{s} g_{0,c}. \end{aligned}$$

Recall that the arrival process of individual customers is a Poisson process. This implies that an individual customer has a larger probability that it arrives during a long frame than during a short frame. In fact, the probability that an arbitrary customer arrives during a frame of length c, denoted by π_c, is given by:

$$\pi_c = \frac{c g_{0,c}}{\sum_{i=1}^{s+1} i g_{0,i}}.$$

Define $\tilde{V}_{\lambda(s)}$ as the random variable that represents the first component of the waiting time in the dynamic arrival slot mechanism. For the generating function of the distribution of $\tilde{V}_{\lambda(s)}$, denoted by $\tilde{V}_{\lambda(s)}(z)$, the following holds:

$$\begin{aligned}
\tilde{V}_{\lambda(s)(z)} &= \sum_{k=0}^{s} P(\tilde{V}_{\lambda(s)} = k) z^k \\
&= \sum_{k=0}^{s} \sum_{i=k+1}^{s+1} \pi_i \frac{1}{i} z^k \\
&= \sum_{k=0}^{s} \left(\sum_{i=k+1}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \right) z^k.
\end{aligned}$$

This result will be used in the next subsection.

The expectation of $\tilde{V}_{\lambda(s)}$, denoted by $\mathbb{E}\tilde{V}_{\lambda(s)}$ is given by:

$$\begin{aligned}
\mathbb{E}\tilde{V}_{\lambda(s)} &= \sum_{k=0}^{s} P(\tilde{V}_{\lambda(s)} = k) k \\
&= \sum_{k=0}^{s} \left(\sum_{i=k+1}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \right) k \\
&= \sum_{i=1}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \sum_{k=0}^{i-1} k
\end{aligned}$$

$$= \sum_{i=1}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \cdot \frac{i(i-1)}{2}.$$

The second moment of $\tilde{V}_{\lambda(s)}$, denoted by $\mathbb{E}\tilde{V}^2_{\lambda(s)}$, is given by:

$$\begin{aligned}
\mathbb{E}\tilde{V}^2_{\lambda(s)} &= \sum_{k=0}^{s} P(\tilde{V}_{\lambda(s)} = k)k^2 \\
&= \sum_{k=0}^{s} \left(\sum_{i=k+1}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \right) k^2 \\
&= \sum_{i=0}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \sum_{k=0}^{i-1} k^2 \\
&= \sum_{i=0}^{s+1} \frac{g_{0,i}}{\sum_{i=1}^{s+1} i g_{0,i}} \frac{i(i-1)(2i-1)}{6}.
\end{aligned}$$

11.3.2 The mean total waiting time

Let's denote the variable that represents the total waiting time of an individual customer in the dynamic arrival slot mechanism by $G_{ind,d}$. The expectation of this variable will be denoted by $\mathbb{E}G_{ind,d}$. The maximum number of slots in a frame will be denoted by $s+1$. In this subsection $\lambda(c)$ will be denoted by λ_c when λ is not used as a subscript.

As mentioned in the previous subsection, in the dynamic arrival slot mechanism, the stationary probability that the length of an arbitrary frame equals c, $c \in \{1, \ldots, s+1\}$ is denoted by $g_{0,c}$.

Individual customers may arrive during a frame of length c with $c \in \{1, \ldots, s\}$ or during a frame of length $s+1$.

In case of an arrival during a frame of length c, $c \leq s$, the mean total individual waiting time equals $\frac{1}{2}(c-1)$. This is because of the fact that there is no work in the system to wait for and consequently, the second component of the waiting time (tree queue waiting time) equals zero. Given the length c of the frame, the first part of the mean waiting time equals $\frac{1}{2}(c-1)$.

In case of an arrival during a frame of length $s+1$, the first part of the mean waiting time equals $\frac{1}{2}s$. The second component of the waiting time, the tree queue waiting time, can be obtained as follows. The mean total amount of work just before the second slot of an arbitrary frame equals $\mathbb{E}G_{\lambda(s)}$. This gives the following:

$$\mathbb{E}\left[G_{\lambda(s)}|\text{ previous frame was of length } s+1\right] = \frac{\mathbb{E}G_{\lambda(s)} - 0 \cdot \sum_{c=1}^{s} g_{0,c}}{1-\sum_{c=1}^{s} g_{0,c}} = \frac{\mathbb{E}G_{\lambda(s)}}{g_{0,s+1}}.$$

Recall that $\hat{p}_n$ denotes the probability that an individual customer which arrived in a batch of size n is *lucky* in the sense that it leaves the tree directly after the first (arrival) slot. Such a customer experiences no tree queue waiting time. $\hat{p}_n$ is given by:

$$\hat{p}_n = P(\text{none of the } n-1 \text{ other customers arrive in same mini-slot}) = \left(\frac{2}{3}\right)^{n-1}, \qquad n \geq 1.$$

So the second part of the mean waiting time of an individual customer that arrived during a frame of length $s+1$ and that arrives in a batch of n customers, equals $(1-\hat{p}_n)\frac{\mathbb{E}G_{\lambda(s)}}{g_{0,s+1}}$. Furthermore, define $q_{c,n}$ as the probability that n customers arrive during a frame of length c, $c=1,\ldots,s+1$ and $n \geq 1$:

$$\hat{q}_{c,n} = e^{-\lambda_{c-1}}\frac{\lambda_{c-1}^n}{n!}, \qquad c=1,\ldots,s+1, \quad n \geq 1.$$

Putting everything together, we can obtain an expression for the mean waiting time of an arbitrary individual customer as follows:

$$\mathbb{E}G_{ind,d} = \frac{g_{0,s+1}\sum_{n=1}^{\infty} n\hat{q}_{s+1,n}\left(\frac{1}{2}s + \frac{(1-\hat{p}_n)\mathbb{E}G_{\lambda(s)}}{g_{0,s+1}}\right) + \sum_{n=1}^{\infty}\sum_{c=1}^{s} n g_{0,c}\hat{q}_{c,n}\frac{c-1}{2}}{\sum_{n=1}^{\infty}\left(n\hat{q}_{s+1,n} + \sum_{c=1}^{s} n\hat{q}_{c,n}\right)}$$

$$= \frac{g_{0,s+1}\sum_{n=1}^{\infty} e^{-\lambda_s}\frac{\lambda_s^n}{n!}n\left(\frac{1}{2}s + \frac{\left(1-\left(\frac{2}{3}\right)^{n-1}\right)\mathbb{E}G_{\lambda(s)}}{g_{0,s+1}}\right) + \sum_{n=1}^{\infty}\sum_{c=1}^{s} n g_{0,c} e^{-\lambda_{c-1}}\frac{\lambda_{c-1}^n}{n!}\frac{c-1}{2}}{\sum_{n=1}^{\infty}\left(e^{-\lambda_s}\frac{\lambda_s^n}{n!}n + \sum_{c=1}^{s} e^{-\lambda_{c-1}}\frac{\lambda_{c-1}^n}{n!}n\right)}$$

$$= \frac{\lambda_s\left(\frac{1}{2}s g_{0,s+1} + \mathbb{E}G_{\lambda_s}\left(1-e^{-\frac{1}{3}\lambda_s}\right)\right) + \sum_{c=1}^{s} g_{0,c}\lambda_{c-1}\frac{c-1}{2}}{\sum_{c=1}^{s+1}\lambda_{c-1}}.$$

11.4 Numerical results on the waiting time

In this section, numerical results for the dynamic arrival slot mechanism are obtained, using the analytical procedures of the previous chapters. In this section, we restrict ourselves to the waiting time results. These results are eventually compared with simulation results. All the calculations are done for the mechanism with $s = 2$. This mechanism is chosen for two reasons. First, the calculations are relatively easy for mechanisms with small s. Another reason is that the mechanism with $s = 2$ has highest capacity among all arrival slot mechanisms.

11.4.1 Determination of $\tilde{P}(z)$

In this subsection, the generating function $\tilde{P}(z)$ will be determined for the dynamic arrival slot mechanism with $s = 2$. From this, $\mathbb{E}P_{\lambda(s)}$ can be determined. Results for $\mathbb{E}G_{\lambda(s)}$, the mean waiting time in the dynamic arrival slot mechanism (with $s = 2$) are obtained using either $\tilde{P}(z)$ or $\mathbb{E}P_{\lambda(s)}$.

In Section 11.2, an expression for $\tilde{P}(z)$ was given. When we specifically consider the case with $s = 2$, we observe the following.

In the expression for $\tilde{P}(z)$, the probabilities $\tilde{b}_{0,1}, \tilde{b}_{0,2}, \ldots, \tilde{b}_{1,2}$ appear. In the balance equations for $p_{0,1}, p_{0,2}$ the probabilities $\tilde{b}_{2,1}, \ldots, \tilde{b}_{2,2}$ appear. So in case $s = 2$ we have to determine $\tilde{b}_{0,1}, \tilde{b}_{0,2}, \tilde{b}_{1,1}, \tilde{b}_{1,2}, \tilde{b}_{2,1}$ and $\tilde{b}_{2,2}$. As argued before, these probabilities depend on the arrival rate μ, but this is not indicated by means of a subscript to keep notation simple. To obtain expressions for the probabilities, the results derived in subsection 9.2.1 are used. The relations are as follows:

$$\begin{aligned}
\tilde{b}_{0,0} &= \tilde{b}_{0,1} = \tilde{b}_{0,2} = 0, \\
\\
\tilde{b}_{1,0} &= b_{\lambda(0),1}, \\
\tilde{b}_{1,1} &= b_{\lambda(1),1}, \\
\tilde{b}_{1,2} &= b_{\lambda(2),1}, \\
\\
\tilde{b}_{2,0} &= b_{\lambda(0),2}, \\
\tilde{b}_{2,1} &= b_{\lambda(1),2}, \\
\tilde{b}_{2,2} &= b_{\lambda(2),2}.
\end{aligned}$$

For $b_{\lambda(0),1}$, $b_{\lambda(1),1}$ and $b_{\lambda(2),1}$ an expression has already been determined in Section 9.2.1. So in fact, the only thing that remains is to obtain expressions for $b_{\lambda(1),2}$ and $b_{\lambda(2),2}$. In general the following holds according to Section 9.2.1:

$$\begin{aligned}
b_{\lambda(s),2} &= (\pi_{\lambda(s)}(2) + \frac{6}{7}\pi_{\lambda(s)}(3))P(\tilde{B}(2) = 2) \\
&\quad + \frac{1}{7}\pi_{\lambda(s)}(3)P(\tilde{B}(3) = 2) + \sum_{n=4}^{7} \pi_{\lambda(s)}(n)P(\tilde{B}(n) = 3).
\end{aligned}$$

Plugging in the correct expressions ultimately yields the following:

$$\begin{aligned}
b_{\lambda(s),2} &= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{1}{6}\frac{2}{9}(\lambda(s))^2 + \frac{1}{9}\frac{2}{9}(\lambda(s))^3 + \frac{1}{54}\frac{38}{81}(\lambda(s))^3 + \frac{1}{4!}\frac{2296}{6561}(\lambda(s))^4 + \right. \\
&\quad \left. \frac{1}{5!}\frac{21400}{59049}(\lambda(s))^5 + \frac{1}{6!}\frac{7360}{59049}(\lambda(s))^6 + \frac{1}{7!}\frac{2240}{177147}(\lambda(s))^7 \right)
\end{aligned}$$

$$= \alpha_{\lambda(s)}^{-1} e^{-\lambda(s)} \left(\frac{1}{27}\lambda(s)^2 + \frac{73}{2187}\lambda(s)^3 + \frac{287}{19683}\lambda(s)^4 + \right.$$

$$\left. \frac{535}{177147}\lambda(s)^5 + \frac{92}{531441}\lambda(s)^6 + \frac{4}{1594323}\lambda(s)^7 \right).$$

Assume that the individual arrival rate per slot, denoted by μ, is known. Further quantities, besides the $b_{k,c}$'s, that appear in the expression for $\tilde{P}(z)$ or in the balance equations, will be discussed briefly:

- $\lambda(0), \lambda(1)$ and $\lambda(2)$ can be computed as $(0 + 1)\mu = \mu$, $(1 + 1)\mu = 2\mu$ and $(1 + 2)\mu = 3\mu$, respectively.

- α_c, for $c = 0, 1, 2$, defined as the arrival probabilities, can be computed as follows:

$$\alpha_c \;=\; 1 - e^{-\lambda(c)}(1 + \frac{\lambda(c)}{3})^3.$$

- $\tilde{B}_c(z)$, $c = 0, 1, 2$, the generating functions of the super customer service times, can be computed as $B_{\lambda(c)}(z)$.

- Using the iterative method to evaluate $B_{\lambda(2)}(z)$, the zeros of the equation $z^2 - \alpha_2 \tilde{B}_2(z) - (1 - \alpha_2)$ can be found. In fact, these zeros are the same as in the static case with $s = 2$. These zeros will be used to obtain two equations.

The normalization condition can be used numerically by applying l'Hôpital's rule. The two balance equations for $p_{0,0}$ and $p_{0,1}$ can be written down directly. This illustrates that constructing the set of 2+2+1 = 5 equations as described at the end of section 11.1.2 is straightforward. So, for any value of μ, $\tilde{P}(z)$ can be found and evaluated. As an illustration, the numerical procedure is carried out for $\mu = 1$. The plot of $\tilde{P}(z)$ is given in figure 11.1.

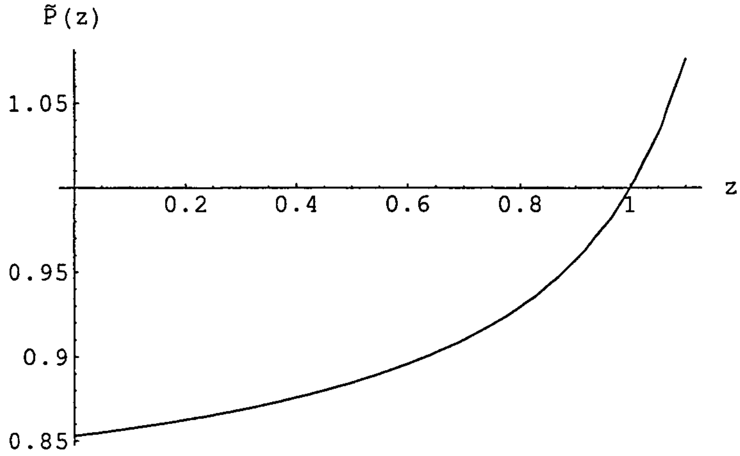

Figure 11.1: Plot of $\tilde{P}(z)$ in case of $\mu = 1$ and $s = 2$.

11.4.2 Numerical results for $\mathbb{E}G_{ind,d}$

For a range of values for μ, $\mathbb{E}P_{\lambda(2)}$ has been calculated. From these results, $\mathbb{E}G_{\lambda(2)}$ can be calculated for various values of μ, using for instance the translation method that has already been outlined in Chapter 7. When this quantity is known, the mean individual waiting time can be calculated. In this subsection we will present a graph in which this (calculated) waiting time is plotted against the arrival rate μ together with simulation results. This makes a comparison possible.

As indicated before, an analytical expression for $\mathbb{E}\tilde{P}_{\lambda(2)}$ has not been derived. To obtain $\mathbb{E}\tilde{P}_{\lambda(2)}$, we use the generating function and determine the derivative in $z = 1$ numerically. Figure 11.1 suggests that this will give an accurate approximation. However, this is not exactly the case. When we "zoom in" on the graph around $z = 1$, we observe that there is a vertical asymptote. See figure 11.2. This can be explained by numerical errors made in the calculations. Consequently, the approximations for $\mathbb{E}\tilde{P}_{\lambda(2)}$ will not be very accurate.
This approximation procedure is carried out for different values of μ. Eventually, this gives approximation results for $\mathbb{E}G_{ind,d}$.

In figure 11.3, the obtained simulation results for $\mathbb{E}G_{ind,d}$ as a function of μ in case of the dynamic arrival slot mechanism with $s = 2$ are plotted. Because of the fact that the simulations produced very accurate results (small confidence intervals), the simulation values can almost be considered as the

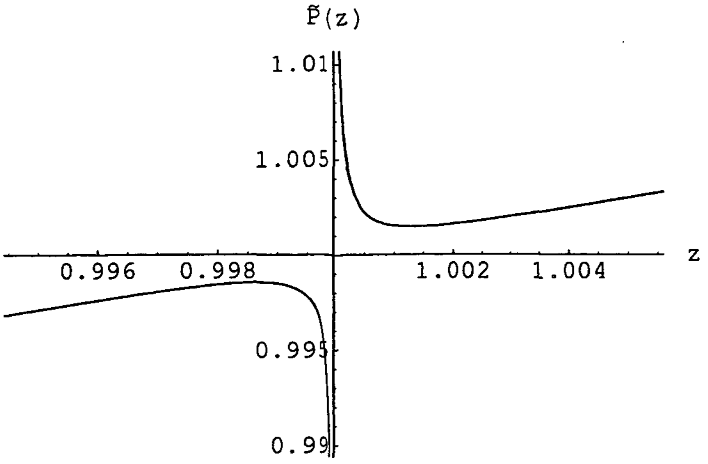

Figure 11.2: Plot of $\tilde{P}(z)$ around $z = 1$ in case of $\mu = 1$ and $s = 2$.

true values. These results can be compared with the approximation results found when using the approximation based on p_0 and p_1. As one can see, the approximation is reasonably good. However, small errors are present.

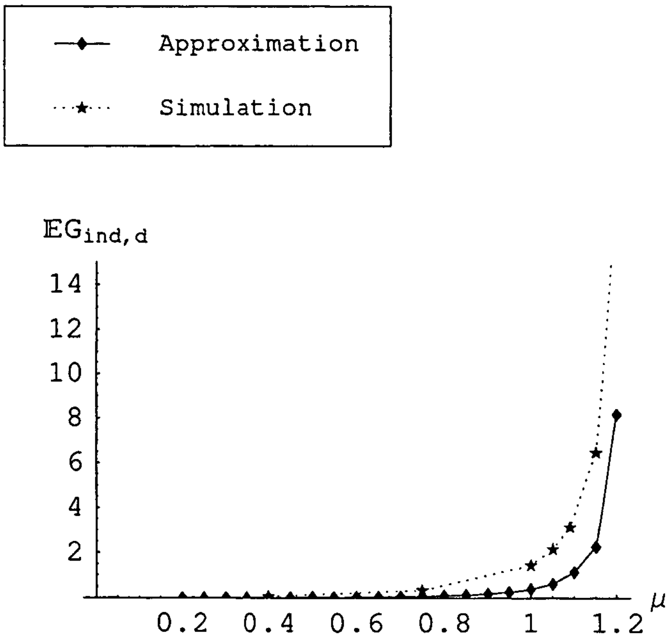

Figure 11.3: Plot of approximation and simulation results for $\mathbb{E}G_{ind,d}$ as a function of μ

Chapter 12

Analysis of the gated mechanism

This chapter concentrates on the waiting time as well as the service time analysis in case of a gated contention resolution mechanism. The first section will focus on approximating the mean tree length. Complications arise. The next section, Section 12.2, is devoted to an analysis of the gated mechanism when the conditional tree length is assumed to be the sum of independent continuous variables. This analysis will provide results that can be used to solve the complications. In Section 12.3, we return to the "discrete" analysis. In this section, the mean waiting time of an individual customer will be determined. In Section 12.4, the mean service time will be discussed. Subsequently, the determination of coefficients used in the approximations of the first four sections, is discussed. Numerical results are presented in Section 12.6.

Section 12.7 has a slightly different subject as the main part of this report. It considers different arrival process for a queue with a gate mechanism. Throughout this report it is assumed that the arrival process of individual customers is a Poisson process with a constant intensity. In Section 12.7, three variants on this arrival process are studied to obtain more insights in the effects of different arrival processes.

12.1 Approximation of the mean tree length

In this section, we will analyze the tree length distribution in the gated mechanism. We want to find an accurate approximation for the mean tree length in this mechanism. Finding an exact expression seems not possible at first sight, so we finally turn to an approximation method, that will be outlined in Section 12.2.

In the gated mechanism, the number of arrivals that enter the system after each gate period is Poisson distributed, but the actual parameter of this distribution depends on the length of the previous gate period. Assume that the individual arrival rate per slot equals μ. Let X_j denote the number of customers that enter the system after the gate opens for the j^{th} time. Let $\tilde{B}(n)$ denote the random variable that represents the length of a tree with n customers in it.

The following holds:

$$\begin{aligned}
P(X_j = k) &= \sum_{i=0}^{\infty} P(X_j = k \mid \tilde{B}(X_{j-1}) = i) P(\tilde{B}(X_{j-1}) = i) \\
&= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} P(X_j = k \mid \tilde{B}(X_{j-1}) = i \text{ and } X_{j-1} = m) \cdot \\
&\quad P(\tilde{B}(X_{j-1}) = i \mid X_{j-1} = m) P(X_{j-1} = m) \\
&= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} P(X_j = k \mid \tilde{B}(X_{j-1}) = i) P(\tilde{B}(m) = i) P(X_{j-1} = m) \\
&= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu}\frac{(i\mu)^k}{k!} P(\tilde{B}(m) = i) P(X_{j-1} = m), \\
&\quad j \geq 1, \qquad k = 0, 1, \ldots
\end{aligned}$$

Taking limits on both sides of the equation above and furthermore introducing p_k as the stationary distribution of the number of customers at the start of a tree with $p_k := \lim_{j\to\infty} P(X_j = k)$ for $k = 0, 1, \ldots$, gives:

$$\begin{aligned}
p_k &= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu}\frac{(i\mu)^k}{k!} P(\tilde{B}(m) = i) p_m, \\
&\quad k = 0, 1, \ldots
\end{aligned}$$

Multiplying both sides of the equation with z^k and summing over $k = 0, 1, \ldots$ yields:

$$
\begin{aligned}
\sum_{k=0}^{\infty} p_k z^k &= \sum_{k=0}^{\infty}\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu}\frac{(i\mu)^k}{k!}P(\tilde{B}(m)=i)p_m z^k \\
&= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu(1-z)}P(\tilde{B}(m)=i)p_m.
\end{aligned}
$$

Define the following generating function:

$$
P(z) \;:=\; \sum_{k=0}^{\infty} p_k z^k = \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu(1-z)}P(\tilde{B}(m)=i)p_m.
$$

The mean number of customers at the beginning of an arbitrary gate opening, denoted by $\mathbb{E}N$, equals:

$$
\begin{aligned}
\mathbb{E}N &= P'(1) \\
&= \sum_{i=0}^{\infty}\sum_{m=0}^{\infty} i\mu P(\tilde{B}(m)=i)p_m \\
&= \mu\sum_{m=0}^{\infty} p_m \sum_{i=0}^{\infty} P(\tilde{B}(m)=i)i \\
&= \mu\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m).
\end{aligned}
$$

It will turn out that $\mathbb{E}\tilde{B}(n)$ as a function of n can be approximated very well by means of a linear function of n. To justify this, a plot is given in figure 12.1 with $\mathbb{E}\tilde{B}(n)$ as a function of n. This plot turns out to be indeed almost linear. For a proof of the asymptotically linear behaviour, see [6]. In that paper, it is proven that:

$$
\mathbb{E}\tilde{B}(n) \;\approx\; \tilde{c_1}n + \tilde{c_0},
$$

$$
\text{with:} \qquad \tilde{c_0} = -\frac{1}{2}, \qquad \tilde{c_1} = \frac{1}{\operatorname{Log} 3}.
$$

The difference between the exact value and the approximation (as a function of n) is plotted in figure 12.2. This turns out to be an oscillating graph. The difference is small except for small values of n.

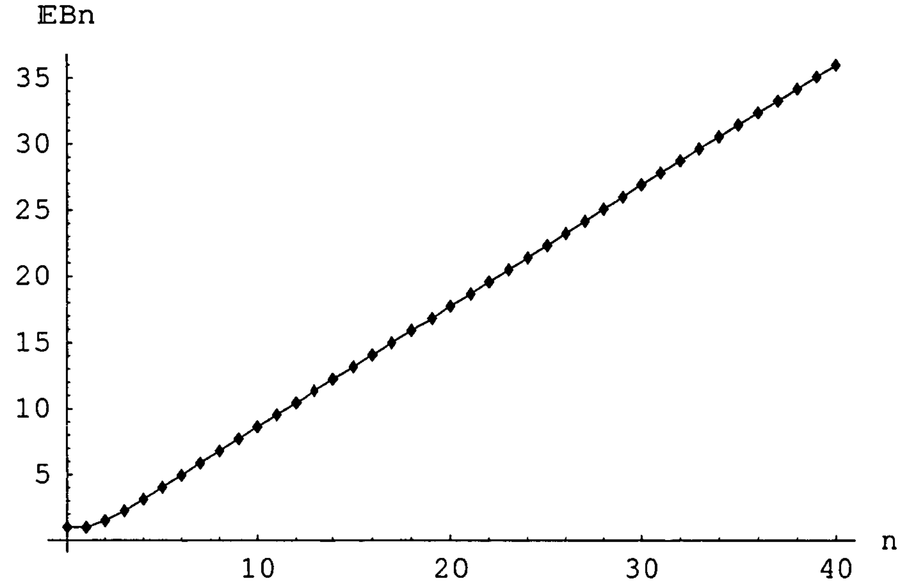

Figure 12.1: $\mathbb{E}\tilde{B}(n)$ as a function of n

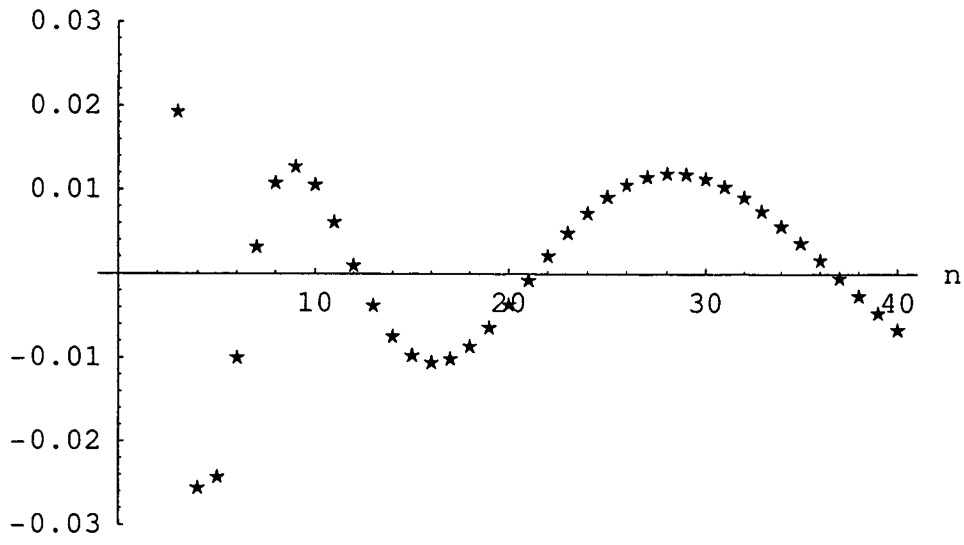

Figure 12.2: $\mathbb{E}\tilde{B}(n) - \left(-\frac{1}{2} + \frac{1}{\text{Log } 3}n\right)$ as a function of n

It is also interesting to have a look at $\frac{\mathbb{E}\tilde{B}(n)}{n}$ as a function of n. The plot is given in figure 12.3. One can see that for small trees it takes on average a little less time per customer to complete the tree. One can give several intuitive reasons for this phenomenon, which are all closely connected and all partly explain the phenomenon.

It is for instance true that the maximum throughput of a contention slot equals 3. 2 is also a relatively high throughput. In contention trees with 2 or 3 customers in it, this high throughput is more often achieved.

Anyway, this phenomenon might be in favour of a contention tree algorithm which handles most of the time only small trees (such as the arrival slot mechanism).

Substituting the linear approximation for $\mathbb{E}\tilde{B}(m)$ yields:

$$\begin{aligned} \mathbb{E}N &= \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) \\ &\approx \mu \sum_{m=0}^{\infty} p_m \left(\tilde{c_1} m + \tilde{c_0}\right) \end{aligned}$$

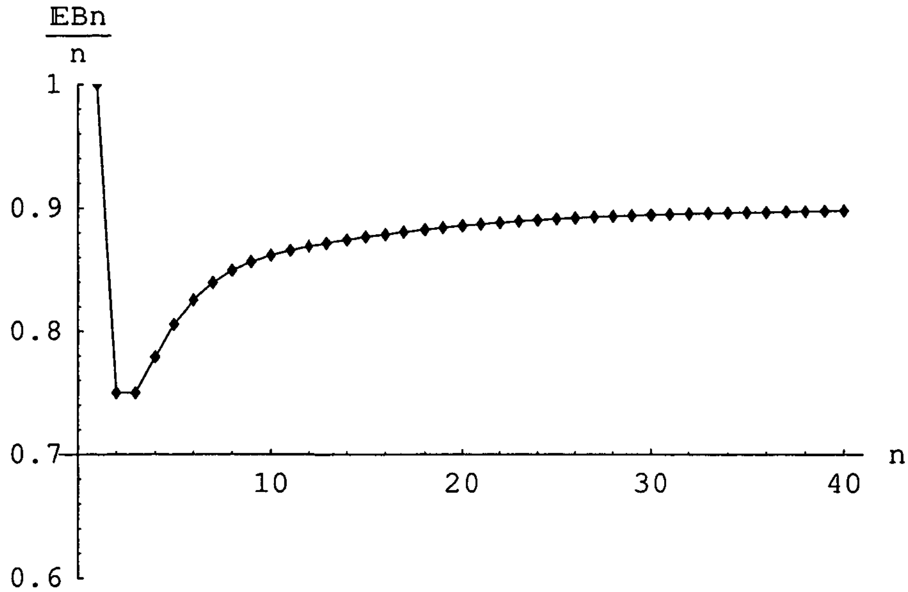

Figure 12.3: $\frac{\mathbb{E}\tilde{B}(n)}{n}$ as a function of n

$$= \mu(\tilde{c_1}\mathbb{E}N + \tilde{c_0}).$$

This gives the following approximation for $\mathbb{E}N$:

$$\mathbb{E}N \approx \frac{\mu\tilde{c_0}}{1-\mu\tilde{c_1}}.$$

Unfortunately, this approximation is not very useful. For all values of μ for which the system is stable, the approximation for $\mathbb{E}N$ is negative. So the approach that has been followed needs at least some modification. As indicated before, the linear approximation for $\mathbb{E}\tilde{B}(n)$ is not accurate for $n = 0, 1$. Therefore, if p_0 and p_1 were known, the inaccurate approximations could be replaced by the exact results. However, it seems difficult to obtain p_0 and p_1 exactly. In the next section, an approximation method will be outlined.

When p_0 and p_1 are known, the approximation for the mean number of customers is improved in the following way:

$$\mathbb{E}N = \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)$$

$$
\begin{aligned}
&\approx \mu\left(p_0 + p_1 + \sum_{m=2}^{\infty} p_m\left(\tilde{c_1} m + \tilde{c_0}\right)\right) \\
&= \mu\left(p_0 + p_1 + \tilde{c_1}\left(\mathbb{E}N - p_1\right) + \tilde{c_0}\left(1 - p_0 - p_1\right)\right).
\end{aligned}
$$

This gives finally the following approximation for $\mathbb{E}N$:

$$
\mathbb{E}N \approx \frac{\mu\left(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right)\right)}{1 - \mu\tilde{c_1}}.
$$

We now come to an approximation for the mean tree length, denoted by $\mathbb{E}\tilde{B}$. We have to take into account that we define a tree to consist of at least one customer. So gate openings where no waiting customers are present and their corresponding "gate periods" of length 1 do not contribute to the mean tree length. Therefore, the resulting approximation for the mean tree length becomes the following:

$$
\begin{aligned}
\mathbb{E}\tilde{B} &:= \left(1 - p_0\right)^{-1} \sum_{m=1}^{\infty} p_m \mathbb{E}\tilde{B}(m) \\
&= \left(1 - p_0\right)^{-1} \left(\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) - p_0\right) \\
&= \left(1 - p_0\right)^{-1} \left(\frac{\mathbb{E}N}{\mu} - p_0\right) \\
&\approx \left(1 - p_0\right)^{-1} \left(\frac{p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right)}{1 - \mu\tilde{c_1}} - p_0\right) \\
&= \frac{p_1 - \tilde{c_1} p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right) + \mu c_1 p_0}{\left(1 - \mu\tilde{c_1}\right)\left(1 - p_0\right)}.
\end{aligned}
$$

12.2 Approximation of p_0 and p_1

In this section, we will approximate p_0 and p_1 by assuming that the tree length is a continuous stochastic variable. The approach is based on [1]. The method will also lead to an approximation for the mean tree length and for the mean number of customers at an arbitrary gate opening.

The same notation as in the previous subsection is used.

We assume the following:

$$\begin{aligned} \tilde{B}(X_j) &= B_1 + \ldots + B_{X_j}, \qquad X_j = 2, 3, \ldots, \\ \tilde{B}(0) &= \tilde{B}(1) = 1. \end{aligned}$$

The B_i are i.i.d. random variables. The distribution of the B_i must be chosen carefully. In [6] it is proven that asymptotically the following holds:

$$\mathbb{E}\tilde{B}(n) \approx \tilde{c_1} n + \tilde{c_0},$$

$$\text{with:} \qquad \tilde{c_0} = -\frac{1}{2}, \quad \tilde{c_1} = \frac{1}{\text{Log } 3},$$

and

$$Var(\tilde{B}(n)) \approx n\tilde{d_1}$$

$$\text{with:} \qquad \tilde{d_1} = \frac{2}{\text{Log } 3} \sum_{i=1}^{\infty} \frac{1}{1+3^i} + \frac{1}{2\text{Log } 3} - \frac{1}{(\text{Log } 3)^2}.$$

This suggests that the distribution of the B_i must be chosen such that for the squared coefficient of variation $c(B_i)$ the following holds:

$$\begin{aligned} c(B_i) &:= \frac{Var(B_i)}{(\mathbb{E}B_i)^2} \\ &= \frac{\tilde{d_1}}{\tilde{c_1}^2} \\ &\approx 0.437124. \end{aligned}$$

Hence, an appropriate distribution for the B_i will be a mixture of an $Erlang(2, \nu)$ and an $Erlang(3, \nu)$ distributed random variable. With probability p, a B_i

will be distributed as $Erlang(2, \nu)$, and with probability $1-p$, a B_i will be distributed as $Erlang(3, \nu)$. The two parameters p and ν can be determined using the conditions on both the expectation and variance of the B_i:

$$\begin{aligned} \mathbb{E}B_i &= p\frac{2}{\nu} + (1-p)\frac{3}{\nu} = \frac{1}{\text{Log } 3}, \\ Var(B_i) &= p^2\frac{2}{\nu^2} + (1-p)^2\frac{3}{\nu^2} = 0.3621. \end{aligned}$$

Solving these two equations yields:

$$\begin{aligned} p &= 0.95461, \\ \nu &= 2.24709. \end{aligned}$$

For the Laplace-Stieltjes transform of the B_i, denoted by $\beta(s)$, the following holds:

$$\beta(s) = p\left(\frac{\nu}{\nu+s}\right)^2 + (1-p)\left(\frac{\nu}{\nu+s}\right)^3.$$

$A(x)$ is defined as the number of arrivals during a period of x slots. For X_{j+1} the following holds:

$$\begin{aligned} X_{j+1} &= A(B_1 + \ldots + B_{X_j}) \qquad \text{if } X_j > 1, \\ X_{j+1} &= A(1) \qquad \text{if } X_j \in \{0, 1\}. \end{aligned}$$

Using this, we can write:

$$\begin{aligned} \mathbb{E}\left[z^{X_{j+1}}\right] &= \mathbb{E}\left[z^{A(B_1+\ldots+B_{X_j})}\right] - P(X_j = 0) - P(X_j = 1)\mathbb{E}\left[z^{A(B_1)}\right] + \\ &\quad (P(X_j = 0) + P(X_j = 1))\,\mathbb{E}z^{A(1)} \end{aligned}$$

$$= \mathbb{E}\left[\beta(\mu(1-z))^{X_j}\right] - P(X_j = 0) - P(X_j = 1)\beta(\mu(1-z)) +$$

$$\left(P(X_j = 0) + P(X_j = 1)\right) e^{-\mu(1-z)}.$$

We define $X(z)$ to be the generating function of the steady-state distribution of the number of customers when the gate opens :

$$X(z) := \sum_{k=0}^{\infty} P(X = k) z^k.$$

Taking limits on both sides of the previous equation, we obtain the following:

$$X(z) = X\left(\beta(\mu(1-z))\right) - X(0) - X'(0)\beta(\mu(1-z)) + \left(X(0) + X'(0)\right) e^{-\mu(1-z)}.$$

Taking the derivative with respect to z on both sides gives another equation:

$$X'(z) = -\mu\beta'(\mu(1-z)X'\left(\beta(\mu(1-z))\right) + \mu\beta'(\mu(1-z))X'(0) +$$

$$\mu e^{-\mu(1-z)}\left(X(0) + X'(0)\right).$$

Both equations can be solved by iteration. Define the following (analogously as in [1]):

$$f(z) = \beta(\mu(1-z)),$$

$$f_j(z) = f(f_{j-1}(z)) \qquad \text{if } j > 0,$$

$$f_0(z) = z.$$

After k iterations the two expressions are the following:

$$
\begin{aligned}
X(z) &= \sum_{i=0}^{k} \left[-X(0) - X'(0)\beta(\mu(1-f_i(z))) + (X(0)+X'(0))\, e^{-\mu(1-f_i(z))}\right] + \\
& \quad X\left(\beta(\mu(1-f_k(z)))\right), \\
X'(z) &= \sum_{i=0}^{k} \left[\left(\prod_{j=1}^{i} f'(f_j(z)) \right) \left(f'(f_i(z))X'(0) + \mu e^{-\mu(1-f_i(z))} \left(X(0)+X'(0)\right)\right) \right] + \\
& \quad \prod_{j=0}^{k} f'(f_j(z)) X'\left(\beta(\mu(1-f_k(z)))\right).
\end{aligned}
$$

It can be shown that $\lim_{k\to\infty} f_k(z) = 1$. This implies that the term $X\left(\beta(\mu(1-f_k(z)))\right)$ in the expression for $X(z)$ tends to $X(1) = 1$ as $k \to \infty$. Furthermore, the last term in the expression for $X'(z)$ tends to zero if the system is stable. To see this, observe that $f'(1) = -\mu\beta'(0) = \rho$, where ρ equals the throughput of the system. In a stable system, it holds that $\rho < 1$. Therefore, the product tends to zero and $X'(1)$ is finite.

We are especially interested in finding $X(0)$ and $X'(0)$. When we take limits for $k \to \infty$ in both equations, we can obtain the probabilities $X(0)$ and $X'(0)$ by substituting $z = 0$ in the two equations and subsequently solve for $X(0)$ and $X'(0)$. The resulting solution will not be presented here. From a numerical point of view, this procedure can be used as well. The infinite sum can be approximated by a truncated sum and the two linear equations can easily be solved.

If this numerical procedure is carried out for a range of values for μ, the results can be presented in a graph. In this graph, both probabilities are given as a function of μ. For convenience, $X(0) + X'(0)$ is plotted as well. The graph is presented in figure 12.4.

Whether the obtained results for $X(0)$ and $X'(0)$ are indeed accurate approximations for the probabilities p_0 and p_1, respectively, is not clear yet. To obtain insight in this matter, a simulation experiment is set up. For several values of μ, one very long run of the gated mechanism is simulated. Subsequently, the two probabilities are estimated by the corresponding fractions. The obtained simulation results can be plotted in a graph as well. This is done in figure 12.5. The graphs of $X(0)$ and $X'(0)$ are plotted as well, to

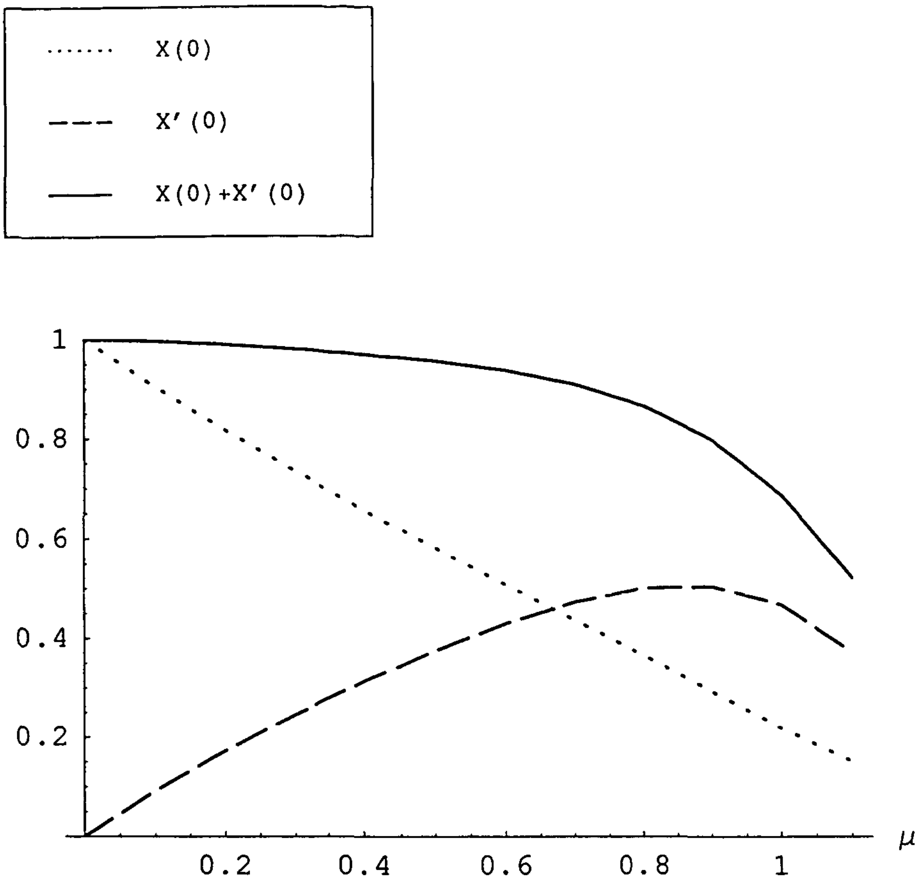

Figure 12.4: X_0 and X_0' as a function of μ.

facilitate the comparison. Inspecting figure 12.5, we can conclude that for small values of μ the approximation is accurate. For larger values of μ, the approximation becomes less where $X(0)$ seems to perform better dan $X'(0)$.

When $X(0)$ and $X'(0)$ are known, the expression for $X'(z)$ can be used to approximate $X'(1)$, which is equal to the expected number of customers at an arbitrary gate opening. In figure 12.6, $X'(1)$ is plotted as a function of the individual arrival rate μ. It must be remarked that truncating the infinite sum becomes a less accurate approximation method for values of μ that are close to the capacity of the system. For the lower range, the approximation is very accurate. $X'(1)$ is only calculated for a couple of values for μ. This explains the angles in the graph. In reality, the graph is smooth.

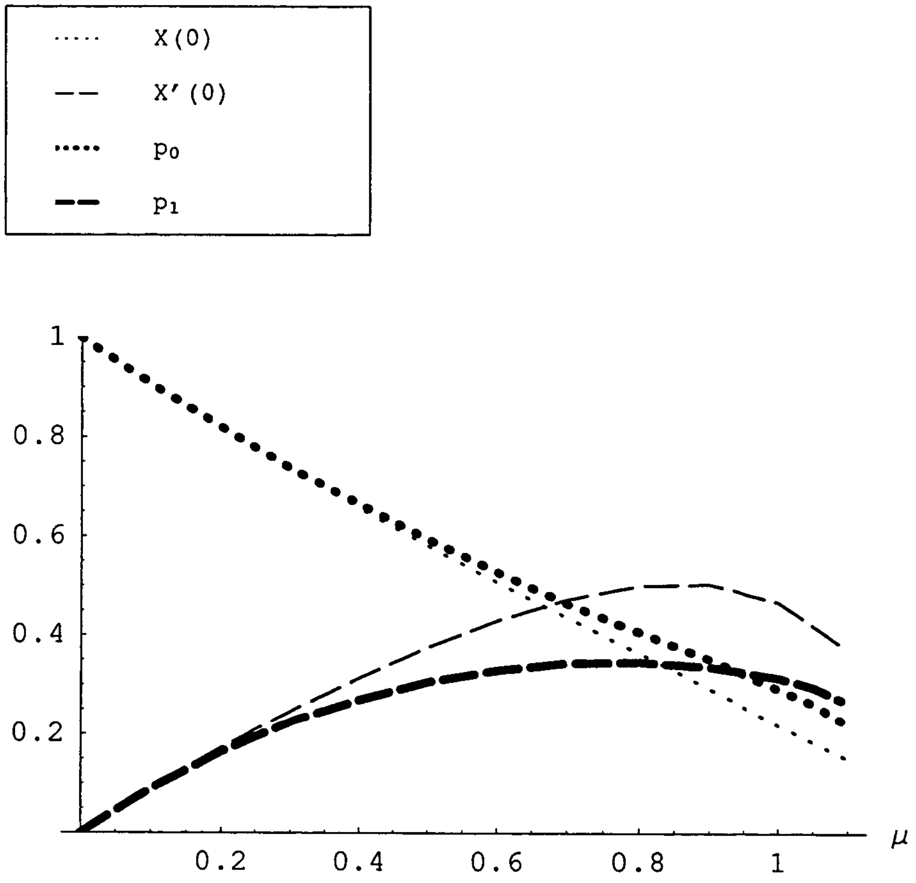

Figure 12.5: Simulation results for p_0 and p_1 as a function of μ.

