

Jari Melasniemi

## **Size-based scheduling under terminal constraints in cellular systems**

**Faculty of Electronics, Communications and Automation**

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 9.8.2010

**Thesis supervisor:**

Prof. Samuli Aalto

**Thesis instructor:**

D.Sc. (Tech.) Pasi Lassila

Author: Jari Melasniemi

Title: Size-based scheduling under terminal constraints in cellular systems

Date: 9.8.2010

Language: English

Number of pages:7+68

Faculty of Electronics, Communications and Automation

Department of Communications and Networking

Professorship: Networking technology

Code: S-38

Supervisor: Prof. Samuli Aalto

Instructor: D.Sc. (Tech.) Pasi Lassila

The traffic volume of mobile data has been increasing while the third generation of mobile communication systems (3G) and its evolution versions such as High Speed Downlink Packet Access (HSDPA) have increased the transmission rates of mobile networks. In cellular networks it is not possible to serve all users simultaneously and the system schedules the transmissions by selecting the service order of users in the system. In HSDPA-like systems the transmissions of users are separated by codes, that is the systems are based on Code Division Multiple Access (CDMA) technology. User devices are categorized by the number of codes each device is able to use at maximum.

In this thesis, the scheduling aspect of improving the performance of wireless systems is examined. The service order of jobs in the system is defined by the scheduling policy. By changing this policy, it is possible to change the behaviour of the system considerably. Earlier it has been proven that the Shortest Remaining Processing Time (SRPT) policy is the optimal way of minimizing the mean delay of the M/G/1 queue. In this thesis, the SRPT policy is under examination when terminal constraints are taken into account. This results in multi-server queuing models for which hardly any optimal scheduling results are known.

The results achieved by simulating the wireless cellular system at flow level show that the performance of the system is improved by using SRPT instead of Processor Sharing, the fair baseline policy. The performance improvement depends on the load of the system together with the number of codes in the system. The performance improvement is higher when the system is under high loads.

Keywords: Scheduling, terminal constraints, SRPT, PS, HSDPA, performance, elastic traffic

Tekijä: Jari Melasniemi		
Työn nimi: Päätelaiterajoitukset huomioiva kokopohjainen aikataulutus solukko-verkoissa		
Päivämäärä: 9.8.2010	Kieli: Englanti	Sivumäärä: 7+68
Elektroniikan, tietoliikenteen ja automaation tiedekunta		
Tietoliikenne- ja tietoverkkotekniikan laitos		
Professori: Tietoverkot		Koodi: S-38
Valvoja: Prof. Samuli Aalto		
Ohjaaja: TkT Pasi Lassila		
<p>Langattomien verkkojen suosio on lisääntynyt samalla, kun tiedonsiirtonopeudet ovat kolmannen sukupolven matkapuhelinverkkojen (3G) ja niiden kehitysversioiden, kuten High Speed Downlink Packet Access (HSDPA), myötä kasvaneet. Solukko-verkoissa kaikkia käyttäjiä ei voida palvella yhtäaikaan, ja järjestelmän on aikataulutettava lähetykset eli tehtävä päätös siitä, missä järjestyksessä käyttäjiä palvellaan. HSDPA-järjestelmissä eri käyttäjien lähetykset erotellaan toisistaan koodien avulla, tällöin puhutaan koodijakokanavoinnista (Code Division Multiple Access, CDMA). HSDPA-järjestelmässä päätelaitteet on ryhmitelty sen perusteella, montako koodia laite pystyy enimmillään käyttämään.</p> <p>Tässä työssä tutkitaan aikataulutuspäätöksen vaikutusta langattoman järjestelmän suorituskykyyn päätelaiterajoitusten vallitessa. Aikataulutuspäätöksen muuttamalla on mahdollista parantaa järjestelmän suorituskykyä, mikä kiinnostaa erityisesti langattomien verkkojen tapauksessa. Aiemmin on osoitettu, että niin kutsuttua Shortest Remaining Processing Time (SRPT) -politiikkaa noudattamalla M/G/1 jonotusjärjestelmän keskiviive voidaan minimoida. Päätelaiterajoitusten huomioiminen johtaa ns. monen palvelimen jonomalleihin, joiden optimaalisia aikataulutuspäätöksiä ei tunneta.</p> <p>Langattomien solukko-verkkoa on simuloitu vuotasolla, ja tulosten perusteella SRPT-politiikkaa käyttämällä järjestelmän suorituskyky vaikuttaa paranevan myös siinä tapauksessa, että päätelaiterajoitukset huomioidaan. Suorituskyvyn muutos riippuu järjestelmän kuormituksesta ja järjestelmässä olevien koodien kokonaismäärästä. Suurin parannus suorituskykyyn saavutetaan hyvin korkeilla kuorman arvoilla.</p>		
Avainsanat: Aikataulutus, terminaalirajoitukset, SRPT, PS, HSDPA, suorituskyky, elastinen liikenne		

## Preface

This thesis has been done in the Department of Communications and Networking (Comnet) at the Aalto University School of Science and Technology. The research has been done within the Advances in Wireless Access (AWA) project funded by TEKES, Nokia Siemens Networks, and Ericsson.

At first, I want to thank my supervisor, Prof. Samuli Aalto, and my instructor, D.Sc. (Tech.) Pasi Lassila, for their continuous support and advice all along the way and specially for the subject for this thesis. The help was always available when needed and discussions during the whole process were very beneficial. This has been an educational experience and I want to thank you for this opportunity.

I want to thank also my parents and friends who have supported me with this work. Without your help this thesis would not have been possible.

Otaniemi, 9.8.2010

Jari Melasniemi

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Abstract (in Finnish)</b>	<b>iii</b>
<b>Preface</b>	<b>iv</b>
<b>Contents</b>	<b>v</b>
<b>Abbreviations</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Research problem . . . . .	3
1.3 Structure of the thesis . . . . .	3
<b>2 Wireless cellular systems</b>	<b>4</b>
2.1 Evolution towards 3G . . . . .	4
2.2 3G . . . . .	6
2.3 HSDPA . . . . .	7
<b>3 Modelling the wireless channel</b>	<b>9</b>
3.1 Path loss . . . . .	9
3.1.1 Free-space path loss model . . . . .	9
3.1.2 Ray tracing model . . . . .	10
3.1.3 Empirical path loss models . . . . .	11
3.1.4 Simplified path loss model . . . . .	11
3.2 Other channel characteristics . . . . .	12
<b>4 General theory</b>	<b>14</b>
4.1 Concepts . . . . .	14
4.2 M/M/1 - FIFO . . . . .	17
4.3 M/M/n . . . . .	18
4.4 Processor Sharing . . . . .	21
4.5 Shortest Remaining Processing Time . . . . .	21
4.6 Elastic traffic . . . . .	22
<b>5 Models</b>	<b>24</b>
5.1 Wireless system model . . . . .	24
5.2 Channel model . . . . .	24
5.3 Analysis of the mean service time . . . . .	25
<b>6 Homogeneous case</b>	<b>28</b>
6.1 Assumptions . . . . .	28
6.2 Policies . . . . .	28
6.2.1 Baseline policy . . . . .	28