In the next section, we will return to the exact analysis, by means of discrete variables rather than continuous variables. We will proceed with the waiting time analysis of an individual customer. The remaining part of the analysis in this chapter is based on the fact that accurate approximations for p_0 and p_1 are available.

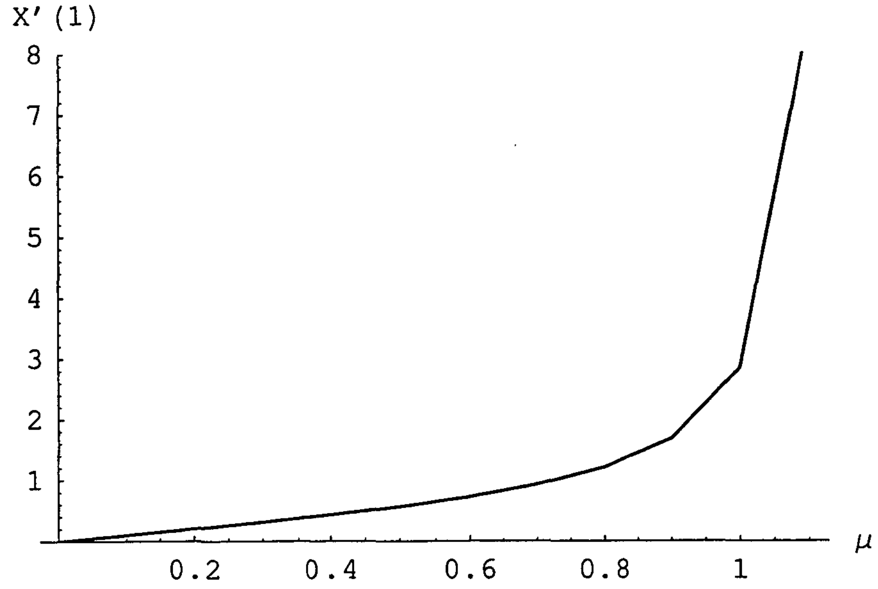

Figure 12.6: X_1' as a function of μ.

12.3 An approximation for the mean waiting time of an individual customer

Define q_i as the stationary probability that an arbitrary tree has length i, $i = 1, 2, \ldots$. The probability q_i is related to the p_m's in the following way:

$$q_i = \sum_{m=0}^{\infty} p_m P(\tilde{B}(m) = i), \qquad i = 1, 2, \ldots$$

The mean waiting time of an individual customer arriving during a tree of length i equals by definition $\frac{1}{2}(i-1)$. This leads to the following formula for the mean waiting time of an arbitrary individual customer, denoted by $\mathbb{E}\tilde{W}_{ind}$:

$$\mathbb{E}\tilde{W}_{ind} = \frac{\sum_{i=0}^{\infty} i q_i \frac{1}{2}(i-1)}{\sum_{i=0}^{\infty} i q_i}$$

$$= \frac{\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} p_m P(\tilde{B}(m) = i)\frac{1}{2}(i^2 - i)}{\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} p_m P(\tilde{B}(m) = i)i}$$

$$= \frac{\frac{1}{2}\left(\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m) - \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)\right)}{\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)}$$

$$= \frac{\frac{1}{2}\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m)}{\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)} - \frac{1}{2}.$$

The denominator of the first part of the expression for $\mathbb{E}\tilde{W}_{ind}$ equals $\frac{\mathbb{E}N}{\mu}$ as indicated before. So for the denominator, an accurate approximation is already available. In the numerator, a quantity appears that is closely related to the second moment of the tree length. How the numerator of the expression for $\mathbb{E}\tilde{W}$ can be approximated will be discussed now.

In [6] it is proven that the variance of $\tilde{B}(m)$ as a function of m, denoted by $Var(\tilde{B}(m))$, behaves asymptotically linear in m with a very small oscillation error:

$$Var(\tilde{B}(n)) \approx n\tilde{d}_1$$

$$\text{with:} \qquad \tilde{d}_1 = \frac{2}{\text{Log } 3}\sum_{i=1}^{\infty}\frac{1}{1+3^i} + \frac{1}{2\text{Log } 3} - \frac{1}{(\text{Log } 3)^2}.$$

Because of the fact that $\mathbb{E}\tilde{B}(n)$ as a function of n approximates a linear curve and since:

$$\mathbb{E}\tilde{B}^2(n) = Var(\tilde{B}(n)) + (\mathbb{E}\tilde{B}(n))^2,$$

we have that $\mathbb{E}\tilde{B}^2(n)$ as a function of n approximates a quadratic curve. To determine the exact values for $\mathbb{E}\tilde{B}^2(n)$, a recursion for $\mathbb{E}\tilde{B}^2(n)$ will be derived. Then $\mathbb{E}\tilde{B}^2(n)$ can be determined for arbitrary n. To obtain the recursion for $\mathbb{E}\tilde{B}^2(n)$ one can proceed as follows:

$$
\begin{aligned}
\mathbb{E}\tilde{B}^2(n) &= \sum_{k=1}^{\infty} P(\tilde{B}(n) = k)k^2 \\
&= \sum_{k=1}^{\infty} \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n P(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3) = k-1)\cdot \\
&\quad ((k-1)^2 + 2(k-1) + 1) \\
&= \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \\
&\quad \left(\mathbb{E}\left[\left(\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)\right)^2\right] + 2\mathbb{E}\left[\tilde{B}(n_1)+\tilde{B}(n_2)+\tilde{B}(n_3)\right] + 1\right) \\
&= \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \Big[\mathbb{E}\tilde{B}^2(n_1) + \mathbb{E}\tilde{B}^2(n_2) + \mathbb{E}\tilde{B}^2(n_3) + \\
&\quad 2\mathbb{E}\tilde{B}(n_1)\mathbb{E}\tilde{B}(n_2) + 2\mathbb{E}\tilde{B}(n_1)\mathbb{E}\tilde{B}(n_3) + 2\mathbb{E}\tilde{B}(n_2)\mathbb{E}\tilde{B}(n_3) + \\
&\quad 2\mathbb{E}\tilde{B}(n_1) + 2\mathbb{E}\tilde{B}(n_2) + 2\mathbb{E}\tilde{B}(n_3) + 1\Big], \qquad n \geq 2, \\
\mathbb{E}\tilde{B}^2(0) &= \mathbb{E}\tilde{B}^2(1) = 0.
\end{aligned}
$$

We stress that the boundary conditions $\mathbb{E}\tilde{B}^2(0) = \mathbb{E}\tilde{B}^2(1) = 0$ are only correct in the context of the recursion that has just been presented. In other situations, we use that $\mathbb{E}\tilde{B}^2(0) = \mathbb{E}\tilde{B}^2(1) = 1$.

Because of the fact that $\mathbb{E}\tilde{B}(n)$ can be obtained via another recursion (see Section 5.2), the above recursion can be used to obtain $\mathbb{E}\tilde{B}^2(n)$. For $n = 1, \ldots, 40$, $\mathbb{E}\tilde{B}^2(n)$ is determined and the results are summarized in figure 12.7.

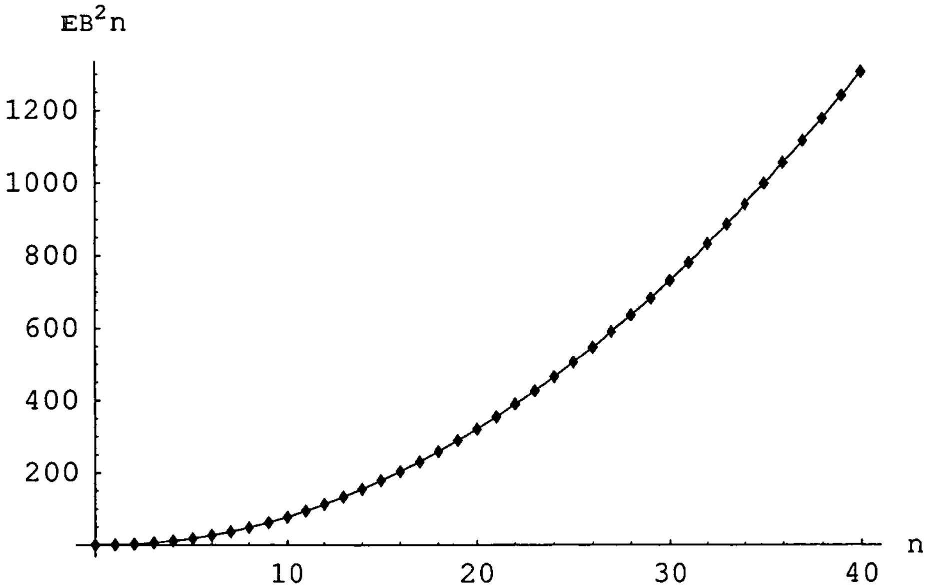

Figure 12.7: $\mathbb{E}\tilde{B}^2(n)$ as a function of n

A second-degree polynomial will be fitted to the curve in the previous plot. How to determine the coefficients of the polynomial accurately will be explained in Section 12.5.

In figure 12.8, the difference between the fitted points and the original points is plotted as a function of n.

The result is a graph that is almost equal to the original curve on all

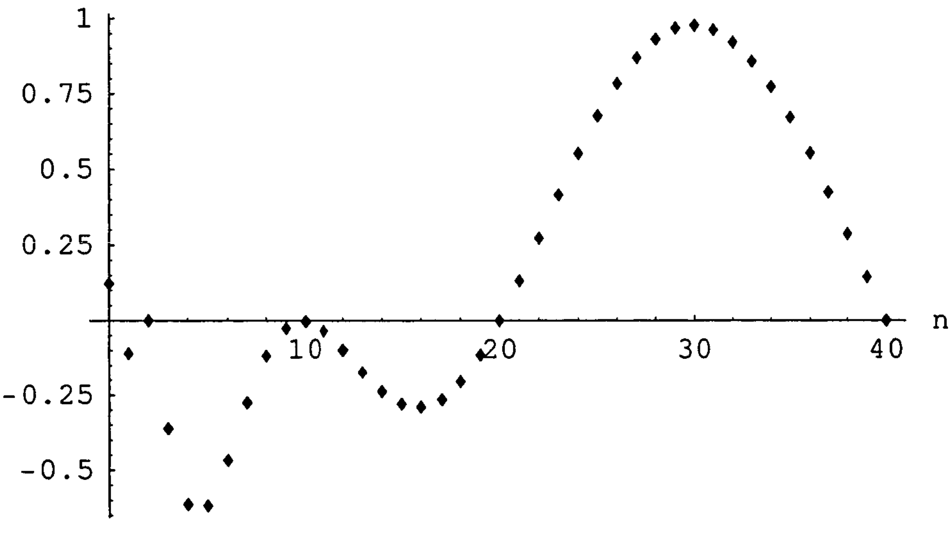

Figure 12.8: Absolute error of the quadratic approximation of $\mathbb{E}\tilde{B}^2(n)$ as a function of n

points. From this result, it can be concluded that $\mathbb{E}\tilde{B}^2(n)$ as a function of n can be aprroximated well by means of a quadratic curve in n, $n \geq 2$:

$$\mathbb{E}\tilde{B}^2(n) \approx \tilde{b_2}n^2 + \tilde{b_1}n + \tilde{b_0}.$$

The determination of the three coefficients will be discussed later.

Plugging this approximation for $\mathbb{E}\tilde{B}^2(n)$ into the sum-expression $\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m)$ yields the following:

$$\begin{aligned}
\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m) &= p_0 + p_1 + \sum_{n=2}^{\infty} p_n \mathbb{E}\tilde{B}^2(n) \\
&\approx p_0 + p_1 + \sum_{n=2}^{\infty} p_n \left(\tilde{b_2} n^2 + \tilde{b_1} n + \tilde{b_0}\right) \\
&= p_0 + p_1 + \tilde{b_2} \sum_{n=2}^{\infty} p_n n^2 + \tilde{b_1} \sum_{n=2}^{\infty} p_n n + \tilde{b_0}\,(1 - p_0 - p_1) \\
&= p_0 + p_1 + \tilde{b_2} \sum_{n=2}^{\infty} p_n n^2 + \tilde{b_1}\,(\mathbb{E}N - p_1) + \tilde{b_0}\,(1 - p_0 - p_1)\,.
\end{aligned}$$

An exact expression for $\sum_{n=2}^{\infty} p_n n^2$ can be derived as follows:

$$\begin{aligned}
\sum_{n=0}^{\infty} p_n n^2 &= \left(\frac{d}{dz}\left(z\frac{d}{dz}\left(\sum_{n=0}^{\infty} p_n z^n\right)\right)\right)_{z=1} \\
&= \left(\frac{d}{dz}\left(z\frac{d}{dz}\left(\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} e^{-i\mu(1-z)} P(\tilde{B}(m)=i) p_m\right)\right)\right)_{z=1} \\
&= \left(\frac{d}{dz}\left(z\left(\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} i\mu e^{-i\mu(1-z)} P(\tilde{B}(m)=i) p_m\right)\right)\right)_{z=1} \\
&= \left(\sum_{i=0}^{\infty}\sum_{m=0}^{\infty}\left(i\mu e^{-i\mu(1-z)} P(\tilde{B}(m)=i) p_m\right) + \left(z i^2 \mu^2 e^{-i\mu(1-z)} P(\tilde{B}(m)=i) p_m\right)\right)_{z=1} \\
&= \left(\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} i\mu P(\tilde{B}(m)=i) p_m\right) + \left(\sum_{i=0}^{\infty}\sum_{m=0}^{\infty} i^2\mu^2 P(\tilde{B}(m)=i) p_m\right) \\
&= \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) + \mu^2 \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m).
\end{aligned}$$

So eventually, we find the following:

$$\sum_{n=2}^{\infty} p_n n^2 = -p_1 + \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) + \mu^2 \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m).$$

Substituting this expression for $\sum_{n=2}^{\infty} p_n n^2$ in the expression for $\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m)$ yields:

$$\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m) = p_0 + p_1 + \tilde{b_2}\left(-p_1 + \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) + \mu^2 \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m)\right) + \tilde{b_1}\left(\mathbb{E}N - p_1\right) + \tilde{b_0}\left(1 - p_0 - p_1\right).$$

Because of the fact that both $\mathbb{E}N$ and $\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)$ can be approximated accurately, this gives an equation which can be solved for $\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m)$ to find the following:

$$\begin{aligned}
\sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m) &= \frac{p_0 + p_1 + \tilde{b_2}\left(-p_1 + \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)\right) + \tilde{b_1}\left(\mathbb{E}N - p_1\right) + \tilde{b_0}\left(1 - p_0 - p_1\right)}{1 - \tilde{b_2}\mu^2} \\
&= \frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right)\left(-p_1 + \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m)\right) + \tilde{b_0}\left(1 - p_0 - p_1\right)}{1 - \tilde{b_2}\mu^2} \\
&= \frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right)\left(-p_1 + \frac{\mu(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}(1 - p_0 - p_1))}{1 - \mu \tilde{c_1}}\right) + \tilde{b_0}\left(1 - p_0 - p_1\right)}{1 - \tilde{b_2}\mu^2}.
\end{aligned}$$

Eventually, $\mathbb{E}\tilde{W}_{ind}$ can be approximated using this result. The final approximation formula for $\mathbb{E}\tilde{W}_{ind}$ is the following:

$$\begin{aligned}
\mathbb{E}\tilde{W}_{ind} &\approx \frac{\frac{1}{2}\frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right)\left(-p_1 + \frac{\mu(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}(1 - p_0 - p_1))}{1 - \mu \tilde{c_1}}\right) + \tilde{b_0}(1 - p_0 - p_1)}{1 - \tilde{b_2}\mu^2}}{\frac{(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}(1 - p_0 - p_1))}{1 - \mu \tilde{c_1}}} - \frac{1}{2} \\
&= \frac{1}{2}\frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right)\left(-p_1 + \frac{\mu(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}(1 - p_0 - p_1))}{1 - \mu \tilde{c_1}}\right) + \tilde{b_0}\left(1 - p_0 - p_1\right)}{1 - \tilde{b_2}\mu^2} \cdot \\
&\quad \frac{1 - \mu \tilde{c_1}}{\left(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right)\right)} - \frac{1}{2}.
\end{aligned}$$

12.4 An approximation for the mean individual service time

Let $\tilde{S}(n)$ denote the random variable that represents the total aggregated service time of a batch consisting of n customers, $n \geq 2$. The aggregated service time of a batch of customers is the sum of all individual service times of those customers. The mean total aggregated service time of an *arbitrary* batch of customers, denoted by $\mathbb{E}\tilde{S}$, is given by:

$$\mathbb{E}\tilde{S} = \sum_{n=0}^{\infty} p_n \mathbb{E}\tilde{S}(n).$$

The aim is to find a good approximation for $\mathbb{E}\tilde{S}$. As we have already seen, $\mathbb{E}\tilde{B}(n)$ as a function of n can be approximated very well by means of a linear function of n. Now, it will be investigated whether $\mathbb{E}\tilde{S}(n)$ can be approximated very well by a *quadratic* function of n.

Therefore, $\mathbb{E}\tilde{S}(n)$ will be calculated exactly for relatively small n. How this can be done will be indicated now.

$\mathbb{E}\tilde{S}(n)$ can be determined recursively, using a formula that has already been presented (in a slightly different formulation) in Section 6.2 as an intermediate result. In Section 6.2, it was also explained how this recursion can be obtained. For an explanation of the following intermediate result, we refer to that section.

$$\begin{aligned}
\mathbb{E}\tilde{S}(n) = & \sum_{\substack{n_1, n_2 \neq 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \cdot \\
& \Big(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_2) + \mathbb{E}\tilde{S}(n_3) + (n_2 + n_3)\mathbb{E}\tilde{B}(n_1) + n_3 \mathbb{E}\tilde{B}(n_2)\Big) \\
& + \sum_{\substack{n_1, n_2 = 1, n_3 \neq 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \Big(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_3) + n_3 \mathbb{E}\tilde{B}(n_1)\Big) \\
& + \sum_{\substack{n_1, n_2 \neq 1, n_3 = 1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n \Big(\mathbb{E}\tilde{S}(n_1) + \mathbb{E}\tilde{S}(n_2) + n_2 \mathbb{E}\tilde{B}(n_1)\Big)
\end{aligned}$$

$$+ \sum_{\substack{n_1,n_2=1,n_3=1 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} \left(\frac{1}{3}\right)^n \left(\mathbb{E}\tilde{S}(n_1)\right) + n,$$

$n \geq 2,$

$$\mathbb{E}\tilde{S}(0) = \mathbb{E}\tilde{S}(1) = 0.$$

We emphasize that the boundary conditions given by $\mathbb{E}\tilde{S}(0) = \mathbb{E}\tilde{S}(1) = 0$ are only valid in the context of this recursion. In all other contexts, we use that $\mathbb{E}\tilde{S}(0) = 0$ and $\mathbb{E}\tilde{S}(1) = 1$.

When using this recursion to determine $\mathbb{E}\tilde{S}(n)$, the values of the $\mathbb{E}\tilde{B}(m)$'s are necessary. These values can be obtained straightforwardly by using the recursion given in Section 5.2. For $n = 1, \ldots, 40$, values for $\mathbb{E}\tilde{S}(n)$ are determined. The results are summarized in a plot, given in figure 12.9.

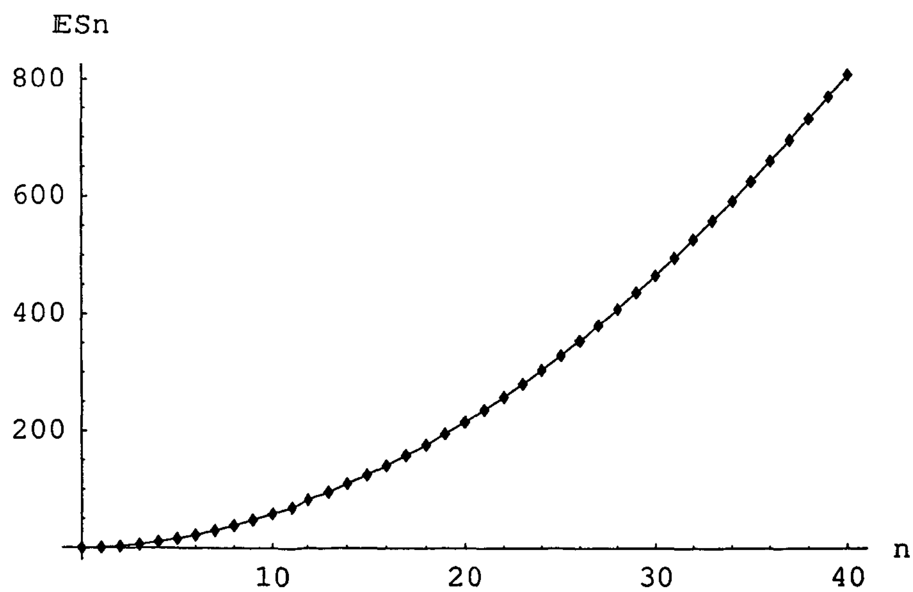

Figure 12.9: $\mathbb{E}\tilde{S}(n)$ as a function of n

In the plot, we can see that $\mathbb{E}\tilde{S}(n)$ as a function of n is certainly not a straight line. A second-degree polynomial will be fitted. How to determine the coefficients of the polynomial accurately will be explained in Section 12.5. In figure 12.10, the difference between the fitted points and the original points is plotted as a function of n.

The graph shows that the difference remains small, although the coefficients

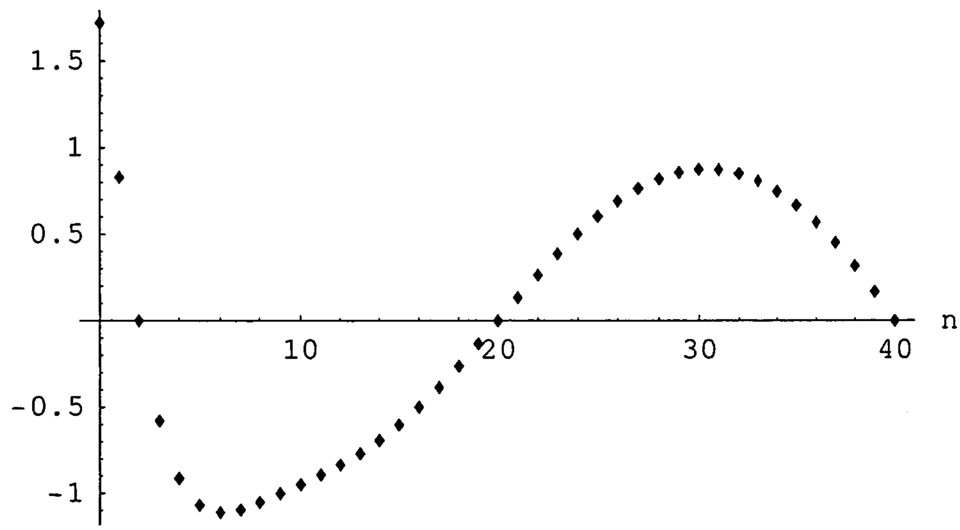

Figure 12.10: Absolute error of the quadratic approximation of $\mathbb{E}\tilde{S}(n)$ as a function of n

are not estimated in the most accurate way. From this result it can be concluded that $\mathbb{E}\tilde{S}(n)$ as a function of n can be approximated well by means of a quadratic curve in n, for $n \geq 2$:

$$\mathbb{E}\tilde{S}(n) \approx \tilde{a_2}n^2 + \tilde{a_1}n + \tilde{a_0}.$$

When this approximation is plugged into the expression for $\mathbb{E}\tilde{S}$, the following is obtained:

$$\mathbb{E}\tilde{S} = p_0 \cdot 0 + p_1 \cdot 1 + \sum_{n=2}^{\infty} p_n \mathbb{E}\tilde{S}(n)$$

$$
\begin{aligned}
&\approx p_1 + \sum_{n=2}^{\infty} p_n \left(\tilde{a_2} n^2 + \tilde{a_1} n + \tilde{a_0}\right) \\
&= p_1 + \tilde{a_2} \sum_{n=2}^{\infty} p_n n^2 + \tilde{a_1} \sum_{n=2}^{\infty} p_n n + \tilde{a_0} \sum_{n=2}^{\infty} p_n \\
&= p_1 + \tilde{a_2} \sum_{n=2}^{\infty} p_n n^2 + \tilde{a_1} \left(\mathbb{E}N - p_1\right) + \tilde{a_0} \left(1 - p_0 - p_1\right).
\end{aligned}
$$

For $\mathbb{E}N$ an approximation has indirectly already been given in the previous section:

$$
\mathbb{E}N \approx \frac{\mu \left(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0} \left(1 - p_0 - p_1\right)\right)}{1 - \mu \tilde{c_1}}.
$$

Now we are still looking for an accurate approximation for $\sum_{n=2}^{\infty} p_n n^2$. In the previous section an exact expression for $\sum_{n=2}^{\infty} p_n n^2$ has been derived:

$$
\sum_{n=2}^{\infty} p_n n^2 = -p_1 + \mu \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}(m) + \mu^2 \sum_{m=0}^{\infty} p_m \mathbb{E}\tilde{B}^2(m).
$$

Plugging in the accurate approximations for both sum-expressions that have been derived in the previous section as well, gives an approximation formula for $\sum_{n=2}^{\infty} p_n n^2$:

$$
\begin{aligned}
\sum_{n=2}^{\infty} p_n n^2 \approx\ & -p_1 + \mu \left(\frac{\left(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0} \left(1 - p_0 - p_1\right)\right)}{1 - \mu \tilde{c_1}} \right) + \\
& \mu^2 \left(\frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right) \left(-p_1 + \frac{\mu(p_0 + p_1 - \tilde{c_1} p_1 + \tilde{c_0}(1 - p_0 - p_1))}{1 - \mu \tilde{c_1}}\right) + \tilde{b_0} \left(1 - p_0 - p_1\right)}{1 - \tilde{b_2} \mu^2} \right).
\end{aligned}
$$

Plugging this approximation into the approximation formula for $\mathbb{E}\tilde{S}$, we finally obtain the following approximation for $\mathbb{E}\tilde{S}$:

$$
\begin{aligned}
\mathbb{E}\tilde{S} &\approx p_1 + \tilde{a_2}\sum_{n=2}^{\infty} p_n n^2 + \tilde{a_1}\left(\mathbb{E}N - p_1\right) + \tilde{a_0}\left(1 - p_0 - p_1\right) \\
&\approx p_1 + \tilde{a_2}\left(-p_1 + \mu\frac{(p_0 + p_1 - \tilde{c_1}p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right))}{1 - \mu\tilde{c_1}}\right) + \\
&\quad \tilde{a_2}\mu^2\left(\frac{p_0 + p_1 + \left(\tilde{b_1} + \tilde{b_2}\right)\left(-p_1 + \frac{\mu(p_0+p_1-\tilde{c_1}p_1+\tilde{c_0}(1-p_0-p_1))}{1-\mu\tilde{c_1}}\right) + \tilde{b_0}\left(1 - p_0 - p_1\right)}{1 - \tilde{b_2}\mu^2}\right) + \\
&\quad \tilde{a_1}\left(\frac{\mu\left(p_0 + p_1 - \tilde{c_1}p_1 + \tilde{c_0}\left(1 - p_0 - p_1\right)\right)}{1 - \mu\tilde{c_1}} - p_1\right) + \tilde{a_0}\left(1 - p_0 - p_1\right).
\end{aligned}
$$

The quantity of interest is the mean service time of an individual customer. This mean service time of an arbitrary individual customer, denoted by $\mathbb{E}\tilde{S}_{ind}$, can be obtained (as we have seen before) by dividing the mean total delay of an arbitrary batch by the mean number of customers that arrives in a batch. This mean number of customers equals $\mu\sum_{n=0}^{\infty} p_n\mathbb{E}\tilde{B}(n) = \mathbb{E}N$. So the mean delay of an arbitrary individual customer satisfies the following:

$$
\mathbb{E}\tilde{S}_{ind} = \frac{\mathbb{E}\tilde{S}}{\frac{\mu(p_0+p_1-\tilde{c_1}p_1+\tilde{c_0}(1-p_0-p_1))}{1-\mu\tilde{c_1}}}.
$$

12.5 Determination and estimation of the unknown coefficients

In this section it will be indicated how all the parameters that appeared in the final approximation for $\mathbb{E}\tilde{S}_{ind}$, presented at the end of the previous section, can be estimated or even determined exactly.

The coefficients $\tilde{c_0}$ and $\tilde{c_1}$ are already known. Since:

$$
\begin{aligned}
\mathbb{E}\tilde{B}^2(n) &= Var(\tilde{B}(n)) + (\mathbb{E}\tilde{B}(n))^2 \\
&\approx n\tilde{d_1} + (\tilde{c_1}n + \tilde{c_0})^2 \\
&= (\tilde{c_1})^2 n^2 + (\tilde{d_1} + 2\tilde{c_0}\tilde{c_1})n + (\tilde{c_0})^2,
\end{aligned}
$$

with

$$\tilde{d_1} = \frac{2}{\text{Log } 3}\sum_{i=1}^{\infty}\frac{1}{1+3^i} + \frac{1}{2\text{Log } 3} - \frac{1}{(\text{Log } 3)^2} \approx 0.362173,$$

it must be the case that:

$$\begin{aligned}
\tilde{b_0} &= (\tilde{c_0})^2 = \frac{1}{4}, \\
\tilde{b_1} &= \tilde{d_1} + 2\tilde{c_0}\tilde{c_1} \approx -0.548066, \\
\tilde{b_2} &= (\tilde{c_1})^2 \approx 0.828535.
\end{aligned}$$

The only coefficients that are still unknown are $\tilde{a_0}$, $\tilde{a_1}$ and $\tilde{a_2}$. When both the quadratic approximation for $\mathbb{E}\tilde{S}(n)$ and the linear approximation for $\mathbb{E}\tilde{B}(n)$ are plugged into the recursion formula for $\mathbb{E}\tilde{S}(n)$ with $n \geq 2$ (see Section 12.4), the following is obtained:

$$\begin{aligned}
\tilde{a_2}n^2 + \tilde{a_1}n + \tilde{a_0} &\approx \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \\
&\quad \Big[\tilde{a_2}n_1^2 + \tilde{a_1}n_1 + \tilde{a_0} + \tilde{a_2}n_2^2 + \tilde{a_1}n_2 + \tilde{a_0} + \tilde{a_2}n_3^2 + \tilde{a_1}n_3 + \tilde{a_0} + \\
&\quad (n_2+n_3)(\tilde{c_1}n_1 + \tilde{c_0}) + n_3(\tilde{c_1}n_2 + \tilde{c_0})\Big] + n \\
&= \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n \\
&\quad \left(3\left(\tilde{a_2}n_1^2 + \tilde{a_1}n_1 + \tilde{a_0}\right) + 3\tilde{c_1}n_1n_2 + 3\tilde{c_0}n_1\right) + n.
\end{aligned}$$

To evaluate the last expression, define the function $s_n(x,y,z)$ for $n \geq 2$ to be the following:

$$s_n(x,y,z) \;:=\; (x+y+z)^n$$

$$= \sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} x^{n_1} y^{n_2} z^{n_3}.$$

The following four relations hold:

$$\begin{aligned}
\sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} \cdot 1 &= s_n(1,1,1) \\
&= 3^n, \\
\sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} n_1 &= \frac{\partial}{\partial x} s_n(1,1,1) \\
&= n3^{n-1}, \\
\sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} n_1 n_2 &= \frac{\partial^2}{\partial x \partial y} s_n(1,1,1) \\
&= n(n-1)3^{n-2}, \\
\sum_{\substack{n_1,n_2,n_3 \\ n_1+n_2+n_3=n}} \frac{n!}{n_1!n_2!n_3!} n_1^2 &= \frac{\partial}{\partial x}\left(x \frac{\partial}{\partial x} s_n\right)(1,1,1) \\
&= n3^{n-1} + n(n-1)3^{n-2} = n(n+2)3^{n-2}.
\end{aligned}$$

These three results can be used to obtain the following final relation:

$$\begin{aligned}
\tilde{a}_2 n^2 + \tilde{a}_1 n + \tilde{a}_0 &\approx \left(\frac{1}{3}\right)^n \big(3\tilde{a}_2 n(n+2)3^{n-2} + 3\tilde{a}_1 n 3^{n-1} + 3\tilde{a}_0 3^n + \\
&\quad 3\tilde{c}_1 n(n-1)3^{n-2} + 3\tilde{c}_0 n 3^{n-1}\big) + n \\
&= \frac{\tilde{a}_2}{3} n(n+2) + \tilde{a}_1 n + 3\tilde{a}_0 + \frac{\tilde{c}_1}{3} n(n-1) + \tilde{c}_0 n + n.
\end{aligned}$$

It can be seen that on both sides of the equation, a quadratic function of n appears. By equating the coefficients of the quadratic term on both sides, the following equation is obtained:

$$
\begin{aligned}
\tilde{a_2} &= \frac{\tilde{a_2}}{3} + \frac{\tilde{c_1}}{3}, \\
\tilde{a_2} &= \frac{\tilde{c_1}}{2} = \frac{1}{2\mathrm{Log}\,3}.
\end{aligned}
$$

It is obvious that using a higher-order approximation in this case does not make sense. The equations that one can obtain by equating the coefficients of these higher-order terms on both sides will show that the coefficients must be all equal to zero.

Given the value of the coefficient $\tilde{a_2}$, the other two coefficients can be approximated accurately using linear regression:

$$
\mathbb{E}\tilde{S}(n) - \frac{1}{2\mathrm{Log}\,3}n^2 = \tilde{a_0} + \tilde{a_1}n + \epsilon, \qquad n = 2, 3, \ldots
$$

In figure 12.11, a graph is given in which the original points are marked with stars. The estimated linear function of n is plotted as a dotted line. This gives an indication of the goodness of fit.

Finally, the estimates for the two coefficients together with their 95%-confidence intervals are the following:

$$
\begin{aligned}
\tilde{a_0} &\approx -7.18 \pm 1.05, \\
\tilde{a_1} &\approx 2.054 \pm 0.044.
\end{aligned}
$$

In the method that has just been outlined, the value of $\tilde{a}_2$ was derived analytically. Subsequently, a regression was carried out to estimate $\tilde{a}_1$ and $\tilde{a}_0$. A slightly different approach is to determine a suitable value for $\tilde{a}_2$ by means of linear regression as well. This will lead to different coefficients. In Section 12.6, it is investigated whether this different approach leads to reasonably different values for the mean service time. Another remark is that the number of observations that is used for the linear regression can be varied. It could be for instance beneficial to have a good (regression) fit for small values

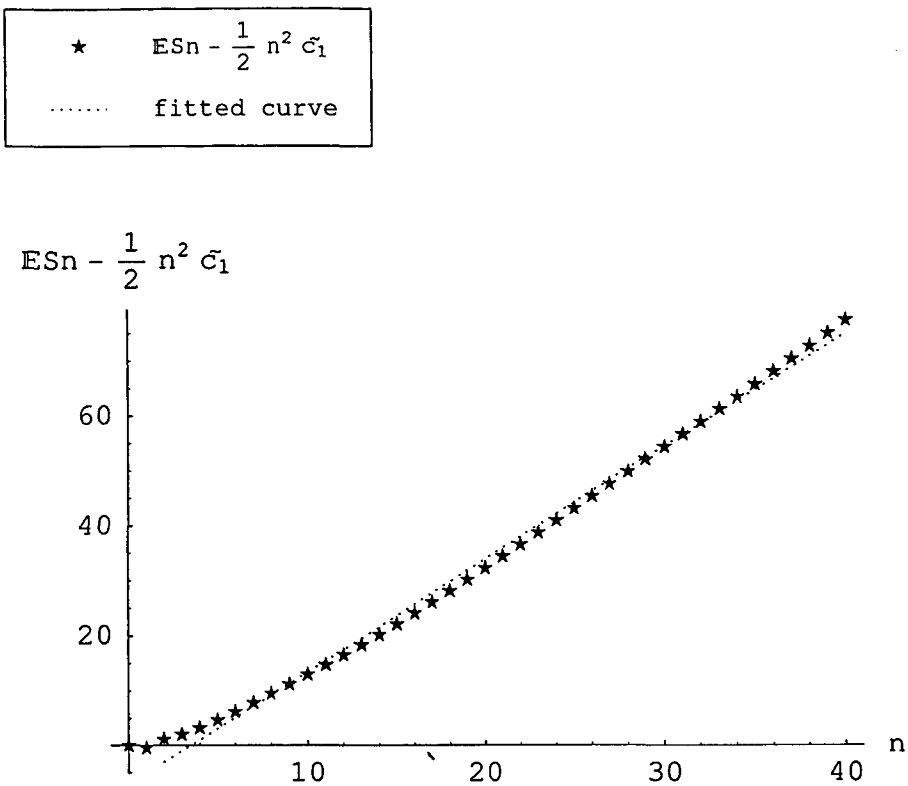

Figure 12.11: Regression results for estimation of $\tilde{a_0}$ and $\tilde{a_1}$

of n since trees with small numbers of initial customers are likely to occur in case the arrival intensity is not close to the capacity. The effect of using less observations to estimate the cōefficients is studied as well in Section 12.6.

12.6 Numerical results

In this section, some numerical results will be presented, which have been obtained by implementing the previously described procedures to determine the mean tree length as well as the mean waiting and service time of an arbitrary individual customer for the gated mechanism. The analytically found results are compared with simulation results for a range of values for μ.

12.6.1 Mean tree length results

The approximation formula that was presented for the mean tree length is used to evaluate approximations for the mean tree length as a function of the arrival rate μ. The formula contains the coefficients c_0 and c_1 for which the value has already been determined. For the determination of p_0 and p_1, two methods are used. The first method is the approximation as described in Section 12.2. The second method uses simulation to estimate the probabilities. The results of both methods are presented in the next two graphs.

In the first graph, figure 12.12, the mean tree length results obtained via the approximation method from Section 12.2 are compared with simulation results on the mean tree length which can be regarded as the true values. In the second graph, presented in figure 12.13, the results obtained via the second method are plotted and again compared with the simulation results on the mean tree length.

Observing figure 12.12, we see that the approximation method is very accurate for values of μ with $\mu \leq 0.7$. For larger values, the approximation becomes worse. It could be expected that as μ approaches the critical value of the capacity of the system, the approximation method is not very useful anymore. The fact that we see already for smaller values of μ a disagreement between exact results and approximations can be explained by two arguments. The first argument is the fact that the probabilities p_0 and p_1 that appear in the approximation formula for the mean tree length have been approximated as well. Inspecting figure 12.5, we see that the approximation of the probabilities becomes worse as well for $\mu > 0.7$. The second argument is of course that the formula for the mean tree length is an approximation formula as well.

To distinguish between the two effects that influence the quality of the approximation method for the mean tree length, figure 12.13 can be observed.

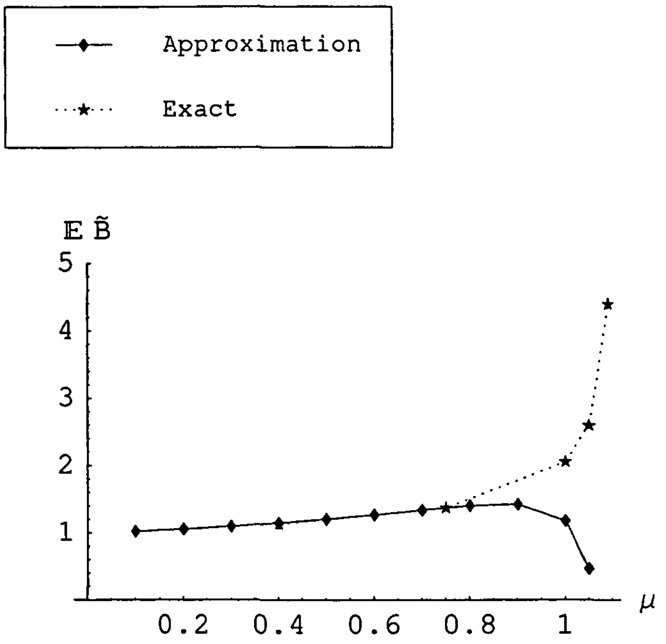

Figure 12.12: $\mathbb{E}\tilde{B}$ as a function of μ.

This approximation graph is obtained without the rough approximation of the probabilities p_0 and p_1. Instead, simulation is used to find the probabilities. These simulation values can almost be considered as the true values, so the influence of the first effect on the quality of the approximation is almost negligible. And indeed, we see a better approximation.

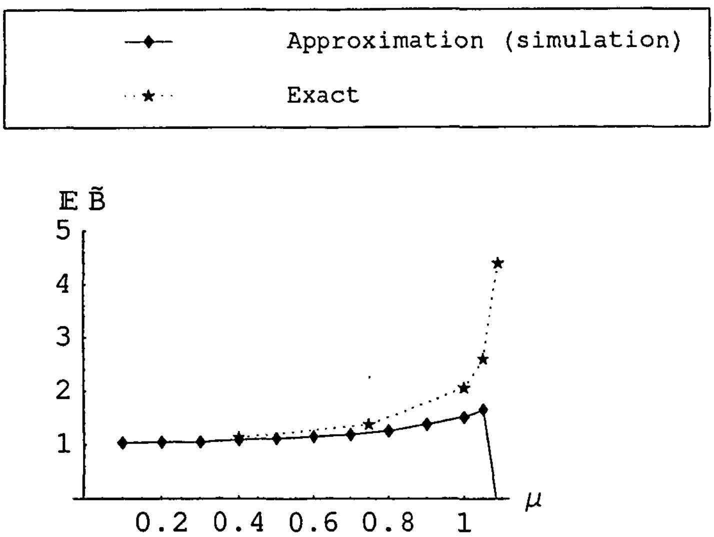

Figure 12.13: $\mathbb{E}\tilde{B}$ as a function of μ.

12.6.2 Mean waiting time results

In this subsection, the same approach is followed as in the previous subsection. Now, the mean waiting time is considered instead of the mean tree length. Again, two graphs are presented. The first figure, 12.14, contains the approximation results and the simulation results on the mean waiting time of an individual customer. Figure 12.15 compares the approximation results based on the true values of p_0 and p_1 with again the simulation results which can be regarded as the exact values of the mean waiting time.

Summarizing the results in both graphs, we can say that the quality of

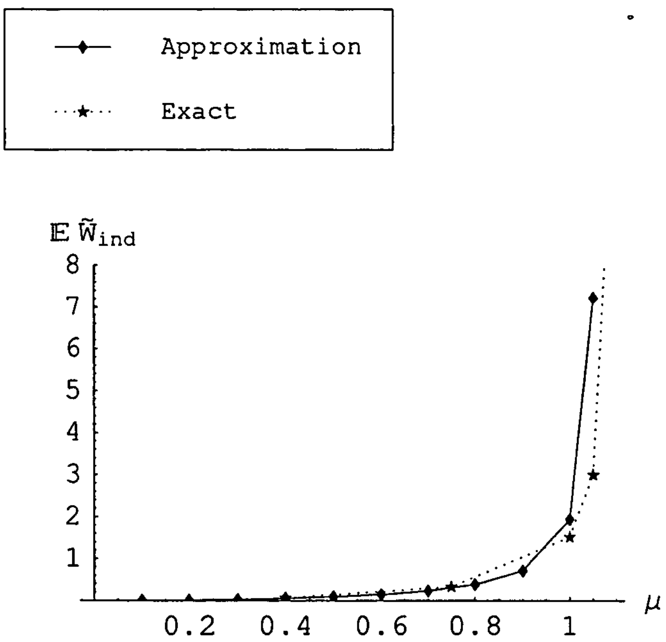

Figure 12.14: $\mathbb{E}\tilde{W}_{ind}$ as a function of μ.

the approximation is good, especially for the lower range of values for μ. The difference between exact values and approximations can be explained analogously as in the previous subsection. Again, the second graph shows a clearly better approximation as the first one, indicating that the error due to the approximated probabilities p_0 and p_1 is prevalent over the other error.

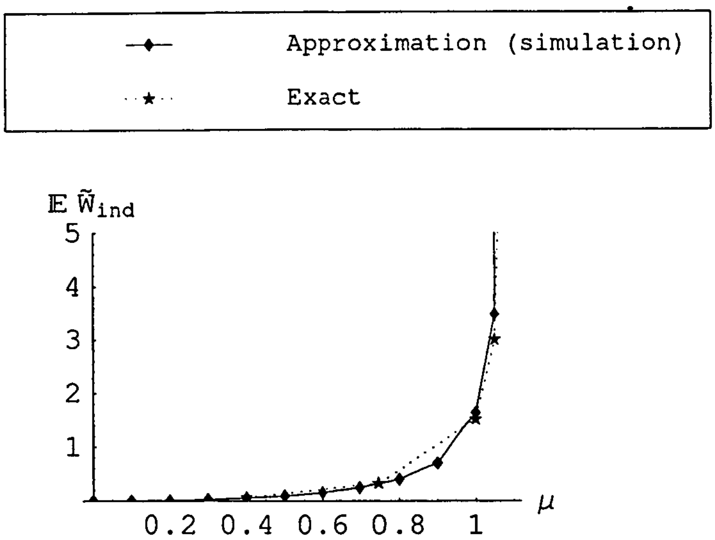

Figure 12.15: $\mathbb{E}\tilde{W}_{ind}$ as a function of μ.

12.6.3 Mean service time results

In this third subsection, the mean service time is considered. The quality of the approximation formula for the mean service time is studied, using the same approach as in the previous two subsections.

At the end of the previous section, it was indicated that determining suitable values for the coefficients $\tilde{a}_0$, $\tilde{a}_1$ and $\tilde{a}_2$ in the approximation formula, could be done in several ways. In the following graph, presented in figure 12.16, three slightly different methods are followed:

1. $\tilde{a}_2 := \frac{c_1}{2}$ and the other two coefficients are estimated using linear regression with observations $\mathbb{E}\tilde{S}(2), \ldots, \mathbb{E}\tilde{S}(40)$.
2. $\tilde{a}_0$, $\tilde{a}_1$ and $\tilde{a}_2$ are all estimated using linear regression with observations $\mathbb{E}\tilde{S}(2), \ldots, \mathbb{E}\tilde{S}(40)$.
3. $\tilde{a}_0$, $\tilde{a}_1$ and $\tilde{a}_2$ are all estimated using linear regression with observations $\mathbb{E}\tilde{S}(2), \ldots, \mathbb{E}\tilde{S}(6)$.

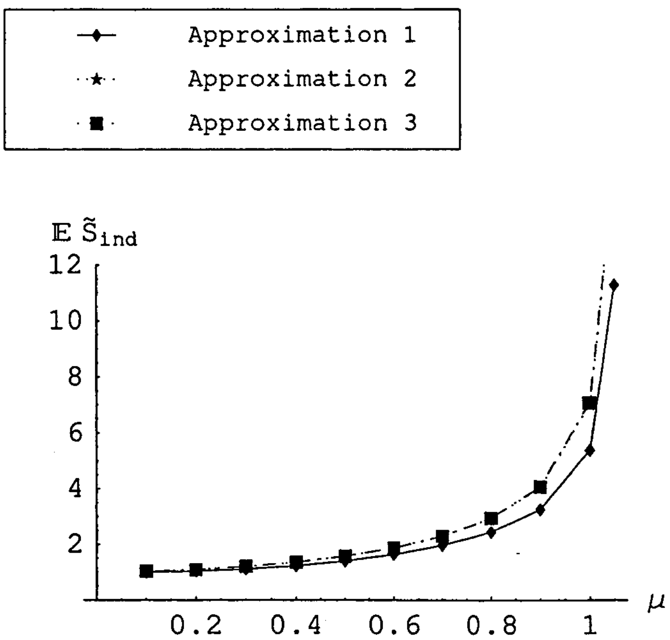

Figure 12.16: $\mathbb{E}\tilde{S}_{ind}$ as a function of μ.

The second approximation nearly coincides with the third. Hence it is not visible in the graph. Concluding, we can say that there is little difference between the three graphs, especially in the light-traffic zone. In the heavy-traffic zone, approximation 1 turns out to perform slightly better than the other two. Therefore, in figure 12.17, this approximation is compared with the simulation results on the mean service time, which can be regarded as the exact values. We see that the absolute error increases with μ.

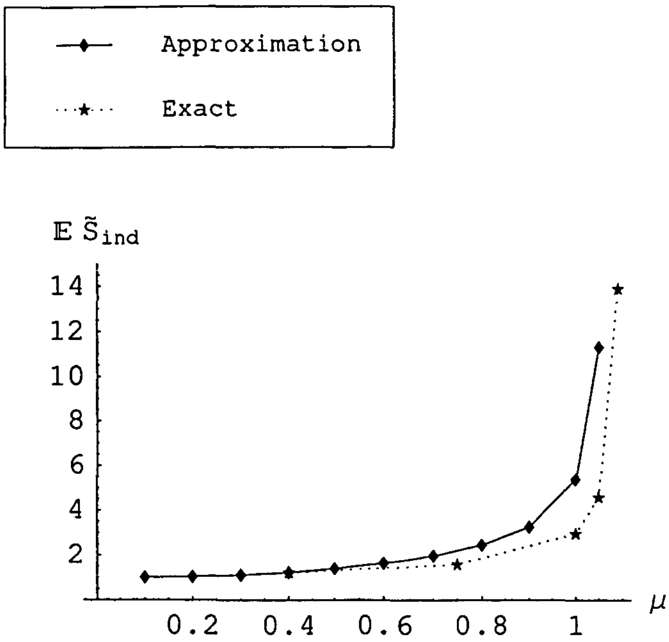

Figure 12.17: $\mathbb{E}\tilde{S}_{ind}$ as a function of μ.

In figure 12.18, we compare the true values with the approximation results that are obtained when the approximation formula is used, but the true values of p_0 and p_1 are substituted. As we have seen before, this leads to better approximation results. However, observe that the quality of the approximation results that are obtained in this way is also dependent on the estimated coefficients.

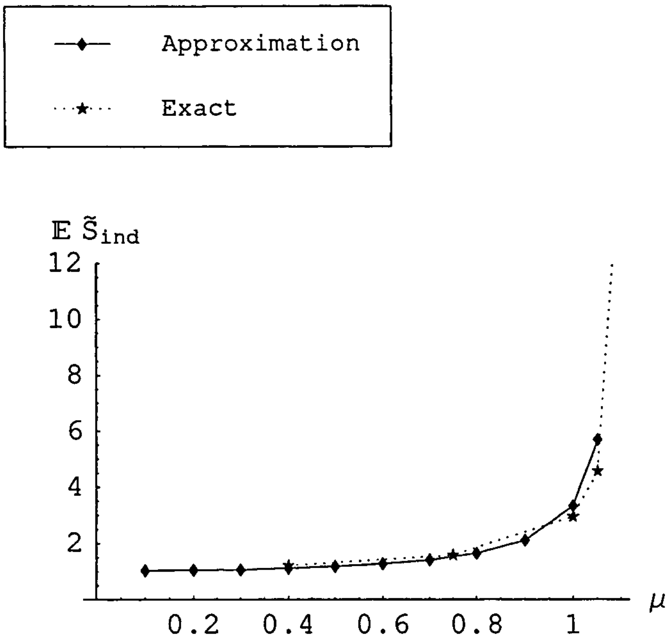

Figure 12.18: $\mathbb{E}\tilde{S}_{ind}$ as a function of μ.

12.7 Different arrival processes for a queue with a gate mechanism

12.7.1 Introduction

This section deals with a slightly different subject as the main part of this report. The arrival process of individual customers was assumed to be a Poisson process with constant arrival rate. This is a model assumption which obviously simplifies reality. In this section, it is studied how the analysis of the gated mechanism changes when we assume a different arrival process. The main motivation for this study is to obtain insight in the effect of different arrival processes on the model outcomes. In particular, we want to know if the currently used Poisson arrival process with constant rate is justified as an approximation. When it turns out that more realistic (and more complicated) arrival processes lead to rather different results, it can be concluded that the analysis is quite sensitive with respect to the choice of the arrival process. In this section, we will study only the equilibrium distribution of the number of customers at the beginning of a gate period to make the comparison between the different arrival processes. This will be outlined in more detail shortly.

In the gate mechanism as described in [1] on "A queue with a gate mechanism", there is a constant Poisson arrival rate λ and the corresponding network is an open network. In this section we will consider some different situations which have in common that they all correspond to a closed network, which probably resembles reality slightly better.

Consider a finite population of N customers. These customers arrive at a service facility according to a certain arrival process. The service facility uses a gate mechanism to allow customers entrance into the service facility. Each time the gate opens, all waiting customers enter the service facility and the gate closes immediately. When there are no waiting customers present, the gate remains open until the first arrival. When there are X_j customers at the moment that the gate opens for the j-th time, then these X_j customers receive service. This takes a (stochastic) amount of time $\tilde{B}(X_j)$. This epoch wil be called the j-th gate period. It will be assumed that

$$\begin{aligned} \tilde{B}(X_j) &= B_1 + \ldots + B_{X_j}, \qquad \text{if } X_j > 0, \\ \tilde{B}(0) &= B_1, \end{aligned}$$

where the B_i are i.i.d. with distribution $B(\cdot)$ chosen to be $Erlang(r, \nu)$.

12.7.2 Different arrival processes

Under the assumptions made in the previous section we are going to study and subsequently compare three different arrival processes. First,these three arrival processes will be discussed briefly. Afterwards, a motivation will be given.

1. **Poisson arrival process with constant intensity**
   Arrivals are generated according to a Poisson process with constant intensity $N\mu$, independent of the actual number of customers being served at the moment. When it is the case that this arrival process generates more than $N - k$ arrivals, this will be translated as that there are exactly $N - k$ arrivals, because of the finite population of N customers.

2. **Poisson arrival process with state-dependent intensity (1)**
When there are currently k customers being served at the facility, arrivals are generated according to a Poisson process with intensity $(N-k)\mu$. When it is the case that this arrival process generates more than $N-k$ arrivals, this will be translated as that there are exactly $N-k$ arrivals, because of the finite population of N customers.

3. **Poisson arrival process with state-dependent intensity (2)**
When there are currently k customers being served at the facility and there are already i customers waiting in front of the gate, with $0 \leq i \leq N-k$, arrivals are generated according to a Poisson process with intensity $(N-k-i)\mu$. This process can be viewed at slightly different. Suppose there are currently k customers at the service facility being served and that it will take a total time t to complete service for all these k customers. Every customer that is currently not being served, arrives during this time t with probability $1-e^{-\mu t}$. So given the time t, the actual number of arrivals is binomially $B(N-k, 1-e^{-\mu t})$ distributed.

In the next three subsections, the stationary distribution of X_j will be determined for the three different arrival processes. As briefly mentioned before, we will finally compare the three stationary distributions. This will be done for a couple of parameter settings. In this way, we obtain insight whether (and for what parameter settings) the three different processes lead to rather different model outcomes.

12.7.3 Poisson arrival process with constant intensity

Independent of the number of customers that is being served at the service facility, the arrival process is a Poisson process with constant rate equal to $N\mu$. When there are k waiting customers when the gate opens for the $(j+1)$-th time, it must have been the case that there were at most $N-k$ customers in the service facility during the previous gate period. (Consequently, in case of more than $N-k$ arrivals, the number of arrivals is truncated to $N-k$.) This leads to the following equations:

$$P(X_{j+1}=k) = \sum_{i=0}^{N-k} P(X_{j+1}=k|X_j=i)P(X_j=i), \qquad k=0,\ldots,N.$$

Now, the conditional probabilities are being considered:

$$
\begin{aligned}
P(X_{j+1}=k|X_j=0) &= P(X_{j+1}=k|X_j=1), \\
P(X_{j+1}=k|X_j=i) &= \int_{t=0}^{\infty} e^{-\mu N t}\frac{(\mu N t)^k}{k!}dB^{i*}(t) \\
&= \int_{t=0}^{\infty} e^{-\mu N t}\frac{(\mu N t)^k}{k!}e^{-\nu t}\frac{(\nu t)^{ir-1}}{(ir-1)!}\nu dt \\
&= \frac{(\mu N)^k}{k!}\frac{(\nu)^{ir}}{(ir-1)!}\int_{t=0}^{\infty} e^{-(\mu N+\nu)t}t^{ir-1+k}dt \\
&= \frac{(\mu N)^k}{k!}\frac{(\nu)^{ir}}{(ir-1)!}\frac{(ir-1+k)!}{(\mu N+\nu)^{ir+k}} \\
&= \binom{ir-1+k}{k}p_0^k(1-p_0)^{ir}, \qquad i=1,...,N-1, \\
&\quad \text{with } p_0=\frac{\mu N}{\mu N+\nu}, \\
P(X_{j+1}=k|X_j=N) &= \begin{cases} 1 \text{ if } k=0; \\ 0 \text{ if } k\neq 0. \end{cases}
\end{aligned}
$$

So, putting things together, we get the following expression:

$$
\begin{aligned}
P(X_{j+1}=k) &= \sum_{i=1}^{N-k}\binom{ir-1+k}{k}p_i^k(1-p_i)^{jr}P(X_j=i)+ \\
&\quad \binom{r-1+k}{k}p_1^k(1-p_1)^r P(X_j=0)+P(X_j=N)I_{\{k=0\}}, \qquad k=0,\dots,N.
\end{aligned}
$$

Now define the stationary distribution: $g_k=P(X_\infty=k)$. Finally we get the following set of equations to be solved:

$$
\begin{aligned}
g_k &= \sum_{j=1}^{N-k}\binom{jr-1+k}{k}p_j^k(1-p_j)^{jr}g_j+ \\
&\quad \binom{r-1+k}{k}p_1^k(1-p_1)^r g_0+g_N I_{\{k=0\}}, \qquad k=0,..N.
\end{aligned}
$$

12.7.4 State-dependent Poisson arrival process (1)

We can go through almost the same analysis as in the previous subsection. The only difference is the Poisson arrival intensity. In this arrival process, the arrival rate depends on the number of customers that is currently being served. When there are k customers currently being served, the arrival rate equals $(N-k)\mu$. We start again with the following equations:

$$P(X_{j+1}=k) = \sum_{i=0}^{N-k} P(X_{j+1}=k|X_j=i)P(X_j=i), \qquad k=0,\ldots,N.$$

Now, the conditional probabilities are being considered:

$$\begin{aligned}
P(X_{j+1}=k|X_j=0) &= P(X_{j+1}=k|X_j=1), \\
P(X_{j+1}=k|X_j=i) &= \int_{t=0}^{\infty} e^{-\mu(N-i)t}\frac{(\mu(N-i)t)^k}{k!}dB^{i*}(t) \\
&= \int_{t=0}^{\infty} e^{-\mu(N-i)t}\frac{(\mu(N-i)t)^k}{k!}e^{-\nu t}\frac{(\nu t)^{ir-1}}{(ir-1)!}\nu dt \\
&= \frac{(\mu(N-i))^k}{k!}\frac{(\nu)^{ir}}{(ir-1)!}\int_{t=0}^{\infty} e^{-(\mu(N-i)+\nu)t}t^{ir-1+k}dt \\
&= \frac{(\mu(N-i))^k}{k!}\frac{(\nu)^{ir}}{(ir-1)!}\frac{(ir-1+k)!}{(\mu(N-i)+\nu)^{ir+k}} \\
&= \binom{ir-1+k}{k}p_i^k(1-p_i)^{ir}, \qquad i=1,\ldots,N-1, \\
&\quad \text{with } p_i = \frac{\mu(N-i)}{\mu(N-i)+\nu}, \\
P(X_{j+1}=k|X_j=N) &= \begin{cases} 1 \text{ if } k=0; \\ 0 \text{ if } k\neq 0. \end{cases}
\end{aligned}$$

From now on, we can proceed exactly the same as in the previous section, eventually leading to the following set of equations to be solved:

$$g_k = \sum_{j=1}^{N-k}\binom{jr-1+k}{k}p_j^k(1-p_j)^{jr}g_j +$$

$$p_1^k(1-p_1)^r g_0 + g_N I_{\{k=0\}}, \qquad k = 0,\dots,N.$$

12.7.5 State-dependent Poisson arrival process (2)

This model has again a state-dependent Poisson arrival process. In this arrival process, the arrival rate depends on both the number of customers that is currently being served and the number of customers that is waiting in front of the gate. As already has been indicated in the model description in subsection 12.7.2, the number of arrivals conditionally on the lenght of the previous gate period and the number of customers k that is currently being served, is binomial distributed. This property will be used in the derivation of the equilibrium equations.

We start again with conditioning on the number of customers one period ago. So

$$P(X_{j+1}=k) = \sum_{i=0}^{N-k} P(X_{j+1}=k|X_j=i)P(X_j=i), \qquad k=0,\dots,N.$$

Now consider the conditional probabilities, given that the j-th gate period was of length t:

$$\begin{aligned} P(X_{j+1}=k|X_j=0,\tilde{B}(X_j)=t) &= P(X_{j+1}=k|X_j=1), \\ P(X_{j+1}=k|X_j=i,\tilde{B}(X_j)=t) &= \binom{N-i}{k}(1-e^{-\mu t})^k e^{-\mu t(N-i-k)}, \qquad i=1,\dots,N. \end{aligned}$$

Now integrating over all values of t leads to the following equations:

$$\begin{aligned} P(X_{j+1}=k|X_j=i) &= \int_{t=0}^{\infty}\binom{N-i}{k}(1-e^{-\mu t})^k e^{-\mu t(N-i-k)}dB^{i*}(t) \\ &= \int_{t=0}^{\infty}\binom{N-i}{k}(1-e^{-\mu t})^k e^{-\mu t(N-i-k)}e^{-\nu t}\frac{(\nu t)^{ir-1}}{(ir-1)!}\nu dt \\ &= \int_{t=0}^{\infty}\binom{N-i}{k}\sum_{n=0}^{k}\binom{k}{n}(-e^{-\mu t})^n e^{-\mu t(N-i-k)}e^{-\nu t}\frac{(\nu t)^{ir-1}}{(ir-1)!}\nu dt \end{aligned}$$

$$
\begin{aligned}
&= \sum_{n=0}^{k} \binom{N-i}{k}\binom{k}{n}(-1)^n \int_{t=0}^{\infty} e^{-\mu t(n+N-i-k)-\nu t}\frac{(\nu t)^{ir-1}}{(ir-1)!}\nu dt \\
&= \sum_{n=0}^{k} \binom{N-i}{k}\binom{k}{n}(-1)^n \frac{(\nu)^{ir}}{(\mu(n+N-i-k)+\nu)^{ir}}, \\
& i = 1,\dots,N.
\end{aligned}
$$

Putting this last result in the previous equation, we obtain the following equations:

$$
\begin{aligned}
P(X_{j+1}=k) &= \sum_{i=1}^{N-k}\sum_{n=0}^{k} \binom{N-i}{k}\binom{k}{n}(-1)^n \frac{(\nu)^{ir}}{(\mu(n+N-i-k)+\nu)^{ir}} P(X_j=i) \\
&+ \sum_{n=0}^{k} \binom{N-1}{k}\binom{k}{n}(-1)^n \frac{(\nu)^{r}}{(\mu(n+N-1-k)+\nu)^{r}} P(X_n=0), \\
& k = 0,\dots,N.
\end{aligned}
$$

Introducing the stationary distribution of the X_j's as $g_k = P(X_\infty = k)$, we get the following set of equations to be solved:

$$
\begin{aligned}
g_k &= \sum_{j=1}^{N-k}\sum_{n=0}^{k} \binom{N-j}{k}\binom{k}{n}(-1)^n \frac{(\nu)^{jr}}{(\mu(n+N-j-k)+\nu)^{rj}} g_j \\
&+ \sum_{n=0}^{k} \binom{N-1}{k}\binom{k}{n}(-1)^n \frac{(\nu)^{r}}{(\mu(n+N-1-k)+\nu)^{r}} g_0, \qquad k = 0,\dots,N.
\end{aligned}
$$

12.7.6 Numerical results

In this subsection, some numerical results on the g_k's, the stationary probabilities, will be presented. Recall that g_k represents the probability that the number of customers at an arbitrary gate opening equals k.

Four different models are compared:

1. Poisson arrival process with constant intensity.
2. Poisson arrival process with state-dependent intensity (1).
3. Poisson arrival process with state-dependent intensity (2).

These are just the three arrival processes as described and analyzed in the previous subsection. The fourth arrival process is an approximation for the "open network" model. We will return to this model later.

The four models are compared in three different situations. These situations are characterized by the choices for the model parameters N, μ, ν and s. The approximation for the open network model is obtained by analyzing the model with a Poisson arrival process with constant intensity but with $2N$ customers instead of N. Only the stationary probabilities $g_0, \ldots, g_N$ are shown in the graph.

The three situations that are considered are characterized by:

1. $N = 100$, $\mu = 1$, $\nu = 70$, $s = 1$.
2. $N = 100$, $\mu = 1$, $\nu = 100$, $s = 1$.
3. $N = 100$, $\mu = 1$, $\nu = 200$, $s = 2$.

The graphs are given in figures 12.21 until 12.30. The first four graphs contain results on situation 1, the next four on situation 2 and the final four on situation 3. After the presentation of the graphs, some remarks will be made on the numerical results.

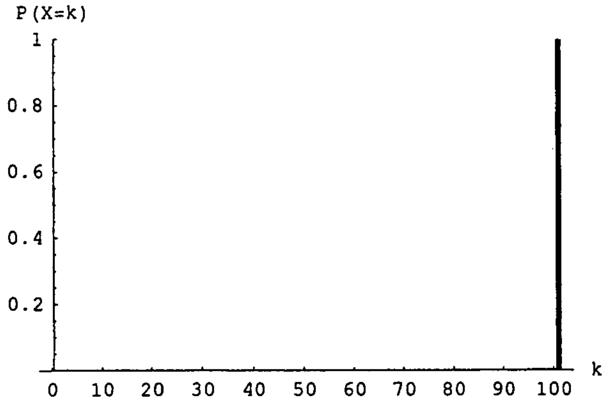

Figure 12.19: Situation 1, Poisson arrival process, with constant intensity.

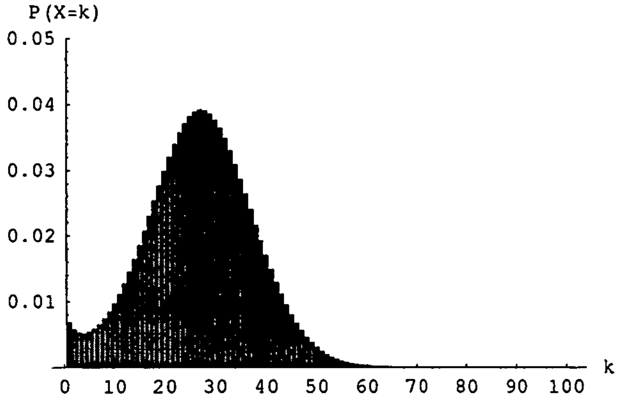

Figure 12.20: Situation 1, Poisson arrival process with state-dependent intensity (1).

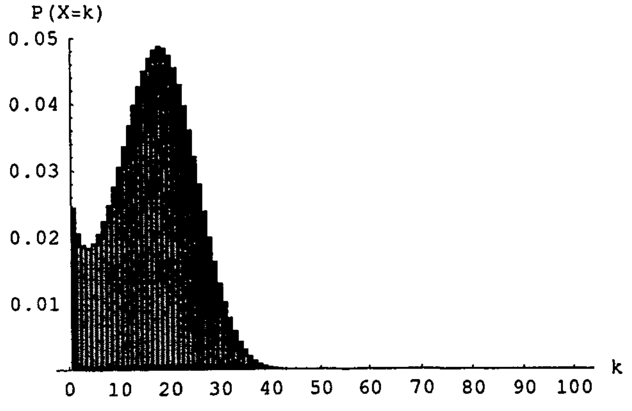

Figure 12.21: Situation 1, Poisson arrival process with state-dependent intensity (2).

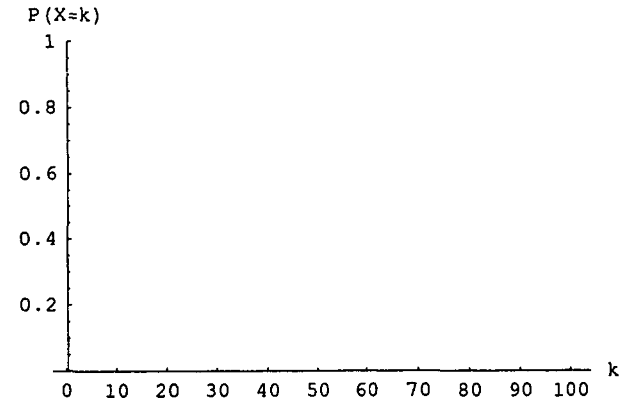

Figure 12.22: Situation 1, Poisson arrival process, "open network".

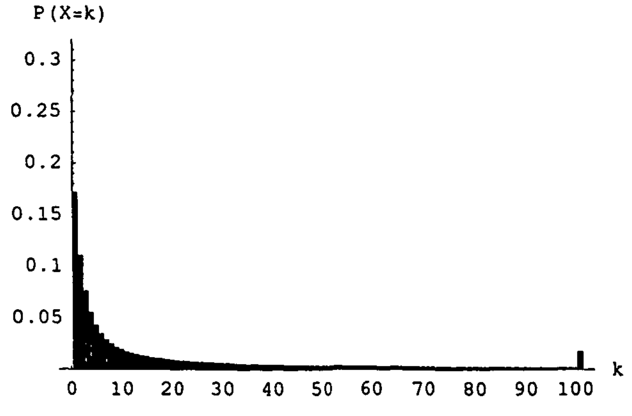

Figure 12.23: Situation 2, Poisson arrival process, with constant intensity.

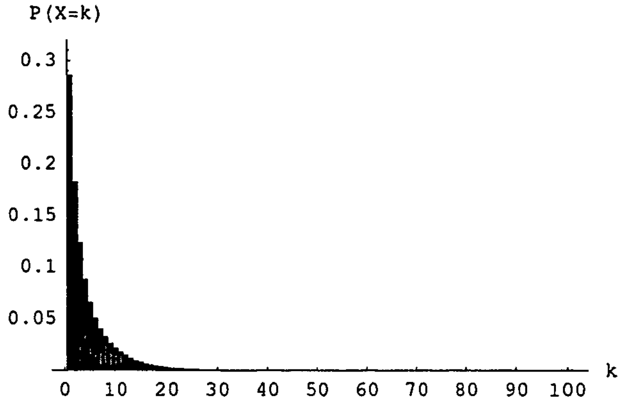

Figure 12.24: Situation 2, Poisson arrival process with state-dependent intensity (1).

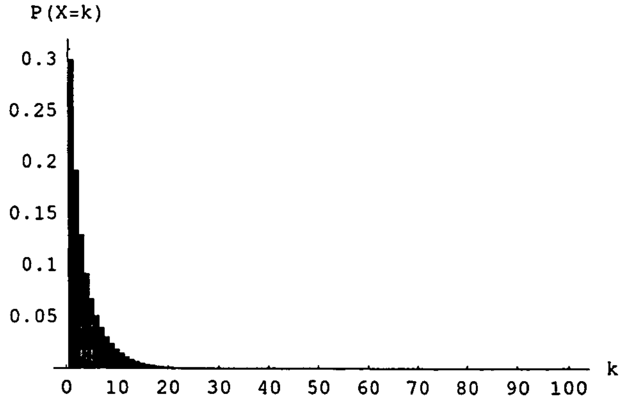

Figure 12.25: Situation 2, Poisson arrival process with state-dependent intensity (2).

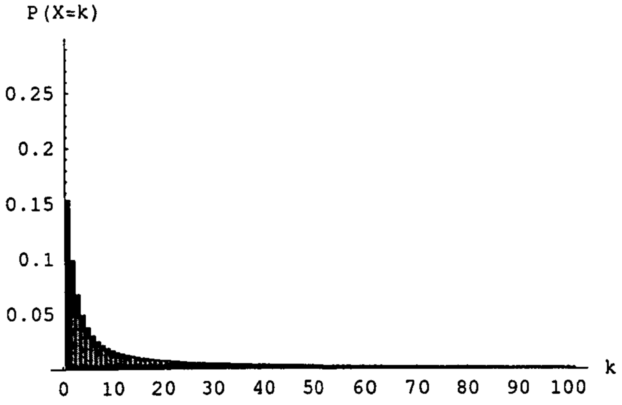

Figure 12.26: Situation 2, Poisson arrival process, “open network”.

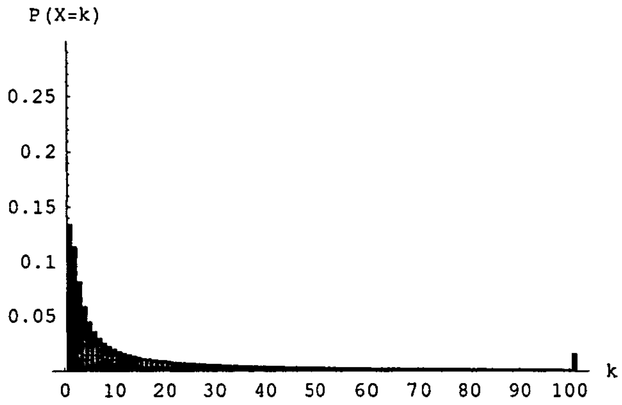

Figure 12.27: Situation 3, Poisson arrival process, with constant intensity.

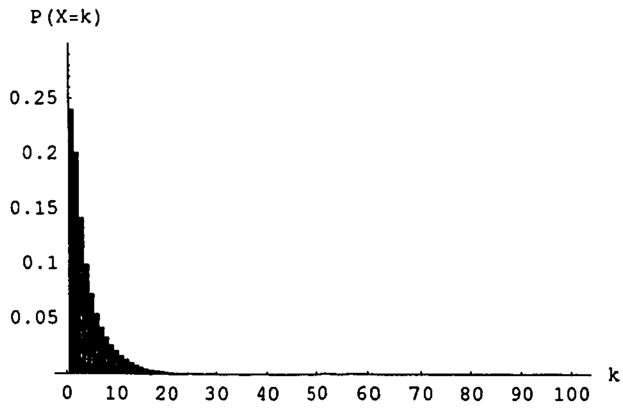

Figure 12.28: Situation 3, Poisson arrival process with state-dependent intensity (1).

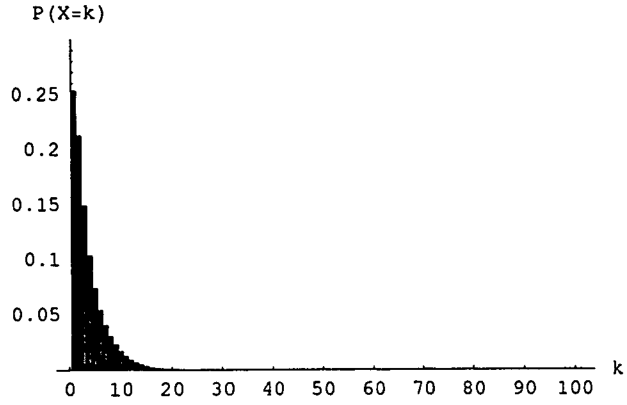

Figure 12.29: Situation 3, Poisson arrival process with state-dependent intensity (2).

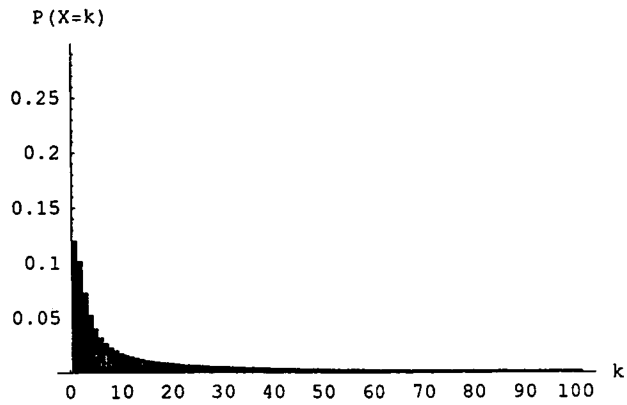

Figure 12.30: Situation 3, Poisson arrival process, "open network".

We will briefly discuss the numerical results that are presented in the figures. The three situations will be treated separately. Subsequently, the three different situations are compared.

When we consider situation 1, the binomial process and the process with varying arrival rate show a similar behaviour where the probability mass of the distribution is approximately centered around 20 and 30 respectively. The other two arrival processes show a completely different distribution. There is no "clock-form" distribution and all g_k's with $k < N$ are approximately zero. This phenomenon can partly be described by the fact that the "effective arrival rate" of individual customers is not the same for all four situations. One can think of the effective arrival rate as the average number of customers that arrive per time unit. Clearly, this effective arrival rate is related to the stationary distribution. When in all four situations the same value of μ is taken, the effective arrival rate in the binomial and varying arrival processes will be significantly lower than in the other two situations. This explains the striking difference in the graphs. Adjusting the values of μ for each situation in order to obtain approximately equal effective arrival rates would result in a more fair comparison between the four situations. However, this is difficult since the effective arrival rate is related to the stationary distribution. With trial and error, one could obtain approximately equal rates. On the other hand, one of the purposes of the comparison between different arrival processes is to study the influence on the effective arrival rate. This influence is of course not measurable when all effective arrival rates are chosen to be equal.

Situations 2 and 3 are nearly the same. The only difference is the service time distribution. When we inspect the graphs that are involved, we see that in both situations the four graphs relate to each other in almost the same way. In the binomial and varying process, the probability mass is slightly more centered towards zero and the tail of the distribution seems less heavy than in the other two processes. This phenomenon can again be described by the difference in effective arrival rates.

When we compare the graphs of situation 2 and 3, we observe that there is almost no difference. The only thing that is structurally different is that in all graphs, g_0 is slightly larger in situation 2 than it is in situation 3. So the resulting stationary distributions when choosing an exponential or Erlang service time distribution (with equal means) are nearly the same.

Summarizing, we can say that in heavy-traffic systems, the three different arrival processes do not lead to significantly different stationary distributions. However, in light-traffic systems, the differences are obviously present.

Chapter 13

Analysis of the non-blocked tree mechanism

This chapter concentrates on the service time analysis in case of a non-blocked tree mechanism. At the end, a numerical scheme for determining the mean tree length as well as the mean service time of an individual customer is presented. The first section, Section 13.1, will be devoted to the derivation of a functional equation, which can be used to obtain the generating function of the tree length. In Section 13.2, the same is done, but now for the generating function of the service time. Some complications arise. In Section 13.3 the focus will be on deriving two functional equations, which are closely related to the mean tree length and the mean service time, respectively. These relations will be given in Sections 13.6 and 13.7, respectively. In Section 13.4, it will be indicated how the two functional equations can be used to obtain function values numerically. In Section 13.5, it will be indicated how the capacity of this system can be determined numerically. Furthermore, numerical results are obtained and compared with simulation results. This is the subject of Section 13.8. In the last section, Section 13.9, the non-blocked tree mechanism with periodic arrival rate is studied by means of simulation.

13.1 The generating function of the tree length

This section is divided into three subsections. The first subsection is dedicated to the derivation of a 2-dimensional functional equation, relating the tree length distribution and the number of initial customers in a tree. The next subsection relates the generating function of the tree length distribution of an arbitrary tree (assuming a Poisson arrival process) to the functional equation. The final subsection shows how the functional equation can be used to determine mean and variance of the tree length distribution.

13.1.1 Derivation of a 2-dimensional functional equation

Define $\hat{B}(n)$ as the random variable that represents the total length of a tree (in slots), when the initial number of customers equals n, $n \geq 2$. One can obtain a recursion for these random variables. It will be explained first how this recursion can be interpreted.

The root of a tree with initially n customers in it, splits with a certain probability in three subtrees. When a subtree consists of at least 2 customers, it will be processed in a next contention slot in the future. In this slot, a Poisson number of new customers joins the "original" customers. Such a subtree can again be seen as the root of a new tree with a certain (random) number of customers in it. When a subtree consists of less than 2 customers, it is clearly finished. This together explains the following recursion.

For all $n \geq 2$ the following recursion holds:

$$
\begin{aligned}
P(\hat{B}(n) = k) \;=\; & \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 \geq 2 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) e^{-3\mu} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \sum_{k_3=0}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\mu^{k_2}}{k_2!} \frac{\mu^{k_3}}{k_3!} \cdot \\
& P(\hat{B}(n_1 + k_1) + \hat{B}(n_2 + k_2) + \hat{B}(n_3 + k_3) + 1 = k) + \\
& 3 \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) e^{-2\mu} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\mu^{k_2}}{k_2!} \cdot \\
& P(\hat{B}(n_1 + k_1) + \hat{B}(n_2 + k_2) + 1 = k) + \\
& 3 \sum_{\substack{n_1 \geq 2, n_2 < 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) e^{-\mu} \sum_{k_1=0}^{\infty} \frac{\mu^{k_1}}{k_1!} P(\hat{B}(n_1 + k_1) + 1 = k) + \\
& \sum_{\substack{n_1 < 2, n_2 < 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) I_{k=1},
\end{aligned}
$$

$k = 1, 2, \ldots,$

with:

$$\xi(n_1, n_2, n_3) = \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n.$$

The following generating function is introduced:

$$\hat{B}_n(z) = \sum_{k=0}^{\infty} P(\hat{B}(n) = k)z^k, \quad |z| \leq 1.$$

Multiplying both sides of the recursive equation with z^k, plugging in the expression for $\xi(n_1, n_2, n_3)$, summing the equations over $k = 1, 2, \ldots$ and finally multiplying both sides with $\frac{3^n}{n!}$ yields:

$$\begin{aligned}
3^n\frac{\hat{B}_n(z)}{n!} &= z \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3\geq 2\\ n_1+n_2+n_3=n}} \frac{e^{-3\mu}}{n_1!n_2!n_3!} \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty}\sum_{k=1}^{\infty} \frac{\mu^{k_1}}{k_1!}\frac{\mu^{k_2}}{k_2!}\frac{\mu^{k_3}}{k_3!} \cdot \\
&\quad P(\hat{B}(n_1+k_1) + \hat{B}(n_2+k_2) + \hat{B}(n_3+k_3) = k-1)z^{k-1} + \\
&\quad 3z \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3< 2\\ n_1+n_2+n_3=n}} \frac{e^{-2\mu}}{n_1!n_2!n_3!} \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k=1}^{\infty} \frac{\mu^{k_1}}{k_1!}\frac{\mu^{k_2}}{k_2!} \cdot \\
&\quad P(\hat{B}(n_1+k_1) + \hat{B}(n_2+k_2) = k-1)z^{k-1} + \\
&\quad 3z \sum_{\substack{n_1\geq 2, n_2< 2, n_3< 2\\ n_1+n_2+n_3=n}} \frac{e^{-\mu}}{n_1!n_2!n_3!} \sum_{k_1=0}^{\infty} \frac{\mu^{k_1}}{k_1!} \sum_{k=1}^{\infty} P(\hat{B}(n_1+k_1) = k-1)z^{k-1} + \\
&\quad \sum_{\substack{n_1< 2, n_2< 2, n_3< 2\\ n_1+n_2+n_3=n}} \frac{1}{n_1!n_2!n_3!} \sum_{k=1}^{\infty} z^k I_{k=1} \\
&= z \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3\geq 2\\ n_1+n_2+n_3=n}} \frac{e^{-3\mu}}{n_1!n_2!n_3!} \sum_{k_1=0}^{\infty}\sum_{k_2=0}^{\infty}\sum_{k_3=0}^{\infty} \frac{\mu^{k_1}}{k_1!}\frac{\mu^{k_2}}{k_2!}\frac{\mu^{k_3}}{k_3!}(n_1+k_1)!(n_2+k_2)!(n_3+k_3)! \cdot \\
&\quad \frac{\hat{B}_{n_1+k_1}(z)}{(n_1+k_1)!}\frac{\hat{B}_{n_2+k_2}(z)}{(n_2+k_2)!}\frac{\hat{B}_{n_3+k_3}(z)}{(n_3+k_3)!} +
\end{aligned}$$

$$3z \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \frac{e^{-2\mu}}{n_1! n_2! n_3!} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\mu^{k_2}}{k_2!} (n_1+k_1)!(n_2+k_2)! \cdot$$

$$\frac{\hat{B}_{n_1+k_1}(z)}{(n_1+k_1)!} \frac{\hat{B}_{n_2+k_2}(z)}{(n_2+k_2)!} +$$

$$3z \sum_{\substack{n_1 \geq 2, n_2 < 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \frac{e^{-\mu}}{n_1! n_2! n_3!} \sum_{k_1=0}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\hat{B}_{n_1+k_1}(z)}{(n_1+k_1)!} (n_1+k_1)! +$$

$$z \sum_{\substack{n_1 < 2, n_2 < 2, n_3 < 2 \\ n_1+n_2+n_3=n}} \frac{1}{n_1! n_2! n_3!}.$$

Introduce $\hat{f}_n(z) = \dfrac{\hat{B}_n(z)}{n!}$ for $n \geq 0$.

Furthermore, define $m_i = n_i + k_i$, $i = 1, 2, 3$.

Finally define the 2-dimensional generating function $\hat{F}(x, z)$ to be:

$$\hat{F}(x, z) = \sum_{n=2}^{\infty} \hat{f}_n(z) x^n.$$

Using these definitions, we obtain the following when we sum the equations, each multiplied with x^n, over $n = 2, 3, \ldots$:

$$\begin{aligned}
\sum_{n=2}^{\infty} \hat{f}_n(z)(3x)^n &= ze^{-3\mu} \sum_{n_1=2}^{\infty} \sum_{k_1=0}^{\infty} \binom{n_1+k_1}{n_1} \mu^{k_1} x^{n_1} \hat{f}_{n_1+k_1}(z) \cdot \\
&\sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{f}_{n_2+k_2}(z) \sum_{n_3=2}^{\infty} \sum_{k_3=0}^{\infty} \binom{n_3+k_3}{n_3} \mu^{k_3} x^{n_3} \hat{f}_{n_3+k_3}(z) + \\
&3ze^{-2\mu} \sum_{n_1=2}^{\infty} \sum_{k_1=0}^{\infty} \binom{n_1+k_1}{n_1} \mu^{k_1} x^{n_1} \hat{f}_{n_1+k_1}(z) \cdot \\
&\sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{f}_{n_2+k_2}(z) \sum_{n_3=0}^{1} \frac{1}{n_3!} x^{n_3} +
\end{aligned}$$

$$3ze^{-\mu}\sum_{n_1=2}^{\infty}\sum_{k_1=0}^{\infty}\binom{n_1+k_1}{n_1}\mu^{k_1}x^{n_1}\hat{f}_{n_1+k_1}(z)\sum_{n_2=0}^{1}\frac{1}{n_2!}x^{n_2}\sum_{n_3=0}^{1}\frac{1}{n_3!}x^{n_3}+$$

$$z\sum_{n_1=0}^{1}\frac{1}{n_1!}x^{n_1}\sum_{n_2=0}^{1}\frac{1}{n_2!}x^{n_2}\sum_{n_3=0}^{1}\frac{1}{n_3!}x^{n_3}-z(1+3x)$$

$$= ze^{-3\mu}\sum_{m_1=2}^{\infty}\sum_{n_1=2}^{\infty}\binom{m_1}{n_1}\mu^{m_1-n_1}x^{n_1}\hat{f}_{m_1}(z)\cdot$$

$$\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{f}_{m_2}(z)\sum_{m_3=2}^{\infty}\sum_{n_3=0}^{\infty}\binom{m_3}{n_3}\mu^{m_3-k_3}x^{n_3}\hat{f}_{m_3}(z)+$$

$$3ze^{-2\mu}\sum_{m_1=2}^{\infty}\sum_{n_1=2}^{\infty}\binom{m_1}{n_1}\mu^{m_1-n_1}x^{n_1}\hat{f}_{m_1}(z)\cdot$$

$$\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{f}_{m_2}(z)(1+x)+$$

$$3ze^{-\mu}\sum_{m_1=2}^{\infty}\sum_{n_1=2}^{\infty}\binom{m_1}{n_1}\mu^{m_1-n_1}x^{n_1}\hat{f}_{m_1}(z)(1+x)^2+z(1+x)^3-z(1+3x)$$

$$= ze^{-3\mu}\sum_{m_1=2}^{\infty}\hat{f}_{m_1}(z)\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)\cdot$$

$$\sum_{m_2=2}^{\infty}\hat{f}_{m_2}(z)\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)\cdot$$

$$\sum_{m_3=2}^{\infty}\hat{f}_{m_3}(z)\left((x+\mu)^{m_3}-m_3x\mu^{m_3-1}-\mu^{m_3}\right)+$$

$$3ze^{-2\mu}\sum_{m_1=2}^{\infty}\hat{f}_{m_1}(z)\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)\cdot$$

$$\sum_{m_2=2}^{\infty}\hat{f}_{m_2}(z)\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)(1+x)+$$

$$3ze^{-\mu}\sum_{m_1=2}^{\infty}\hat{f}_{m_1}(z)\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)(1+x)^2+$$

$$z(1+x)^3-z(1+3x).$$

Using the definition of $\hat{F}(x,z)$, we obtain:

$$
\begin{aligned}
\hat{F}(3x,z) &= ze^{-3\mu}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)^3 + \\
&\quad 3ze^{-2\mu}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)^2(1+x) + \\
&\quad 3ze^{-\mu}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)(1+x)^2 + z\,(1+x)^3 - z\,(1+3x) \\
&= z\left(e^{-\mu}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)+(1+x)\right)^3 - z\,(1+3x)\,.
\end{aligned}
$$

13.1.2 Relation between $\hat{F}(x,z)$ and the tree length GF

Define $\hat{B}$ as the random variable that represents the total length of an arbitrary tree (in slots). For the generating function of the tree length distribution, denoted by $\hat{B}(z)$, the following holds:

$$
\begin{aligned}
\hat{B}(z) &= \sum_{k=0}^{\infty} P(\hat{B}=k)z^k \\
&= \sum_{k=0}^{\infty}\sum_{n=0}^{\infty} e^{-\mu}\frac{\mu^n}{n!}P(\hat{B}(n)=k)z^k \\
&= \sum_{k=0}^{\infty}\sum_{n=2}^{\infty} e^{-\mu}\frac{\mu^n}{n!}P(\hat{B}(n)=k)z^k + \sum_{k=0}^{\infty}\sum_{n=0}^{1} e^{-\mu}\frac{\mu^n}{n!}P(\hat{B}(n)=k)z^k \\
&= e^{-\mu}\sum_{n=2}^{\infty}\hat{f}_n(z)\mu^n + e^{-\mu}\,(1+\mu z) \\
&= e^{-\mu}\left(\hat{F}(\mu,z)+1+\mu z\right).
\end{aligned}
$$

This shows that the generating function of the tree length is closely related to the function $\hat{F}(x,z)$ and so the functional equation for $\hat{F}(x,z)$ gives a method to determine numerically the generating function $\hat{B}(z)$ and its derivatives in $z = 1$. How this method works, will be outlined in the next (sub)sections.