6.2.2	SRPT policy . . . . .	29
6.3	Implementation of the simulator . . . . .	29
6.3.1	Supporting functions . . . . .	29
6.3.2	Simulate() function . . . . .	30
6.4	Verification of the simulator . . . . .	31
6.5	Results . . . . .	33
6.5.1	Transient period . . . . .	33
6.5.2	SRPT compared with PS . . . . .	34
6.5.3	Modified SRPT with selections based on bit sizes . . . . .	35
6.5.4	Effect of $K$ for the performance improvement . . . . .	38
6.6	Summary . . . . .	39
<b>7</b>	<b>Heterogeneous case</b>	<b>40</b>
7.1	Assumptions . . . . .	40
7.2	Policies . . . . .	40
7.2.1	Baseline policy . . . . .	40
7.2.2	SRPT policy . . . . .	42
7.3	Implementation of the simulator . . . . .	43
7.4	Verification of the simulator . . . . .	43
7.5	Results . . . . .	44
7.5.1	Transient period . . . . .	44
7.5.2	SRPT compared with PS . . . . .	45
7.5.3	The conditional mean delay with respect to the size and distance	48
7.6	Summary . . . . .	56
<b>8</b>	<b>Summary</b>	<b>58</b>
8.1	Conclusions . . . . .	58
8.2	Future research . . . . .	59
	<b>References</b>	<b>60</b>
	<b>Appendix A: Source code of the simulateSRPT() function</b>	<b>61</b>
	<b>Appendix B: Source code of the simulatePS() function</b>	<b>66</b>

## Abbreviations

1G	First generation
2G	Second generation
3GPP	3 <sup>rd</sup> Generation Partnership Project
3GPP2	3 <sup>rd</sup> Generation Partnership Project number 2
8PSK	Eight-phase shift keying
AMC	Adaptive Modulation and Coding
AMPS	Advanced Mobile Phone Service
ARIB	Association of Radio Industries and Businesses
ARQ	Automatic Repeat Request
CDMA	Code Division Multiple Access
CS	Circuit Switched
D-AMPS	Digital Advanced Mobile Phone Service
DPS	Discriminatory Processor-Sharing
EDGE	Enhanced Data rates for Global Evolution
EGPRS	Enhanced General Packet Radio Service
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplex
FIFO	First In First Out
GMSK	Gaussian minimum shift keying
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communication
HARQ	Hybrid Automatic Repeat Request
HSCSD	High Speed Circuit Switched Data
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
IP	Internet Protocol
IS-95	Interim Standard 95
LIFO	Last In Last Out
MAP	Mobile Application Part
NMT	Nordic Mobile Telephone
OSI	Open Systems Interconnection model
PDF	Personal Digital Cellular
PF	Proportional Fair
PS	Processor Sharing
QoS	Quality of Service
SRPT	Shortest Remaining Processing Time
TACS	Total Access Communications System
TCP	Transmission Control Protocol
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
UMTS	Universal Mobile Telecommunications System
UTRA	Universal Terrestrial Radio Access

# 1 Introduction

This section will give a short overview of the topic related to this thesis. At first some background information will be given and after that the research topic will be clarified. At the end of this chapter, the structure of this thesis will be described.

## 1.1 Background

Mobile technology has developed significantly during the last decades. Much has happened since the launch of the first mobile communication network. This happened in the 1980s and roughly ten years later it was time for the next generation mobile communication networks. The main difference between these two network generations is that while the first generation (1G) mobile networks were based on purely analogue transmissions, the second generation (2G) networks are fully digital. With the digitalization the capacity of networks became much higher than before and overall the efficiency of the network got better.

Even if 2G was able to fix many problems related to the first generation networks it was not enough as such for the growing needs of the customers. 2G networks were at first only circuit based like the first generation analogue networks. This meant that also the data traffic was transferred through the circuit switched (CS) technology. Constantly growing traffic demands and low data rates in 2G drove the evolution onwards. High Speed Circuit Switched Data (HSCSD) and the General Packet Radio Service (GPRS) were the first additions to 2G systems. HSCSD and GPRS tried to use the existing technology more efficiently than before. Since this was not enough, Enhanced Data rates for Global Evolution (EDGE) was introduced. EDGE was based on a totally new modulation technology and with that it became possible to achieve much higher data rates than with the previous versions of 2G. These modifications and extensions are often called 2.5G systems.

Modifications of 2G systems were not enough for the market and the development of the technology continued further in a direction where EDGE was leading it already. The third generation of mobile communication systems was introduced again roughly ten years after the previous generation. The designing of the 3G system was started from scratch so there were no limitations of previous technologies. 3G networks were designed from the beginning to carry also data traffic much more efficiently than 2G systems.

Transfer rates in 3G networks are much higher than in 1G and 2G systems. In 2.5G systems theoretical transfer rates are at maximum a couple of hundreds of kilobits per second while in 3G it is, in theory, possible to achieve even megabits per second depending on the channel conditions and the amount of traffic in the network. 3G technology is still developing and new versions of 3G networks have been published almost every year. The most significant extensions to 3G are High Speed Downlink Packet Access (HSDPA) and High Speed Uplink Packet Access (HSUPA) which increased transfer rates even further. These extensions are often called 3.5G.

Mobile data communication has become more and more popular in recent years.

3G technology has made relatively fast mobile data connections broadly available and the cost of the connection has become competitive. Mobile devices can make good use of available 3G resources and the prices of the devices are near to customers' financial standings. Service providers have discovered this willingness for mobile communications and thereupon introduced mobile versions for many existing services. Also, in addition to just modifying old services, totally new mobile services have been launched.

The traffic volume of the mobile data is increasing all the time and prioritizing, for example, real-time traffic has become topical. With scheduling policies it is possible to favour certain jobs in the system. For example real-time traffic, which usually consists of rather small packets, can be prioritized to assure promised Quality of Service (QoS). At the same time, some other traffic type will experience longer delays. In most cases, this is acceptable and can improve the quality of user experience.

Scheduling policies are part of queueing theory and performance analysis. Queueing theory tries to examine and analyse the behaviour of systems at flow or packet level. These results can be used when real world networks and systems are dimensioned so that a desired performance level is achieved. A common scheduling policy, at least for comparing purposes, is Processor Sharing (PS). In PS, each user or job in the system is served simultaneously and the service rate depends on the number of jobs or users in the system. When the system is almost empty, all users experience a higher service rate than when the load of the system is higher. Several other scheduling policies have been developed in addition to the PS policy. In the 1960s, Schrage proved [13] that Shortest Remaining Processing Time (SRPT) is the optimal policy for minimizing the mean delay of the M/G/1 system. SRPT means that the job which has the shortest remaining processing time is served first. SRPT has been under examination since the publication of the optimality result.

The high increase in data traffic in recent years has made different scheduling policies interesting. Systems and networks can be rationalized by changing the scheduling policy. However, SRPT has not been adopted widely in real systems. It is a prevailing opinion that large jobs would starve under SRPT and this opinion has been the winning one against all known advantages of SRPT compared with for example PS even though the degree of unfairness of large jobs under SRPT is shown to be surprisingly low [3].

The performance of the cellular system can be improved by applying channel-aware scheduling like Proportional Fair (PF) scheduling. Channel-aware scheduling policies are based on measuring the channel conditions and adjusting the scheduling based on measurements. In the HSDPA systems, where the PF policy is used, the time slot is scheduled to the user device with the highest momentary receiving rate proportionally to its average receiving rate. However, the performance improvement of channel-aware scheduling has only small effect since the quality of the measured channel data may not always be usable. This can be the case if the channel is behaving so randomly that it is almost impossible to predict the future behaviour. In this thesis, the channel-aware scheduling is not considered. Instead, we focus on the impact of flow-level size information.

## 1.2 Research problem

The impact of size-based scheduling on the flow level performance of elastic data traffic in wireless downlink data channels has been examined in [2]. In this thesis, the same aspect of a single cell in a cellular mobile network is examined. The flow level delay or just delay, the total time needed for the transmission, is the main performance measure used in this thesis. In [2], the examined system corresponds to an early evolution version 1xEV-DO of CDMA2000 networks. In that system, one user was scheduled in a time slot which can be modelled with an M/G/1 queuing system. The model used in this thesis is a developed version of the model used in [2]. Now many users are scheduled in a time slot and with queuing theory terms this means a multiserver problem.