13.1.3 The mean and variance of the tree length

In this subsection, it will be outlined how $\hat{B}'(1)$ and $\hat{B}''(1)$ can be found numerically. These quantities can be used to determine eventually the mean and variance of $\hat{B}$, in which we are interested.

In the previous subsection, we related $\hat{B}(z)$ to $\hat{F}(\mu, z)$. Differentiating this equation once with respect to z and subsequently substituting $z = 1$ yields the following:

$$\hat{B}'(1) = e^{-\mu}\left(\frac{\partial \hat{F}}{\partial z}(\mu, 1) + \mu\right).$$

Now we return to the more general problem of evaluating $\hat{R}(x) := \frac{\partial \hat{F}}{\partial z}(x, 1)$ for arbitrary x. This problem will be tackled by using the functional equation that was found at the end of subsection 13.1.1. For convenience, we give the relevant equation again:

$$\hat{F}(3x, z) = z\left(e^{-\mu}\left(\hat{F}(x+\mu, z) - \hat{F}(\mu, z) - x\frac{\partial \hat{F}}{\partial x}(\mu, z)\right) + (1+x)\right)^3 - z\,(1+3x).$$

If we take the derivative with respect to z and substitute $z = 1$, we obtain the following:

$$\begin{aligned}\frac{\partial \hat{F}}{\partial z}(3x, 1) &= \left(e^{-\mu}\left(\hat{F}(x+\mu, 1) - \hat{F}(\mu, 1) - x\frac{\partial \hat{F}}{\partial x}(\mu, 1)\right) + (1+x)\right)^3 + \\ &\quad 3\left(e^{-\mu}\left(\hat{F}(x+\mu, 1) - \hat{F}(\mu, 1) - x\frac{\partial \hat{F}}{\partial x}(\mu, 1)\right) + (1+x)\right)^2 \cdot \\ &\quad e^{-\mu}\left(\frac{\partial \hat{F}}{\partial z}(x+\mu, 1) - \frac{\partial \hat{F}}{\partial z}(\mu, 1) - x\frac{\partial^2 \hat{F}}{\partial x \partial z}(\mu, 1)\right) \\ &\quad - (1+3x).\end{aligned}$$

Using the definition of $\hat{F}(x,z)$, it follows directly that

$$\begin{aligned}\hat{F}(x,1) &= e^x - 1 - x,\\ \frac{\partial \hat{F}}{\partial x}(x,1) &= e^x - 1.\end{aligned}$$

Using this together with the definition of $\hat{R}(x)$ yields the following equation:

$$\begin{aligned}\hat{R}(3x) &= \left(e^{-\mu}\left(e^{x+\mu} - 1 - x - \mu - e^{\mu} + 1 + \mu - x\left(e^{\mu} - 1\right)\right) + (1+x)\right)^3 + \\ &\quad 3e^{-\mu}\left(\hat{R}(x+\mu) - \hat{R}(\mu) - x\frac{d\hat{R}}{dx}(\mu)\right)\cdot \\ &\quad \left(e^{-\mu}\left(e^{x+\mu} - 1 - x - \mu - e^{\mu} + 1 + \mu - x\left(e^{\mu} - 1\right)\right) + (1+x)\right)^2 \\ &\quad - (1+3x) \\ &= e^{3x} - (1+3x) + 3e^{2x}e^{-\mu}\left(\hat{R}(x+\mu) - \hat{R}(\mu) - x\frac{d\hat{R}}{dx}(\mu)\right).\end{aligned}$$

In this equation, two unknown functions appear. If we take the derivative with respect to x on both sides of this functional equation, a second equation is obtained. Define $\hat{Q}(x) := \hat{R}'(x)$. This leads to the following equation:

$$\begin{aligned}3\hat{Q}(3x) &= 3e^{3x} - 3 + 6e^{2x}e^{-\mu}\left(\hat{R}(x+\mu) - \hat{R}(\mu) - x\hat{Q}(\mu)\right) + \\ &\quad 3e^{2x}e^{-\mu}\left(\hat{Q}(x+\mu) - \hat{Q}(\mu)\right), \\ \hat{Q}(3x) &= e^{3x} - 1 + e^{2x}e^{-\mu}\left(2\hat{R}(x+\mu) - 2\hat{R}(\mu) - 2x\hat{Q}(\mu) + \hat{Q}(x+\mu) - \hat{Q}(\mu)\right).\end{aligned}$$

Eventually we have two functional equations with two yet unknown functions. Recall that we were originally interested in determining $\hat{R}(\mu)$. In Section 13.4, it will be outlined how these two functional equations can be used to determine $\hat{R}(\mu)$.

We will now proceed with $\hat{B}''(1)$, completely analogously as in the case of $\hat{B}'(1)$.

We differentiate the equation for $\hat{B}(z)$ twice with respect to z and subsequently substitute $z = 1$. This yields the following:

$$\hat{B}''(1) = e^{-\mu}\frac{\partial^2 \hat{F}}{\partial z^2}(\mu, 1).$$

We will again investigate the more general problem of evaluating $\hat{K}(x) := \frac{\partial^2 \hat{F}}{\partial z^2}(x, 1)$ for arbitrary x. Furthermore, define $\hat{L}(x) := \hat{K}'(x)$. Differentiating the functional equation 13.1 twice with respect to z, substituting $z = 1$ and using the definitions of $\hat{K}(x)$ and $\hat{L}(x)$ yields:

$$\begin{aligned}
\hat{K}(3x) &= 2\cdot 3\left(e^{-\mu}\left(\hat{F}(x+\mu,1)-\hat{F}(\mu,1)-x\frac{\partial \hat{F}}{\partial x}(\mu,1)\right)+(1+x)\right)^2\cdot\\
&\quad e^{-\mu}\left(\frac{\partial \hat{F}}{\partial z}(x+\mu,1)-\frac{\partial \hat{F}}{\partial z}(\mu,1)-x\frac{\partial^2 \hat{F}}{\partial x\partial z}(\mu,1)\right)+\\
&\quad 6\left(e^{-\mu}\left(\hat{F}(x+\mu,1)-\hat{F}(\mu,1)-x\frac{\partial \hat{F}}{\partial x}(\mu,1)\right)+(1+x)\right)\cdot\\
&\quad e^{-2\mu}\left(\frac{\partial \hat{F}}{\partial z}(x+\mu,1)-\frac{\partial \hat{F}}{\partial z}(\mu,1)-x\frac{\partial^2 \hat{F}}{\partial x\partial z}(\mu,1)\right)^2+\\
&\quad 3\left(e^{-\mu}\left(\hat{F}(x+\mu,1)-\hat{F}(\mu,1)-x\frac{\partial \hat{F}}{\partial x}(\mu,1)\right)+(1+x)\right)^2\cdot\\
&\quad e^{-\mu}\left(\frac{\partial^2 \hat{F}}{\partial z^2}(x+\mu,1)-\frac{\partial^2 \hat{F}}{\partial z^2}(\mu,1)-x\frac{\partial^3 \hat{F}}{\partial x\partial z^2}(\mu,1)\right)\\
&= 6e^{2x}e^{-\mu}\left(\hat{R}(x+\mu)-\hat{R}(\mu)-xQ(\mu)\right)+\\
&\quad 6e^{x}e^{-2\mu}\left(\hat{R}(x+\mu)-\hat{R}(\mu)-xQ(\mu)\right)^2+
\end{aligned}$$

$$3e^{2x}e^{-\mu}\left(\hat{K}(x+\mu)-\hat{K}(\mu)-x\hat{L}(\mu)\right).$$

If we take the derivative with respect to x on both sides of the functional equation, we obtain a second equation:

$$\begin{aligned}\hat{L}(x) &= 12e^{2x}e^{-\mu}\left(\hat{R}(x+\mu)-\hat{R}(\mu)-xQ(\mu)\right)+6e^{2x}e^{-\mu}\left(\hat{Q}(x+\mu)-Q(\mu)\right)+\\ &\quad 6e^{x}e^{-2\mu}\left(\hat{R}(x+\mu)-\hat{R}(\mu)-xQ(\mu)\right)^2+\\ &\quad 12e^{x}e^{-2\mu}\left(\hat{R}(x+\mu)-\hat{R}(\mu)-xQ(\mu)\right)\left(\hat{Q}(x+\mu)-Q(\mu)\right)+\\ &\quad 6e^{2x}e^{-\mu}\left(\hat{K}(x+\mu)-\hat{K}(\mu)-x\hat{L}(\mu)\right)+3e^{2x}e^{-\mu}\left(\hat{L}(x+\mu)-\hat{L}(\mu)\right).\end{aligned}$$

When we regard $\hat{R}(x)$ and $\hat{Q}(x)$ as known functions, we have again obtained two functional equations in two unknown functions. We are interested in evaluating $\hat{K}(\mu)$. How this can be done, will be outlined later.

13.2 The generating function of the individual service time

This section is divided into several subsections. The first subsection is devoted to the derivation of a 2-dimensional functional equation, relating the service time distribution and the number of initial customers in a tree. The second subsection relates the generating function of the service time distribution of a customer in an arbitrary tree (assuming a Poisson arrival process) to this functional equation. Analogously as in the previous section, the third subsection shows how mean and variance of the service time distribution can be numerically approximated.

The analysis assumes a depth-first tree protocol.

13.2.1 Derivation of a 2-dimensional functional equation

Define $\hat{D}(n)$ as the random variable that represents the individual service time of a customer, that is going to compete in a slot which contains in total

n customers, $n \geq 2$. A recursion will be derived for $\hat{D}(n)$.

Assume that a certain customer is competing together with $n-1$ other customers. After the first branching step, the customer can be in either of the three branches. When a customer is in a branch, it can be there alone, or together with other customers. This distinction will be made when deriving the recursion, because of the fact that when a customer is alone, it does not have to wait for other branches to be completed. Furthermore, when a customer is in the second or third branch, it is of interest whether the previous branch or branches contained more than one customer or not. This is because of the fact that new customers only may arrive when a branch consists of more than one customer. Otherwise, the branch is terminated directly. A last remark is that sometimes symmetry arguments can be exploited, avoiding the use of unnecessarily many sum notations. Putting all these considerations together, we obtain the following recursion:

$$
\begin{aligned}
P(\hat{D}(n)=k) \quad = \quad & \sum_{\substack{n_1\geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_1}{n}\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}P(\hat{D}(n_1+k_1)=k-1)+ \\
& 3\sum_{\substack{n_1=1, n_2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_1}{n}P(\hat{D}(n_1)=k-1)+ \\
& \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_2}{n}\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\cdot \\
& \sum_{k_2=0}^{\infty} e^{-\mu}\frac{\mu^{k_2}}{k_2!}P(\hat{D}(n_2+k_2)+\hat{B}(n_1+k_1)=k-1)+ \\
& \sum_{\substack{n_1<2, n_2\geq 2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_2}{n}\sum_{k_2=0}^{\infty} e^{-\mu}\frac{\mu^{k_2}}{k_2!}P(\hat{D}(n_2+k_2)=k-1)+ \\
& \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3\geq 2 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_3}{n}\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\sum_{k_2=0}^{\infty} e^{-\mu}\frac{\mu^{k_2}}{k_2!}\cdot \\
& \sum_{k_3=0}^{\infty} e^{-\mu}\frac{\mu^{k_3}}{k_3!}P(\hat{D}(n_3+k_3)+\hat{B}(n_1+k_1)+\hat{B}(n_2+k_2)=k-1)+ \\
& 2\sum_{\substack{n_1<2, n_2\geq 2, n_3\geq 2 \\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_3}{n}\sum_{k_2=0}^{\infty} e^{-\mu}\frac{\mu^{k_2}}{k_2!}\cdot
\end{aligned}
$$

$$\sum_{k_3=0}^{\infty} e^{-\mu}\frac{\mu^{k_3}}{k_3!}P(\hat{D}(n_3+k_3)+\hat{B}(n_2+k_2)=k-1)+$$

$$\sum_{\substack{n_1<2,n_2<2,n_3\geq 2\\ n_1+n_2+n_3=n}} \xi(n_1,n_2,n_3)\frac{n_3}{n}\sum_{k_3=0}^{\infty} e^{-\mu}\frac{\mu^{k_3}}{k_3!}P(\hat{D}(n_3+k_3)=k-1),$$

$$k=1,2,\ldots,$$

with:

$$\begin{aligned}
\hat{D}(0) &= \hat{D}(1)=0,\\
P(\hat{D}(n)=0) &= 0, \qquad n=2,3,\ldots,\\
\xi(n_1,n_2,n_3) &= \frac{n!}{n_1!n_2!n_3!}\left(\frac{1}{3}\right)^n.
\end{aligned}$$

The following generating function is introduced:

$$\hat{D}_n(z) = \sum_{k=0}^{\infty} P(\hat{D}(n)=k)z^k, \quad |z|\leq 1.$$

The same procedure is followed as in the previous section. We multiply both sides of the recursive equation with z^k, plug in the expression for $\xi(n_1,n_2,n_3)$, sum the equations over $k=1,2,\ldots$ and finally multiply both sides of the equation with $\frac{3^n}{n!}$. This yields:

$$\begin{aligned}
3^n\frac{\hat{D}_n(z)}{n!} &= z\sum_{\substack{n_1\geq 2,n_2,n_3\\ n_1+n_2+n_3=n}} \frac{n_1}{n}\frac{e^{-\mu}}{n_1!n_2!n_3!}\sum_{k_1=0}^{\infty}\sum_{k=1}^{\infty}\frac{\mu^{k_1}}{k_1!}\cdot\\
&\quad P(\hat{D}(n_1+k_1)=k-1)z^{k-1}+\\
&\quad 3z\sum_{\substack{n_1=1,n_2,n_3\\ n_1+n_2+n_3=n}} \frac{n_1}{n}\frac{1}{n_1!n_2!n_3!}\sum_{k=1}^{\infty}\cdot\\
&\quad P(\hat{D}(n_1)=k-1)z^{k-1}+
\end{aligned}$$

$$z \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 \\ n_1+n_2+n_3=n}} \frac{n_2}{n} \frac{e^{-2\mu}}{n_1! n_2! n_3!} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \sum_{k=1}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\mu^{k_2}}{k_2!} \cdot$$

$$P(\hat{D}(n_2+k_2) + \hat{B}(n_1+k_1) = k-1) z^{k-1} +$$

$$z \sum_{\substack{n_1 < 2, n_2 \geq 2, n_3 \\ n_1+n_2+n_3=n}} \frac{n_2}{n} \frac{e^{-\mu}}{n_1! n_2! n_3!} \sum_{k_2=0}^{\infty} \sum_{k=1}^{\infty} \frac{\mu^{k_2}}{k_2!} \cdot$$

$$P(\hat{D}(n_2+k_2) = k-1) z^{k-1} +$$

$$z \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 \geq \\ n_1+n_2+n_3=n}} \frac{n_3}{n} \frac{e^{-3\mu}}{n_1! n_2! n_3!} \sum_{k_1=0}^{\infty} \sum_{k_2=0}^{\infty} \sum_{k_3=0}^{\infty} \sum_{k=1}^{\infty} \frac{\mu^{k_1}}{k_1!} \frac{\mu^{k_2}}{k_2!} \frac{\mu^{k_3}}{k_3!} \cdot$$

$$P(\hat{D}(n_3+k_3) + \hat{B}(n_1+k_1) + \hat{B}(n_2+k_2) = k-1) z^{k-1} +$$

$$2z \sum_{\substack{n_1 < 2, n_2 \geq 2, n_3 \geq 2 \\ n_1+n_2+n_3=n}} \frac{n_3}{n} \frac{e^{-2\mu}}{n_1! n_2! n_3!} \sum_{k_2=0}^{\infty} \sum_{k_3=0}^{\infty} \sum_{k=1}^{\infty} \frac{\mu^{k_2}}{k_2!} \frac{\mu^{k_3}}{k_3!} \cdot$$

$$P(\hat{D}(n_3+k_3) + \hat{B}(n_2+k_2) = k-1) z^{k-1} +$$

$$z \sum_{\substack{n_1 < 2, n_2 < 2, n_3 \geq 2 \\ n_1+n_2+n_3=n}} \frac{n_3}{n} \frac{e^{-\mu}}{n_1! n_2! n_3!} \sum_{k_3=0}^{\infty} \sum_{k=1}^{\infty} \frac{\mu^{k_3}}{k_3!} \cdot$$

$$P(\hat{D}(n_3+k_3) = k-1) z^{k-1},$$

$n = 2, 3, \ldots \qquad k = 1, 2, \ldots$

Introduce $\hat{h}_n(z) = \dfrac{n\hat{D}_n(z)}{n!}$ for $n \geq 0$.

Furthermore, define $m_i = n_i + k_i$, $i = 1, 2, 3$.

Finally define the 2-dimensional generating function $\hat{H}(x, z)$ to be:

$$\hat{H}(x, z) \;=\; \sum_{n=2}^{\infty} \hat{h}_n(z) x^n.$$

Now each equation is multiplied with nx^n. Using the previous definitions, we obtain the following when we sum the equations over $n = 2, 3, \ldots$:

$$
\begin{aligned}
\sum_{n=2}^{\infty} \hat{h}_n(z)\,(3x)^n \;=\; & ze^{-\mu} \sum_{n_1=2}^{\infty} \sum_{k_1=0}^{\infty} \binom{n_1+k_1}{n_1} \mu^{k_1} x^{n_1} \hat{h}_{n_1+k_1}(z) \cdot \\
& \sum_{n_2=0}^{\infty} \frac{x^{n_2}}{n_2!} \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} + 3zx \left(\sum_{n_2=0}^{\infty} \frac{x^{n_2}}{n_2!} \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} - 1 \right) + \\
& ze^{-2\mu} \sum_{n_1=2}^{\infty} \sum_{k_1=0}^{\infty} \binom{n_1+k_1}{n_1} \mu^{k_1} x^{n_1} \hat{f}_{n_1+k_1}(z) \cdot \\
& \sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{h}_{n_2+k_2}(z) \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} + \\
& ze^{-\mu} \sum_{n_1=0}^{1} \frac{x^{n_1}}{n_1!} \sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{h}_{n_2+k_2}(z) \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} + \\
& ze^{-3\mu} \sum_{n_1=2}^{\infty} \sum_{k_1=0}^{\infty} \binom{n_1+k_1}{n_1} \mu^{k_1} x^{n_1} \hat{f}_{n_1+k_1}(z) \cdot \\
& \sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{f}_{n_2+k_2}(z) \sum_{n_3=2}^{\infty} \sum_{k_3=0}^{\infty} \binom{n_3+k_3}{n_3} \mu^{k_3} x^{n_3} \hat{h}_{n_3+k_3}(z) + \\
& 2ze^{-2\mu} \sum_{n_1=0}^{1} \frac{x^{n_1}}{n_1!} \sum_{n_2=2}^{\infty} \sum_{k_2=0}^{\infty} \binom{n_2+k_2}{n_2} \mu^{k_2} x^{n_2} \hat{f}_{n_2+k_2}(z) \cdot \\
& \sum_{n_3=2}^{\infty} \sum_{k_3=0}^{\infty} \binom{n_3+k_3}{n_3} \mu^{k_3} x^{n_3} \hat{h}_{n_3+k_3}(z) + \\
& ze^{-\mu} \sum_{n_1=0}^{1} \frac{x^{n_1}}{n_1!} \sum_{n_2=0}^{1} \frac{x^{n_2}}{n_2!} \sum_{n_3=2}^{\infty} \sum_{k_3=0}^{\infty} \binom{n_3+k_3}{n_3} \mu^{k_3} x^{n_3} \hat{h}_{n_3+k_3}(z) \\
=\; & ze^{-\mu} \sum_{m_1=2}^{\infty} \sum_{n_1=2}^{\infty} \binom{m_1}{n_1} \mu^{m_1-n_1} x^{n_1} \hat{h}_{m_1}(z) \sum_{n_2=0}^{\infty} \frac{x^{n_2}}{n_2!} \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} + \\
& 3zx \left(\sum_{n_2=0}^{\infty} \frac{x^{n_2}}{n_2!} \sum_{n_3=0}^{\infty} \frac{x^{n_3}}{n_3!} - 1 \right) +
\end{aligned}
$$

$$ze^{-2\mu}\sum_{m_1=2}^{\infty}\sum_{n_1=2}^{\infty}\binom{m_1}{n_1}\mu^{m_1-n_1}x^{n_1}\hat{f}_{m_1}(z)\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{h}_{m_2}(z)\cdot$$

$$\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}+$$

$$ze^{-\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{h}_{m_2}(z)\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}+$$

$$ze^{-3\mu}\sum_{m_1=2}^{\infty}\sum_{n_1=2}^{\infty}\binom{m_1}{n_1}\mu^{m_1-n_1}x^{n_1}\hat{f}_{m_1}(z)\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{f}_{m_2}(z)\cdot$$

$$\sum_{m_3=2}^{\infty}\sum_{n_3=2}^{\infty}\binom{m_3}{n_3}\mu^{m_3-n_3}x^{n_3}\hat{h}_{m_3}(z)+$$

$$2ze^{-2\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{m_2=2}^{\infty}\sum_{n_2=2}^{\infty}\binom{m_2}{n_2}\mu^{m_2-n_2}x^{n_2}\hat{f}_{m_2}(z)\cdot$$

$$\sum_{m_3=2}^{\infty}\sum_{n_3=2}^{\infty}\binom{m_3}{n_3}\mu^{m_3-n_3}x^{n_3}\hat{h}_{m_3}(z)+$$

$$ze^{-\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{n_2=0}^{1}\frac{x^{n_2}}{n_2!}\cdot$$

$$\sum_{m_3=2}^{\infty}\sum_{n_3=2}^{\infty}\binom{m_3}{n_3}\mu^{m_3-n_3}x^{n_3}\hat{h}_{m_3}(z)$$

$$= ze^{-\mu}\sum_{m_1=2}^{\infty}\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)\hat{h}_{m_1}(z)\sum_{n_2=0}^{\infty}\frac{x^{n_2}}{n_2!}\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}+$$

$$3zx\left(\sum_{n_2=0}^{\infty}\frac{x^{n_2}}{n_2!}\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}-1\right)+$$

$$ze^{-2\mu}\sum_{m_1=2}^{\infty}\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)\hat{f}_{m_1}(z)\cdot$$

$$\sum_{m_2=2}^{\infty}\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)\hat{h}_{m_2}(z)\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}+$$

$$ze^{-\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{m_2=2}^{\infty}\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)\hat{h}_{m_2}(z)\sum_{n_3=0}^{\infty}\frac{x^{n_3}}{n_3!}+$$

$$ze^{-3\mu}\sum_{m_1=2}^{\infty}\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)\hat{f}_{m_1}(z)\cdot$$

$$\sum_{m_2=2}^{\infty}\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)\hat{f}_{m_2}(z)\cdot$$

$$\sum_{m_3=2}^{\infty}\left((x+\mu)^{m_3}-m_3x\mu^{m_3-1}-\mu^{m_3}\right)\hat{h}_{m_3}(z)+$$

$$2ze^{-2\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{m_2=2}^{\infty}\left((x+\mu)^{m_2}-m_2x\mu^{m_2-1}-\mu^{m_2}\right)\hat{f}_{m_2}(z)\cdot$$

$$\sum_{m_3=2}^{\infty}\left((x+\mu)^{m_3}-m_3x\mu^{m_3-1}-\mu^{m_3}\right)\hat{h}_{m_3}(z)+$$

$$ze^{-\mu}\sum_{n_1=0}^{1}\frac{x^{n_1}}{n_1!}\sum_{n_2=0}^{1}\frac{x^{n_2}}{n_2!}\sum_{m_3=2}^{\infty}\left((x+\mu)^{m_3}-m_3x\mu^{m_3-1}-\mu^{m_3}\right)\hat{h}_{m_3}(z).$$

Using the definition of $\hat{H}(x,z)$, we obtain the following:

$$\begin{aligned}
\hat{H}(3x,z) &= ze^{-\mu}e^{2x}\left(\hat{H}(x+\mu,z)-\hat{H}(\mu,z)-x\frac{\partial\hat{H}}{\partial x}(\mu,z)\right)+3zx\left(e^{2x}-1\right)+\\
&\quad ze^{-2\mu}e^{x}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)\cdot\\
&\quad \left(\hat{H}(x+\mu,z)-\hat{H}(\mu,z)-x\frac{\partial\hat{H}}{\partial x}(\mu,z)\right)+\\
&\quad z(1+x)e^{-\mu}e^{x}\left(\hat{H}(x+\mu,z)-\hat{H}(\mu,z)-x\frac{\partial\hat{H}}{\partial x}(\mu,z)\right)+\\
&\quad ze^{-3\mu}\left(\hat{F}(x+\mu,z)-\hat{F}(\mu,z)-x\frac{\partial\hat{F}}{\partial x}(\mu,z)\right)^2.
\end{aligned}$$

$$\left(\hat{H}(x+\mu,z) - \hat{H}(\mu,z) - x\frac{\partial \hat{H}}{\partial x}(\mu,z)\right) +$$

$$2z(1+x)e^{-2\mu}\left(\hat{F}(x+\mu,z) - \hat{F}(\mu,z) - x\frac{\partial \hat{F}}{\partial x}(\mu,z)\right) \cdot$$

$$\left(\hat{H}(x+\mu,z) - \hat{H}(\mu,z) - x\frac{\partial \hat{H}}{\partial x}(\mu,z)\right) +$$

$$z\,(1+x)^2\,e^{-\mu}\left(\hat{H}(x+\mu,z) - \hat{H}(\mu,z) - x\frac{\partial \hat{H}}{\partial x}(\mu,z)\right).$$

This equation can be slightly simplified. However, this is not done here.

13.2.2 Relation between $\hat{H}(x,z)$ and the service time GF

Define $\hat{D}$ as the random variable that represents the individual service time of an arbitrary customer. Unfortunately, the (stationary) distribution of the number of (competing) customers in an arbitrary slot is difficult to obtain.

In an arbitrary slot, new arrivals occur according to a Poisson process with constant rate μ. However, some of the customers that collided in a previous slot are also allowed to compete in this slot. For the moment, it seems impossible for us to obtain the exact distribution of the number of customers in an arbitrary slot. Therefore, we are going to use an approximation.

Think of the following. Every customer that is single in a certain mini-slot immediately leaves the tree. Now think of the situation that such a *lucky* customer does not leave the tree, but competes again in a certain slot in the future, just as it would have done if it *was* involved in a collision. In this mechanism, the (stationary) distribution of the number of customers in an arbitrary slot is Poisson with parameter $\frac{3}{2}\mu$. To see that this distribution is Poisson, one can reason as follows. At the beginning of a tree, the number of customers is Poisson distributed. With probabilities $\frac{1}{3}$, $\frac{1}{3}$ and $\frac{1}{3}$ this Poisson number distributes itself over the three following slots. In these three slots, a Poisson number of fresh arrivals is added. This implies that the number of customers in these three slots is still Poisson, only with a different parameter. Continuing in this way, we can argue that the stationary distribution of the number of customers in an arbitrary slot is Poisson. The rate of $\frac{3}{2}\mu$

can easily be determined by using a balance argument. Unfortunately, this situation does not resemble the true situation, in which lucky customers leave immediately. However, the assumption that the number of customers in an arbitrary slot is Poisson does not seem too bad. Therefore, we adopt it for the moment. It is obvious that the corresponding arrival rate must be in between μ and $\frac{3}{2}\mu$. In fact, under the assumption of a Poisson distribution, the exact arrival rate can be found by using a balance argument. This will be explained now.

For completeness, the individual rate of new customers (per slot) equals μ. The total arrival rate of customers in an arbitrary slot will be denoted by $(1+\gamma)\,\mu$. Consider two consecutive slots. The total arrival rate in the second slot can be decomposed in an arrival rate of fresh customers, equal to μ, and an arrival rate of customers that collided in the first slot. This second arrival rate can easily be determined. The expected number of customers $\tilde{n}$, that collide in the first mini-slot of the first slot is given by

$$\begin{aligned}\tilde{n} &= \frac{1}{3}(1+\gamma)\,\mu - e^{-\frac{1}{3}(1+\gamma)\mu}\frac{1}{3}(1+\gamma)\,\mu \\ &= \frac{1}{3}(1+\gamma)\,\mu\left(1-e^{-\frac{1}{3}(1+\gamma)\mu}\right).\end{aligned}$$

Obviously, $\mu+\tilde{n}$ must equal $(1+\gamma)\,\mu$. This gives us an equation, which can be used to solve for γ (as a function of μ). This has been done with Mathematica. The resulting graph is plotted in figure 13.1.
As we can see in the figure, the graph is approximately linear in μ. A good approximation turns out to be $\gamma(\mu)=0.115\mu$.

So, if we approximate the arrival process in an arbitrary slot by a Poisson process with the correct arrival rate, the following holds for the generating function of the service time distribution of an arbitrary customer, denoted by $\hat{D}(z)$:

$$\begin{aligned}\hat{D}(z) &= \sum_{k=0}^{\infty} P(\hat{D}=k)z^k \\ &\approx \sum_{k=0}^{\infty}\sum_{n=0}^{\infty} e^{-(1+\gamma)\mu}\frac{((1+\gamma)\,\mu)^n}{n!}P(\hat{D}(n)=k)z^k \\ &= \sum_{k=0}^{\infty}\sum_{n=2}^{\infty} e^{-(1+\gamma)\mu}\frac{((1+\gamma)\,\mu)^n}{n!}P(\hat{D}(n)=k)z^k +\end{aligned}$$

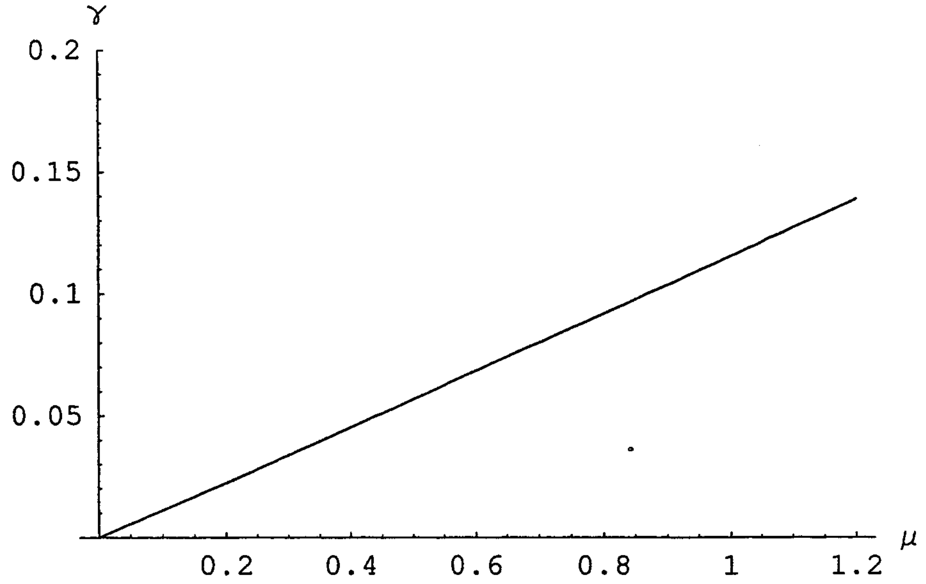

Figure 13.1: γ as a function of μ.

$$
\begin{aligned}
& \sum_{k=0}^{\infty}\sum_{n=0}^{1} e^{-(1+\gamma)\mu}\frac{((1+\gamma)\mu)^n}{n!}P(\hat{D}(n)=k)z^k \\
= \;& e^{-(1+\gamma)\mu}\sum_{n=2}^{\infty}\hat{h}_n(z)\left((1+\gamma)\mu\right)^n + e^{-(1+\gamma)\mu}\left(1+(1+\gamma)\mu z\right) \\
= \;& e^{-(1+\gamma)\mu}\left(\hat{H}((1+\gamma)\mu, z) + 1 + (1+\gamma)\mu z\right).
\end{aligned}
$$

We observe that under the Poisson approximation, the generating function of the individual service time is directly related to the function $\hat{H}(x,z)$. So the functional equation for $\hat{H}(x,z)$ gives an opportunity to approximate the mean and variance of $\hat{D}$, in which we are interested. The mean and variance can be evaluated by using the functional equations that were obtained in the previous section. This method is very similar as in the case of the tree length distribution. However, there seems to be one essential difference.

This difference is that in this case, we do not start with one functional equation with $\hat{H}(x,z)$ as the only unknown function, but with two functional equations with $\hat{F}(x,z)$ and $\hat{H}(x,z)$ as unknowns. But in fact, we can regard $\hat{F}(x,z)$ as a known function, because it can be approximated and evaluated by using the first functional equation, given at the end of 13.1.1. Considering it this way, there is no initial difference.

The method for evaluating the mean and variance of $\hat{D}$ consists again of the evaluation of the first and second derivative of the generating function of $\hat{D}$ in $z = 1$. The next subsection will briefly discuss this method. However, it will not be worked out in detail.
Due to the fact that we are able to find exact results on the first moment of the individual service time by means of the "aggregated service time" approach that will be outlined in Section 13.3.2, it is possible to make a comparison between the exact mean service time and the approximation that is found when using the approximated generating function. This will give an indication on how accurate the approximation is and if it makes sense to use this approach to find higher moments of the individual service time. Of course, simulation is another method of verifying the approximations for higher moments of the service time. The conjecture is that the approximation is accurate.

13.2.3 The mean and variance of the service time

Analagously as in the case of the tree length, the functional equation for $\hat{H}(x, z)$ can be used to obtain approximations for $\hat{D}'(1)$ and $\hat{D}''(1)$. For completeness, we summarize the procedure that must be followed in order to obtain an approximation for $\hat{D}'(1)$.

We treat $\hat{F}(x, z)$ as a known function. First, take in the functional equation given in 13.1 the derivative with respect to z and substitute $z = 1$. Simplify this equation by using that $\hat{H}(x, 1)$ is known. Subsequently, take in the obtained 1-dimensional functional equation the derivative with respect to x to obtain a second functional equation. We are interested in evaluating $\frac{\partial \hat{H}}{\partial z}(\mu, 1)$. At this point, we can directly use the procedure that will be described in Section 13.4.

To obtain an approximation for $\hat{D}''(1)$ is completely analogous as in the tree length case.

13.3 Derivation of two functional equations

In this section two functional equations will be derived. These two equations will be related to the mean tree length and the mean service time, respectively. This will be done in the next two sections.

13.3.1 Mean tree length functional equation

Recall that $\hat{B}(n)$ was defined as the random variable that represents the total length of a tree (measured in slots), when the number of customers in the first slot equals $n, n \geq 2$. For $\hat{B}(n)$, a recursive formula can be derived. The reasoning behind this recursive formula is exactly the same as presented in subsection 13.1.1. Therefore, we refer to this subsection for an explanation.

The expectation of $\hat{B}(n)$, denoted by $\mathbb{E}\hat{B}(n)$, satisfies the following recursion:

$$\mathbb{E}\hat{B}(n) = 3 \sum_{\substack{n_1 \geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} \mathbb{E}\hat{B}(n_1 + k_1) \right) + 1,$$

$$n \geq 2,$$

with:

$$\mathbb{E}\hat{B}(0) = \mathbb{E}\hat{B}(1) = 0,$$

$$\xi(n_1, n_2, n_3) = \frac{n!}{n_1! n_2! n_3!} \left(\frac{1}{3}\right)^n.$$

The recursion can now be written in a slightly different way:

$$3^n \frac{\mathbb{E}\hat{B}(n)}{n!} = 3 \sum_{\substack{n_1 \geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \frac{1}{n_1! n_2! n_3!} \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} \mathbb{E}\hat{B}(n_1 + k_1) \right) + 3^n \frac{1}{n!},$$

$$n \geq 2.$$

Multiplying both sides of the recursive equation with x^n and summing over $n = 2, 3, \ldots$ yields:

$$\sum_{n=2}^{\infty} \frac{\mathbb{E}\hat{B}(n)}{n!} (3x)^n = 3 \sum_{n=2}^{\infty} \sum_{\substack{n_1 \geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!} \frac{x^{n_2}}{n_2!} \frac{x^{n_3}}{n_3!} \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} \mathbb{E}\hat{B}(n_1 + k_1) \right) +$$

$$\sum_{n=2}^{\infty}(3x)^n\frac{1}{n!}.$$

Define the generating function $G_\mu(x) := \sum_{n=2}^{\infty}\frac{\mathbb{E}\hat{B}(n)}{n!}x^n$. Clearly, the generating function depends on μ, hence a subscript is added. This leads to the following equation:

$$\begin{aligned} G_\mu(3x) &= 3\sum_{n_1=2}^{\infty}\sum_{n_2=0}^{\infty}\sum_{n_3=0}^{\infty}\frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!}\left(\sum_{k_1=0}^{\infty}e^{-\mu}\frac{\mu^{k_1}}{k_1!}\mathbb{E}\hat{B}(n_1+k_1)\right)+\sum_{n=2}^{\infty}(3x)^n\frac{1}{n!} \\ &= 3e^{2x}\sum_{n_1=2}^{\infty}\left(\sum_{k_1=0}^{\infty}e^{-\mu}\frac{\mu^{k_1}}{k_1!}\frac{x^{n_1}}{n_1!}(n_1+k_1)!\frac{\mathbb{E}\hat{B}(n_1+k_1)}{(n_1+k_1)!}\right)+e^{3x}-3x-1. \end{aligned}$$

Introducing the summation variable $m_1 = n_1 + k_1$ gives the following:

$$\begin{aligned} G_\mu(3x) &= 3e^{2x}\sum_{m_1=2}^{\infty}\left(\sum_{n_1=2}^{\infty}e^{-\mu}\mu^{m_1-n_1}x^{n_1}\binom{m_1}{n_1}\frac{\mathbb{E}\hat{B}(m_1)}{m_1!}\right)+e^{3x}-3x-1 \\ &= 3e^{2x}e^{-\mu}\sum_{m_1=2}^{\infty}\frac{\mathbb{E}\hat{B}(m_1)}{m_1!}\left(\sum_{n_1=2}^{\infty}\mu^{m_1-n_1}x^{n_1}\binom{m_1}{n_1}\right)+e^{3x}-3x-1 \\ &= 3e^{2x}e^{-\mu}\sum_{m_1=2}^{\infty}\frac{\mathbb{E}\hat{B}(m_1)}{m_1!}\left((x+\mu)^{m_1}-m_1x\mu^{m_1-1}-\mu^{m_1}\right)+e^{3x}-3x-1 \\ &= 3e^{2x}e^{-\mu}\left(G_\mu(x+\mu)-G_\mu(\mu)-xG'_\mu(\mu)\right)+e^{3x}-3x-1. \end{aligned}$$

How this functional equation can be solved (numerically) will be discussed later in Section 13.4.

Observe that this functional equation is identical to the functional equation for $\hat{R}(x)$, given in subsection 13.1.3. Moreover, the generating function $G_\mu(x)$ turns out to be identical to $\hat{R}(x)$.

13.3.2 Mean service time functional equation

Define $\hat{S}(n)$ as the random variable that represents the aggregate service time of all customers in a (non-blocked) tree (measured in slots), when the number of customers in the first slot equals $n, n \geq 2$. The expectation of $\hat{S}(n)$, denoted by $\mathbb{E}\hat{S}(n)$, satisfies the following recursion. For some insight in this recursion, we refer to the beginning of subsection 13.2.1.

$$\begin{aligned}
\mathbb{E}\hat{S}(n) &= 3 \sum_{\substack{n_1 \geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^k}{k!} \mathbb{E}\hat{S}(n_1 + k_1) \right) + \\
& \quad 3 \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 \\ n_1+n_2+n_3=n}} \xi(n_1, n_2, n_3) \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} n_2 \mathbb{E}\hat{B}(n_1 + k_1) \right) + n \\
& \quad n \geq 2,
\end{aligned}$$

with:

$$\mathbb{E}\hat{S}(0) = \mathbb{E}\hat{S}(1) = 0.$$

The recursion can now be written in a sligthly different way:

$$\begin{aligned}
3^n \frac{\mathbb{E}\hat{S}(n)}{n!} &= 3 \sum_{\substack{n_1 \geq 2, n_2, n_3 \\ n_1+n_2+n_3=n}} \frac{1}{n_1! n_2! n_3!} \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} \mathbb{E}\hat{S}(n_1 + k_1) \right) + \\
& \quad 3 \sum_{\substack{n_1 \geq 2, n_2 \geq 2, n_3 \\ n_1+n_2+n_3=n}} \frac{1}{n_1! n_2! n_3!} \left(\sum_{k_1=0}^{\infty} e^{-\mu} \frac{\mu^{k_1}}{k_1!} n_2 \mathbb{E}\hat{B}(n_1 + k_1) \right) + 3^n \frac{n}{n!}, \\
& \quad n \geq 2.
\end{aligned}$$

Multiplying both sides of the recursive equation with x^n and summing over $n = 2, 3, \ldots$ yields:

$$
\begin{aligned}
\sum_{n=2}^{\infty} \frac{\mathbb{E}\hat{S}(n)}{n!}(3x)^n &= 3\sum_{n=2}^{\infty} \sum_{\substack{n_1\geq 2, n_2, n_3\\ n_1+n_2+n_3=n}} \frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\mathbb{E}\hat{S}(n_1+k_1)\right)+ \\
& 3\sum_{n=2}^{\infty} \sum_{\substack{n_1\geq 2, n_2\geq 2, n_3\\ n_1+n_2+n_3=n}} \frac{1}{n_1!n_2!n_3!}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}n_2\mathbb{E}\hat{B}(n_1+k_1)\right)+ \\
& \sum_{n=2}^{\infty}(3x)^n\frac{n}{n!}.
\end{aligned}
$$

Define the generating function $H_\mu(x) := \sum_{n=2}^{\infty}\frac{\mathbb{E}\hat{S}(n)}{n!}x^n$. This leads to the following equation:

$$
\begin{aligned}
H_\mu(3x) &= 3\sum_{n_1=2}^{\infty}\sum_{n_2=0}^{\infty}\sum_{n_3=0}^{\infty}\frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\mathbb{E}\hat{S}(n_1+k_1)\right)+ \\
& 3\sum_{n_1=2}^{\infty}\sum_{n_2=2}^{\infty}\sum_{n_3=0}^{\infty}\frac{x^{n_1}}{n_1!}\frac{x^{n_2}}{n_2!}\frac{x^{n_3}}{n_3!}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}n_2\mathbb{E}\hat{B}(n_1+k_1)\right)+ \\
& \sum_{n=2}^{\infty}(3x)^n\frac{n}{n!} \\
&= 3e^{2x}\sum_{n_1=2}^{\infty}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\frac{x^{n_1}}{n_1!}(n_1+k_1)!\frac{\mathbb{E}\hat{S}(n_1+k_1)}{(n_1+k_1)!}\right)+ \\
& 3x\left(e^x-1\right)e^x\sum_{n_1=2}^{\infty}\left(\sum_{k_1=0}^{\infty} e^{-\mu}\frac{\mu^{k_1}}{k_1!}\frac{x^{n_1}}{n_1!}(n_1+k_1)!\frac{\mathbb{E}\hat{B}(n_1+k_1)}{(n_1+k_1)!}\right)+3x\left(e^{3x}-1\right).
\end{aligned}
$$

Introducing the summation variable $m_1 = n_1 + k_1$ finally yields (analogously as in the previous case) the following functional equation:

$$
\begin{aligned}
H_\mu(3x) &= 3e^{2x}e^{-\mu}\left(H_\mu(x+\mu)-H_\mu(\mu)-xH'_\mu(\mu)\right)+ \\
& 3xe^x\left(e^x-1\right)e^{-\mu}\left(G_\mu(x+\mu)-G_\mu(\mu)-xG'_\mu(\mu)\right)+3x\left(e^{3x}-1\right).
\end{aligned}
$$

13.4 How to evaluate $G_\mu(\mu)$ and $H_\mu(\mu)$?

In this section it will be indicated how the two functional equations can be used to approximate $G_\mu(x_0)$ and $H_\mu(x_0)$. As will turn out later, it is important to find accurate approximations of $G_\mu(\mu)$ and $H_\mu(\mu)$ if one is interested in the mean individual service time. In this section, first the equation containing only $G_\mu(\cdot)$ will be discussed. After that, the equation containing both $G_\mu(\cdot)$ and $H_\mu(\cdot)$ will be discussed.