In HSDPA systems, the transmissions of different users are separated by codes which is the main idea behind the CDMA technology. The scheduler will allocate these codes to users and this way share the channel. In this thesis, terminal constraints are taken into account which means that users are not able to use all the codes in the system at a time. All users are assumed to have similar characteristics in the so-called homogeneous case and each of users can use the same number of codes. In the heterogeneous case it is possible to have users with different characteristics in the system.

There are no analytical results for the heterogeneous case and both the SRPT and PS policies must be examined by simulations in the multi-server case. The PS policy shares the capacity fairly among users based on the number of codes each user is able to use. This differs from the basic PS policy which shares the time fairly among users and PS policy used in this thesis can be thought to be closer to the so-called Discriminatory Processor-Sharing (DPS) policy in case of multiple servers. The performance improvement of the SRPT policy comparing with the PS policy is examined in the sense of the flow level delay. When analytical results are not available, results are produced by simulations.

## 1.3 Structure of the thesis

This thesis is divided into eight sections. Section 2 is an introduction to how cellular networks have developed from the first generation towards 3G and HSDPA. Section 3 introduces different path loss models and other fading models which can be used for modelling the wireless channel. At the beginning of Section 4 some important concepts are introduced and after that the topic-related theory is covered. Section 5 introduces the model which is used in this thesis and the assumptions made. Sections 6 and 7 are about the actual work and results. Section 8 will summarize the results found in this thesis. Also, suggestions for further research will be given. The source codes of the SRPT and PS simulators done during this thesis are given in Appendix A and B.

## 2 Wireless cellular systems

In this section the wireless system related to this thesis will be covered. At first the evolution of mobile communication networks from the first generation to the third generation will be introduced. The 3G networks and the HSDPA extension will be discussed in more detail since the topic of this thesis relates to systems like HSDPA.

### 2.1 Evolution towards 3G

The first mobile cellular telecommunication systems were introduced in the 1980s. This was the first time when cellular radio networks were implemented for practical use. These early cellular networks had much higher capacity than those non-cellular mobile networks which existed at that time. This was the start of the first generation of mobile cellular telecommunication. The main idea in cellular networks is to divide the coverage area into small cells and then use same frequencies in different cells around the area. This main idea has not changed from the early days even though nowadays mobile communication systems are mostly digital while the first ones were fully analogue. The amount of data traffic has increased significantly since the time of the first generation systems which carried in practice only voice traffic.

There were many competing standards in the first generation networks but none of them became dominant. The most widely used standards were Nordic Mobile Telephone (NMT), Total Access Communications System (TACS), and Advanced Mobile Phone Service (AMPS). It must be noted that in addition to the aforementioned there were some standards that were used only in single countries. NMT was mainly used in Scandinavia and in central and southern European countries but NMT networks were also launched in Eastern Europe in the latter half of the 1990s. TACS is based on AMPS and it was used in U.K., in some Middle East countries, and in southern Europe. In America, the Far East, Australia, and New Zealand AMPS was used widely. The first commercial cellular network in Japan was MCS provided by NTT. [11]

Digitalization is the main difference when moving from the first generation towards 2G of mobile cellular systems. With digital radio transmission much higher capacity can be achieved in comparison with analogue transmission. In 2G systems, the same frequency channel can be divided among users which use either different codes or different time slots for sending.

Standardization of 2G systems was done by different participants around the world. As a result, four main standards were formulated. These are the Global System for Mobile communication (GSM) and its derivatives, Digital AMPS (D-AMPS), Code Division Multiple Access (CDMA), the Interim Standard 95 (IS-95), and Personal Digital Cellular (PDC).

GSM is the most popular one of these standards. It was first introduced in Europe and then spread quickly all over the world. GSM or some of its variants are, at least in some form, in use also in the Americas even though they have not reached a dominant position there. GSM is based on a Time Division Multiple Access (TDMA) system which means that the transmission time is split into short

slots and these time slots are allocated to different users. Every user uses the same frequency band, one at a time. The first GSM version uses the 900 MHz band, later derivatives use in addition bands of 400 MHz and 1800 MHz. The 1800 MHz band was taken into use because of the limited capacity of the 900 MHz band. In urban areas, there were more users than the 900 MHz band was able to serve. By using a higher frequency the coverage area will become smaller which is the reason why lower frequencies are used in rural areas. After the initialization of the 1800 MHz band, user devices were able to use both networks, 900 MHz and 1800 MHz, and chose the best one available. [11]

IS-95, developed by Qualcomm, got a firm foothold in North America among 2G technologies. IS-95 uses CDMA which differs from TDMA in such that CDMA shares the same frequency band with different codes to different users. IS-95 is used also outside North America, mainly in East Asian countries, such as South Korea, Hong Kong, and Singapore. IS-95 is also known by the name cdmaOne and it has been an early version of the 3G standard CDMA2000. [11]

PDC networks were standardized in 1991 and they were spread widely in Japan. PDC offered higher data rates than the previous mobile networks. One very popular service related to PDC was *i-mode*, developed by NTT DoCoMo. *I-mode* made it possible to charge customers based on the actual amount of transferred bits instead of time like traditionally in circuit switched networks. Another much liked service was the possibility to use e-mails in mobile networks. Every user had their own e-mail address in format <mobile\_number>@docomo.ne.jp. [11]

Improvements that have been introduced to GSM networks after the first release will be gone through next. In this thesis, the main focus will be on systems like HSDPA. For that reason the evolution towards HSDPA will be discussed in more detail.

The basic GSM air interface provided only 9.6 kbps data rate for one user. This is not enough, for example, for a smooth web browsing experience. The low data rate was the most significant problem in the basic GSM. There was room for improvements in the efficiency of the air interface usage. HSCSD is the simplest way of increasing data rates. HSCSD provides a possibility to use several time slots at the same time and transfer rate increases linearly as a function of time slots used. The first HSCSD specification allows using 4+4 time slots at most (downlink+uplink) [7]. In practice, many user equipment support only some limited versions of HSCSD, like 1+3 or 2+2 time slots. HSCSD was easy to implement in networks because it could be done with software updates. Also, old user equipments worked even though new equipment was able to use many time slots simultaneously and get higher data rates. Overall, HSCSD was not a very efficient way of improving data rates. HSCSD is still circuit switched, which means that allocated time slots are reserved even if the user has nothing to transmit. This wastes scarce radio resources and is definitely not the optimal way of improving low data rates. HSCSD is however suitable, and even better than packet switched solutions, for real-time applications. The circuit is already open when there is something to send and no additional delays will be added. [11]

User device manufacturers were not very interested in HSCSD and most of them

moved directly to the GPRS system. GPRS is packet switched which means that there is no need for reserving any circuit beforehand just to be sure. For this reason GPRS is not as suitable for real-time applications as HSCSD. The peak data rate for GPRS is 115 kbps with error correction. This is the theoretical maximum and it can be achieved only in optimal radio conditions using eight downlink time slots. In the usual case, around 10 kbps per time slot is achieved. Even though adding GPRS support to an existing network is more expensive than HSCSD, it was seen by the operators as a required step towards 3G technology and more and more data-oriented traffic. [11]

EDGE, originally Enhanced Data rates for GSM Evolution, is the third improvement to GSM. EDGE differs from the basic GSM system with its new modulation method, Eight-phase shift keying (8PSK). Adding the support for the new modulation method is possible by updating the software of the base station and the old Gaussian minimum shift keying (GMSK) can be maintained unchanged. This gives a possibility for users to use their old equipment as long as they do not want to use higher data rates. The 8PSK modulation is usable only within a relatively short range from the base station so it is not suitable as the only modulation method in the whole network. It is also possible to combine EDGE with HSCSD and GPRS. The combination of EDGE and GPRS is called Enhanced GPRS (EGPRS) and with that the maximum data rate can be even 384 kbps. It is possible to achieve this transfer rate only near the base station using all radio resources of a frequency carrier. [11]