13.4.1 Evaluation of $G_\mu(\mu)$

The equation that is being considered is the following:

$$G_\mu(3x) = 3e^{2x}e^{-\mu}\left(G_\mu(x+\mu) - G_\mu(\mu) - xG'_\mu(\mu)\right) + e^{3x} - 3x - 1.$$

A second equation can be obtained by differentiating the equation above:

$$3G'_\mu(3x) = 3e^{2x}e^{-\mu}\left(2G_\mu(x+\mu) - 2G_\mu(\mu) - 2xG'_\mu(\mu) + G'_\mu(x+\mu) - G'_\mu(\mu)\right) + 3e^{3x} - 3,$$

$$G'_\mu(3x) = e^{2x}e^{-\mu}\left(2G_\mu(x+\mu) - 2G_\mu(\mu) - 2xG'_\mu(\mu) + G'_\mu(x+\mu) - G'_\mu(\mu)\right) + e^{3x} - 1.$$

Now, a set of (linear) equations will be constructed. Substitute $x_1 = \frac{1}{3}\mu$ in both equations. This gives 4 yet unknown quantities: $G_\mu(\mu)$, $G'_\mu(\mu)$, $G_\mu(\frac{4}{3}\mu)$, $G'_\mu(\frac{4}{3}\mu)$. Substitute $x_2 = \frac{4}{9}\mu$ in both equations. This gives rise to 2 extra unknowns: $G_\mu(\frac{13}{9}\mu)$, $G'_\mu(\frac{13}{9}\mu)$. Substitute $x_3 = \frac{13}{27}\mu$. Etc.

Proceeding in this way, we have that $x_n = \frac{1}{2}\frac{3^n-1}{3^n}\mu$. After n steps, the result is $2n$ equations and $2n+2$ unknowns. The following holds:

$$\lim_{n\to\infty} x_n = \frac{1}{2}\mu.$$

So, we have the following:

$$G'_\mu(3x_n) \approx G'_\mu(\frac{3}{2}\mu),$$
$$G_\mu(\frac{3}{2}\mu) \approx G_\mu(3x_n) + \left(\frac{3}{2}\mu - 3x_n\right) G'_\mu(3x_n).$$

For large n, these approximations are nearly exact and can therefore be used as 2 extra equations. Furthermore, substitution of $x_0 = \frac{1}{2}\mu$ gives another 2 equations. Finally, we have $2n+4$ equations and also $2n+4$ unknowns. Solving this set gives an approximation for $G_\mu(\mu)$.

An interesting question is whether this approximation method converges as $n \to \infty$. Is it stable? What is the convergence ratio? Is the numerical procedure stable? For the moment only a plot is given below, in which the approximation for $G_1(1)$, denoted by $\tilde{G}_1(1)_n$, is given as a function of n. This plot indicates a fast convergence but also numerical problems for large n, probably due to underflow errors.

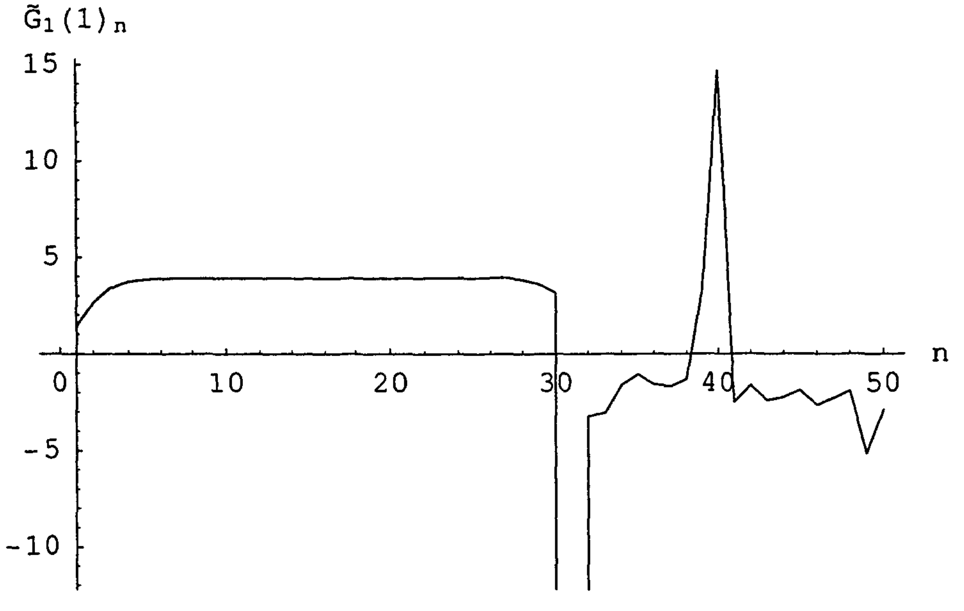

Figure 13.2: $\tilde{G}_1(1)_n$ as a function of n

13.4.2 Evaluation of $H_\mu(\mu)$

In this subsection, we return to the second functional equation. Recall that this second functional equation was the following:

$$\begin{aligned} H_\mu(3x) &= 3e^{2x}e^{-\mu}\left(H_\mu(x+\mu) - H_\mu(\mu) - xH'_\mu(\mu)\right) + \\ &\quad 3xe^{x}\left(e^{x}-1\right)e^{-\mu}\left(G_\mu(x+\mu) - G_\mu(\mu) - xG'_\mu(\mu)\right) + 3x\left(e^{3x}-1\right). \end{aligned}$$

We are interested in approximating $H_\mu(\mu)$. Differentiation of the equation above with respect to x gives another equation. So finally we have four equations: the two original functional equations for $G_\mu(3x)$ and $H_\mu(3x)$ and the two "derivatives" of these two equations. If we proceed in exactly the same way as in the previous case, but now using these *four* equations instead of only the first two, we finally obtain a set of $4n+8$ equations with the same number of unknowns. Obviously, this set contains the previous set of $2n+4$ equations. Solving this set gives an approximation for $H_\mu(\mu)$.

For this approximation, also a graph is presented (figure 13.3), in which the approximation for $H_1(1)$, denoted by $\tilde{H}_1(1)_n$, is given as a function of n. This plot indicates again a fast convergence but also numerical problems for large n, probably due to underflow errors.

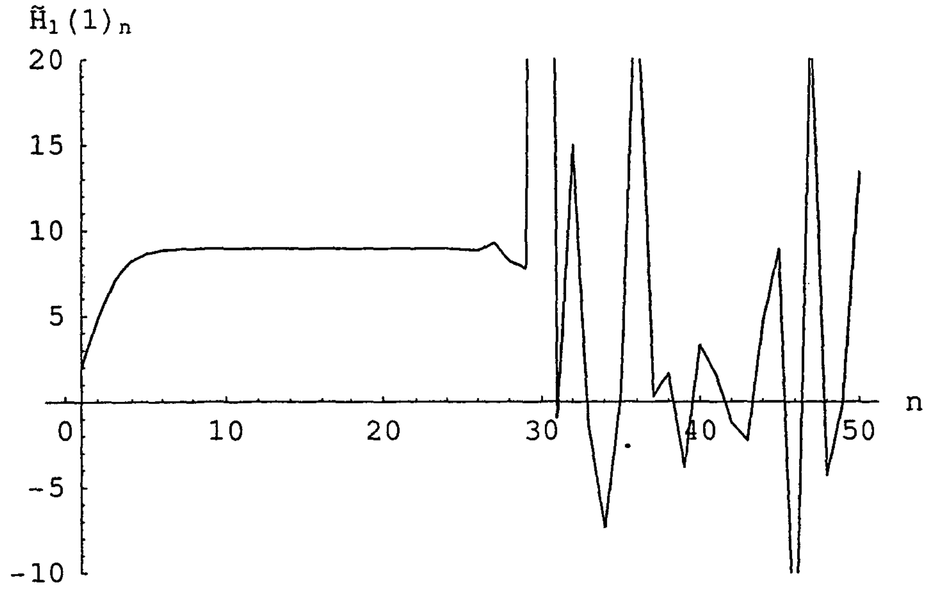

Figure 13.3: $\tilde{H}_1(1)_n$ as a function of n

13.5 Determination of the capacity

In the previous section, a method for approximating $G_\mu(\mu)$ was given. As has been indicated already, $G_\mu(\cdot)$ depends on μ. For values of μ which are above the capacity of the system, the generating function $G_\mu(\cdot)$ is not well-defined. Therefore, a numerical method to determine capacity of the non-blocked tree mechanism can be the following.

Make a plot of $G_\mu(\mu)$ as a function of μ. Observe where this function explodes, i.e. has a vertical asymptote. The value at which this asymptote occurs can be interpreted as the capacity of the mechanism.

Below, two figures are given. First a plot for $0 \leq \mu \leq 1.4$ is shown. After that, a more detailed plot is given. In figure 13.5 one can see that the capacity is found to be approximately 1.2048.

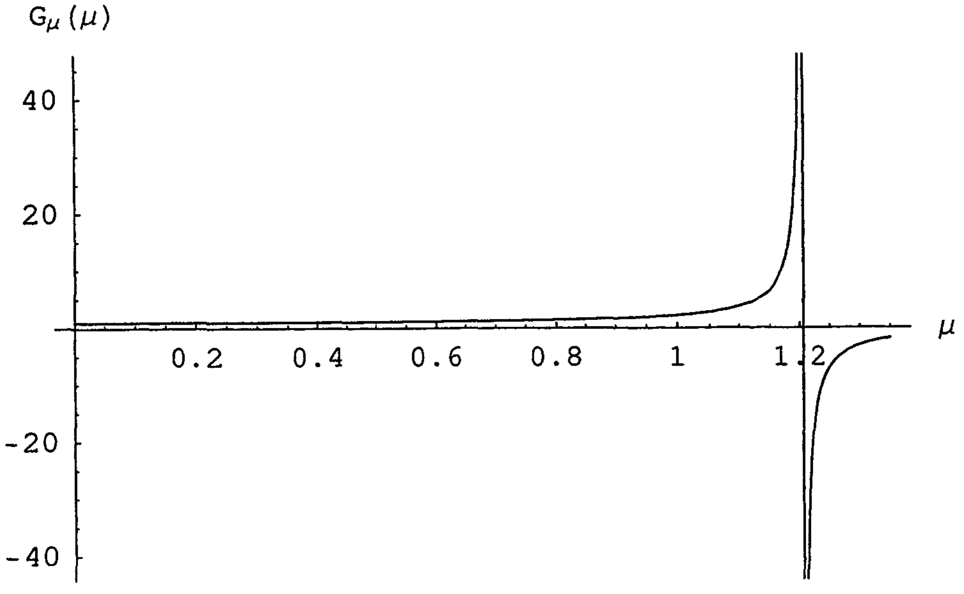

Figure 13.4: $G_\mu(\mu)$ as a function of μ

13.6 Determination of the mean tree length

In Section 13.4, a method for approximating $G_\mu(\mu)$ was given. In this section, $G_\mu(\mu)$ will be related directly to the mean tree length.

Consider again the case in which the arrival process of individual customers is a Poisson process with rate μ (per slot). Let $\hat{B}$ denote the random variable that represents the total tree length of an arbitrary batch of customers (including zero-customer batches) that arrives just after the completion of a tree. The expectation of $\hat{B}$, denoted by $\mathbb{E}\hat{B}$, satisfies the following:

$$\begin{aligned}
\mathbb{E}\hat{B} &= e^{-\mu}\cdot 1 + \mu e^{-\mu}\cdot 1 + \sum_{n=2}^{\infty} e^{-\mu}\frac{\mu^n}{n!}\mathbb{E}\hat{B}_n \\
&= e^{-\mu} + \mu e^{-\mu} + e^{-\mu}G_\mu(\mu) \\
&= e^{-\mu}\left(1+\mu+G_\mu(\mu)\right).
\end{aligned}$$

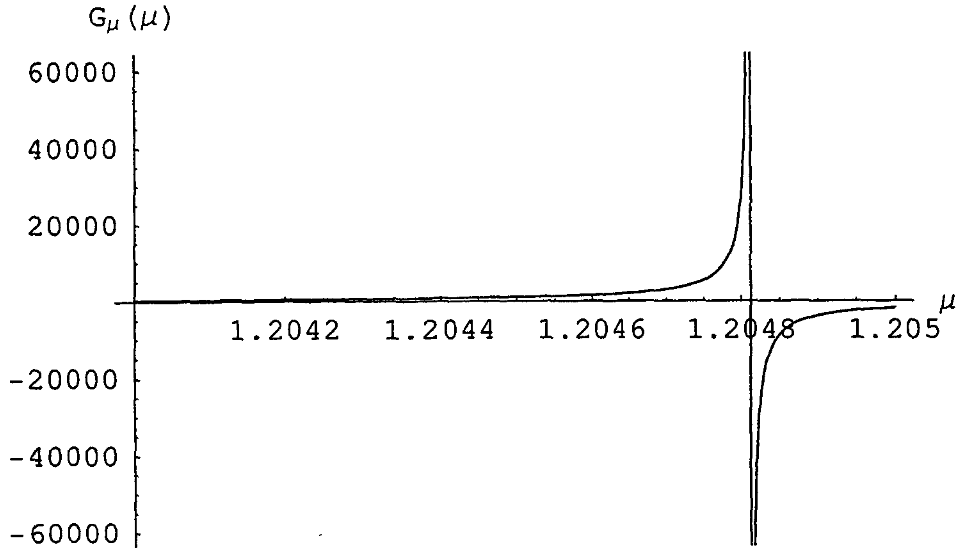

Figure 13.5: $G_\mu(\mu)$ as a function of μ

13.7 Determination of the mean individual service time

A method for approximating both $G_\mu(\mu)$ and $H_\mu(\mu)$ has already been outlined. In this section, this last quantity will be related directly to the mean individual service time.

Consider again the case in which the arrival process of individual customers is a Poisson process with rate μ (per slot). Let $\hat{S}$ denote the random variable that represents the total internal delay of an arbitrary batch of customers that arrives just after the completion of a tree. The expectation of $\hat{S}$, denoted by $\mathbb{E}\hat{S}$, is given by:

$$
\begin{aligned}
\mathbb{E}\hat{S} &= e^{-\mu} \cdot 0 + \mu e^{-\mu} \cdot 1 + \sum_{n=2}^{\infty} e^{-\mu} \frac{\mu^n}{n!} \mathbb{E}\hat{S}_n \\
&= \mu e^{-\mu} + e^{-\mu} H_\mu(\mu) \\
&= e^{-\mu} (\mu + H_\mu(\mu)).
\end{aligned}
$$

The mean delay of an arbitrary individual customer, denoted by $\mathbb{E}\hat{S}_{ind}$, can be obtained by dividing the mean total delay of an arbitrary batch by the mean number of customers that arrives in a batch. Obviously, this mean number of customers equals $\mu\mathbb{E}\hat{B}$. So the mean delay of an arbitrary individual customer satisfies the following relation:

$$\begin{aligned}
\mathbb{E}\hat{S}_{ind} &= \frac{\mathbb{E}\hat{S}}{\mu\mathbb{E}\hat{B}} \\
&= e^{-\mu}\frac{\mu + H_\mu(\mu)}{\mu e^{-\mu}(1+\mu+G_\mu(\mu))} \\
&= \frac{\mu + H_\mu(\mu)}{\mu(1+\mu+G_\mu(\mu))}.
\end{aligned}$$

13.8 Numerical results

In this section, some numerical results will be presented. These results have been obtained by implementing the previously described procedures to determine the mean tree length as well as the mean individual service time for the gated mechanism. For a range of values for μ, the analytically found results are compared with simulation results.

In figure 13.6, a plot of the mean tree length as a function of the arrival rate μ is given. Simulation results for several values of μ are presented in the same graph with stars. An important remark is that $\mathbb{E}\hat{B}$ is not exactly the mean tree length as calculated during the simulations, due to the fact that zero customers are not "included" during the simulation. The corrected mean tree length is plotted and this explains why the stars are present *on* the curve.

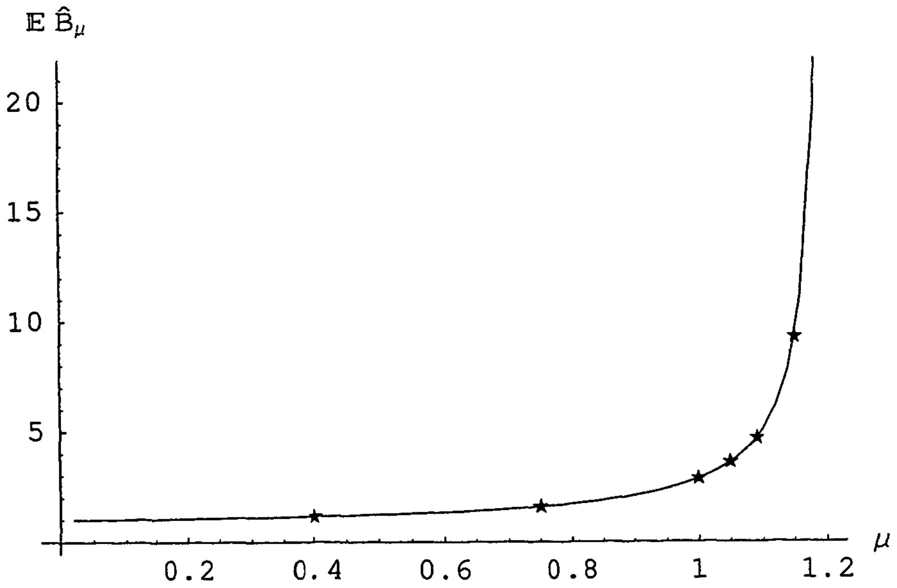

Figure 13.6: $\mathbb{E}\hat{B}$ as a function of μ

In figure 13.7, a plot of the mean individual service time as a function of the arrival rate μ is given. Simulation results for several values of μ are presented in the same graph with stars.

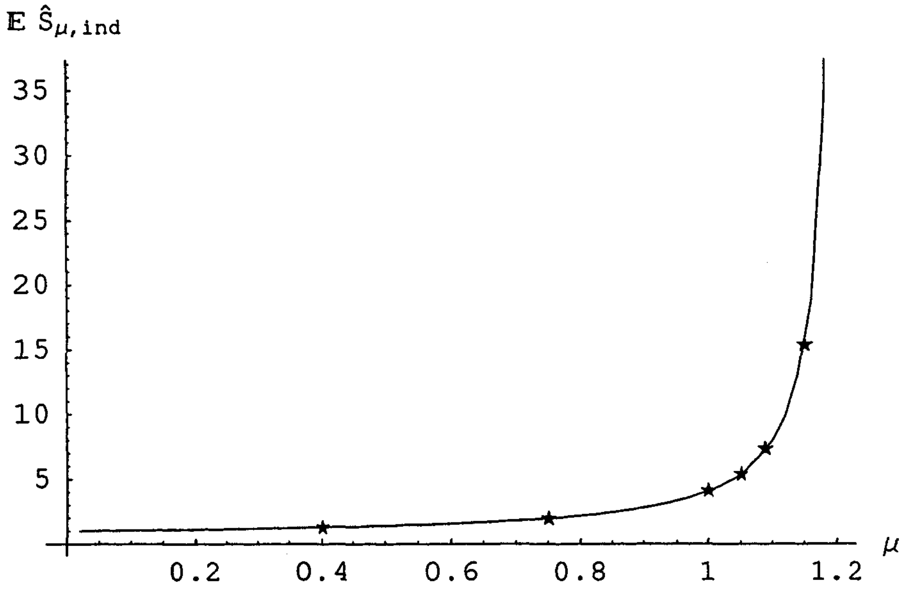

Figure 13.7: $\mathbb{E}\hat{S}_{ind}$ as a function of μ

13.9 The non-blocked tree mechanism with periodic arrival rates

In this section, a slightly more general model will be considered. The model in the first part of this chapter assumed Poisson arrivals in every slot with a known *constant* rate μ. In this section the non-blocked tree mechanism with Poisson arrivals with (known) periodic arrival rates is considered. Such a mechanism will be called a periodic non-blocked tree mechanism. This will be explained in more detail first.

Assume that in every slot the number of arrivals is Poisson distributed with a certain arrival rate. Furthermore, assume that the arrival rates are periodic with period p, in the sense that every p slots, exactly the same arrival rate occurs. Consequently, the arrival rates of consecutive slots (starting with slot 1) can be represented as a sequence $\mu_1, \ldots, \mu_p, \mu_1, \ldots$. Notice that the original arrival process is contained in this set of periodic arrival processes.

For a queuing system like the (periodic) non-blocked tree mechanism, important performance measures include the capacity and mean individual service time. Two very interesting questions are the following:

1. Do there exist periodic mechanisms which have a higher capacity than the original mechanism?

2. Do there exist periodic mechanisms which have a smaller mean individual service time than the original mechanism when the mean arrival rate is fixed at $\overline{\mu}$?

For the moment, only the second question will be investigated.

A simulation study is done to investigate the second question. p is chosen equal to 2. Two cases are investigated. In the first case $\mu_1 + \mu_2$ is fixed at 2, in the second case, $\mu_1 + \mu_2$ is fixed at 2.30. For a range of values of μ_1 the mean individual service time as well as the mean service time of a super customer are estimated and plotted together with their confidence intervals in the graphs 13.8 until 13.11. The confidence intervals are depicted with dotted and dashed lines.

As one can see in the graphs, the mean service times are roughly spoken decreasing as a function of μ_1, for $0 \leq \mu_1 \leq \frac{1}{2}(\mu_1 + \mu_2)$. In the interesting part of the graphs, i.e. in the regions with high values of μ_1 it is not exactly clear if the function is really decreasing and what value of μ_1 corresponds to the minimum mean service time. Therefore, for these values of μ_1, extensive simulations are done, in order to obtain very accurate estimates of the mean service time.

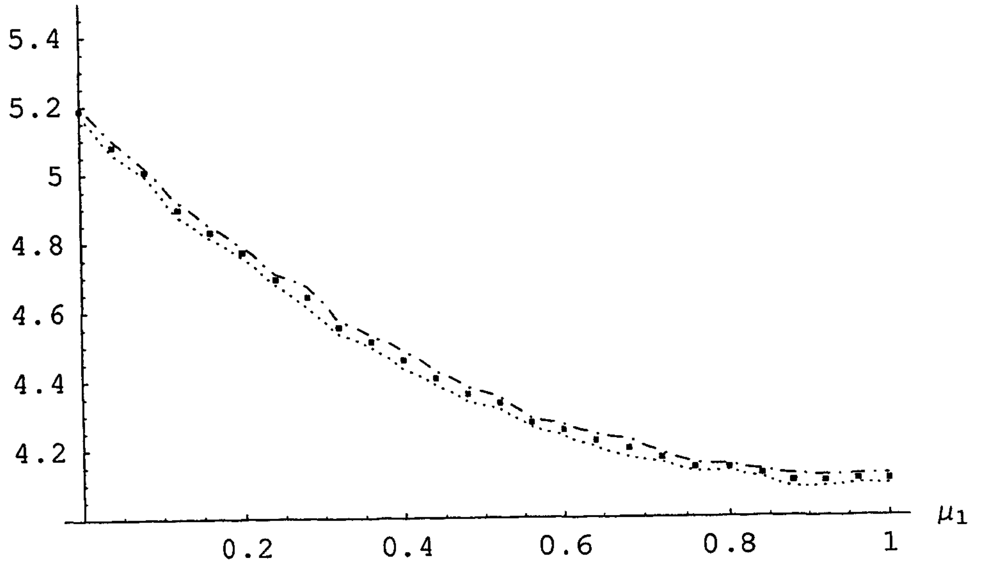

Figure 13.8: Estimated mean individual service time as a function of μ_1 with $\mu_1 + \mu_2 = 2$.

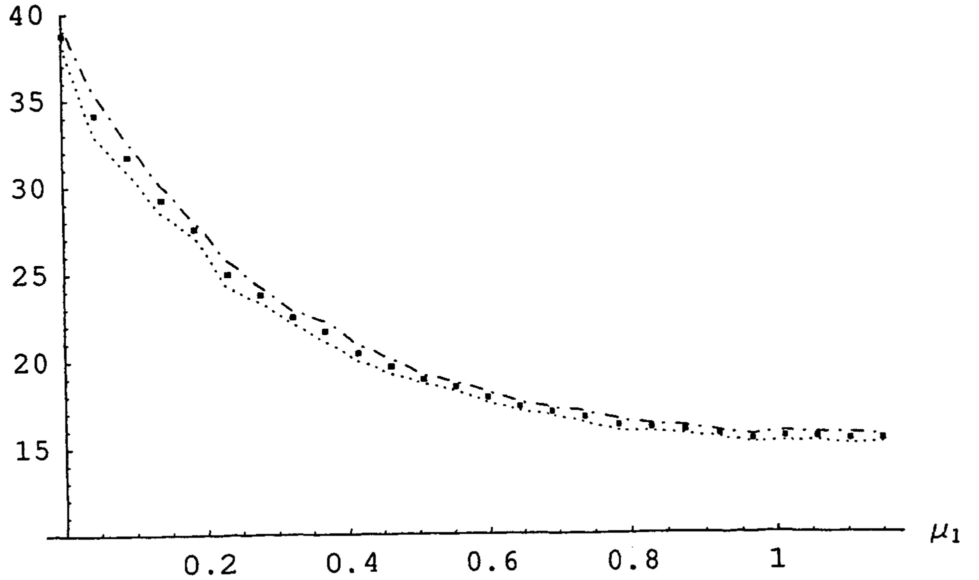

Figure 13.9: Estimated mean individual service time as a function of μ_1 with $\mu_1 + \mu_2 = 2.30$.

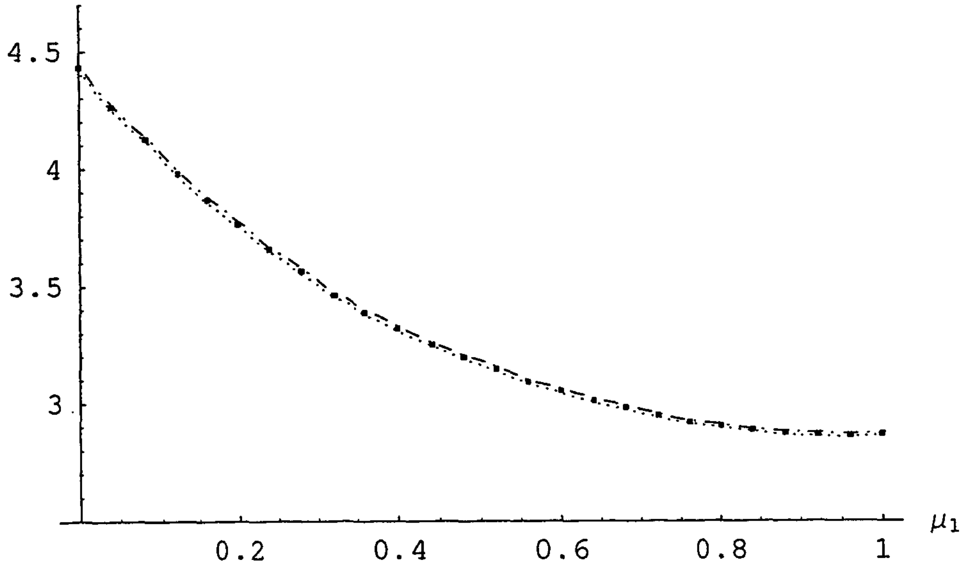

Figure 13.10: Estimated mean service time of a super customer as a function of μ_1 with $\mu_1 + \mu_2 = 2$.

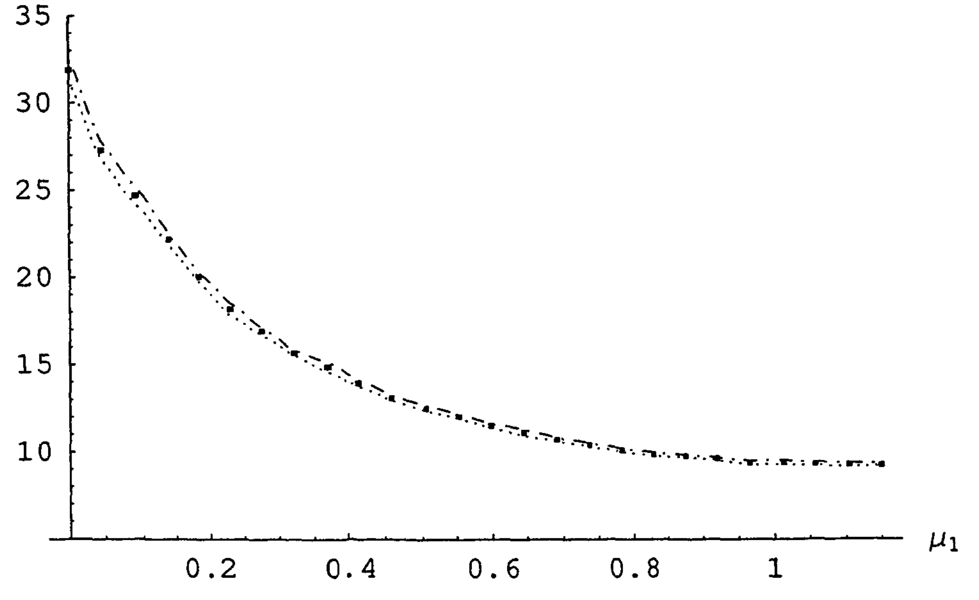

Figure 13.11: Estimated mean service time of a super customer as a function of μ_1 with $\mu_1 + \mu_2 = 2.30$.

In the two graphs 13.12 and 13.13, the results that are obtained via this extensive simulation are plotted. The estimates for the mean individual as well as the mean super customer service time in case of $\mu_1 + \mu_2 = 2$ are given, together with their confidence intervals. As one can conclude, both functions are decreasing in μ_1.

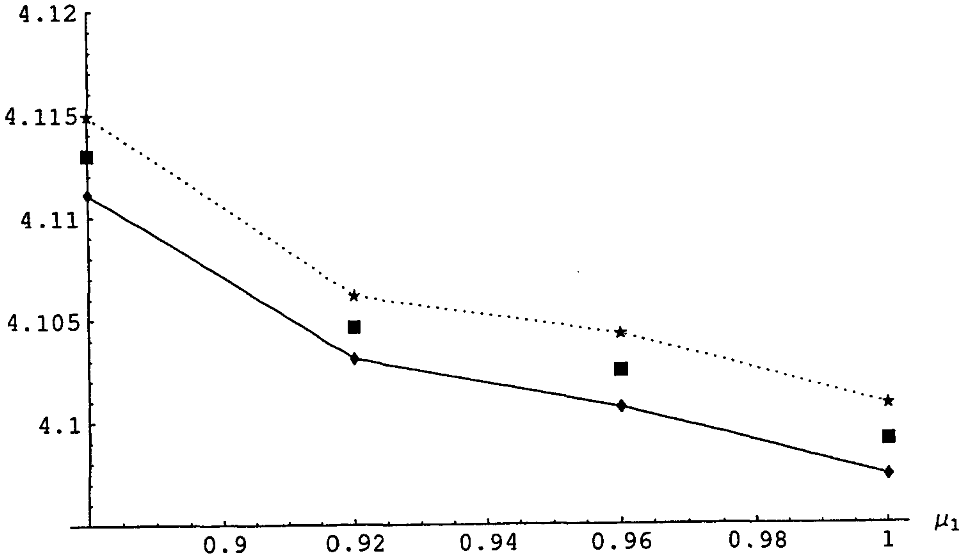

Figure 13.12: Estimated mean individual service time as a function of μ_1 with $\mu_1 + \mu_2 = 2$.

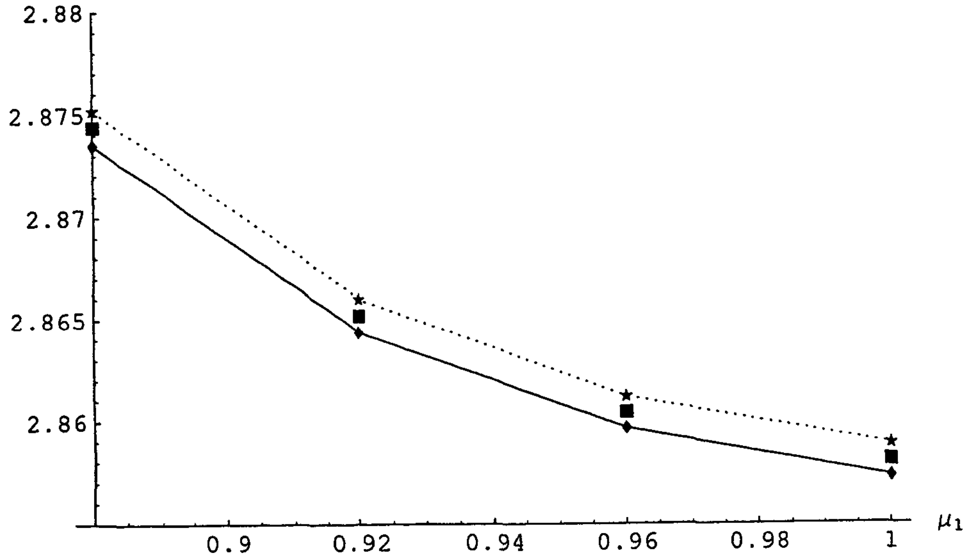

Figure 13.13: Estimated mean service time of a super customer as a function of μ_1 with $\mu_1 + \mu_2 = 2$.

Chapter 14

A comparison through simulation

14.1 Introduction

In this chapter, the four different contention resolution procedures, as briefly described in Section 2.2, will be compared through simulation. The comparison is made with respect to a couple of performance measures. One of these performance measures is the capacity of contention resolution procedure. Other measures include the expectation and variance of the waiting and sojourn time distribution of an individual customer. Simulations are not only done for *ternary* contention trees, but also for q-ary contention trees with $q = 2, 3, 4$. For the moment, only the ternary results are presented.

The simulations have taken place in a relatively early phase of our investigations of the models. At that moment, most of the results were not obtained yet by analysis. However, this is not the case anymore. Most of the estimates that have been obtained via simulation, have been obtained by analysis as well. It is obviously interesting and useful to make the comparison between analysis and simulation. These comparisons have been made already and the results are placed in the various chapter that were devoted to the analysis of the models.

However, the comparison between different contention resolution procedures has not yet taken place. That is the main subject of this chapter. The next section will be devoted to the introduction of a couple of performance measures that will be used to make the comparison between the different contention resolution procedures.

We will end this section with some information on how the simulations are carried out exactly. All simulation programs are written in Turbo Pascal 7.0. They have all been extensively verified. For all estimates, confidence intervals have been determined. Simulation parameters as run length and number of runs have each time been chosen such that the confidence intervals became acceptably small. On the other hand, the total run time of one simulation was required not to be to large. For almost all simulation results that are presented (graphically) without their confidence intervals, it holds that they can almost be regarded as the true values since the width of the confidence intervall is very small. However, when the confidence interval indicated that the estimate was not very accurate, we have presented these estimates graphically together with their confidence intervals.

14.2 Performance measures

14.2.1 Capacity of the system

As mentioned before, the four systems will be compared according to their capacity. The capacity of a system is the maximum arrival rate of individual customers for which the system is still stable. For all four systems, the capacity can be obtained via analysis. For a capacity analysis concerning the arrival-slot mechanism, see Chapter 8. It turns out that it is possible to approximate these theoretical values very well through simulation. How this is done will now be briefly described. The procedure is the same for all four models.

One run of a simulation program consists of a number of subruns. A subrun terminates when the system is empty again. To approximate the capacity, a run is started with a certain value for λ. During this run and especially during a subrun, the number of customers in the system will be observed. When this number continues growing during the (sub)run it is likely that the current value of λ exceeds the capacity. This will result in not finishing the started run, because of the fact that one of the subruns will not finish. In a next run a somewhat smaller value of λ is tried. If the run finishes, λ is increased. So the capacity of the system is approximated with a kind of binary search. It turns out that this approximation method is quite accurate (within one percent of the real capacity), although one can never say for sure that a subrun would not have ended eventually when it is aborted. And the other way around, it is also the case that a finite subrun does not directly imply that the system is stable.

14.2.2 Waiting, service and sojourn times of individual customers

Other interesting performance measures are the expectation and variance of the waiting time, the service time and the sojourn time of individual customers. Note that analyzing the first two moments of the waiting, service and sojourn time is sufficient to obtain estimates for expectation and variance. Observe that some of these quantities are dependent on the actual tree protocol: the order in which contention trees are processed. As mentioned before, two commonly used methods are *breadth-first* and *depth-first.* For both methods, results will be presented, so that a comparison between depth-first and breadth-first is possible as well. Estimators for the first moments will be denoted by $\overline{W}_{ind}$, $\overline{B}_{ind}$ and $\overline{S}_{ind}$ respectively. Estimators for the variances will be denoted by $\hat{Var}(W_{ind})$, $\hat{Var}(B_{ind})$ and $\hat{Var}(S_{ind})$, respectively.

Recall that the waiting time of an individual customer was defined to consist of two parts. First a customer has to wait for an arrival slot to enter the system. This was called the "waiting room time".

- In the gated mechanism this can take a very long time.
- In the non-blocked mechanism this takes zero slots.
- In the static arrival-slot mechanism this takes at most s slots.
- In the dynamic arrival-slot mechanism this takes at most s slots.

Second a customer has to wait for the tree it participates in to be processed. This was called the "tree queue time".

- In the gated mechanism this waiting time is zero.
- In the non-blocked mechanism this time is also zero.
- In the arrival-slot mechanism this waiting time can be zero if the customer is lucky, but in general this time is positive and corresponds to the waiting time in the tree queue. It is a matter of definition whether the arrival slot in which the tree is formed is included in the waiting time or not. In the former case the arrival slot is seen as a part of the service time and not of the waiting time.

- The situation for the dynamic mechanism is similar to that of the (static) periodic arrival slot mechanism.

Estimators for the first moments of the two parts of the waiting time of an individual customer will be denoted by $\overline{W}_{1,ind}$ and $\overline{W}_{2,ind}$, respectively. Estimators for the second moments of the two parts of the waiting time of an individual customer will be denoted by $\hat{Var}(W_{1,ind})$ and $\hat{Var}(W_{2,ind})$, respectively.

14.2.3 Waiting, service and sojourn times of super customers

It is also interesting to examine the expectation and variance of the waiting, service and sojourn time of super customers. These quantities do not depend on the tree protocol that is used.

In the gated mechanism as well as the non-blocked mechanism, the waiting time equals zero by definition. In these two mechanisms the (mean) sojourn time corresponds to the (mean) length (measured in slots) of a tree. In the arrival-slot mechanism the mean waiting time is in general non-zero. The service time of a super customer in the arrival slot mechanism consists of slots that are occupied by the own tree as well as of slots that are used for new arrivals to enter the system. Therefore, for both the total service time and the pure service time, estimators for the expectation and variance of these variables will be presented.

The estimators for the expectations of waiting time, service time, pure service time and sojourn time will be denoted by $\overline{W}_{super}$, $\overline{B}_{super}$, $\overline{B}_{pure,super}$ and $\overline{S}_{super}$, respectively.

The estimators for the variances of waiting time, service time, pure service time and sojourn time will be denoted by $\hat{Var}(W_{super})$, $\hat{Var}(B_{super})$, $\hat{Var}(B_{pure,super})$ and $\hat{Var}(S_{super})$, respectively.

14.3 Capacity results

In this section (simulation) results will be presented for q-ary trees. Because of the fact that in case of the gated mechanism the capacity approximates Log q very closely, this number will be displayed. The non-blocked results are obtained via simulation. Because of the fact that the capacities for the static and dynamic arrival slot mechanism are equal, they are presented only for the static case. In case of the (static) arrival slot mechanism with ternary trees, the analytical results that are obtained are given, instead of the simulation results. When looking at table 14.1, there are a few remarkable

Mechanism	$q=2$ μ_{max}	$q=3$ μ_{max}	$q=4$ μ_{max}
Gated	0.3466	0.3662	0.3466
Non-blocked	0.360	0.40	0.40
$s=1$	0.420	0.4012	0.368
$s=2$	0.427	0.4132	0.378
$s=3$	0.419	0.4080	0.374
$s=4$	0.410	0.4017	0.369
$s=20$	0.363	0.3753	0.352
$s=100$	0.350	0.3680	0.348
$s=2000$	0.347	0.3662	0.347

Table 14.1: Capacity results in case of q-ary contention trees

observations. When $q = 4$, the non-blocked mechanism really outperforms the other mechnanisms. When q equals 2 or 3, the mechanism with $s = 2$ has the highest capacity of all mechanisms. Furthermore, the gated mechanism reaches a maximum capacity for $q = 3$. In fact, this maximum is over *all* values of q. It is also true that the case $q = 3$ dominates the case $q = 4$ for all studied mechanisms. But for binary trees some mechanisms have a higher capacity and some have a lower capacity than their ternary equivalent.

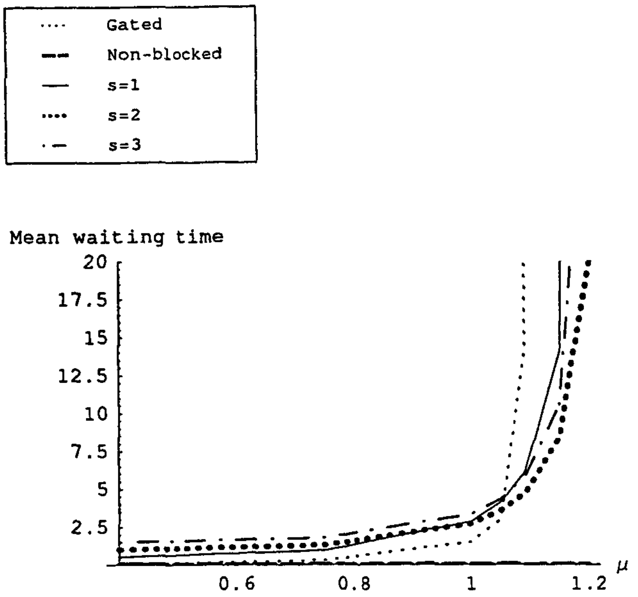

Figure 14.1: Estimated mean waiting time as a function of μ

14.4 Individual performance measures

14.4.1 Mean waiting, service and sojourn times

In this subsection, simulation results for the mean waiting, service and sojourn time of individual customers are presented.

The figures 14.1 until 14.9 contain graphs of the expectations of waiting, service and sojourn times. The first three graphs summarize *mean waiting time* results, the next three graphs present *mean service time* results and the final three present *mean sojourn time* results. We remark that in these nine graphs the *breadth-first* results are presented.

In the three waiting time graphs, the (estimates of the) mean waiting time of an individual customer in several contention resolution mechanisms is plotted as a function of μ, the arrival intensity per slot. The same is done for the service time and sojourn time. Subsequently, these result will be discussed.