## 2.2 3G

The next generation mobile telecommunications network was called the Universal Mobile Telecommunications System (UMTS). Its standardization was started by the European Telecommunications Standards Institute (ETSI) in the same year as GSM was commercially launched. 3G networks were supposed to be able to offer the same quality of sound as the wired phone network and offer higher data rates with a more efficient use of the frequency band than 2G systems. 3G systems were specified to be able to offer at least data rates of 144 and 384 Kbps. ETSI did not develop the 3G system alone, as some research programs funded by The European Commission participated in the development process as well. The WCDMA was selected as the 3G radio standard of ETSI and Association of Radio Industries and Businesses (ARIB) in 1996 and 1997. [11]

A few years after the decision regarding 3G technology by ETSI an institution for advocating 3G specifications, the <sup>rd</sup> Generation Partnership Project (3GPP), was founded by several telecommunication companies. ETSI Universal Terrestrial Radio Access (UTRA) was taken as a base radio system by the 3GPP association. Enhanced GSM/GPRS Mobile Application Part (MAP) core network was also developed. Nowadays the development work of 3G systems is done more and more within the telecommunications industry itself and the role of standardization organizations has been reduced. Associations produce specification proposals for the standardization organizations to get the formal approval. Companies have more

available resources for development process than intergovernmental organizations which means a faster development process of new standards. [11]

As it was in case of 2G mobile communication systems, a worldwide standard for the third generation systems did not exist. In the United States proposals like CDMA2000 were attractive because CDMA2000 was backward compatible with the widely used IS-95 system. It was possible to run these two systems of different generations on the same time at the same frequency band. 3<sup>rd</sup> Generation Partnership Project number 2 (3GPP2) was launched to further develop the specification of the CDMA2000 system.

In this thesis, only the development branch of WCDMA will be discussed. More information about the other competing 3G technologies can be found, for example, from the book *Introduction to 3G Mobile Communications, 2nd edition* [11] by Juha Korhonen.

The radio interface of WCDMA can be either synchronous or asynchronous. In a synchronous network, all base stations are time synchronized to each other. With time synchronization the radio interface can be used more efficiently but requirements for the hardware are also higher. For the synchronization, for example, the Global Positioning System (GPS) can be used even if there are challenges in using a GPS in high-block city centres. In the asynchronous mode, no time synchronization is used. The ETSI/ARIB proposal, like many others, used the asynchronous mode. The Korea TTA I and CDMA2000 proposals include also the synchronous mode of the network. [11]

The UTRA system, developed by 3GPP, is based on the core network of GSM and it encompasses modes for both Frequency Division Duplex (FDD) and Time Division Duplex (TDD). In the FDD mode, the downlink and uplink use separate frequency bands for transmissions. In the TDD mode both the downlink and uplink use the same frequency carrier. This means that the capacity of these links can be different since time slots in the radio frame can be dynamically allocated. One time slot must always be allocated either to the uplink or downlink, as the communication between the user device and the base station needs a return channel. [11]

## 2.3 HSDPA

High Speed Downlink Packet Access is developed by 3GPP in Release 5 enhancing the downlink data rates of a 3G system. The main reason for developing HSDPA was that data rates in Release 99 networks were too low for multimedia applications. As the name says, HSDPA increases only data rates for the downlink direction, and the theoretical maximum throughput of HSDPA is 14.4 Mbps in Release 5. For increasing data rates of the uplink direction, HSUPA was developed. In this thesis, only HSDPA will be introduced.

Release 5 capable user devices include both Hybrid ARQ (HARQ) and Adaptive Modulation and Coding (AMC) functionalities. HARQ is a link adaption scheme in which link layer acknowledgements are used for retransmission decisions because RLC layer retransmissions are too slow for high-speed data transmissions. RLC layer retransmissions are used in Release 99 systems. In HARQ, the retransmissions are

located closer to the physical layer. AMC means that the shared channel transport format depends on the quality of the channel and it is possible to change the format in every frame. In good channel conditions AMC uses higher-order modulation and less redundancy than in poor conditions. In Release 5 two modulation schemes, QPSK and 16QAM, are used. [1, 11]

HSDPA is based on shared data channels which means that the channel is shared among all active HSDPA user devices. Using the shared channel means that maximum transfer delays cannot be guaranteed which may make HSDPA unsuitable for services which have strict real-time requirements. In HSDPA, the resource allocation (i.e., scheduling) is done in every frame which means that the capacity of the network is allocated every 2 ms. [1, 11]

CDMA is used in the HSDPA systems to share the channel capacity for users. There is a number of codes in the system and these codes are allocated to users based on some scheduling policy. The base station may serve only one user at a time at full transfer rate to minimize the inter-cell interference. There is totally 12 different categories, shown in Table 1, in Release 5 HSDPA specification [1]. The number of codes and used modulation together defines the maximum data rate which the user is able to achieve under good channel conditions. Parameters which are used in simulations have been selected based on values in Table 1.

Table 1: HSDPA user categories [1]

Category	Max number of codes	Modulation	Maximum data rate
Category 1	5	QPSK & 16-QAM	1.2 Mbps
Category 2	5	QPSK & 16-QAM	1.2 Mbps
Category 3	5	QPSK & 16-QAM	1.8 Mbps
Category 4	5	QPSK & 16-QAM	1.8 Mbps
Category 5	5	QPSK & 16-QAM	3.6 Mbps
Category 6	5	QPSK & 16-QAM	3.6 Mbps
Category 7	10	QPSK & 16-QAM	7.3 Mbps
Category 8	10	QPSK & 16-QAM	7.3 Mbps
Category 9	15	QPSK & 16-QAM	10.2 Mbps
Category 10	15	QPSK & 16-QAM	14.4 Mbps
Category 11	5	QPSK only	900 kbps
Category 12	5	QPSK only	1.8 Mbps

### 3 Modelling the wireless channel

This section will present different path loss models and give a short overview to other fading models. Path loss is the most important component for modelling the wireless channel for this thesis so it is discussed in more detail in its own section.

#### 3.1 Path loss

Path loss is one of the notable components when we are trying to model how electromagnetic waves will attenuate when they propagate through an air interface. Path loss represents how much a transmitted signal will attenuate during the transmission. The main component of path loss is the line-of-sight attenuation (also called free space loss) which represents how much signal strength will reduce from transmitter to receiver through a line-of-sight path. The effects of the path loss appear over long distances, and usually this means at least hundreds of metres.

There are many models which try to represent channel conditions as well as possible. The simplest models do not take into account anything else but the free space loss. More advanced models use more components to model the air interface and those methods can thereby give more accurate results. Those models need more parameters and it can sometimes be difficult to find out correct and realistic values for a specific scenario.

Besides analytical models, empirical path loss models have been implemented as well. A complex environment can be modelled more accurately with empirical models than with analytical models. Empirical models are based on measurements done in real environments and it must be noted that measurements are absolutely valid as such only in the original area where measured. When applying those results somewhere else, environmental differences must be taken into account. Things like the average height of buildings or the number of cars in an urban area can change or contort the results unpredictably. The magnitude of the difference may be approximated based on other empirical models and environments where they apply. [6]

##### 3.1.1 Free-space path loss model

Free-space path loss is the loss that the signal will experience when going through a line-of-sight path from a sender to a receiver. Free-space path loss does not take into account any other components which a receiver might detect. Thus for example reflections, scattering, or diffraction effects are not taken into account when calculating path loss.

Based on these assumptions, the free-space path loss model is simple and analytically derivable. A complex scale factor which defines the received signal  $r(t)$  as a function of the transmitted signal  $u(t)$  has been developed for the free-space path loss as [6, p. 31]

$$r(t) = \text{Re} \left\{ \frac{\lambda \sqrt{G_t} e^{-j2\pi d/\lambda}}{4\pi d} u(t) e^{j2\pi f_c t} \right\}, \quad (1)$$

where  $\sqrt{G_l}$  is the product of the transmit and receive antenna field radiation patterns in the line-of-sight direction, the term  $e^{-j2\pi d/\lambda}$  means the phase shift which stems from the distance  $d$  between the transmitter and receiver that the signal travels. The wavelength of the signal is denoted with  $\lambda$ .

The power of a transmitted signal is  $P_t$ . The ratio of received to transmitted power  $\frac{P_r}{P_t}$  can be derived from Equation (1), see [6], and it is given by

$$\frac{P_r}{P_t} = \left[ \frac{\sqrt{G_l}\lambda}{4\pi d} \right]^2. \quad (2)$$

It is possible to represent the received power in dBm (dB value relative to mW) as

$$P_r\text{dBm} = P_t\text{dBm} + 10 \log_{10}(G_l) + 20 \log_{10}(\lambda) - 20 \log_{10}(4\pi) - 20 \log_{10}(d). \quad (3)$$

By this model free-space path loss is defined in dB as

$$P_l\text{dB} = 10 \log_{10} \frac{P_t}{P_r} = -10 \log_{10} \frac{G_l\lambda^2}{(4\pi d)^2}. \quad (4)$$

### 3.1.2 Ray tracing model

Ray tracing widens the free space path loss model by taking into account not only the line-of-sight signal but also those signals which have been reflected, diffracted, or scattered on the way to the receiver. The same signal can go through many ways and each of them can be delayed in time, attenuated in power, and shifted in phase with respect to the line-of-sight signal. The receiver sees the transmitted signal together with all those differently distorted signals summed. [6]

Ray tracing starts from the assumption that there is a finite number of reflected signals and their locations and dielectric properties are known. Maxwell's equations with appropriate boundary conditions can then be used for defining the details of the multipath propagation. In ray tracing geometric equations are used instead of pure Maxwell's equations and wavefronts are approximated as simple particles. Although the approximation produces some error, the comparison of the ray tracing against empirical data shows that ray tracing approximation is usable in rural areas and along city streets with the assumption that both the transmitter and the receiver are close to the ground. For indoor use, the ray tracing model needs to be adjusted with diffraction coefficients. [6]

With the help of computer software it is possible to make a three-dimensional model from the surrounding environment based on, for instance, aerial photographs or architectural drawings, and then use ray tracing for modelling the signal progress. This way one can be assured that the model is suitable and accurate enough in the environment where it will be used. In one general ray tracing model, it is possible to include all attenuated, diffracted, and scattered multipath components of the original signal so only one model is needed. In some cases it is possible to use statistical approximations. For example, when the surfaces of the reflector are not

smooth or the number of reflectors is high, statistical approximations are needed. It might be that the transmitter or the receiver is moving, and due to that the characteristics of the multiple paths vary with time. It is possible that some of these characteristics are not exactly known over time so statistical models must be used. [6]

### 3.1.3 Empirical path loss models

Most mobile systems are usually used in a complex urban environment for which analytical modelling with free-space or ray tracing would be difficult. For that purpose empirical path loss models, based on measurements done in the field, have been developed. Depending on the models there may be some limitations, for example at which frequency band or which kind of a geographical area the results are valid.

Usually empirical models define  $P_r/P_t$  as a function of distance, mainly because of its ease of measurability. Measured  $P_r/P_t$  includes also components other than just path loss since it is impossible to separate the effects of shadowing or multipath effects just by measuring the received signal. Multipath effects are usually removed by averaging the received power measurements and the relative path loss at a certain distance over several wavelengths. To get even more general results averaging can be extended to all measurements which are available from similar environments. [6]

### 3.1.4 Simplified path loss model

It is not always worth trying to model the surrounding environment as accurately as possible, and it may be enough for the model to have a rough estimation of the attenuation. Sometimes it can be almost impossible to define all needed variables so accurately that the result would be certainly usable. A simplified path loss model is a compromise between the ease of use and accuracy of the results. The simple path loss model is suitable for more general modelling whereas more complex models should be used if there are tight specifications which must be met.

A generally used path loss model is

$$P_r = P_t K \left[ \frac{d_0}{d} \right]^\alpha. \quad (5)$$

The received power, represented in Equation (6), has been solved from Equation (5).

$$P_r \text{dBm} = P_t \text{dBm} + K \text{dBm} - 10\alpha \log_{10} \left[ \frac{d}{d_0} \right]. \quad (6)$$

In this model,  $K$  is a constant that represents characteristics of the antenna and the average attenuation on a channel. In other words,  $K$  is the path loss at reference distance  $d_0$  and  $\alpha$  is the path loss exponent which depends on the propagation environment. In case that propagation follows free-space or two-ray model,  $\alpha$  is between two and four.

Path loss exponent  $\alpha = 2$  would be usable for modelling a flat environment without many buildings whereas  $\alpha = 4$  is more suitable for modelling urban areas. For more complex environments more sophisticated methods such as minimum mean-square error can be used to define  $\alpha$ . [6]

## 3.2 Other channel characteristics

In this section, three other channel characteristics are briefly discussed that affect the received signal quality.

**Multipath** Multipath effects were mentioned in the case of ray tracing in Section 3.1.2. A multipath channel can be modelled with statistical characterization and based on that it is possible to describe its properties.

Signals can travel to the receiver in different ways. For the case where a single pulse is transmitted over the multipath channel the received signal will appear as a pulse train. Each received pulse corresponds to the line-of-sight component or a distinct multipath component. The time between the first and last pulse defines the spreading of the delay. [6]

The multipath channel has time-varying characteristics which means that either the transmitter or the receiver is moving and the location of reflectors in the transmission path will change over time. This can be modelled by measuring the pulses sent from a moving transmitter. It is possible to measure changes in the amplitudes, delays, and number of multipath components for each pulse. [6]

**Slow fading/Shadowing** Slow fading, also called shadowing, refers to the random variation which the signal experiences due to a blockage from the object in the signal path. This kind of variation can also be caused by changes in reflecting surfaces and scattering objects. Usually slow (compared with the transfer time of a single symbol) changes in the surrounding environment cause slow fading effects to the transmitted signal. Since in the general case the location, size, and dielectric properties of the blocking objects are unknown, only statistical models can be used for modelling the slow fading. [6]

The most common model for slow fading is so-called log-normal shadowing. This model has been examined in more detail in the book *Wireless Communications* [6] by Andrea Goldsmith.

**Fast fading** Fast fading can occur when the receiver is moving and multipath effects exist. Fast fading is fading which has rapid fluctuations within the symbol duration. In other words, fast fading occurs when the coherence time of the channel is smaller than the symbol period of the transmitted signal. Due to Doppler spreading some frequency dispersion or time selective fading is caused. [4]

In case of fast fading the received signal is the sum of a number of signals reflected from local surfaces. These signals sum in a constructive or destructive way, depending on the relative phase shift. The speed of receiver, frequency of

transmission and physical length of the transmission path have an effect on phase relationships. [4]

## 4 General theory

The characteristics of different kinds of systems can be represented with queueing theory methods. In data networks cases, queueing theory may be used to illustrate traffic flows at packet or flow level. Various buffers and queues exist in networks and with the help of the analytical results from queueing theory different metrics can be analysed. These analytical results can be used in designing and dimensioning real world networks. In this section, some relevant results from queueing theory are reviewed from the point of view of this thesis. *Queueing Systems, Volume I: Theory* [9] and *Volume II: Computer Applications* [10] by Kleinrock, L. have been used as the main sources for this section.