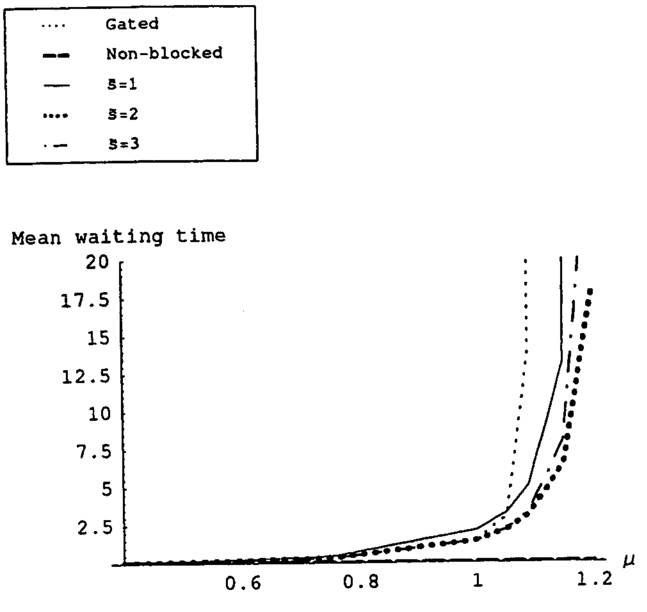

Figure 14.2: Estimated mean waiting time as a function of μ

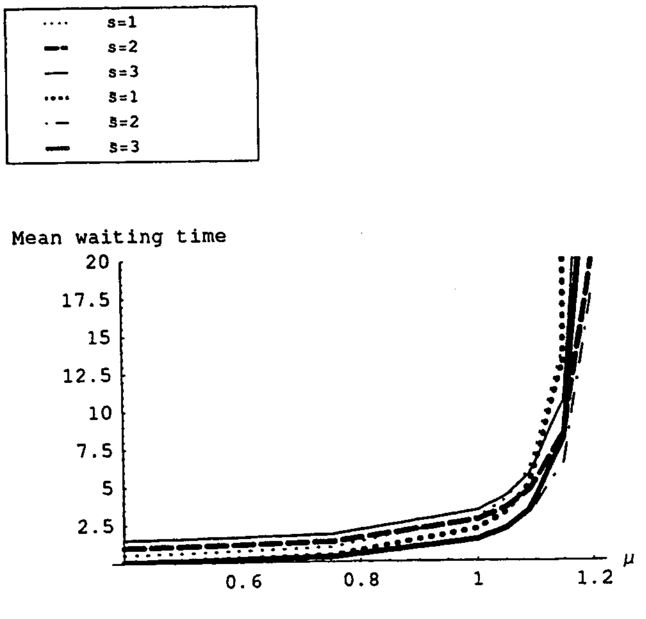

Figure 14.3: Estimated mean waiting time as a function of μ

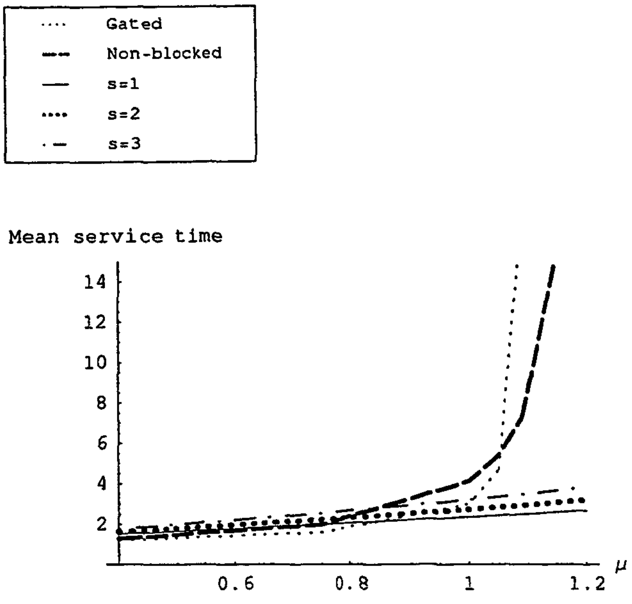

Figure 14.4: Estimated mean service time as a function of μ

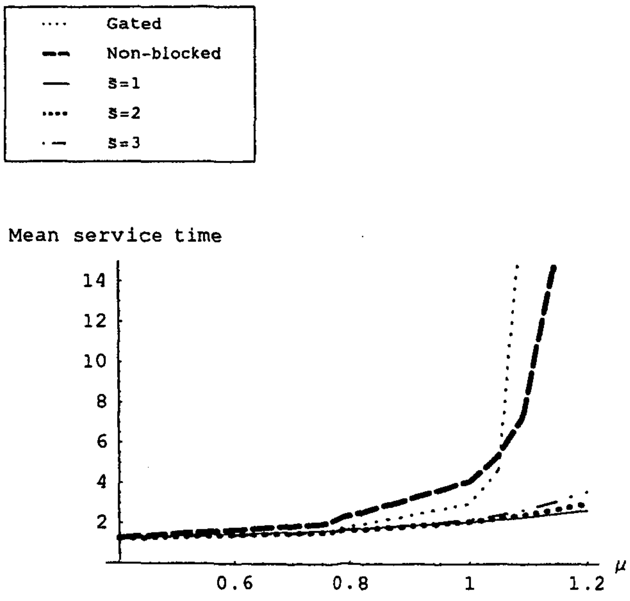

Figure 14.5: Estimated mean service time as a function of μ

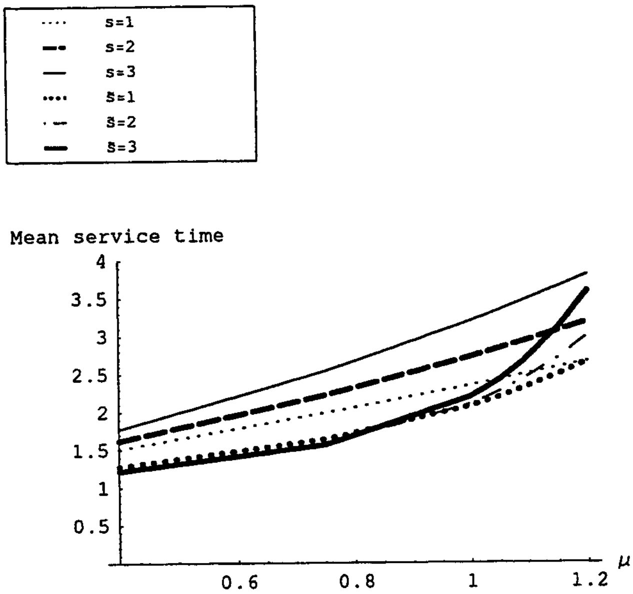

Figure 14.6: Estimated mean service time as a function of μ

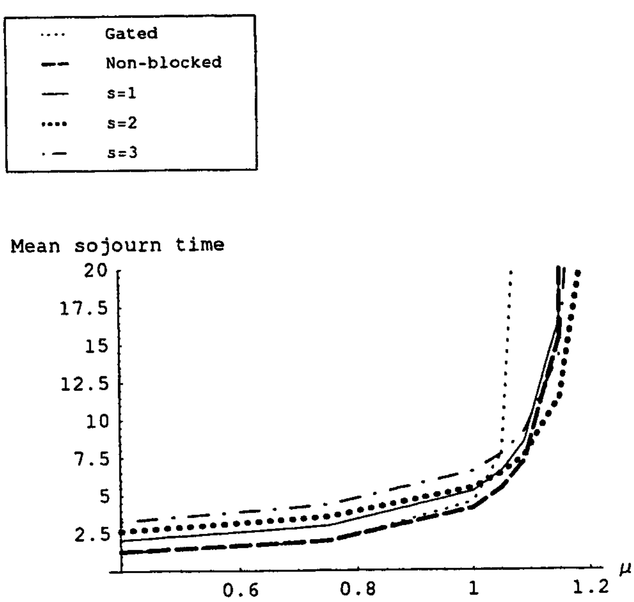

Figure 14.7: Estimated mean sojourn time as a function of μ

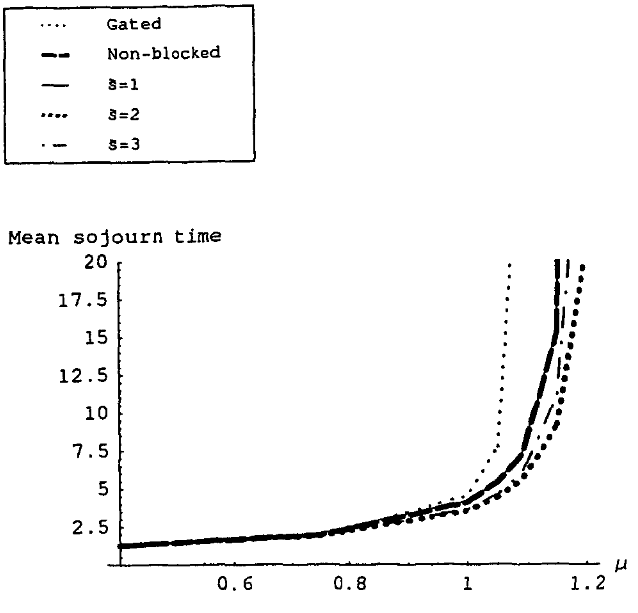

Figure 14.8: Estimated mean sojourn time as a function of μ

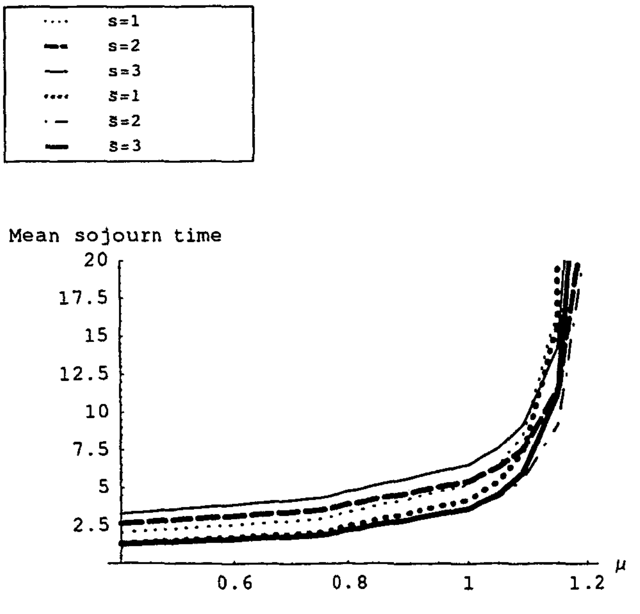

Figure 14.9: Estimated mean sojourn time as a function of μ

Waiting time results
When we observe figure 14.1, we see that the gated mechanism performs very well in the ligth-traffic zone. In this zone, the mechanism with $s = 1$ has a good performance as well. This is intuitively clear. In the heavy-traffic zone, the capacity of the system is eventually the important factor that determines the performance of a system. Therefore, we see that the mechanism with $s = 2$ outperforms all the other mechanisms, except the non-blocked mechanism, in which individual customers experience by definition a waiting time of zero slots.

In the second graph, figure 14.2, we observe that the dynamic arrival slot mechanism with $s = 2$ outperforms all other mechanisms (except the non-blocked mechanism) in both light-traffic and heavy-traffic regions. This arrival slot mechanism has the highest capacity, which is important for a good performance in the heavy-traffic zone, and this mechanism "imitates" the gated mechanism in the light-traffic zone.

Service time results
In figure 14.4, we see for the three arrival slot mechanisms a linear behaviour of the service time as a function of μ, where the performance shrinks with increase of s. This phenomenon has already been described during the analytic part of this report. For small values of μ the non-blocked and gated mechanism show a slightly better performance than the three arrival slot mechanisms.

When we inspect figure 14.5, we observe that indeed the mean service time of the dynamic mechanisms in the light-traffic zone has decreased to the level of the gated mechanism. Only in the heavy-traffic zone, there is a small difference in the three mechanisms and furthermore a small curve is present in this zone, which must "compensate" for the decrease in the light-traffic zone, since the mean values in the heavy-traffic region have not altered that much in comparison with the static mechanisms.

Figure 14.6 confirms that in the high-traffic zone, the static and dynamic variant of the same arrival slot mechanism does not differ very much with respect to mean service times.

Sojourn time results
The most interesting comparison between the different contention resolution procedures is made with respect to the mean sojourn time. When we inspect figure 14.7, we see again that the non-blocked and gated mechanism are su-

perior to the static arrival slot mechanisms in light traffic. In heavy traffic, the mechanisms are ordered with respect to their capacity.

Figure 14.8 is the most interesting figure. In this figure, the dynamic arrival slot mechanisms are compared with the non-blocked and gated mechanism with respect to the mean sojourn time. The outcome of this comparison is promising. We see that the arrival slot mechanism with $s = 2$ outperforms all other mechanisms over the total region of intensity values, so both in light and in heavy traffic intensities. The other arrival slot mechanisms show a good performance as well. The mechanism with $s = 3$ is for all intensities the second best mechanism. The mean sojourn times in the non-blocked mechanism and the arrival slot mechanism with $s = 1$ nearly coincide for every value of μ.

Breadth-first versus depth-first

We will now present some graphs which compare the results of the breadth-first and depth-first tree protocol. Because of the fact that the waiting time results are independent of the tree-processing protocol, only service time results are compared.

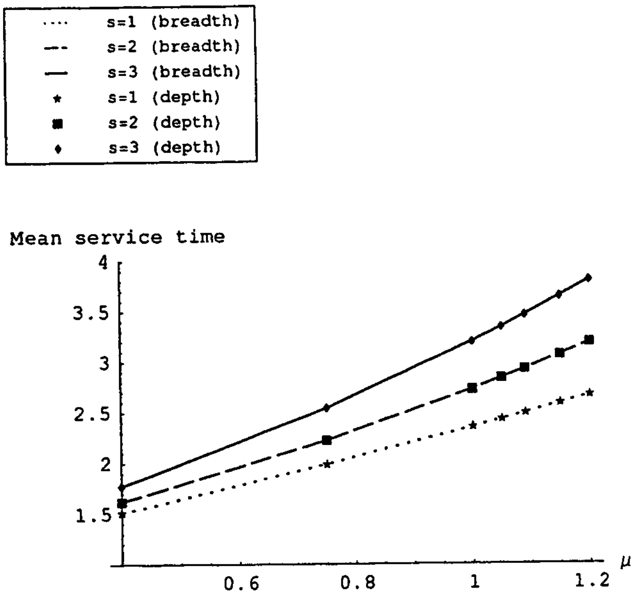

Figure 14.10: Comparison of breadth-first and depth-first mean service times in the static arrival slot mechanism

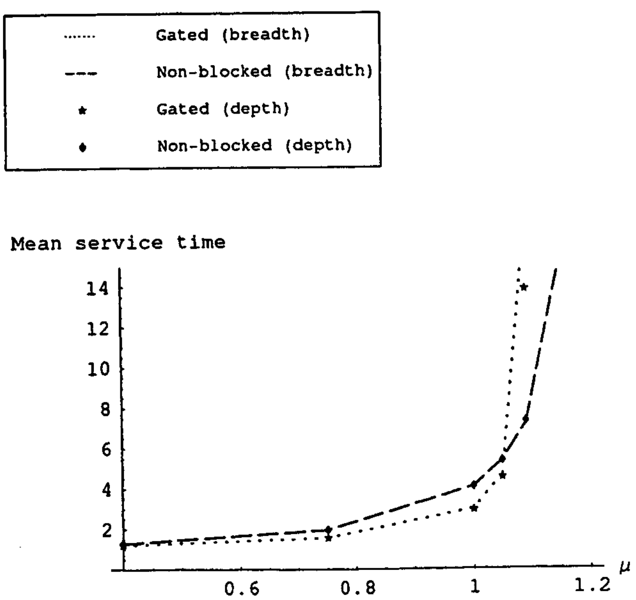

Figure 14.11: Comparison of breadth-first and depth-first mean service times in the gated and non-blocked mechanisms

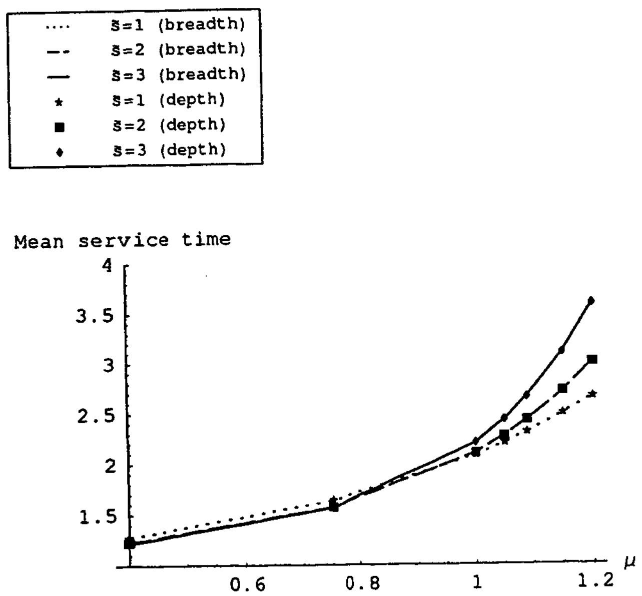

Figure 14.12: Comparison of breadth-first and depth-first mean service times in the dynamic arrival slot mechanism

Inspecting figure 14.10, we can remark that for the static arrival slot mechanisms, there is no significant difference between breadth-first and depth-first tree protocols with respect to the mean sojourn times.

Figure 14.11 shows that also for the gated and non-blocked mechanism, there is almost no difference in the mean service time of both tree protocols. For the gated mechanism, depth-first seems to be slightly better, as far as the *mean service time* is considered.

The last graph, figure 14.12 shows that in case of a dynamic arrival slot mechanism, there is again no significant difference between depth-first and breadth-first.

The lack of difference between both tree-protocols can be explained intuitively, at least for the arrival slot mechanisms. Due to the nature of the arrival slot mechanisms (with small value of s), the initial number of customers in a tree is small for almost all of the trees that are being processed. As an illustration, for $\mu = 1$ and $s = 2$ we have that the probability that the initial number of customers is more than 4 equals 0.184. Therefore, after the arrival slot has taken place and a super customer is formed, most of the times this super customer consists of only one active subtree with 2 customers in it. Depth-first or breadth-first is clearly equivalent for such trees.

In case of a gated mechanism wiht heavy traffic, large trees can occur. For large trees, depth-first seems to be slightly better as far as the *mean* service time is considered, because of the fact that the depth-first protocol ensures that the first departures of individual customers take place very soon instead. This can be seen as applying the "shortest processing time first" rule in the context of contention trees.

14.4.2 Variance of waiting, service and sojourn time

In this subsection, the simulation results for the variances of the performance measures are graphically presented. These results are obtained by estimating the first and second moment of the waiting, service and sojourn time. The variances follow directly from these estimates. In general, the confidence intervals corresponding to the estimates for the second moments are larger than those corresponding to the estimates for the first moments. Therefore, comparing the breadth-first and depth-first results concerning the second moments is more difficult. However, it seems the case when inspecting the simulation results that the breadht-first and depth-first results do not differ too much, where in general the depth-first estimates are a little smaller than the breadth-first estimates. In this subsection, the breadth-first results will be presented.

Figures 14.13 through 14.29 contain graphs of the variances of waiting, service and sojourn times. The first six graphs summarize waiting time results, the next five graphs present *mean* service time results and the last six show *mean* sojourn time results.

In the six waiting time graphs, the (estimates of the) variance of the waiting time of an individual customer in several contention resolution mechanisms is plotted as a function of μ, the arrival intensity per slot. To present the results transparently, each graph is presented twice, where the first graph zooms in on the lower values of μ and the second graph presents an overview of the results for the whole range of arrival rates. The same is done for the service time and sojourn time respectively.

After the presentation of the graphs, the results will be discussed briefly. The main focus will be on the comparison between the several mechanisms. Each graph is accompanied by a legend. The static arrival slot mechanisms are indicated by $s = \ldots$, where the dynamic arrival slot mechanisms are indicated by $\tilde{s} = \ldots$.

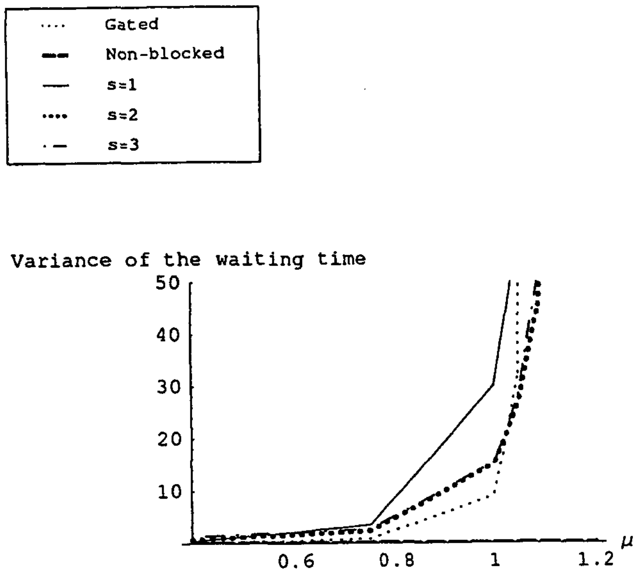

Figure 14.13: Estimated variance of the waiting time as a function of μ

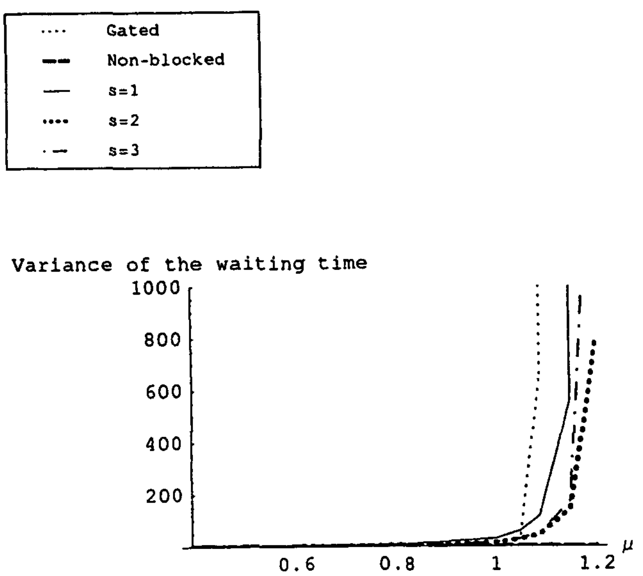

Figure 14.14: Estimated variance of the waiting time as a function of μ

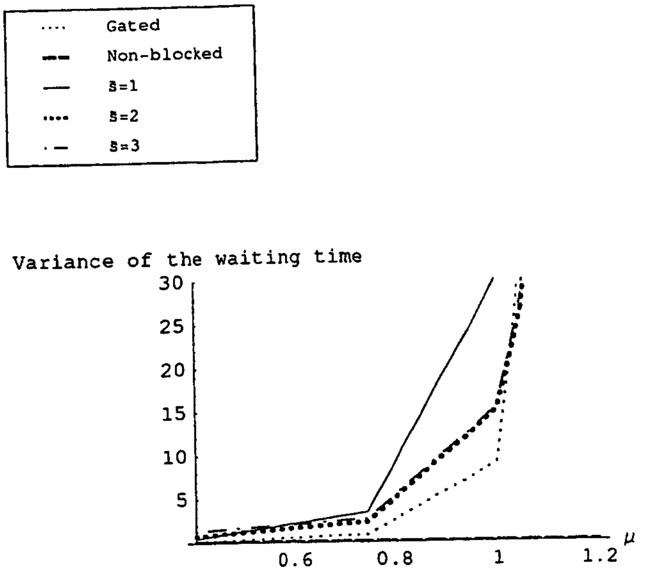

Figure 14.15: Estimated variance of the waiting time as a function of μ

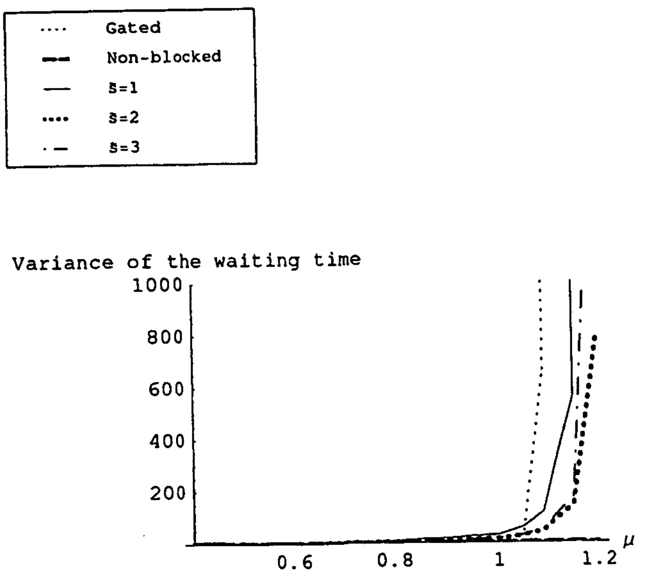

Figure 14.16: Estimated variance of the waiting time as a function of μ

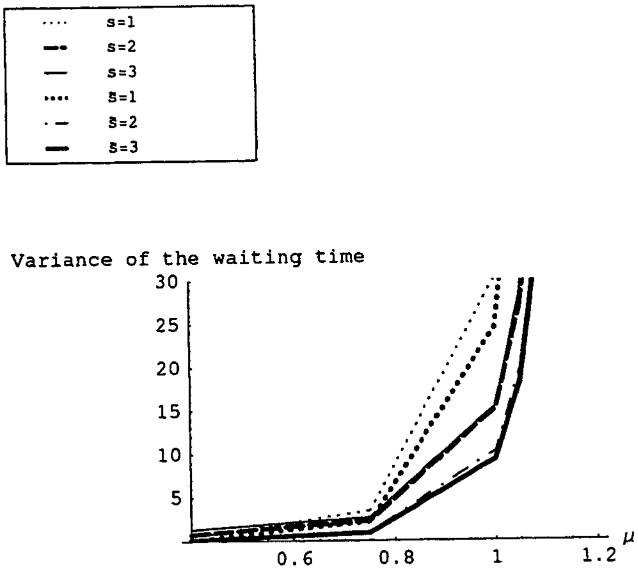

Figure 14.17: Estimated variance of the waiting time as a function of μ

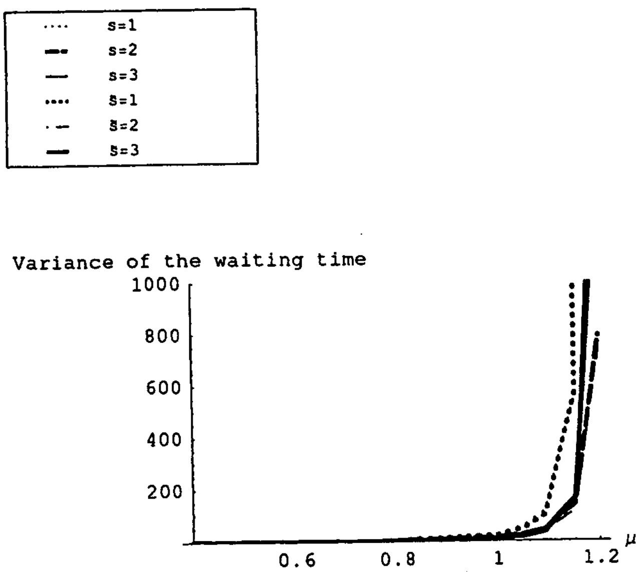

Figure 14.18: Estimated variance of the waiting time as a function of μ

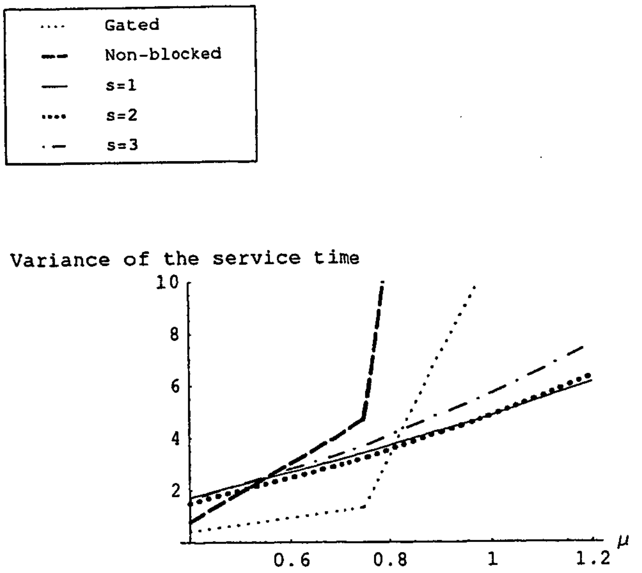

Figure 14.19: Estimated variance of the service time as a function of μ

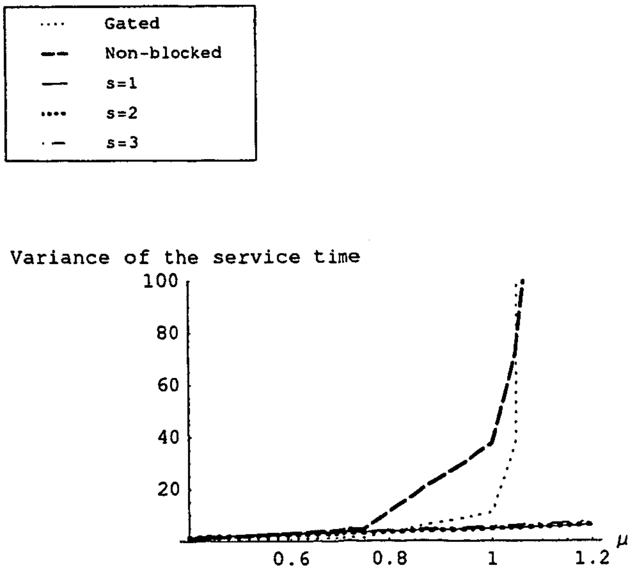

Figure 14.20: Estimated variance of the service time as a function of μ

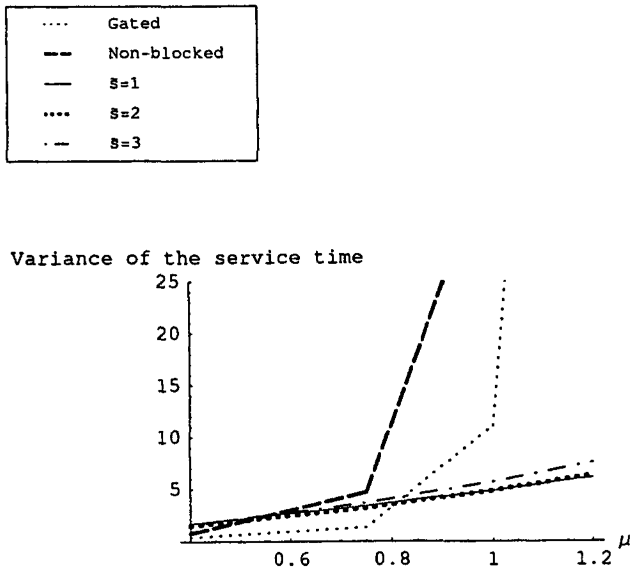

Figure 14.21: Estimated variance of the service time as a function of μ

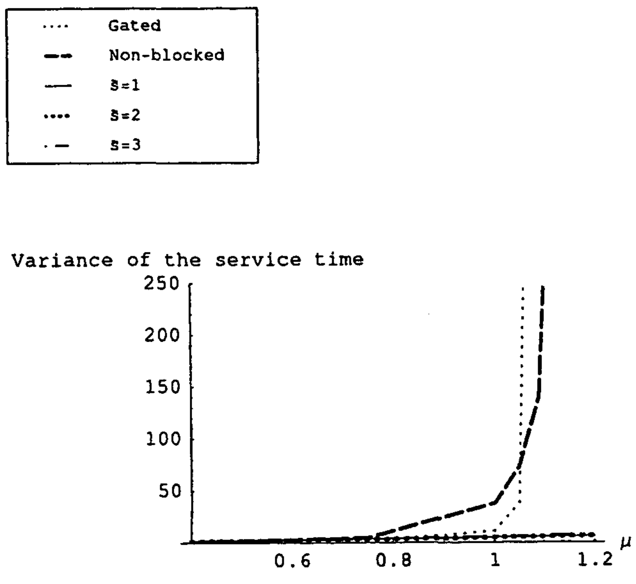

Figure 14.22: Estimated variance of the service time as a function of μ

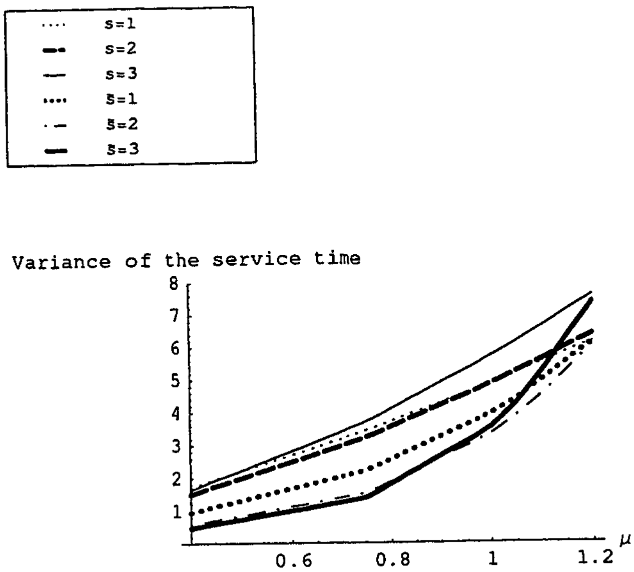

Figure 14.23: Estimated variance of the service time as a function of μ

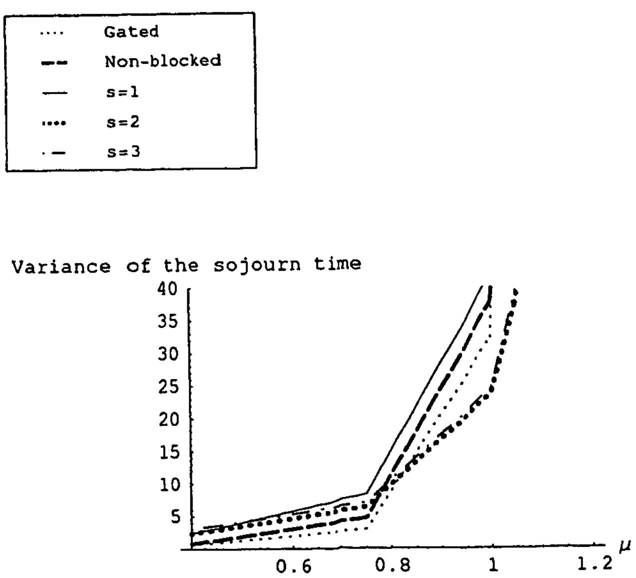

Figure 14.24: Estimated variance of the sojourn time as a function of μ

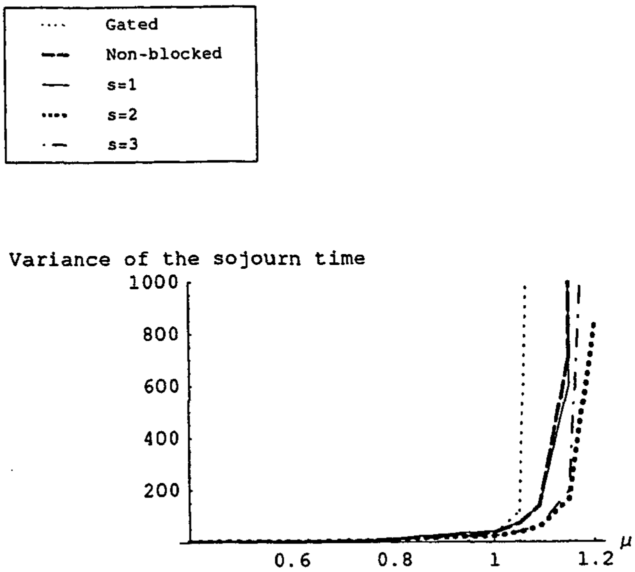

Figure 14.25: Estimated variance of the sojourn time as a function of μ

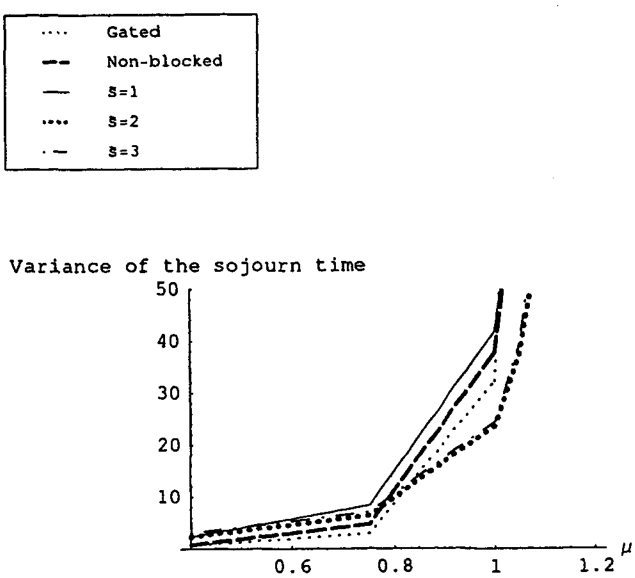

Figure 14.26: Estimated variance of the sojourn time as a function of μ

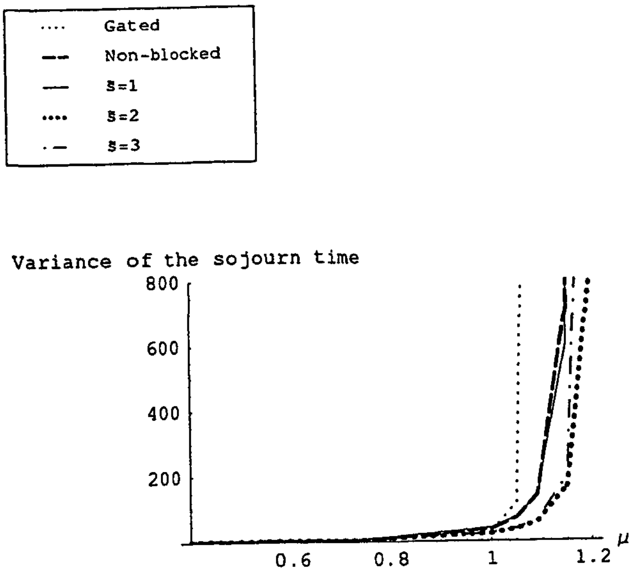

Figure 14.27: Estimated variance of the sojourn time as a function of μ

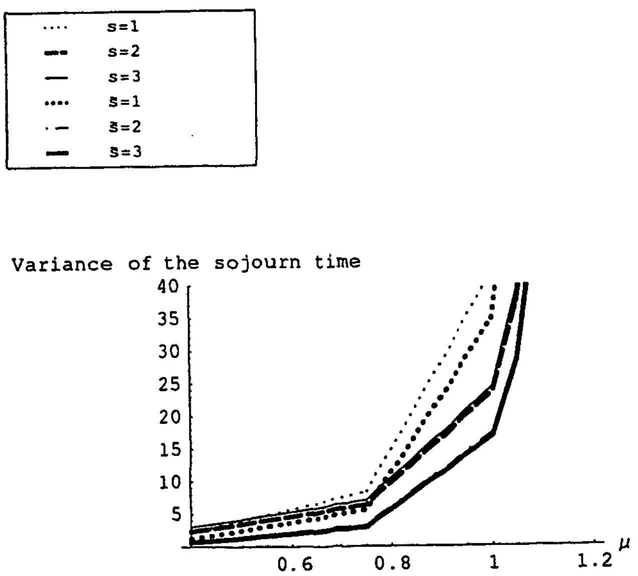

Figure 14.28: Estimated variance of the sojourn time as a function of μ

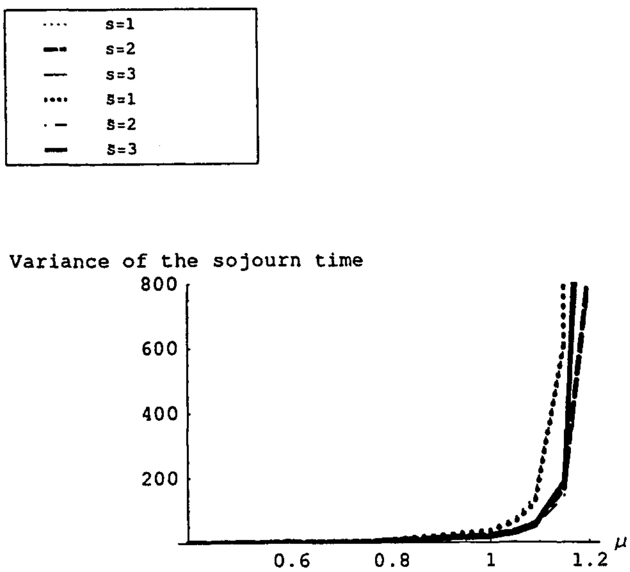

Figure 14.29: Estimated variance of the sojourn time as a function of μ

Discussing the results that were presented in the previous graphs, we will focus on the sojourn time results. The waiting and service time results can be observed in the graphs, but they will not be elaborately discussed here.

An interesting remark can be made on the variance of the service time in the gated mechanism compared with the static arrival slot mechanisms. The corresponding graph is given in figure 14.19. In this graph, we see a considerably smaller variance for the gated mechanism as compared with the arrival slot mechanisms in the light-traffic zone. This can be explained as follows. In the gated mechanism, the number of customers that is involved in a tree is small most of the time. In the arrival slot mechanisms, the arrivals are accumulated over a period of $s + 1$ slots. In light traffic, this period of $s + 1$ slots will be longer than most gate periods. Consequently, there will be more variance in the number of customers involved in a tree, leading to larger variances of the service time.

In figure 14.24, we see in fact a similar behaviour as in the corresponding mean sojourn time graph. In the light-traffic zone, the gated and non-blocked mechanism outperform the static arrival slot mechanism. When the intensity μ increases, the capacity of the system becomes relevant and determines the performance with respect to the variance of the sojourn time completely. This is confirmed in figure 14.25.

Observing the most interesting graphs, presented in figures 14.26 and 14.27, we see that with respect to variance of the sojourn time, the gated mechanims and the non-blocked mechanism outperform the dynamic arrival slot mechanism in case of light traffic. This is different as in the *mean* sojourn time case, where we saw that over the whole range of arrival intensities, the dynamic arrival slot mechanism was not outperformed. However, in case of heavy traffic, we see a considerably smaller variance for the dynamic arrival slot mechanisms.

The last two graphs emphasize again that the dynamic mechanisms lead to smaller variances than the static equivalents.

14.5 Performance measures for super customers

In this section, some of the simulation results on super customers are summarized in graphs. Three graphs will be presented in which successively the estimated mean waiting time, mean service time and mean sojourn time of a supercustomer in various arrival slot mechanisms are plotted as a function of μ, the arrival intensity per slot.

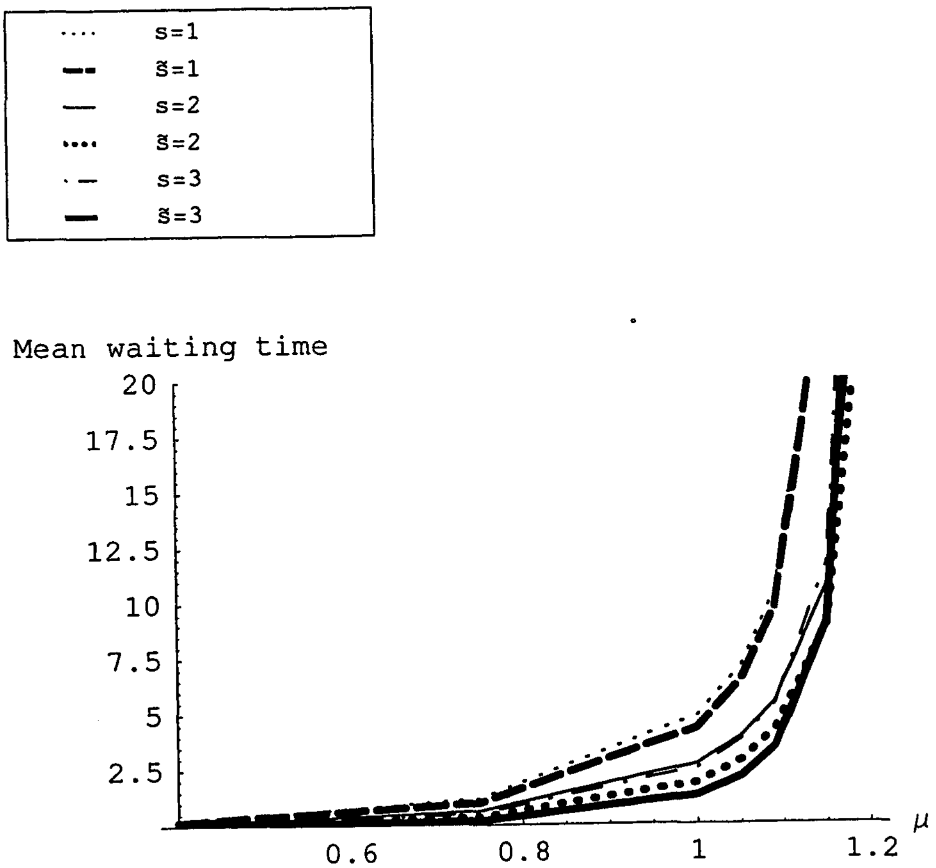

Figure 14.30: Estimated mean waiting time as a function of μ

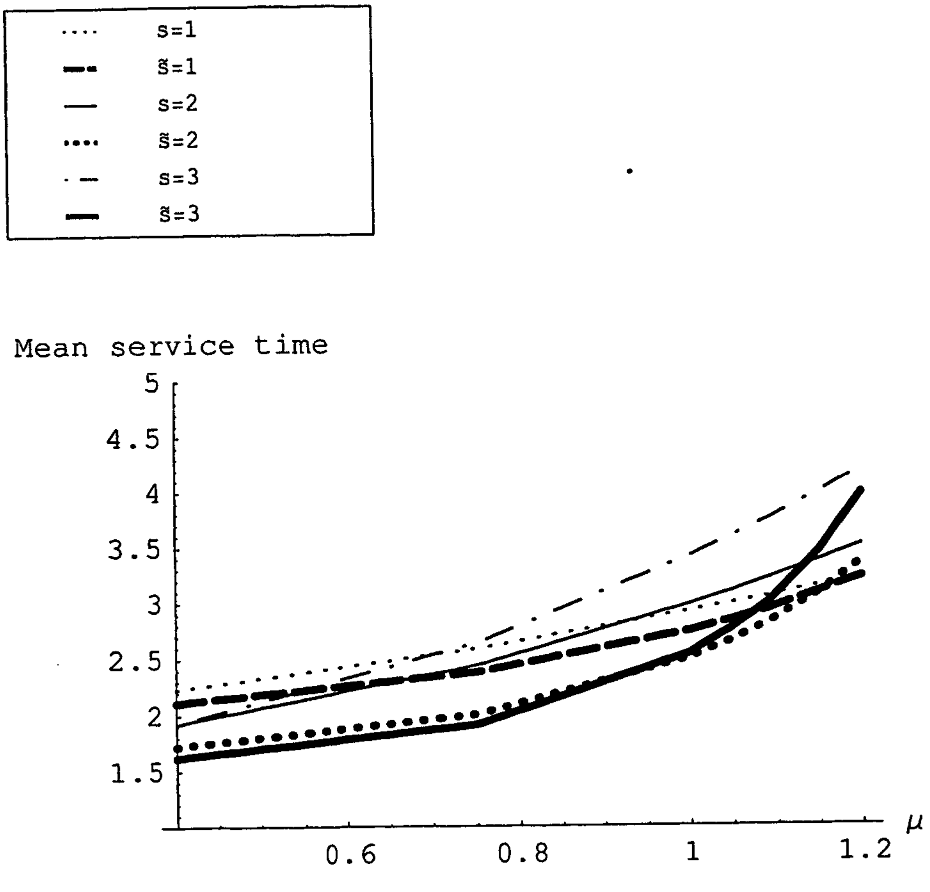

Figure 14.31: Estimated mean service time as a function of μ

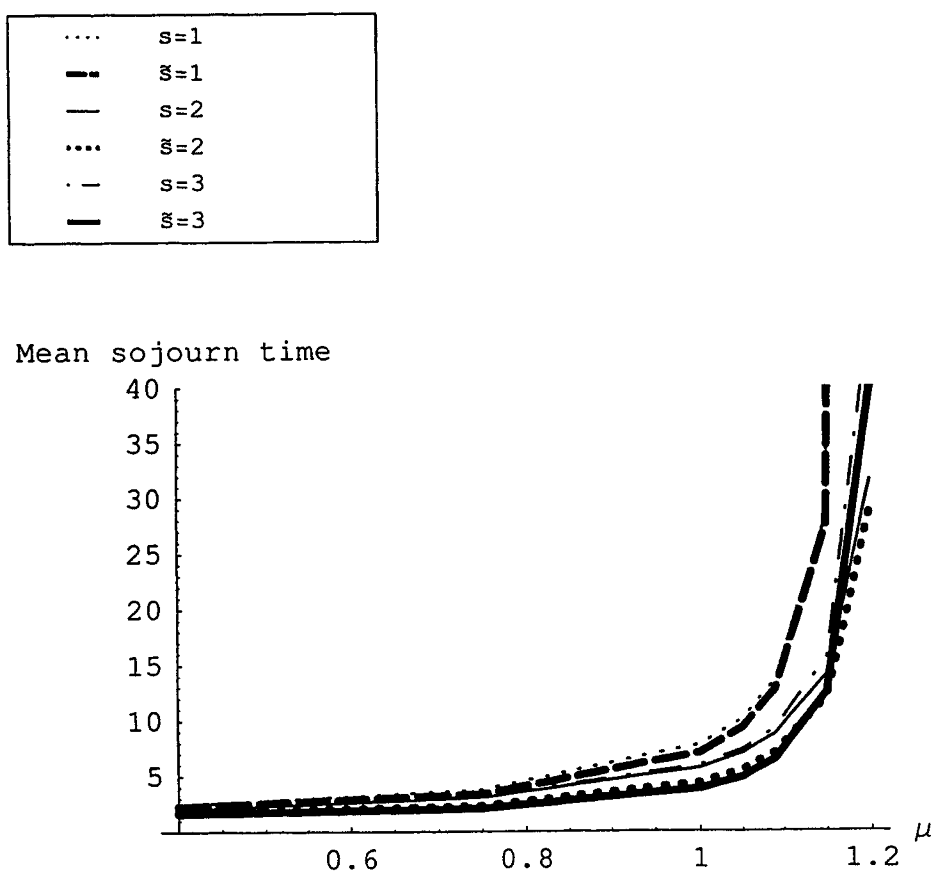

Figure 14.32: Estimated mean sojourn time as a function of μ

Chapter 15

Contention resolution together with data transfer

15.1 Introduction

In the previous chapters, contention resolution was studied without considering the presence of data slots. This was done for the sake of simplicity. As mentioned already in Chapter 1, contention resolution is only one part of the total mechanism. Transmitting data (cells) over the same channel is the other part. In this chapter, contention resolution and sending data by means of data slots are linked together. Furthermore, it will be studied how the positive results for the arrival slot mechanisms can be used in a more realistic model in which data slots are also present.