### 4.1 Concepts

In this section, some concepts which relate closely to this thesis are introduced. Deep analysis or proofs will not be included and concepts will be covered at a general level which will be enough for understanding the rest of the thesis.

**Kendall's notation** In queueing systems so-called Kendall's notation is often used. This notation represents the characteristics of the system and distributions used for the arrival and service process. Kendall's notation has the form A/B/C/D/E, in which letters are defined as follows: (A) arrival process, (B) service process, (C) number of servers, (D) number of system places, and (E) size of customer population. It is not necessary to use and denote all parameters every time. In that case, the default value (infinity) is assumed for D and E. To avoid confusions, both D and E must be visible in case we want to have the default value for D and non-default value for E. The most commonly used symbols for arrival and service processes are M for exponential interarrival distribution (Poisson process, introduced in more detail later in this section), D for constant interarrival times and G for the general case. More symbols for distributions can be introduced as needed. For C, D, and E, simple numerical values are used.

**Scheduling disciplines** The service order of jobs in the system is defined by the scheduling discipline. It is possible to change the behaviour of the system considerably by changing the scheduling discipline.

First In First Out (FIFO) is the most popular scheduling discipline. Under FIFO, jobs are served in the same order where they have arrived into the system. The first job arrived will also leave the system first, in other words FIFO is an order preserving discipline.

Last In Last Out (LIFO) is an opposite version for the FIFO discipline. Under LIFO, the last job arrived is always served first.

Processor Sharing (PS) shares the capacity equally to all jobs in the system and each job is served all the time. The service rate depends on the number of jobs in the system. PS will be examined in more detail in Section 4.4.

Shortest Remaining Processing Time (SRPT) is a scheduling discipline which serves always the shortest job first. SRPT is a so-called pre-emptive discipline which means that the service can be interrupted by an arriving shorter job and resumed at a later time. SRPT will be examined in more detail in Section 4.5.

**Little's law** Little's law is one of the most important laws in queueing theory. It gives the relation between  $E[N]$ , the mean number of customers in the system,  $E[T]$ , the mean sojourn time (average delay), and  $\lambda$ , the arrival rate of customers to the system. Little's law states that

$$E[N] = \lambda E[T]. \quad (7)$$

Little's law supposes that the system is stable and it will become empty every now and then. When arrival rate equals  $\lambda$ , also the departure rate will be  $\lambda$ , based on the assumption that the system is stable.

**Markov process** A Markov process is a stochastic process which has so-called Markovian property. This property means that the current state of the process contains all the information that is needed to characterize the future of the process and it does not matter how the current state has been reached. In other words, given the current state, the future and past of a Markov process are independent. The same is represented in mathematical form in Equation (8). This equation holds for the stochastic process which has a discrete value space. The parameter space (values of  $t$ ) can be either discrete or continuous.

$$\begin{aligned} P\{X_{t_n} \leq x_n | X_{t_{n-1}} = x_{n-1}, \dots, X_{t_1} = x_1\} = \\ P\{X_{t_n} \leq x_n | X_{t_{n-1}} = x_{n-1}\}, \quad \forall n, \forall t_1 < \dots < t_n \end{aligned} \quad (8)$$

where  $X_{t_n}$  is the current state of the process and  $X_{t_{n-1}} \dots X_{t_1}$  are considered as states in the past. It is assumed that the stochastic process is defined at those time values.

Markov processes can be used for modelling queueing systems. Based on the characteristics of the Markov process it is possible to calculate probabilities of different states of the process. This information can be used for analysing the system in question. In this thesis, the Markov process is used when different queueing models are examined and equilibrium probabilities of those are derived.

**Birth-death process** A birth-death process is a Markov process with a couple of added assumptions. The birth-death process has a discrete state space which means that the system has clearly separated states compared to continuous state space where the state can change smoothly. The state transition diagram of the birth-death process is presented in Figure 1. The discrete state space together with the assumption that state transitions can happen only between neighbouring states differentiate birth-death processes from the generic Markov process. With the

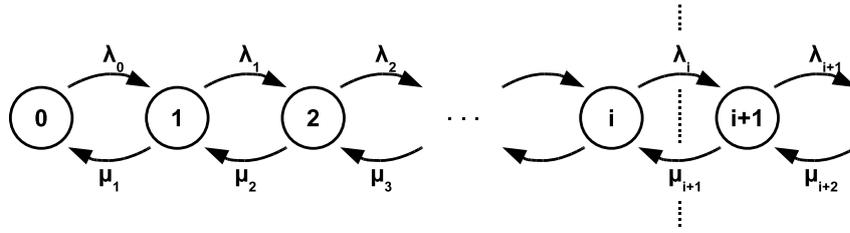


Figure 1: Birth-death process.

notation used in Figure 1 this means that the possible state transitions are  $i \rightarrow i+1$  and  $i \rightarrow i-1$ .

The birth-death process has two special cases: a pure birth process and a pure death process. In a pure birth process,  $i \rightarrow i+1$  is the only state transition which can occur. This means that there are no departures in the system and the initial user population increases forever. Respectively in a pure death process only departures occur and the only state transition is  $i \rightarrow i-1$ . In this case, the user population is given in the system and little by little the amount of users in the system decreases.

**Poisson process** The Poisson process is often used in queueing theory when the arrival process of customers is modelled. Customers may represent, for example arriving packets or calls. The Poisson process is a potential model when there is a need for modelling arrivals from a large population of independent sources. Arrivals in the Poisson process can be visualized as in Figure 2. Arrows denote arrivals and  $N(t)$  refers to the number of arrivals in the time interval  $[0, t]$ , or more generally  $[t_1, t_2]$ . The Poisson process can be characterized as a pure birth process like described previously. In the case of a Poisson process, every  $\lambda_i = \lambda$  with the notation in Figure 1.

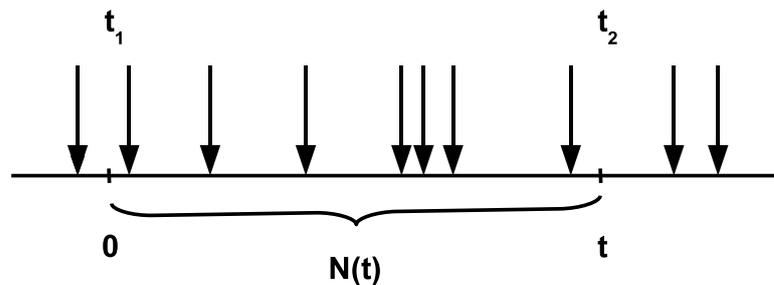


Figure 2: Poisson process.

In a Poisson process with rate  $\lambda$ , the time between two arrivals is exponentially distributed and arrivals are independent. Due to the independence of arrivals,  $N(t)$  has a Poisson distribution with parameter  $\lambda t$  and

$$P\{N(t) = k\} = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \quad k = 0, 1, 2, \dots$$

The Poisson process has some properties which may be used when different kind of systems are examined. It is possible to merge two Poisson processes (with intensities  $\lambda_1$  and  $\lambda_2$ ) and the result is also a Poisson process (with intensity  $\lambda = \lambda_1 + \lambda_2$ ). It has also been proven that if arrivals are selected with a probability  $p$  from a Poisson process (intensity  $\lambda$ ), the resulting process is also a Poisson process, with intensity  $p\lambda$ . A split of a Poisson process has the same characteristics and the resulting processes remain as Poisson processes. These properties will not be proven or discussed in more detail in this thesis.