Every access network has a couple of characteristics: a network contains a (finite) number of customers, which generate data with a certain intensity. When a customer wants to send data, he makes a request and finally becomes involved in a contention resolution procedure. After that, the customer is allocated a number of slots, corresponding to the size of his last request. Note that his last request may differ from his initial one, because of the fact that a customer is allowed to update his request during the contention resolution procedure. In a realistic access network, a frame consists of 18 time slots. From now on it is assumed that every slot of a frame is either a *data slot* or a *contention slot*. At the beginning of each frame, the Head End determines which of these 18 slots will be used as data or contention slots. This must be done in a sensible way, because the stability of the network depends on how this is done. When considering the stability of an access network, contention resolution and data transfer cannot be treated independently. The number of slots in a frame is fixed and this fixed number of slots must be divided over contention resolution and data transfer in a feasible way such that both contention resolution and data transfer are stable processes. So the stability condition will depend on both the intensity of the data generating processes as well as the number of customers in the system.

In general, at the beginning of each frame two decisions must be made:

1. *How many contention slots will be present in the upcoming frame?*
   When making this decision, some information may be important:

   - The total amount of data cells that is currently in the data queue, waiting to be sent.
   - The (recent) history of the contention resolution.
   - The actual scheduling of both data and contention slots in the near past.

2. *Given the number of contention slots in the next frame, how must these be distributed over the frame?*
   Allocation of slots means in every case determining the exact placement of the contention slots in a frame. In case of an arrival slot mechanism, there is the additional issue whether a contention slot becomes an arrival slot or not.

   The reason why it is important to allocate the slots in a sensible manner is that it is possible in this way to control the arrival process of the customers, at least to some extent. When it is the case that no new arrivals are allowed to enter a certain contention slot, it becomes less relevant when this contention slot exactly takes place. When making the allocation decision, it seems important to consider when the previous contention slots took place.

So at the beginning of each frame two decisions must be made. It is of course possible to base these decisions on the actual state of the system at the beginning of each frame and this will in fact lead to the best performance, if done correctly of course. A somewhat simpler approach is to make these decisions irrespective of the actual state of the system, but only considering the *expected* behaviour of the system. This does in no way mean that the number of slots in each frame dedicated to contention resolution (and to data transfer) has to be fixed. Of course the allocation has to be feasible with respect to stability. Now the term *allocation strategy* will be introduced.

An allocation strategy is a rule that determines at the beginning of each frame how each slot in the next frame will be used.

An allocation strategy might take into consideration the actual state of the system. This will be called a *dynamic* strategy. When only the allocation of slots that took place in the previous frame(s) is considered, a strategy will be called a *static* strategy.

The quantities that can be fully observed by the Head End and therefore can be incorporated in a dynamic allocation strategy are at least the following:

- In all mechanisms, a quantity that one can fully observe is the length of the data queue.
- In case of an arrival slot mechanism one can observe the length of the tree queue.
- In case of a gated or a non-blocked mechanism, one can observe the length of the current tree until now.

The question is whether incorporating all such kind of information in the scheduling of contention and data slots leads to a much better performance, compared with a static allocation strategy. It is certainly true that scheduling rules become far more difficult when incorporating all this information. For the moment, we concentrate on static strategies. This is the subject of the next section.

15.2 Static allocation of contention slots

15.2.1 Introduction

In this section we will consider so-called static allocation strategies. So the following question will be considered: what is an intelligent way to allocate the contention slots beforehand, when the stochastic behaviour of the system is known? We will restrict ourselves for the moment to allocation strategies that are periodic in the sense that after a certain number of frames the allocation strategy repeats itself. This is not really a restriction.

Let N_f denote the number of slots in a frame. Let K denote the number of frames after which the strategy repeats itself and let n_c be the number of contention slots that has been allocated in these K frames. (This implies an average number of contention slots per frame equal to $\frac{n_c}{K}$. This value must be based on the characteristics of the network.) Furthermore, define n_d to be the number of data slots used in these K frames. Now we come to the question how to choose the *integers* K and n_c and after that, how to allocate these n_c contention slots to the KN_f slots. This choice will depend on the objective. Three possible objectives are the following:

1. Maximizing the capacity of the system.

2. Minimizing the mean sojourn time of individual customers.

3. Minimizing the variance of the sojourn time of individual customers.

In the next two subsections, the first two objectives will be considered separately. From now on, we assume again that the arrival process of customers that want to participate in the contention resolution is a Poisson process.

15.2.2 Maximizing capacity

One possible objective is to maximize the capacity of the contention resolution procedure. Capacity is measured in terms of customers per contention slot. In the previous chapters we have seen that the arrival slot mechanism outperforms the gated mechanism as well as the non-blocked mechanism as far as capacity is concerned. But we are considering a slightly different model now, because of the fact that the time between two consecutive contention slots does not have to be constant anymore. This is a very important difference.

How do the three contention resolution procedures perform in this different setting? Before this issue is examined, the previously discussed setting (without the data slots) will be slightly generalized, so that in the new setting we can sometimes rely on the results found in the previous chapters, although the models under consideration seem to be different from those in the previous chapters.

A couple of remarks can be made:

1. In the previous chapters, it was assumed that there was zero time between two consecutive slots. This can be generalized to the case that the time between two consecutive slots is constant, but not necessarily zero. This generalization can roughly be seen as a change of the arrival rate per slot (proportionally to the time between two slot beginnings). Only the first component of the waiting time slightly changes. How this affects the results will be outlined in the future.

2. When comparing the different contention resolution mechanisms, it was implicitly assumed that the number of contention slots per time unit was the same for all mechanisms under consideration. This observation will be used sometimes to compare mechanisms in the new setting.

Now we return to the question how the three different mechanisms will perform in the new setting with data slots in between the contention slots.

1. **Gated mechanism**
   One can argue that the capacity of the gated mechanism does not change in this slightly different model. When a gate period takes very long (and this is the case when the contention resolution system is nearly unstable), the actual allocation of contention slots does not matter.

2. **Non-blocked mechanism**
   In the non-blocked mechanism it is not so obvious what happens to the capacity when the time between contention slots does not have to be constant anymore. Intuition suggests that the capacity is maximized when the inter-slot time is constant. If this is the case, then the capacity remains the same as found in the previous chapters. A simulation study, described in Section 13.9, also supports the conjecture that the capacity is maximized when inter-slot time is constant.

3. **Arrival slot mechanism**
   In case of the arrival slot mechanism it can be shown that the capacity can actually increase when compared with the situation with constant time between slots. For simplicity, consider the static arrival slot mechanism which has the same capacity as the dynamic arrival slot mechanism. Consider again figure 8.5, given in Section 8.2. In this graph, the capacity of the arrival slot mechanism is plotted as a function of s. In this graph one can see that the optimal value for s is approximately $s = 1.8$. Furthermore, one can see that for all values of s between 1.6 and 2, the capacity for the system is at least as high as for the system with $s = 2$. The question is how this can be used in the new model to even increase the capacity. This question will be answered now, by means of an example.

   To approximate the system with $s = 1.8$, on average 5 arrival slots out of every 14 contention slots (including the arrival slots) are scheduled. This gives the desired ratio $\frac{14-5}{5} = 1.8$. Furthermore, schedule the arrival slots equally spaced over the K frames, taking into consideration that after K frames the pattern repeats itself. To do this exactly it has to be the case that KN_f is a multiple of 5. The other contention slots can be scheduled in between. The actual allocation of these slots does not alter the capacity of the system. In this way, the system with $s = 1.8$ is implemented in a certain way.

In the model as described in the previous chapters (without data slots), we saw that $s = 1.8$ was the optimal choice. In that context, a non-integer s seemed irrelevant, but it has just been indicated how a non-integer mechanism can be implemented in the setting *with* data slots. This does certainly not directly imply that the mechanism with $s = 1.8$ is always the best choice among all the arrival slot mechanisms in this new setting with interleaving data slots. Now the difficulties of comparing different arrival slot mechanisms in the setting with data slots in between will be discussed.

As was argued before, the time between two contention slots does not have to be constant anymore in this new model. But for the moment, we will restrict ourselves to mechanisms with a constant time between *arrival* slots. As it will turn out later, this is not a restriction as far as maximizing the capacity of the system is concerned. Furthermore, we drop the restriction that the time line is slotted and assume continuous time. In this setting the question how to choose the integers K and n_c optimally and how to allocate the slots optimally will be answered first.

We assume that the average number of contention slots per frame has already been chosen based upon the network characteristics, and is equal to r, so $n_c = rK$.

Now consider the static arrival slot mechanism with $\frac{1}{s+1}$ arrival slot per contention slot, where s does not have to be an integer. Scheduling (on average) r contention slots per frame, this means that the distance between two consecutive arrival slots is equal to $\frac{N_f K}{n_c \frac{1}{s+1}} = \frac{N_f(s+1)}{r}$, independent of n_c and K. Assuming continuous time, the arrival slots can be scheduled such that this distance is achieved. This distance between two arrival slots implies an arrival rate of $\mu \cdot \frac{N_f(s+1)}{r}$ per arrival slot. Because of the fact that the number of contention slots per time unit is fixed, we can rely on the capacity analysis of the arrival slot mechanism. The outcome of that analysis was that the arrival slot mechanism with $s \approx 1.8$ is the optimal mechanism. From now on, this optimal value will be denoted by s_{opt}. One can argue that for the optimal distance d_{opt} between two arrival slots the following must hold:

$$\begin{aligned} d_{opt} &= \frac{N_f(s_{opt}+1)}{r} \\ &\approx 2.8\frac{N_f}{r}. \end{aligned}$$

Given r, one can always find integer n_c and K such that $r = \frac{n_c}{K}$.

When we assume again the slotted time axis, things become slightly more complicated. The distance between two consecutive arrival slots must be integer in this case. So $\frac{N_f(s+1)}{r}$ must be integer. This induces a restriction on s, given the values of N_f and r. So the optimal $s = 1.8$ is in general not feasible anymore. In fact, there are two candidates s_1 and s_2 for the optimal value of s that must be considered. Those two values satisfy the following:

$$\begin{aligned} s_1 &= \frac{m \cdot r}{N_f}, \\ s_2 &= \frac{(m+1) \cdot r}{N_f}, \end{aligned}$$

with m the integer satisfying the following:

$$m < \frac{N_f(s_{opt}+1)}{r} \leq m+1.$$

Which one is best, s_1 or s_2, can be determined by evaluating the objective for both values.

What can we say about the optimal value of the capacity, given r and N_f? Observing figure 8.5, one can roughly say that the graph is symmetric closely around $s = s_{opt}$. It is the case that the absolute difference between s_{opt} and the optimal feasible s is at most $\frac{1}{2} \cdot \frac{r}{N_f}$. Here $\frac{r}{N_f}$ is just the fraction of slots in a frame that is used for contention. In a real network this fraction is typically small, let's say varying between 0.05 and 0.2. This implies that the optimal feasible s is at most 0.1 away from s_{opt}. Inspecting figure 8.5 again, we can see that in this region the graph is very flat, implying a nearly optimal value. This value is also larger than the optimal value for the non-blocked system and the gated mechanism.

Until now, only arrival slot mechanisms with equidistant arrival slots have been considered. This makes that the results found in the previous chapters can be used. An interesting question that will be answered shortly is whether allowing *unequally* spaced arrival slots results in mechanisms with an even higher capacity. We stress that the average number of contention slots per frame is kept constant in comparison with the "equally spaced" mechanisms in order to make a fair comparison with respect to capacity. It will be shown that the best thing to do is to schedule the arrival slots equidistant.

Consider a time interval of T slots. Assume that this interval of T slots contains a arrival slots, c non-arrival contention slots and d dataslots, with $a + c + d = T$. Then the following question will be answered:

> Given the numbers a,c and d, how must the a arrival slots be divided over this fixed time interval consisting of T slots such that the capacity is maximized?

When we mention capacity in this context, it is assumed that the slotted time axis consists of identically scheduled, connected intervals of length T. Because of the periodicity, one can see this as a partitioning problem: the T slots have to be split up into a disjoint intervals. So we search for a partitioning $x_1, \ldots, x_a$ of the T slots, such that the capacity is maximized. x_i can be seen (disregarding the beginning and end of the interval of T slots) as the total number of slots between the i^{th} and $i+1^{th}$ arrival slot, including the i^{th} arrival slot. When the arrival rate of individual customers equals μ, this implies an arrival rate of μx_i for the $i+1^{th}$ arrival slot. An arrival rate of μx_i implies an arrival probability of a super customer denoted by $\alpha_{\mu x_i}$. The random variable that represents the service time of an arriving super customer depends also on the arrival rate, so this variable will be denoted by $B_{\mu x_i}$. The mean total amount of work $\mathbb{E}B_a$, that arrives during the a slots, when using a partitioning $x_1, \ldots, x_a$ satisfies the following:

$$\mathbb{E}\tilde{B}_a = \sum_{i=1}^{a} \alpha_{\mu x_i} \mathbb{E}B_{\mu x_i}.$$

The problem that has to be solved is the following:

$$\begin{aligned} &\text{Maximize } \mu \qquad \text{subject to} \\ &\sum_{i=1}^{a} \mu x_i = \mu T, \\ &\sum_{i=1}^{a} \alpha_{\mu x_i} \mathbb{E}B_{\mu x_i} < c, \\ &x_i \geq 1, \quad i = 1, \ldots, a. \end{aligned}$$

Define the following function for $x \in \mathbb{R}^+ \cup \{0\}$:

$$c(x) := \alpha_x \cdot \mathbb{E}B_x.$$

The graph of this function for $x \in \mathbb{R}^+ \cup \{0\}$ is given in figure 15.1.

As we consider the plot, the conjecture is that $c(x)$ is a convex function for all $x \in \mathbb{R}^+ \cup \{0\}$. However, this conjecture will unfortunately turn out to be false.

When the expression for $\mathbb{E}B_x$ as presented in Section 5.4 is plugged in the expression for $c(x)$, we obtain the following:

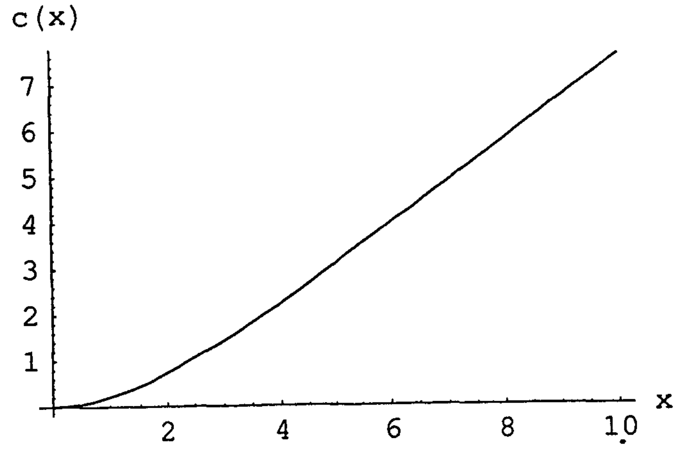

Figure 15.1: Graph of $c(x) = \alpha_x \mathbb{E} B_x$

$$
\begin{aligned}
c(x) &:= \alpha_x \cdot \mathbb{E} B_x \\
&= \alpha_x \alpha_x^{-1} e^{-x} \left(\tilde{F}(x) - (e^x - 1 - x) \right) \\
&= e^{-x} \left(\tilde{F}(x) - (e^x - 1 - x) \right).
\end{aligned}
$$

Recall that $c(x)$ is convex on $\mathbb{R}^+ \cup \{0\}$ if and only if $c''(x) \geq 0$ on $\mathbb{R}^+ \cup \{0\}$. For $c''(x)$ we have the following:

$$
\begin{aligned}
c''(x) &= e^{-x} \left(\left(\tilde{F}(x) - (e^x - 1 - x) \right) - 2 \left(\tilde{F}'(x) - (e^x - 1) \right) + \left(\tilde{F}''(x) - (e^x) \right) \right) \\
&= e^{-x} \left(\tilde{F}(x) - 2\tilde{F}'(x) + \tilde{F}''(x) + x - 1 \right).
\end{aligned}
$$

Using the expression for $\tilde{F}(x)$ as found in Section 5.4, expressions for $\tilde{F}'(x)$ as well as for $\tilde{F}''(x)$ can be derived. The results are summarized below:

$$
\begin{aligned}
\tilde{F}(x) &= \sum_{i=0}^{\infty} 3^i e^{x(1-\frac{1}{3^i})} \left(e^{\frac{x}{3^i}} - \frac{x}{3^i} - 1 \right), \\
\tilde{F}'(x) &= \tilde{F}(x) + \sum_{i=0}^{\infty} e^{x(1-\frac{1}{3^i})} \frac{x}{3^i},
\end{aligned}
$$

$$\tilde{F}''(x) \;=\; \tilde{F}'(x) + \sum_{i=0}^{\infty} e^{x(1-\frac{1}{3^i})}\left((1-\frac{1}{3^i})\frac{x}{3^i} + \frac{1}{3^i}\right).$$

When we plug these expressions in the last expression for $c''(x)$, after some manipulation the following is obtained:

$$\begin{aligned}
c''(x) &= e^{-x}\left(\sum_{i=0}^{\infty} e^{x(1-\frac{1}{3^i})}\left(-\frac{x}{3^i} + (1-\frac{1}{3^i})\frac{x}{3^i} + \frac{1}{3^i}\right) + x - 1\right) \\
&= e^{-x}\left(\sum_{i=0}^{\infty} e^{x(1-\frac{1}{3^i})}\left(-\frac{x}{3^{2i}} + \frac{1}{3^i}\right) + x - 1\right) \\
&= \sum_{i=1}^{\infty} e^{-\frac{1}{3^i}}\left(-\frac{x}{3^{2i}} + \frac{1}{3^i}\right) \\
&=
\end{aligned}$$

Observing the last expression, it is not immediately clear is this expression becomes negative for some $x \geq 0$. Therefore, we have included a plot of this expression as a function of x. This plot is included in figure 15.2.

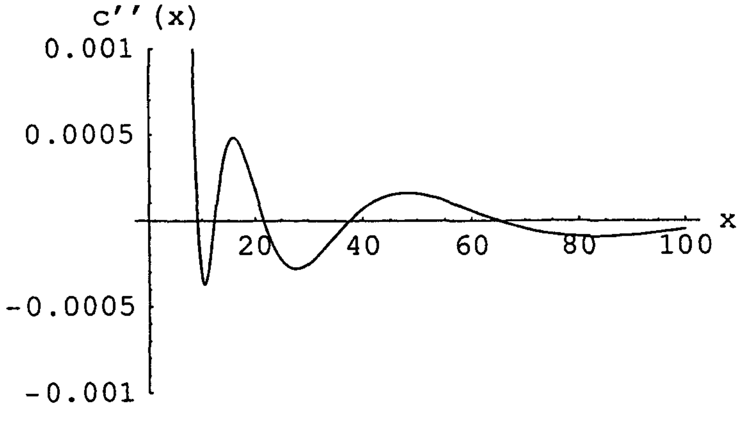

Figure 15.2: Graph of $c''(x)$ as a function of x.

As one can see in the figure, the function shows an oscillating behaviour. Furthermore, it can be concluded that $c''(x)$ is periodically negative. However, it must be remarked that $c''(x)$ becomes only slightly negative. We have not proven analytically that the function oscillates indeed, but the plot that is included is very accurate. The final conclusion is that $c(x)$ is an increasing function that is "almost" convex. Only in some areas, $c(x)$ is slightly concave. Overall, the function is almost linear. This can already be seen in the plot of $c''(x)$, where $c''(x)$ tends to zero as x increases. For completeness, we have included a plot of $c'(x)$ as a function of x, see figure 15.3.

The expression for $c'(x)$ is the following:

$$\begin{aligned} c'(x) &= e^{-x}\left(\tilde{F}'(x) - (e^x - 1) - \left(\tilde{F}(x) - (e^x - 1 - x)\right)\right) \\ &= e^{-x}\left(\sum_{i=0}^{\infty} e^{x(1-\frac{1}{3^i})}\frac{x}{3^i} - x\right). \end{aligned}$$

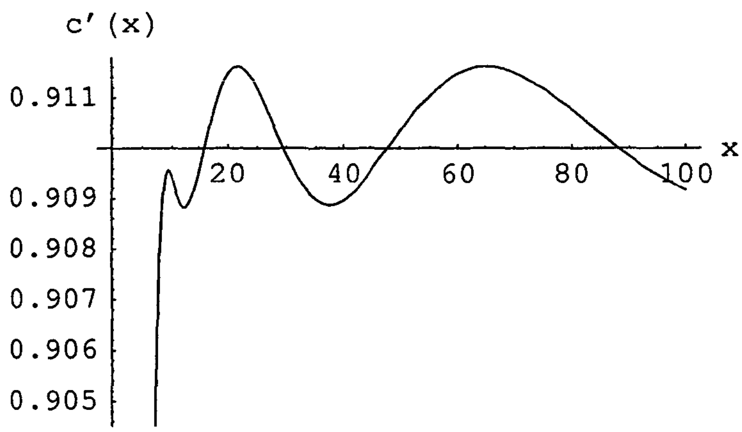

Figure 15.3: Graph of $c'(x)$ as a function of x.

In figure 15.3 we see again an oscillating behaviour with a very small amplitude. So, indeed $c'(x)$ is nearly constant as $x \to \infty$. Again, we have not proven this analytically.

Unfortunately, $c''(x)$ is not convex for $x \geq 0$. Therefore, we are formally not allowed to apply Jensen's inequality for convex functions (see [5]) to $c(x)$. Since $c''(x)$ is convex for small x and for larger x becomes almost linear, we will "cheat" a little bit here. When we nevertheless apply Jensen's inequality for convex functions to $c(x)$, this gives the following:

$$\forall x_1, \ldots, x_a \in \mathbb{R}^+ \cup \{0\} \quad : \quad a \cdot c \left(\frac{\sum_{i=1}^{a} x_i}{a} \right) \leq \sum_{i=1}^{a} c(x_i).$$

When we apply this result to the second constraint of the maximization problem, we have that:

$$\forall \mu x_1, \ldots, \mu x_a \in \mathbb{R}^+ \cup \{0\} \quad : \quad \sum_{i=1}^{a} c \left(\frac{\sum_{i=1}^{a} \mu x_i}{a} \right) \leq \sum_{i=1}^{a} c(\mu x_i).$$

This implies that μ is maximized by choosing $x_1 = \ldots = x_a = \frac{T}{a}$. Because of the fact that $T \geq a$, all the x_i's will satisfy $x_i \geq 1$. This means exactly that the arrival slots should be chosen in an equidistant manner, in order to maximize the capacity. However, since $c(x)$ is not completely convex, this conclusion is formally not true. But from a practical point of view, we assume from now on that equidistant scheduling is optimal, since this simplifies the conclusion significantly and what is more: equidistant scheduling results in nearly optimal results. This is true due to the fact that $c''(x)$ becomes only slightly negative periodically. Therefore, each equidistant schedule has only a negligibly small deviation from the corresponding optimal schedule.

The final conclusion can be that if one is interested in maximizing the capacity, then one must choose an arrival slot mechanism. It has been indicated how the corresponding optimal feasible value for s can be determined. Furthermore, the arrival slots must be scheduled in an equidistant manner.

15.2.3 Minimizing the mean sojourn time

Another objective when allocating the contention slots could be minimizing the mean sojourn time of individual customers. Given a certain mechanism, the mean sojourn time of a customer clearly depends on the actual value of μ, the arrival rate per slot. Therefore, the choice of a mechanism depends in general on the actual value of μ.

Some remarks can be made beforehand:

1. In the previous subsection it has been indicated how the capacity is maximized. This implies that for arrival rates that are close to this capacity, the mean sojourn times are minimized in the same way. The actual scheduling of the non-arrival slots was not discussed in the previous section, since that is irrelevant for the capacity of the system. For the mean sojourn time, the scheduling of these slots *is* important and therefore needs some attention.

2. When requiring equidistantly scheduled contention slots, it was shown that the dynamic arrival slot mechanism is a nearly optimal choice when minimizing the mean sojourn time for both light-traffic and heavy-traffic systems. This would imply that independent of the value of μ, one must always choose a dynamic arrival slot mechanism in this special case. Only the value of the optimal feasible value for s has to be determined.

3. The practical implementation of the dynamic arrival slot mechanism is not a problem because of the fact that the number of contention slots scheduled in the upcoming frame does *not* depend on the state of the system. Only which customers are actually allowed to compete in a certain slot depends on the state of the system, but that is the case for all contention slots.

Remark 2 is very important. It is interesting whether unequally spaced contention slots create an opportunity for mechanisms with a smaller mean sojourn time for lower regions of the arrival intensity. This question is relevant for all four types of mechanisms. Considering the first remark, for high values of the arrival intensity, an arrival slot mechanism with equidistantly scheduled arrival slots is optimal.

We now discuss the question whether unequally spaced contention slots give better sojourn time results for some range of values for μ for each of the four mechanisms separately. It is assumed that the average number of contention slots per frame, denoted by r, is fixed over time and known.

Gated mechanism

In the gated mechanism there is never more than one tree being processed at a time. Consider the case in which all the contention slots are scheduled equidistantly. This mechanism will be compared with another one, which will be described now. When a tree is not completed yet, it is advantageous to finish it as soon as possible. That means that the contention slots should be scheduled as close to each other as possible, incorporating the minimum amount of time that must be in between two consecutive contention slots. When the contention slots dedicated to the current tree are scheduled closely after each other and a new tree is started at the same time as it would have started in the equidistant mechanism, this modified mechanism certainly outperforms the equidistant mechanism. There is only the problem that at the beginning of a frame, the Head End does not know (and no one else either), whether the tree will finish in the upcoming frame, and therefore it may happen that a contention slot will be scheduled that is not going to be used for the current tree anymore.

A simulation study will give insight in performance measures such as mean waiting time and mean sojourn time for this modified mechanism. Observe that capacity does not alter in this modified mechanism. Section 15.3 will be devoted to simulation results.

Non-blocked mechanism

For the non-blocked mechanism, simulation results presented earlier in this report indicated that for both capacity and mean sojourn time it was optimal to schedule the slots equidistantly. But in that simulation study, equidistantly scheduling was compared with periodically unequidistantly scheduling. An alternative scheduling approach that not has been analyzed yet, is the following.

When *during* a tree the slots are scheduled equidistantly and close to each other, but *after* the completion of a tree a relatively long period with no contention slots is scheduled, this might be better than simply scheduling all contention slots at the same distance from each other. Of course, the average number of contention slots per frame must be kept constant in order to make a fair comparison. An interesting question is which (constant) distance between contention slots "belonging" to the same tree is optimal with respect to some performance measure. A simulation study might give insight in this matter. The following approach will be used.

The distance d (measured in slots) between to consecutive contention slots that are dedicated to the same tree, is kept constant within a certain mechanism. The period directly after the finishing of a tree will dynamically vary over time, in order to keep the average number of contention slots per frame constant. Mechanisms with different values of d but the same average number of contention slots per frame can be compared by means of simulation. The comparison can be made on several earlier mentioned performance measures, including capacity. It is probably not true that capacity remains constant when d is varied. The simulation results are discussed in section 15.3.

Static arrival slot mechanism
For the static arrival slot mechanism, we have seen that for every value of μ equidistant arrival slots (almost) minimize the mean total amount of work that must be completed in the remaining contention slots (excluding the arrival slots). Reasoning in this way, one can say that in order to minimize the mean second component of the waiting time of an individual customer, equidistant slots are also optimal.

In order to minimize the first component of the waiting time it is again optimal to choose the arrival slot in an equidistant manner, due to the fact that variation in the time between two consecutive arrival slots will lead to larger mean waiting times, due to an effect that is known as a time biasing effect.

Because of the fact that the mean individual service time is also approximately linear in the arrival rate (experienced in a certain arrival slot), it also seems optimal to choose the arrival slots in an equidistant manner in order to minimize the mean service time, again due to time biasing effects.

Concluding, one can say that the arrival slots must be chosen equidistantly in order to optimize the model. For the other contention slots, it is optimal (with respect to mean sojourn time) to choose them as close to the previous arrival slot as possible, incorporating the minimal distance that is required due to the propagation delay that takes place.

Some more words can be spent on the scheduling of the non-arrival contention slots. Consider for instance the mechanism with $s = 2$. This mechanism implies a ratio of 1 arrival slot per 3 contention slots. However, this does not imply that the scheduling must take place in a periodic manner. So it is not necessary to alternate between one arrival slot and two non-arrival contention slots. In fact, as long as there is "work" in the system, it is optimal to schedule non-arrival contention slots uninterruptedly at the required minimum distance. Obviously, this may interfere with the scheduling of the arrival slots in the sense that the minimum distance is violated. However, the minimum distance between two consecutive contention slots is only needed (from a practical point of view) for the feedback of HE to arrive at the collided customers. When it is absolutely sure that collided customers in a certain slot are not going to compete in the next contention slot, the distance between those two slots can be less than the required minimum distance.

Obviously, scheduling consecutive non-arrival contention slots at the minimum distance implies a frequent use of contention slots per time unit. In fact, this will normally be above the required average number of contention slot per time unit that was externally given. However, if the system is stable, there will be times that the system is empty. During these times, no contention slots are needed and in the long run, the required average number of contention slots per slot will be satisfied. In fact, the average number of effectively used contention slots per frame will even be a bit lower. This gives opportunities to slightly better performing mechanisms.

One possible improvement can be made by scheduling the arrival slots a little closer to each other. This results in a slightly higher average number of contention slots per frame. That this modification leads indeed to an improvement can be proven as follows.

Consider the original situation with a distance x_0 between two consecutive arrival slots. This situation will be compared with the situation with a distance $\frac{x_0}{(1+\beta)}$, where $\beta > 0$. The arrival rate of customers per slot equals μ. The mean total amount of work that arrives per slot in the first situation equals $\frac{\alpha_{x_0\mu}\mathbf{E}B_{x_0\mu}}{x_0} = \frac{c(x_0\mu)}{x_0}$. In the second situation, this amount of work per slot equals $\frac{c(\frac{x_0\mu}{(1+\beta)})}{\frac{x_0}{(1+\beta)}} = (1+\beta)\frac{c(\frac{x_0\mu}{(1+\beta)})}{x_0}$. We will show that $d(x) := \frac{c(x)}{x}$ is an increasing function of x for $x > 0$. This implies that the mean total amount of work per slot in the second situation is less than that in the first situation.

$$d'(x) = \frac{c'(x)x - c(x)}{x^2}.$$

Since the denominator is positive for $x > 0$, we are left to prove that the numerator is positive as well for $x > 0$. Clearly, we have that $c'(0) \cdot 0 - c(0) = 0$. The derivative of the denominator with respect to x equals $c''(x)x + c'(x) - c'(x) = xc''(x)$. We have already proven in subsection 15.2.2 that for $x > 0$, it holds that $c''(x) > 0$. So $xc''(x) > 0$ for $x > 0$.

Another possible improvement that makes use of the fact that the average number of effectively used contention slots per time unit is less than the prescribed maximum will be discussed in combination with the dynamic arrival slot mechanism.

Observe that the implementation of the *static* arrival slot mechanism as just described is in fact *dynamic* since it keeps track of the state of the system just as in the dynamic arrival slot mechanism. However, the dynamic scheduling takes only place with respect to the non-arrival contention slots. This is a main difference with the dynamic arrival slot mechanism.

Dynamic arrival slot mechanism

We will now discuss the dynamic arrival slot mechanism and how this mechanism can be (statically) implemented in the setting with data slots in between the contention slots. This mechanism is considered more closely, because of the fact that in the old setting it outperformed, at least according to simulation results, the other mechanisms for *all* values of μ, i.e in case of both light and heavy traffic.

As indicated already, the static arrival slot mechanism can be implemented in the setting with data slots in between. It is the question whether the dynamic scheduling of the arrival slots "adds" something in the sense that it contributes significantly to smaller mean sojourn times or variances. Recall that the average number of contention slots per frame r is determined externally and that it is known. In case that the system is non-empty, the mechanism behaves by definition exactly the same as the static arrival slot mechanism. It has already been outlined how the optimal scheduling in that case should take place. However, when the system becomes empty, the scheduling of the next arrival slot might be done dynamically. Obviously, in order to minimize waiting times, this slot has to be scheduled very soon. But we also face the condition of a given average number of contention slots per frame. This condition might be violated if the new arrival slot is scheduled too close to the last contention slot. This violation can be corrected by scheduling the non-arrival contention slots slightly more apart from each other in order to keep the average number of contention slots per frame small enough. But how do sojourn times react to this scheduling?

So the scheduling rules in this mechanism become more difficult. As we have indicated already, in the implementation of the static arrival slot mechanism that has just been described, the average number of effectively used contention slots per frame is slightly less than the prescribed maximum, at least for a stable system. One modified mechanism which made use of that has already been described. Another possible modification of the implementation of the static arrival slot mechanism would be the following. Each time that the system becomes empty, the next arrival slot is scheduled a number of slots earlier in order to shorten the waiting time of the new customers that will arrive. The characteristics of the system determine how close such an arrival slot can be scheduled after the slot in which the system became empty.

This last modification shows both a dynamic scheduling of the non-arrival contention slots and the arrival contention slots. When we compare the two proposed modifications, it is not clear which of the two leads to the smallest individual mean sojourn time. This will certainly depend on the characteristics of the network. In heavy-traffic systems, both modifications will not differ that much, leading to approximately the same mean sojourn times. We have not studied the performance of these modified mechanisms in detail.

A last remark is that the differences in performance between all those modifications of the arrival slot mechanisms are probably very small compared with the differences that occur between the three "basic" mechanisms: the gated, non-blocked and arrival slot mechanism.

15.3 Simulation results

15.3.1 Introduction

In this section, several simulation results will be presented. The main purpose of this section is to study the impact of a required minimum distance between two consecutive contention slots on the performance of the various contention resolution mechanisms. In the previous section, it has been outlined in which way the gated and the static arrival slot mechanism should be optimally implemented when the minimal distance d and the average number of contention slots r per frame are known. For the non-blocked mechanism, it has been indicated that it is not clear beforehand what the optimal scheduling looks like for given values of d and r.

We have chosen to do a simulation study instead of an analytic approach for two reasons. The first reason is that for the non-blocked mechanism, it seems quite difficult to obtain analytical results. For the gated mechanism and the non-blocked mechanism, it seems possible to adjust the analysis that has been done on the "basic" mechanisms to obtain results for these adjusted mechanisms. However, the analysis for the gated mechanism was not exact, but only an approximation. The second reason was time. The time that was left, was thought to be too short to investigate it thoroughly by means of analysis.

So we will present simulation results on the mean individual sojourn time in the three mentioned mechanisms, where the arrival slot mechanism with $s = 2$ is chosen as a representative for the class of static arrival slot mechanisms. Some additional remarks on the simulation of this mechanism will be made shortly. These results will be presented in subsection 15.3.2. Furthermore, in subsection 15.3.3 simulation results with respect to the *variance* of the individual sojourn time are presented.

Given the value of the arrival intensity μ, the simulation results have been obtained for a range of values for d. The average number of contention slots per frame is kept fixed through all simulations and is equal for all three mechanisms. Several arrival intensities have been considered. For each arrival rate the (estimated) mean sojourn time and the (estimated) variance of the sojourn time can be plotted as a function of d. This leads to a graphical comparison between the three mechanisms. Furthermore, the performance of the non-blocked mechanism can be inspected on its own.

We have chosen for a frame length of 18 slots. This corresponds to a realistic access network. We have chosen to fix the average number of contention slots per frame at exactly 1 slot per frame. The mimimum distances that are considered are $d = 1, 2, 3, 6, 9, 12, 15, 18$. Observe that the case in which $d = 18$ corresponds to the originally analyzed model. Three different arrival intensities, given in customers *per frame*, are studied: $\mu = 0.75$, $\mu = 1.09$ and $\mu = 1.15$. The last intensity is above the capacity of the gated mechanism. Therefore this mechanism is not simulated for $\mu = 1.15$. In all graphs that are presented, the simulation values are marked and a 95%-confidence interval is graphically presented by means of two lines that form a sort of confidence region. To represent the mean sojourn time results, two graphs are given for each intensity. Each second graph is obtained out of the first by zooming in on the region in which some mechanisms are "close" to each other. In case of the variance of the sojourn time, sometimes only one graph is given.

The arrival slot mechanism with $s = 2$ and a minimum distance d is implemented (and simulated accordingly) in two essentially different ways. These two different implementations will be discussed now briefly.

1. The first implementation assumes an alternation between one arrival slot and two non-arrival contention slots. These last two slots are scheduled at a distance d. The distance between the last of these two contention slots and the next arrival slot will typically be at least 18 slots, so equal to or larger than the minimum distance.

2. The second implementation schedules as many non-arrival contention slots as possible in between two consecutive arrival slots, all at a distance d apart from each other. The first of these slots is scheduled d slots apart from the first arrival slot. As soon as an arrival slot has taken place, this scheduling takes place again, just as after the previous arrival slot. Obviously, when there is no work in the system anymore, scheduling these non-arrival contention slots does not take place, at least until the next arrival slot has taken place.

It is clear that the second implementation leads to smaller mean sojourn times than the first implementation. Furthermore, the second implementation makes a more fair comparison between arrival slot mechanism and gated and non-blocked mechanism possible, since the first implementation clearly does not take as much advantage of the possibility of scheduling contention slots close to each other as the second implementation. In all following graphs, the results of both implementations will be presented. In the discussion and comparison, the second implementation will be considered only.

15.3.2 Results on the mean sojourn time

We will now present the first two graphs on the mean sojourn time, which correspond to $\mu = 0.75$.

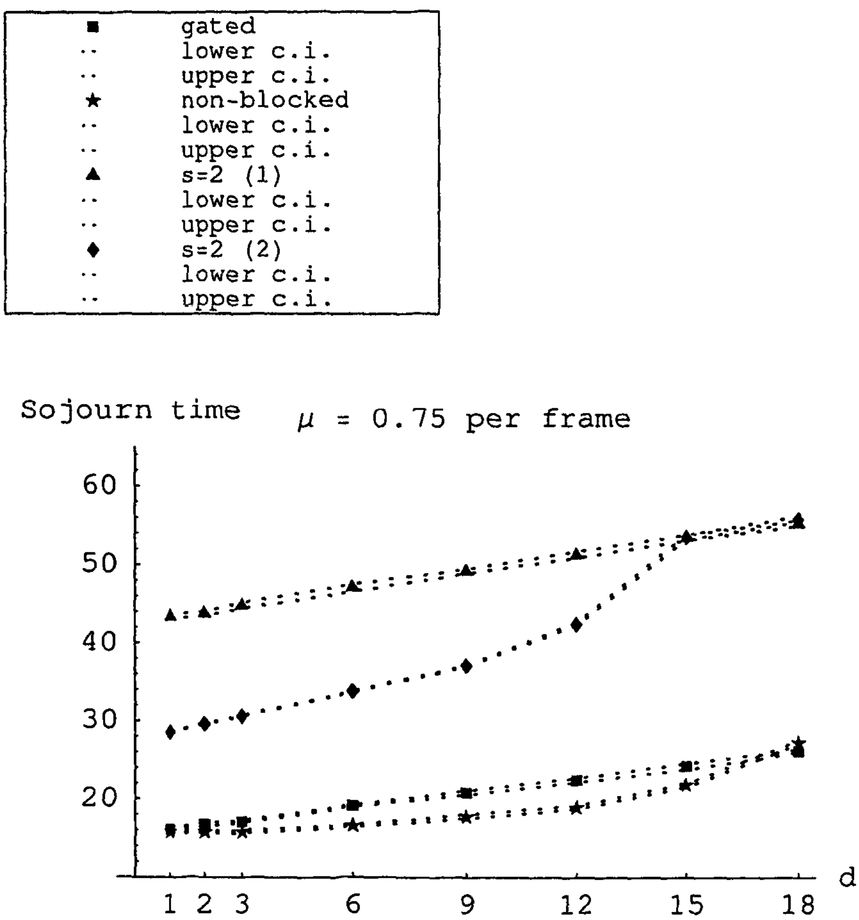

Figure 15.4: Mean sojourn times as a function of the minimum distance between two contention slots.

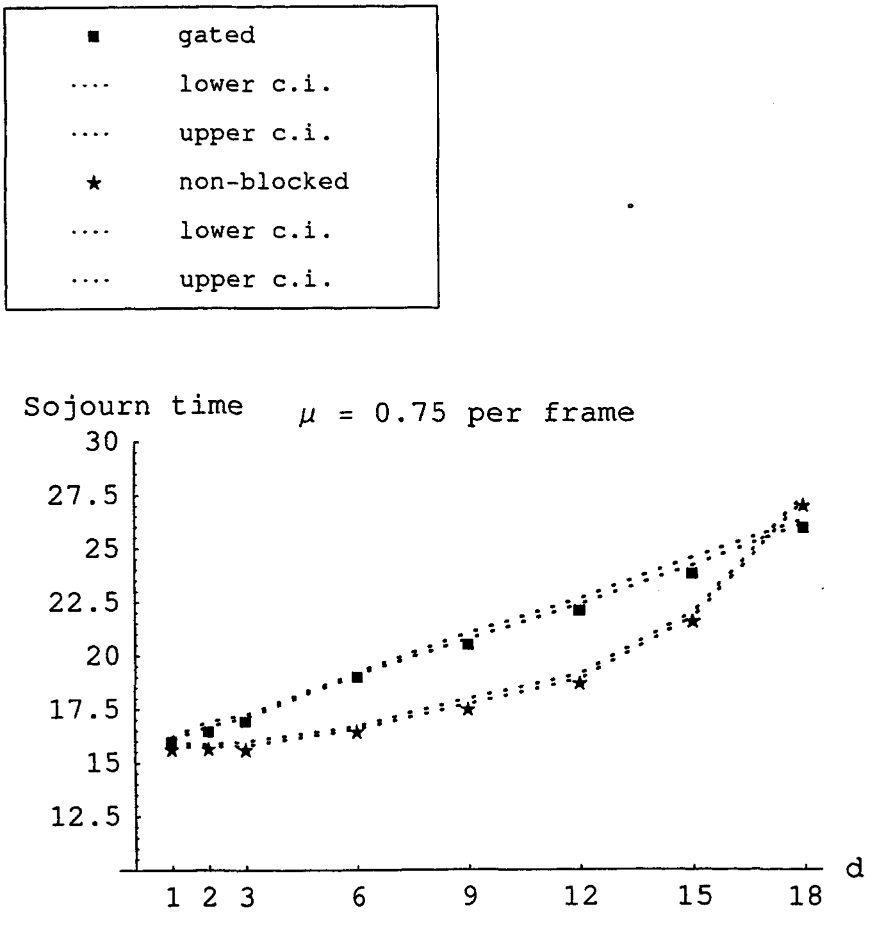

Figure 15.5: Mean sojourn times as a function of the minimum distance between two contention slots.