## 4.2 M/M/1 - FIFO

M/M/1 is Kendall's notation for a system which has just one server and both arrivals and service times are exponentially distributed. The size of a user population and the number of system places are not defined so they are assumed to be infinite. Because of an unlimited number of queueing places, every job will get service at some point of time and none are dropped out. Jobs are served in the FIFO order so that the job arrived first will also get the service at first. The queueing system, with an arrival rate  $\lambda$  and service rate  $\mu$ , can be visualized like in Figure 3 where the circle represents the server and the boxes are queueing places. Arrows symbolize traffic flows through the system.

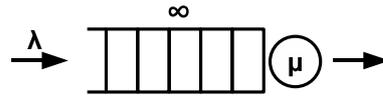


Figure 3: M/M/1 queueing system.

The M/M/1 queueing model can be drawn as a Markov process like in Figure 4. If there is a constant  $\lambda$  between the states, inter arrival times obey an exponential distribution with the mean value  $1/\lambda$ . The service rate  $\mu$  of the M/M/1 system is also constant and state independent. The load of the system is defined as  $\rho = \lambda/\mu$ .

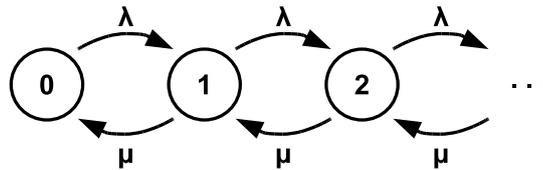


Figure 4: The Markov process of the M/M/1 queue.

Next the equilibrium distribution for the M/M/1 queueing model will be derived. The local balance equation and the normalizing condition are presented in Equations (9) and (10). In both cases, the equilibrium probability for the state  $i$  is denoted by  $\pi_i$  which gives directly the probability that the system will be in that state.

Local balance equation:

$$\begin{aligned}
 \pi_i \lambda &= \pi_{i+1} \mu \\
 \Rightarrow \pi_{i+1} &= \frac{\lambda}{\mu} \pi_i = \rho \pi_i \\
 \Rightarrow \pi_i &= \rho^i \pi_0, \quad i = 0, 1, 2, \dots
 \end{aligned} \tag{9}$$

Normalizing condition which will fix the solution with the assumption  $\rho < 1$ :

$$\begin{aligned}
 \sum_{i=0}^{\infty} \pi_i &= \pi_0 \sum_{i=0}^{\infty} \rho^i = 1 \\
 \Rightarrow \pi_0 &= \left( \sum_{i=0}^{\infty} \rho^i \right)^{-1} = \left( \frac{1}{1-\rho} \right)^{-1} = 1 - \rho
 \end{aligned} \tag{10}$$

The equilibrium queue length distribution can be derived from the local balance equations and the normalizing condition. It can be shown that the system is stable when  $\rho < 1$ . The equilibrium distribution is a geometric distribution and the probability that the system is in state  $i$  is

$$P\{X = i\} = \pi_i = (1 - \rho)\rho^i, \quad i = 0, 1, 2, \dots$$

The mean value for the queue length is thus

$$E[X] = \frac{\rho}{1 - \rho}.$$

The total time a job is in the system is called delay ( $D$ ). The delay includes the waiting time ( $W$ ), in case the job does not get the service immediately, and the actual service time ( $S$ ), the time the job is in the service state. With this notation the delay can be represented in the form  $D = W + S$ . By applying Little's law (Equation (7)) the expected value of  $D$  can be derived as follows

$$E[D] = \frac{E[X]}{\lambda} = \frac{1}{\lambda} \frac{\rho}{1 - \rho} = \frac{1}{\mu} \frac{1}{1 - \rho} = \frac{1}{\mu - \lambda}.$$

The expected value for the waiting time  $W$  can be derived in the same way:

$$E[W] = E[D] - E[S] = \frac{1}{\mu} \frac{1}{1 - \rho} - \frac{1}{\mu} = \frac{1}{\mu} \frac{\rho}{1 - \rho}.$$

### 4.3 M/M/n

Compared to the M/M/1 system examined in Section 4.2, the only difference in M/M/n is the number of servers in the system. In the M/M/n queuing model,  $n$  servers are assumed and other assumptions remain as in the M/M/1 system. M/M/n is not a loss system so every job that is received by the system is served at some point in time. First  $n$  jobs get service immediately and after that new jobs are queued until some server is freed from its previous job.

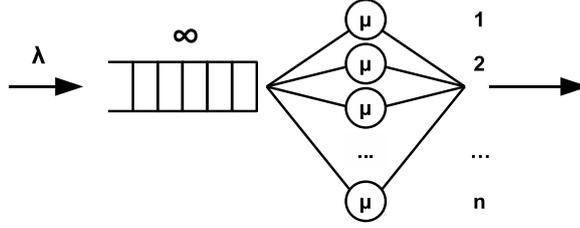


Figure 5: M/M/n queue.

The queueing system, with arrival rate  $\lambda$  and service rates  $\mu$ , is visualized in Figure 5 where the circles represents servers ( $n$  pieces), the boxes are queueing places, and the arrows symbolize traffic flows through the system.

Figure 6 shows the Markov process of the M/M/n queueing model. There is a constant  $\lambda$  between the states, and mean arrival times obey an exponential distribution with mean value  $1/\lambda$ . The service rate of one server is  $\mu$  which also obeys exponential distribution with the mean value  $1/\mu$ . The service rate of the system as a whole depends on the state of the system. The service rate of the system is the service rate of one server ( $\mu$ ) multiplied by the number of occupied servers. From state  $n$  onwards the service rate is constant. The load of the system is defined as

$$\rho = \lambda/(n\mu). \quad (11)$$

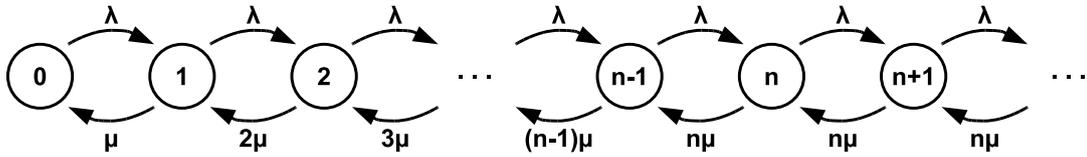


Figure 6: The Markov process of the M/M/n queue.

The equilibrium distribution for the M/M/n queueing model will be derived next. Local balance equations and the normalizing condition are presented in Equations (12)-(14). In both cases, the equilibrium probability for state  $i$  is denoted by  $\pi_i$ .

The analysis of the M/M/n queue is split into two cases where  $i \leq n$  and  $i > n$ . Both cases will be analysed next.