Inspecting the first figure, 15.4, we can conclude that the static arrival slot mechanism has a bad performance compared with the other two mechanisms over the whole range of d. The second graph shows in more detail that the non-blocked mechanism is superior to the gated mechanism for all investigated values of d, except $d = 18$. Furthermore, we see that the mean sojourn time in the non-blocked mechanism is an increasing function of the minimum distance d for $d \geq 2$, at least for this arrival rate. However, it is not exactly clear if the function is increasing over the whole range of values for d. It seems to be the case that $d = 2$ shows a slightly better performance than $d = 1$, but due to simulation uncertainty, it is not sure if this is truly the case.

The next two figures show the performance (with respect to the mean sojourn time) of the three mechanisms for $\mu = 1.09$. In figure 15.6 we see that for this arrival rate, the gated mechanism is significantly worse than the other two, which can be explained by the fact that $\mu = 1.09$ is close to the capacity of the gated mechanism. The other two mechanisms are "close to each other" over the whole range of d. Figure 15.7 gives more insight in this. Here we see that mean sojourn time in the arrival slot mechanism is an increasing function of d, which is intuitively clear. The mean sojourn time in the non-blocked mechanism however, is first *decreasing* until $d = 12$ and for $d \geq 12$ it is increasing. This phenomenon can be explained by considering the two extreme cases $d = 1$ and $d = 18$. The latter has the disadvantage that the distance between two slots is large, resulting in a slow processing of the tree and consequently resulting in large service times. The mechanism with $d = 1$ does not have this disadvantage at all, but due to the fact that slots that belong to the same tree are scheduled so close to each other, there elapses a relatively long time between the end of a tree and the beginning of a new tree. This results in a long mean waiting time for the customers that are present in this initial slot of a tree. Furthermore this initial number will be relatively large. As we have indicated several times, small trees are advantageous with respect to mean service time. Concluding, we can argue that it might be the case that there is some intermediate value of d which is the optimal trade off between advantages and disadvantages of the two extreme cases that are just described. As we have seen, this turns out to be the case.

Comparing the arrival slot mechanism with the non-blocked mechanism, we see that for large values of d the non-blocked mechanism slightly outperforms the static mechanism. However for small values of d the arrival slot mechanism clearly outperforms the non-blocked mechanism. Furthermore, we see a huge difference in performance between the two implementations of the arrival slot mechanism for small values of d.

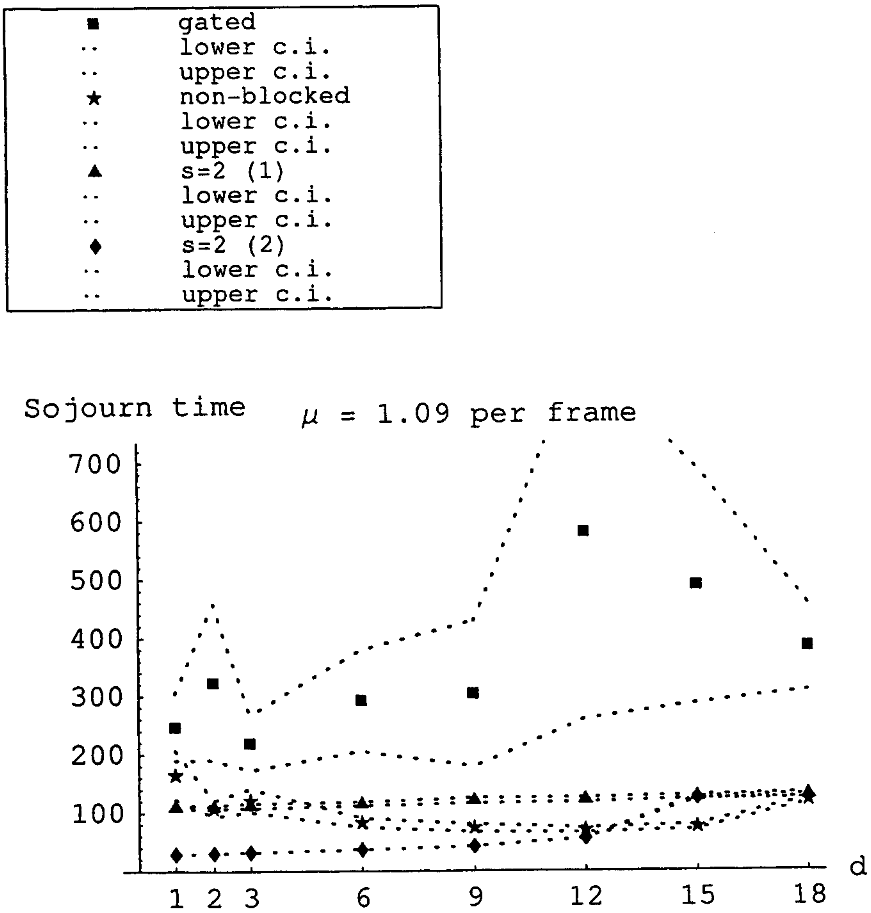

Figure 15.6: Mean sojourn times as a function of the minimum distance between two contention slots.

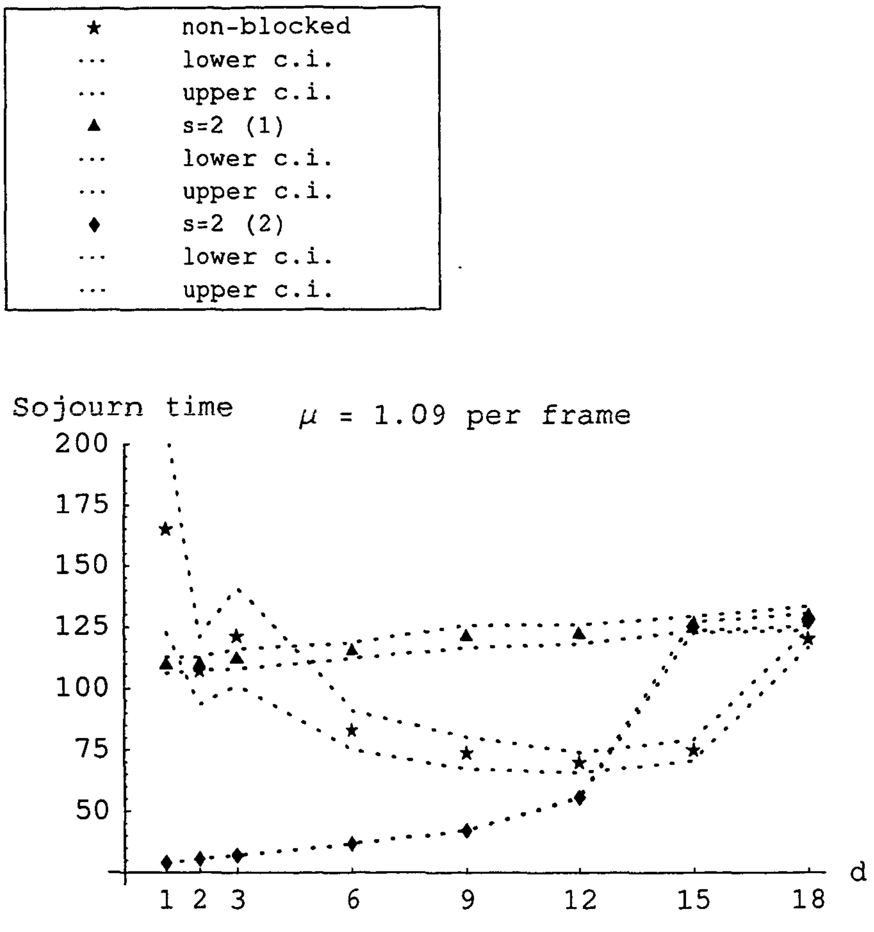

Figure 15.7: Mean sojourn times as a function of the minimum distance between two contention slots.

The next two graphs show the performance in case of an arrival rate μ equal to 1.15. Figure 15.8 shows a huge difference in performance between the non-blocked and the arrival slot mechanism for small values of d. Figure 15.9 shows an interesting picture. We see for this arrival intensity, $d = 15$ is the "optimal distance" for the non-blocked mechanism. For this value of d the performance is even slightly better than that of the arrival slot mechanism. For $d = 18$ however, the arrival slot mechanism is again better than the non-blocked mechanism. Recall that $d = 18$ corresponds to the originally analyzed model, since a minimum distance d equal to 18 and an average number of contention slots per frame (of length 18) r equal to 1 corresponds to equidistant contention slots. Therefore, for intensities that exceed 1.15, the arrival slot mechanism will outperform the non-blocked mechanism for $d = 18$. Together with the information in this graph, it is probably the case that for most of these intensities the outperformance is over the whole range of d.

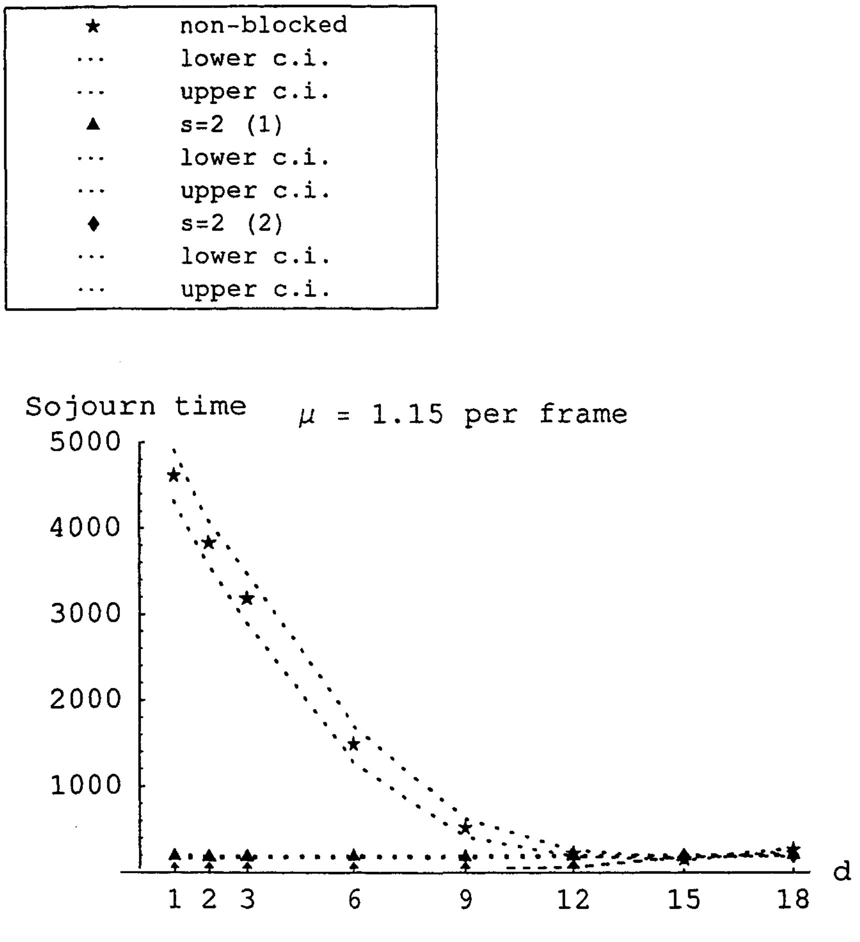

Figure 15.8: Mean sojourn times as a function of the minimum distance between two contention slots.

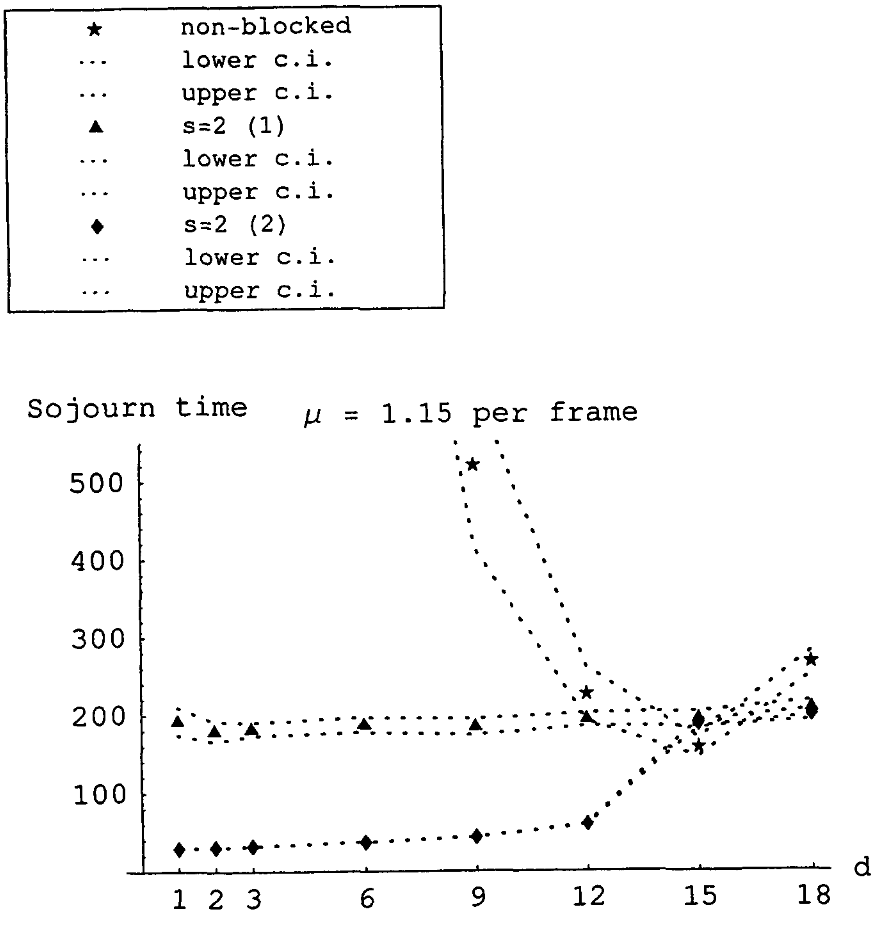

Figure 15.9: Mean sojourn times as a function of the minimum distance between two contention slots.

For each combination of arrival rate and minimum distance d, one of the three mechanisms is the "optimal choice" with respect to minimizing the mean sojourn time. So we have a function that gives for each pair (d, μ) the optimal mechanism. If we represent the gated, non-blocked and arrival slot mechanism by $0, 1$ and 2, respectively, we can summarize the simulation results by means of a three-dimensional plot which shows an array of function values. We will base our final recommendations on this plot. An important remark is that only integer values for the mechanisms make sense. To visualize the structure, all integer points are connected by means of planes.

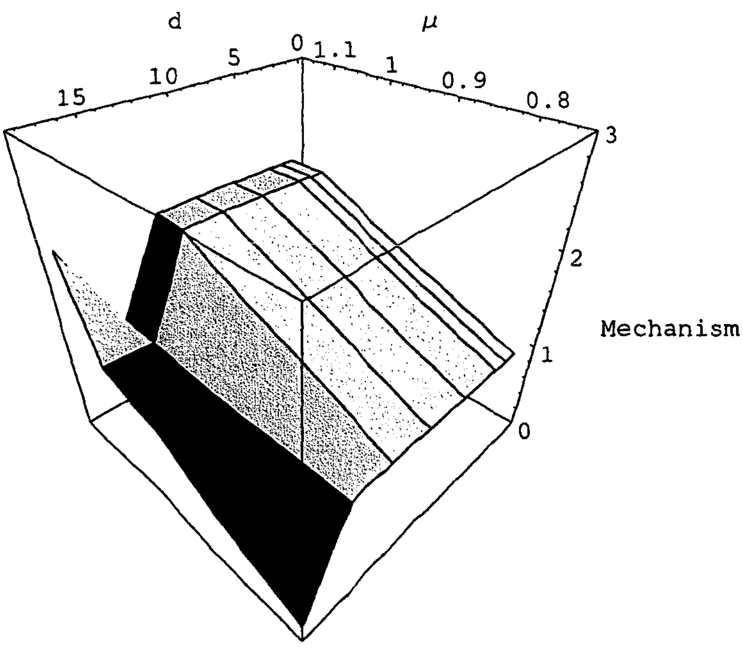

Figure 15.10: Optimal mechanism as a function of μ and d.

15.3.3 Results on the variance of the sojourn time

The next four graphs represent results on the variance of the sojourn time for the various mechanisms. Figure 15.11 shows that for an arrival rate $\mu = 0.75$, the gated mechanism shows the best overall performance. For small values of d, the non-blocked and arrival slot mechanism (with the second implementation) perform slightly better. Only for large values of d we see a significant outperformance by the gated mechanism.

The second graph, given in figure 15.12, shows huge differences in the variances of the mechanisms. The gated mechanism has such a large variance for all d that the corresponding graph is in fact not present in the figure. The non-blocked mechanism shows a minimum variance in case of $d = 9$. For small and large values of d, the variance is much larger than this value. The both arrival slot implementations however, show a significantly better performance. The variance in case of the second implementation is in fact negligible compared with the variances in the other mechanisms. Only for large d the variance in this mechanism rapidly increases to the level of the variance in the first implementation. This is completely logical since for these values of d, both implementations result in the same mechanism.

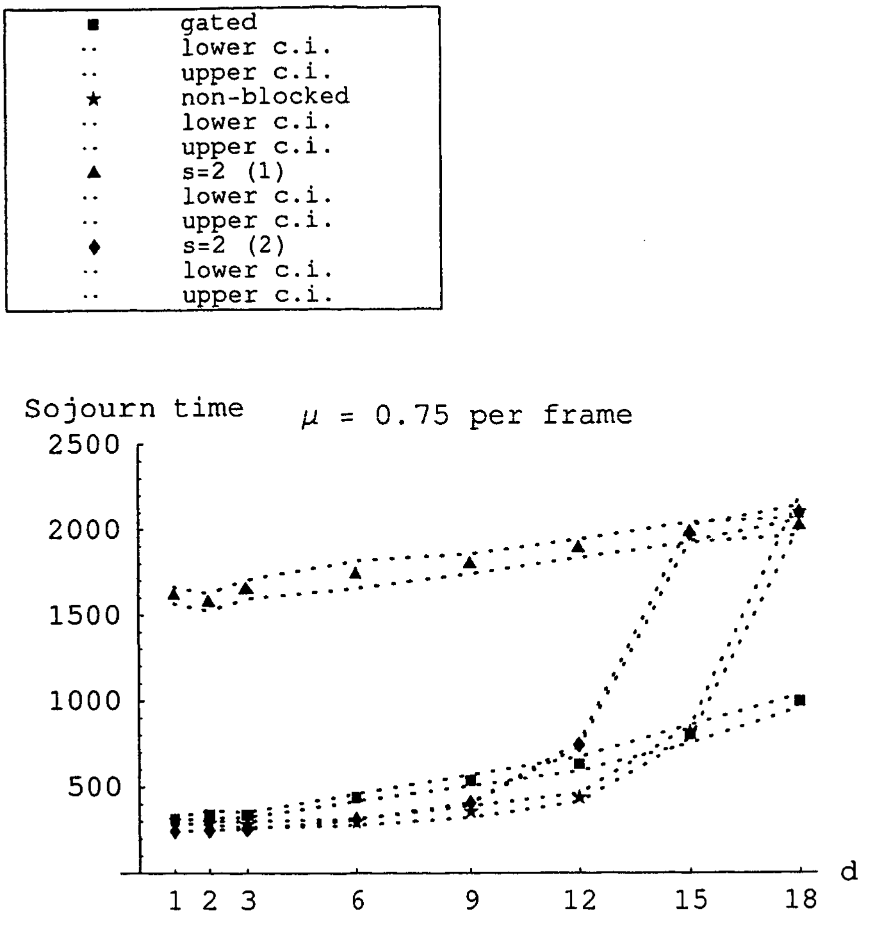

Figure 15.11: Variance of the sojourn times as a function of the minimum distance between two contention slots.

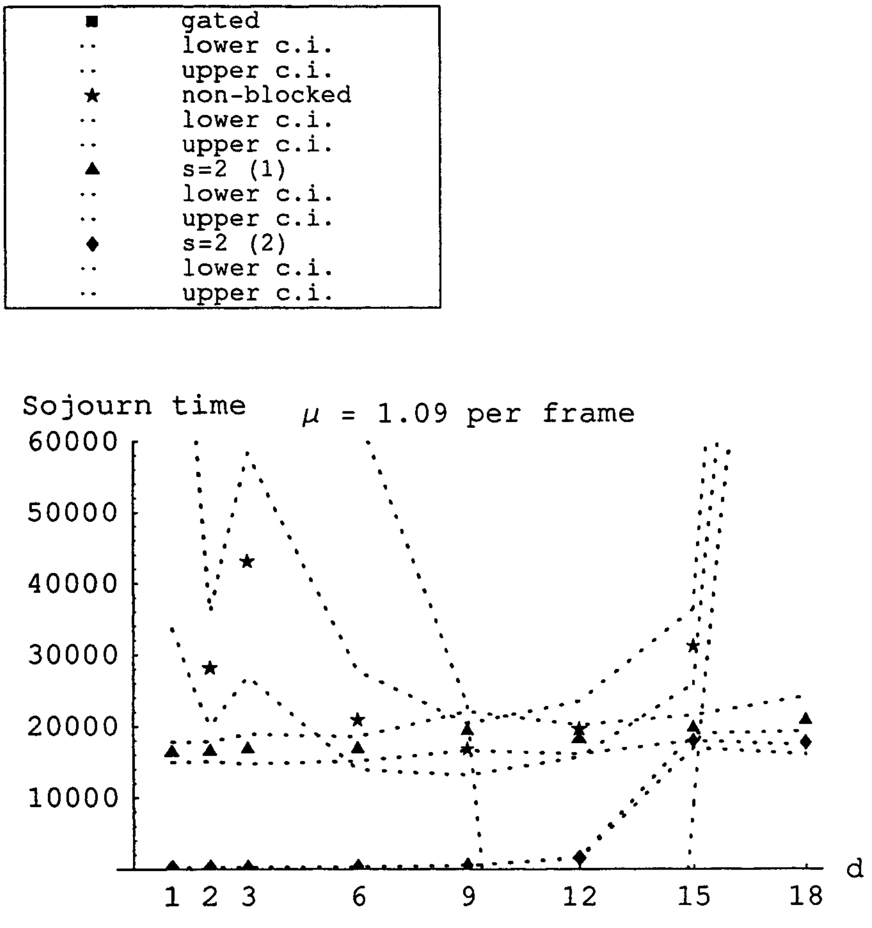

Figure 15.12: Variance of the sojourn times as a function of the minimum distance between two contention slots.

The next two graphs are the final two of this section. They present the results in case of a high arrival intensity, $\mu = 1.15$, which is above the capacity of the gated mechanism. The first graph, depicted in figure 15.13, shows that for this arrival rate the variance in the non-blocked mechanism is again highly dependent on the distance d. The variance is minimized for $d = 15$. For this value of d in fact, the mean sojourn time is minimized as well as we have seen before. It is only for this value of d that the variance in the non-blocked mechanism approximates the variance in case of both arrival slot implementations. For other values there is again a huge difference in performance. And we can argue that as the intensity is further increased, this outperformance becomes only bigger.

In figure 15.14, we have zoomed in on the two arrival slot implementations only. This shows again the huge difference in variance between the second implementation and the first one, where the latter is again outperforming the non-blocked mechanism over the whole range of values for d.

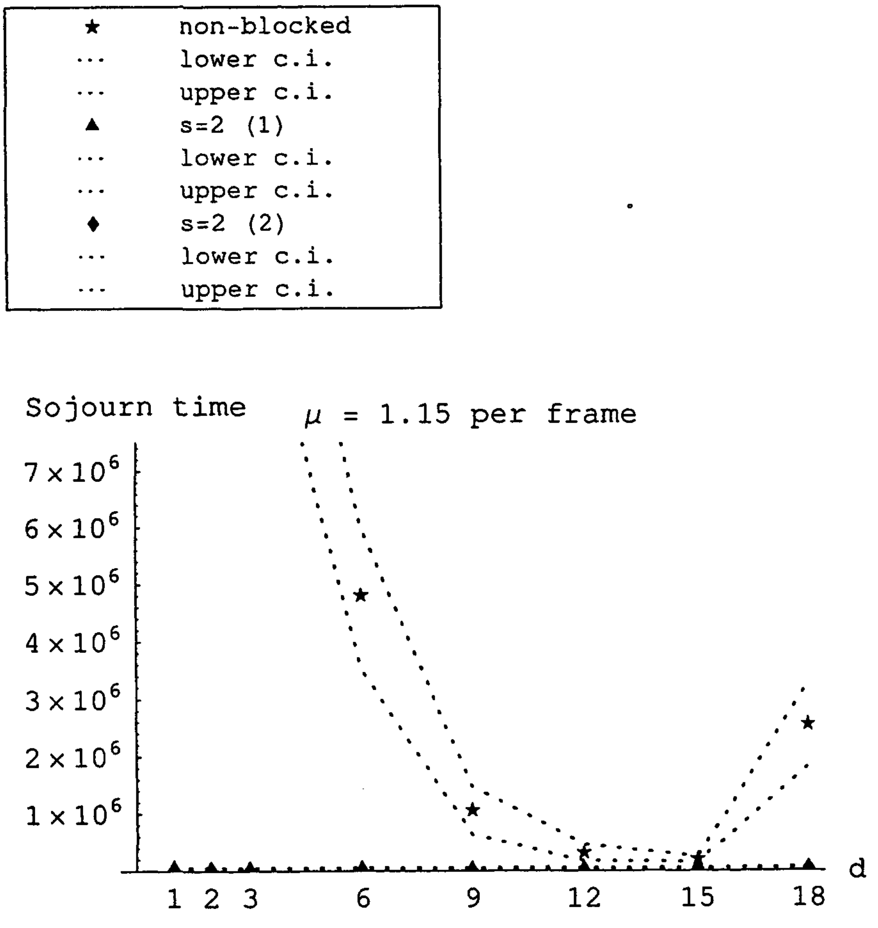

Figure 15.13: Variance of the sojourn times as a function of the minimum distance between two contention slots.

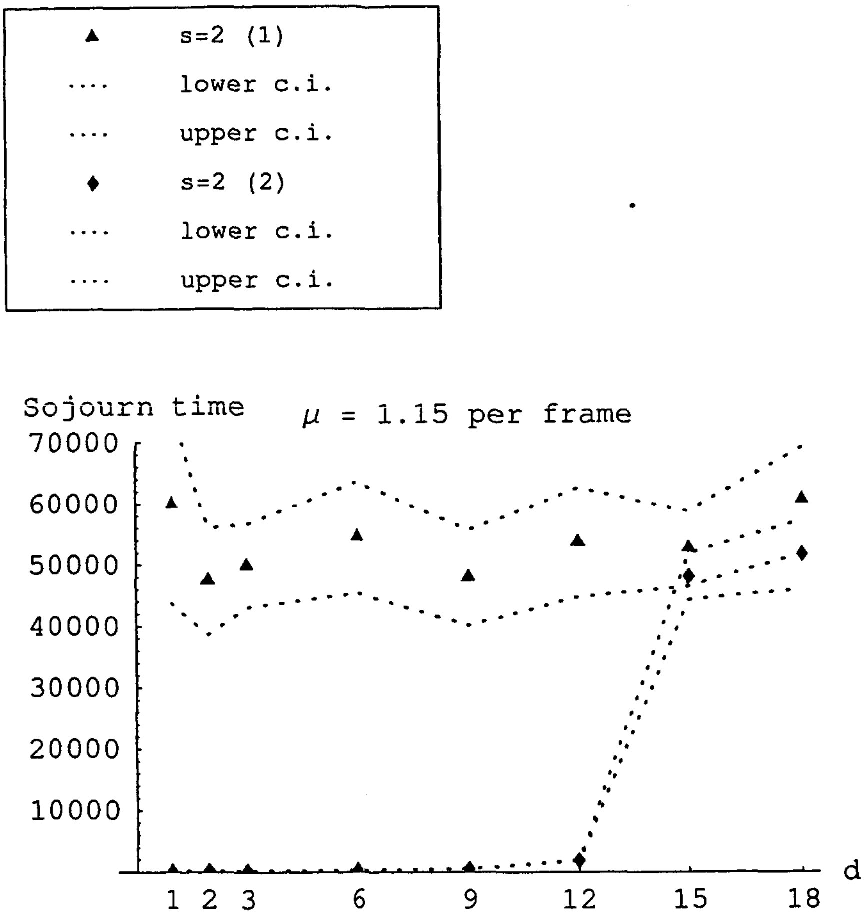

Figure 15.14: Variance of the sojourn times as a function of the minimum distance between two contention slots.

15.4 Conclusions

15.4.1 Introduction

After a thorough study, both by means of analysis and simulation, concluding remarks and recommendations can be made. This will be the subject of this final section.

One of our main purposes was to investigate whether and under what conditions a contention resolution mechanism based on the arrival slot mechanism outperforms the two existing contention resolution mechanisms, known as the gated and the non-blocked mechanism. To reach this final point of comparison, we have first studied the various mechanisms in a simplified context. The simplifications with respect to the "real world" made it possible to do most of the study by means of analysis. The results in this simplified mathematical model turned out to be promising in favour of the arrival slot mechanisms. Subsequently, we have adjusted the simplified mathematical model so that it became more realistic again. These adjustments gave rise to questions concerning the (optimal) implementation of all contention resolution mechanisms. Eventually we have investigated the (optimal) implementation and performance of the various contention resolution mechanisms in this more realistic context.

We will now summarize the results that we have found and discuss the implications of these results, leading to some final recommendations. The focus will be on the results that we have found in the model in which the data slots are incorporated. One subsection is devoted to the drawbacks of the models that we have analyzed. Furthermore, the practical relevance of all obtained results will be indicated. This relevance is present at two levels. First, we observe the possibilities and restrictions of a "real" access network and see how far the recommendations that we have given on the contention resolution stage can be realized in a real access network. Second, we will dwell on whether the outperformance of the newly implemented contention resolution mechanisms, that was clearly present in the mathematical models, is still present after the implementation. Consequently, we will try to answer the question if it is worth implementing such a new procedure.

We have tried to split up the discussion of these topics over the next subsections. However, some of the conclusions cannot be seen apart from for instance crucial model assumptions or practical restrictions.

15.4.2 Model shortcomings

A common aspect of all models that we have investigated is the assumption of a Poisson arrival process of individual customers. This arrival process was assumed to have a constant rate μ. The main reason for this choice was twofold. We thought that it was not too unrealistic to model the arrival process as a Poisson process and the analysis of the model became tractable due to this model assumption. However, one can place some doubts by the first motivation. In Section 12.7, the effect of different arrival processes in case of a gated mechanism was studied.

In a real access network, the number of customers is obviously finite. These customers generate data, all on an individual basis, according to some process which may vary over the individual customers as well. As soon as a customer has generated some data, it tries to send a request. The contention resolution phase is started. However, once an individual customer is involved in a contention resolution procedure, it can update its already involved request each time that it generates new data at its own network terminal. So this does not result in a new request arrival. In our models we have treated the requests as individual customers. When we assume for instance that the data generating process for each network terminal takes place according to a Poisson process with some constant intensity, which is in fact a commonly used (but questionable) approximation of reality, how does the arrival process of requests, which is a derived process of the data generating process, in the several contention resolution models look like? This is a difficult question. We will briefly provide some intuition. You can think of each Poisson data arrival as having a certain probability that it is also counted as a request arrival. This probability will not be constant over time. Accumulating this over all customers that are present in the network results in a non-homogeneous Poisson process. For a large number of customers, approximating this non-homogeneous process by a homogeneous Poisson process does not seem too bad. Another way of looking at it is without considering the underlying data generating processes in detail. It does make sense to assume that a certain individual wants to send data at moments that are generated according to a Poisson process or at least a process what looks like it. Of course, the actual amount of data may vary significantly per request. Accumulating these processes of all individuals will result in a process that can be approximated reasonably well by means of a (homogeneous) Poisson process.

Another drawback of all models that we have considered is that in fact all models isolate the contention resolution phase from the actual data transmission phase. In a real access network, a customer that has been involved in the contention resolution phase for a long time is likely to have a huge amount of data to be transmitted. It might be important for the overal performance of the access network to dynamically vary the number of contention slots and data slots in a frame according to the actual amount of data that is waiting to be transmitted. In analyzing and optimizing the contention resolution only, we forget about the second stage. In general, the optimal strategy for the contention resolution phase in isolation will not be optimal for the whole access network performance. However, analyzing and optimizing the two stages together is much more difficult and desires more knowledge about the data generating processes and their relation to stability of the network. Furthermore, it is plausible that the optimal contention resolution mechanism when analyzed in isolation shows a good performance in a model with data transmission included as well.

15.4.3 Performance comparison and recommendations

In this subsection, we will formulate several conclusions and recommendations based on the performance comparison of the various contention resolution mechanisms.

In this report, we have introduced several performance measures. For the final conclusions, we will mainly restrict ourselves to the mean sojourn time. It turns out that considering this performance measure leads to almost the same conclusions as when considering the variance of the sojourn time. Furthermore, when considering the mean sojourn time for different arrival rates, the capacity of the various systems is taken into account as well, since arrival rates that are above the capacity of a mechanism result in infinite mean sojourn times.

When we consider the mean sojourn time as the performance measure and try to give an answer to the question which mechanism leads to the best performance and how this mechanism must be implemented in a certain access network, two additional things are important to know. First, it is important to know what the minimal distance is that must be present between two consecutive contention slots. This distance is closely related to the so-called propagation delay that was introduced at the beginning of this report. Furthermore, it is important to have a good knowledge of the (average) arrival rate (per frame) of requests in the network that is considered.

To avoid an unnecessarily long discussion, we will summarize the general recommendations in table 15.1, where we consider three arrival rates which can be thought of as representing a network with medium, heavy and very heavy traffic, respectively. Furthermore, we consider three different minimum distances. For each of the resulting nine combinations, it is indicated which of the three considered mechanisms is "optimal" with respect to the mean sojourn time. This table is based on the numerical results that were summarized in figure 15.10 in subsection 15.3.2.

The three mechanisms that we have compared are the following:

1. The gated mechanism.
2. The non-blocked mechanism.
3. The *second* implementation of the static arrival slot mechanism with $s = 2$.

Of course, these three mechanisms do not cover the set of all mechanisms that are optimal for a certain combination of arrival rate and minimum distance. Two important remarks can be made on this issue.

Among all arrival slot mechanism with integer s, the mechanism with $s = 2$ shows the best performance if the arrival rate is not too small. However, it has already been indicated that the arrival slot mechanism with $s \approx 1.8$ is even slightly better for these arrival rates. For ease of implementation, we have considered $s = 2$. This does not affect our final conclusions.

Another remark is that we have not considered the dynamic arrival slot mechanisms in our comparison. We have indicated in section 15.2.3 how such a mechanism could be implemented in the context with data slots present. As said before, we have not studied in detail under what conditions such an implementation of the dynamic arrival slot mechanism is superior to a modified implementation of the static arrival slot mechanism. However, our conjecture is that the differences between all modifications of the arrival slot mechanisms are negligibly small when compared with the differences between the three considered mechanisms. Therefore, the conclusions on when to implement which of the three basic mechanisms are probably not affected by this simplification.
The recommendations that are given in table 15.1, can be seen as general guidelines. For heavy and very heavy traffic, the arrival slot mechanism is the optimal choice, due to the superior capacity of the mechanism. In case

Arrival rate	$d = 1$	$d = 9$	$d = 18$
Medium	Non-blocked	Non-blocked	Gated
Heavy	Arrival slot	Arrival slot	Non-blocked
Very heavy	Arrival slot	Arrival slot	Arrival slot

Table 15.1: Optimal mechanism with respect to mean sojourn time for different network characteristics.

of non-heavy traffic, this mechanism is not the best choice. However, as we argued before in subsection 15.2.3, a dynamic scheduling of the arrival slots will probably improve the performance of the arrival slot mechanism. This will in particular be the case in light traffic situations. In light traffic, the system "empties" quite often and hence a dynamic scheduling of arrival slots, i.e. an advanced scheduling of an arrival slot in case of an empty system, will lead to significantly smaller waiting times.

15.4.4 Practical relevance

So far, we have discussed the results that were obtained in models that were supposed to *resemble* a realistic access network. Recommendations for the implementation of variants of the arrival slot mechanism have been made. Furthermore, we have indicated in which way the various mechanisms, including the gated and non-blocked mechanism, should be implemented in a real access network. In this subsection, we will briefly discuss the practical relevance of such implementations by considering the advantages and disadvantages in a real access network. The main discussion point is whether implementing an arrival slot mechanism is worthwhile from an overall viewpoint.

In this report, we have studied and analyzed several mathematical models. The results were often in favour of the arrival slot mechanism. We have stressed before that there were some important model shortcomings. A pessimistic point of view is that the model choices are in favour of the arrival slot mechanisms and that the advantages are not present anymore in a real access network. However, we have started our reasoning without any mathematical model. Our initial conjecture has been that the basic principle of an arrival slot mechanism is superior to both the gated and non-blocked mechanism, since it can be seen as a sort of trade-off between the two existing mechanisms, combining the advantages of both mechanisms. This is especially true for the dynamic arrival slot mechanism. To support this initial conjecture, we have formulated mathematical models and subsequently analyzed them. The main point that we want to stress is that the idea of using arrival slots has intrisic value, that is independent of the model choice. The results found in the models that we have analyzed are, at least in our opinion, very robust. So we want to state that in real access networks, which obviously are all different from each other and from the analyzed models, the arrival slot mechanism will in general remain valuable.

From the viewpoint of implementation of an arrival slot mechanism, we can simply put that there are no real problems. Both static and arrival slot mechanisms can be implemented straightforwardly, since there are no essentially different implementation issues compared with the gated and the non-blocked mechanism. The scheduling rules for an arrival slot mechanism are in general more complicated than those for the gated and non-blocked mechanism, but this is no main problem. For the dynamic arrival slot mechanism, it is obviously essential that the Head End can immediately observe when the system is empty. This can indeed be observed, analogously as in the gated mechanism, by keeping track of the tree structures.

The final conclusion is that it is in our opinion definitely worthwile implementing all the given recommendations (both the introduced arrival slot mechanism and the improved implementation of the other two mechanisms), based on the mathematical results in this report. By investigating some questions that were left open in this report, there might even be space for more improvement.

Bibliography

[1] Boxma, O.J. (2000) A queue with a gate mechanism. *Internal Pelican report*

[2] Denteneer, D. and Pronk, V. (2000) On the numbers of contenders in a contention tree. *Internal Pelican report*

[3] Shankar N, S. , Borst, S. and Adan, I. (2000) A queueing theoretic approach to contention resolution protocol in HFC access networks. *Internal Pelican Report*

[4] Cohen, J.W. (1982) The Single Server Queue. *North Holland, Amsterdam 2^{nd} revised edition.*

[5] Wolff, R.W. (1989), Stochastic Modeling and the Theory of Queues. *Prentice Hall.*

[6] Janssen, A.J.E.M. and De Jong, M.J.M. (2000) Analysis of contention tree-algorithms. *Manuscript*

[7] Pronk, S.P.P. (2000) Description of the DVB/DAVIC MAC Protocol for HFC Networks. *Philips manuscript*

[8] Golmie, N. , Saintillan, Y. and Su, D.H. (1999) A review of contention resolution algorithms for IEEE 802.14 networks. *IEEE Communications Surveys*

[9] Adan, I. and Zhao, Y. (1996) Analyzing $GI|E_r|1$ queues. *Operations Research Letters 19 (1996) 183-190.*

[10] Abramson, N. (1970) The ALOHA system - Another alternative for computer communications. *Proc. 1970 Fall Joint Comput. Conf. AFIPS Conf. Vol. 37.*

[11] (1985) Special Issue on Random Access Communications. *IEEE Transactions on Information Theory (March 1985).*

[12] Limb, J.O. and Sala, D. (1997) A protocol for efficient transfer of data over hybrid fiber / coax systems. *IEEE/ACM Transactions on Networking 5, 872-881.*

[13] Kelly, F. (1999) Mathematical modelling of the Internet. *http://www.statslab.cam.ac.uk.*

[14] Bruneel, H. and Kim, B.G. (1993) Discrete-Time Models for Communication Systems including ATM. *Kluwer Academic, Boston, 1993.*

[15] Takagi, H. (1993) Queueing Analysis: a Foundation of Performance Evaluation, Vol. 3: Discrete-Time Systems *Elsevier, Amsterdam, 1993.*

[16] Boxma, O.J. and Groenendijk, W.P. (1988) Waiting times in discrete-time cyclic-service systems. *IEEE Transactions on Communications, Vol. COM-36, No. 2 (February), 164-170.*

Appendix A

Solution of a special functional equation

In this appendix, the following functional equation will be studied:

$$\tilde{H}(3x) = \tilde{f}(x) + \tilde{g}(x)\tilde{H}(x).$$

In this equation, $\tilde{H}(\cdot)$ is the yet unknown function. Both $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ are known, well-behaved functions. The solution and the conditions for existence of a solution will be investigated.

Assume for the moment that a solution exists. The functional equation can be solved by iteration. One iteration yields the following equation:

$$\tilde{H}(3x) = \tilde{f}(x) + \tilde{g}(x)\left(\tilde{f}(\frac{x}{3}) + \tilde{g}(\frac{x}{3})\tilde{H}(\frac{x}{3})\right).$$

Continuing in this fashion, we obtain after n iterations the following:

$$\tilde{H}(3x) = \left(\prod_{j=1}^{n} \tilde{g}(\frac{x}{3^j})\right) \tilde{H}(\frac{x}{3^n}) + \sum_{i=0}^{n} \left(\prod_{j=1}^{i} \tilde{g}(\frac{x}{3^j})\right) \tilde{f}(\frac{x}{3^i}).$$

When the limit is taken for $n \to \infty$ we obtain the following:

$$\tilde{H}(3x) = \lim_{n\to\infty} \left[\left(\prod_{j=1}^{n} \tilde{g}(\frac{x}{3^j})\right) \tilde{H}(\frac{x}{3^n})\right] + \sum_{i=0}^{\infty} \left(\prod_{j=1}^{i} \tilde{g}(\frac{x}{3^j})\right) \tilde{f}(\frac{x}{3^i}).$$

If the expressions on the right-hand side are convergent, then the above equation can be used to evaluate $\tilde{H}(3x)$ numerically for arbitrary x. Whether or not the expressions on the right-hand side are indeed convergent depends on the functions $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$ and the convergence ratio of $\tilde{H}(\frac{x}{3^n})$ as $n \rightarrow \infty$.

When we consider the first expression on the right-hand side, this expression can be seen as a product of two terms. If both $\lim_{n\rightarrow\infty}\left[\left(\prod_{j=1}^{n}\tilde{g}(\frac{x}{3^j})\right)\right]$ and $\lim_{n\rightarrow\infty}\left[\tilde{H}(\frac{x}{3^n})\right]$ exist, then the boundary condition for $\tilde{H}(0)$ is still needed for the evaluation.

Appendix B

Closed-form expressions for $f_n(z)$

In this appendix closed-form expressions for $f_n(z)$ are given for $n = 0, \dots, 6$. These expressions can be inverted to obtain the probabilities $P(\tilde{B}(n) = k)$. The latter is also done for $n = 2, \dots, 6$.

$$
\begin{aligned}
f_0(z) &= 1 \\
f_1(z) &= 1 \\
f_2(z) &= \frac{z}{3-z} \\
f_3(z) &= \frac{z(3+5z)}{3(-9+z)(-3+z)} \\
f_4(z) &= \frac{z(-99z - 15z^2 + 10z^3)}{3(-27+z)(-9+z)(-3+z)^2} \\
f_5(z) &= \frac{z(27z + 159z^2 + 22z^3)}{3(-81+z)(-27+z)(-3+z)^2} \\
f_6(z) &= \frac{-2z(-905418z^2 - 1198476z^3 + 515187z^4 - 20493z^5 - 1629z^6 + 77z^7)}{9(-243+z)(-81+z)(-27+z)(-9+z)^2(-3+z)^3}
\end{aligned}
$$

$$
\begin{aligned}
P(\tilde{B}(2) = k) &= 2\,(3)^{-k} \\
P(\tilde{B}(3) = k) &= 2\,(3)^{-2k}\left(-8 + (3)^{1+k}\right) \\
P(\tilde{B}(4) = k) &= \frac{1}{8}(3)^{-3k}\left(1131 - 512(3)^k + 7(3)^{1+2k} + 8k(3)^{1+2k}\right)
\end{aligned}
$$

$$P(\tilde{B}(5)=k) = \frac{5}{104}(3)^{-4k}\left(-32256+4901(3)^{k+1}-751(27)^{k}+104k(3)^{1+3k}\right)$$

$$P(\tilde{B}(6)=k) = \frac{1}{4326400}((3)^{-5k}(91508222421-1490944000(3)^{3+k}+63713000(3)^{3+2k}$$
$$-163645807(3)^{1+4k}+819200000(27)^{k}+1040k(9893(3)^{2+4k}+819200(27)^{k})$$
$$+3785600(k)^{2}(3)^{1+4k}))$$