The local balance equation, when  $i \leq n$ :

$$\begin{aligned} \pi_i \lambda &= \pi_{i+1} (i+1) \mu \\ \pi_{i+1} &= \pi_i \frac{\lambda}{(i+1) \mu} = \pi_i n \rho \frac{1}{i+1} \\ \Rightarrow \pi_i &= \pi_0 \frac{(n\rho)^i}{i!} \end{aligned} \quad (12)$$

The local balance equation, when  $i > n$ :

$$\begin{aligned}
\pi_i \lambda &= \pi_{i+1} n \mu \\
\pi_{i+1} &= \pi_i \frac{\lambda}{n \mu} = \pi_i \rho \\
\Rightarrow \pi_i &= \rho^{i-n} \pi_n \\
\Rightarrow \pi_i &= \rho^{i-n} \pi_0 \frac{(n\rho)^n}{n!} \\
\pi_i &= \pi_0 \frac{n^n \rho^i}{n!}
\end{aligned} \tag{13}$$

The normalizing condition will fix the solution as follows.

$$\begin{aligned}
\sum_{i=0}^{\infty} \pi_i &= 1 \\
\pi_0 \left( \sum_{i=0}^{n-1} \frac{(n\rho)^i}{i!} + \sum_{i=n}^{\infty} \frac{n^n \rho^i}{n!} \right) &= 1 \\
\pi_0 \left( \sum_{i=0}^{n-1} \frac{(n\rho)^i}{i!} + \frac{n^n \rho^n}{n!} \sum_{i=n}^{\infty} \rho^{i-n} \right) &= 1
\end{aligned} \tag{14}$$

With a substitution of  $\sum_{i=n}^{\infty} \rho^{i-n} = \frac{1}{1-\rho}$  the normalizing condition will get the form

$$\pi_0 = \left( \sum_{i=0}^{n-1} \frac{(n\rho)^i}{i!} + \frac{(n\rho)^n}{n!(1-\rho)} \right)^{-1}. \tag{15}$$

In Equation (16), the equilibrium probability for case of M/M/n queue is represented when  $i \leq n$ , Equation (17) is for case  $i > n$ . These equations have been derived by combining Equations (12), (13), and (15).

$$\pi_i = \frac{\frac{(n\rho)^i}{i!}}{\left( \sum_{i=0}^{n-1} \frac{(n\rho)^i}{i!} + \frac{(n\rho)^n}{n!(1-\rho)} \right)}, \quad i \leq n \tag{16}$$

$$\pi_i = \frac{\frac{n^n \rho^i}{n!}}{\left( \sum_{i=0}^{n-1} \frac{(n\rho)^i}{i!} + \frac{(n\rho)^n}{n!(1-\rho)} \right)}, \quad i > n \tag{17}$$

## 4.4 Processor Sharing

Processor Sharing is a scheduling policy in which each job in a system gets the same amount of service per time unit. When a new job arrives into the system, the service rate of every job decreases. Respectively, when one of the jobs becomes ready, the service rate of every remaining job increases. PS is a fair policy since every job is treated equally and no priorities are supported. In practice, PS can be implemented with the round-robin principle where each user gets the full service rate one by one for a short while. Round-robin approximates the PS policy when the service period is short comparing to the total service time.

When examining M/M/1 queues, the queue length distribution does not change if the queueing policy is changed from FIFO to PS. The birth-death process of the M/M/1-PS queue for the number of customers in the system is the same as in case of M/M/1-FIFO queue which leads to that the equilibrium distribution for M/M/1-PS queue is the same as described in Section 4.2.

PS has the so-called insensitivity property which means that the mean number of jobs in the system does not depend on the distribution of the service process. It is enough that the mean values of distributions remain the same. Also the mean delay has the same property by Little's law. With Kendall's notation the insensitivity property means that the results derived for the PS policy, for example, in case of an M/M/1 queueing system are valid also for a more general M/G/1 system.

The conditional mean delay  $E[T(x)]$  of a flow with service time  $x$  is by [10] in one server case as follows

$$E[T(x)] = \frac{x}{1 - \rho} \quad (18)$$

This equation is linear with respect to the service time  $x$  so that a job that is twice as large as some other will take twice as long to get served in the system on average.

The mean delay of a flow  $E[T]$  can be derived from Equation (18) by integrating from zero to infinity

$$E[T] = \int_0^{\infty} E[T(x)] f_S(x) dx = \frac{E[S]}{1 - \rho} \quad (19)$$

where  $E[S]$  is the mean service time of a particular flow and  $f_S(x)$  is the pdf of service time  $S$ .

## 4.5 Shortest Remaining Processing Time

The Shortest Remaining Processing Time (SRPT) policy means that the customer with the least amount of work left will be served next. This policy needs to know the job size in advance and jobs must be such that the service can be interrupted and after a while continued from where it was stopped, a so-called pre-emptive system. It has been shown by Schrage [13] that the SRPT policy is optimal with respect to minimizing steady-state mean sojourn time when the M/G/1 system is examined.

The optimality property for the M/G/1 system is easy to understand by thinking about the length of the queue which should be minimized. The fastest way of

shortening the queue is to take out the shortest ones at first. When thinking about the service time, the shortest actually means the fastest. With the SRPT policy, only one job in the system is served at a time, not all like in case of PS.

The queueing time has to be also counted when the mean delay is formulated. The mean delay consists of the waiting time (when the job is not served but is still in the system) and the service time. The formula for conditional mean delay in the M/G/1 system has been derived in [3], originates from [14], and the result is represented in Equation (20). The derivation of this equation is beyond the scope of this thesis and is thus left out. The first part of the equation is about the waiting time of the job and the latter part defines the service time.

$$\begin{aligned} E[T^{SRPT}(t)] &= \frac{\lambda(\int_0^t u^2 f(u) du + t^2(1 - F(x)))}{2(1 - \rho(t))^2} + \int_0^t \frac{1}{1 - \rho(s)} ds \\ &= \frac{\lambda E[(S \wedge t)^2]}{2(1 - \rho(t))^2} + \int_0^t \frac{1}{1 - \rho(s)} ds \end{aligned} \quad (20)$$

where  $\lambda$  is the arrival rate,  $f(t)$  is the probability density function of the service time distribution, and  $\rho(t)$  is the load made up of jobs up to service requirement  $t$ . The term of  $\rho(t)$  has been written out in Equation (21).

$$\rho(t) = \lambda \int_0^t s f(s) ds \quad (21)$$

As already mentioned, SRPT is the optimal policy in case of one server when the mean delay is minimized. The case where more servers are included is more challenging. For a static case, where no new jobs will arrive, SRPT is proven to be the optimal policy [12]. This stands for the case where the system must serve some finite amount of jobs. For the dynamic case with new arrivals in case of more than one server, no analytical results were found. SRPT is assumed to be one of the candidates for the optimal policy in this case. It may as well be that the optimal solution does not even exist or it is some policy other than SRPT in the dynamic case with more than one server.

## 4.6 Elastic traffic

With elastic traffic we mean flow-like traffic which can represent, for example, a file transfer through a network. Delays and the behaviour of individual packets are not the focus of attention when elastic traffic is examined. The transfer of the whole flow is interesting and the transmission can even last for a while as long as it will finish in a reasonable time. Elastic traffic can be used in packet switched networks, while in the CS networks it is not possible to put a circuit on hold or share the capacity between users of the network without closing the circuit.

Elastic traffic can be implemented with a protocol on top of the third layer in Open System Interconnection model (OSI model), network layer, which is responsible for transferring, for example, whole files through the network. The most popular implementation is Transmission Control Protocol (TCP) over Internet Protocol (IP).

In this thesis, the traffic is assumed to be elastic, thus it is possible to use also preemptive policies like SRPT. Elastic traffic in the case of an M/G/1 system in cellular networks has also been studied in [2].

## 5 Models

In this section, the models used in this thesis are introduced. The modelled system is described in three parts. In Section 5.1 the wireless system model and the traffic model is introduced. Section 5.2 is about modelling the wireless channel and Section 5.3 is about mean service time formulas and calculating the probabilities when the user population is classified based on the distance and the size of the flows.

### 5.1 Wireless system model

The starting point of this thesis is scheduling inside one cell in a cellular system like HSDPA. Cell radius is denoted by  $r_1$ . The assumed traffic in the system consists of elastic flows, for example file transfers using TCP. Only the size of each flow (total amount of bits to be transferred) is in focus. Flows arrive into the system according to a Poisson process with rate  $\lambda$  and the sizes of flows ( $X$ ) are independent and identically distributed.

Terminals are assumed to be independently and uniformly distributed in the cell area. The base station of the cell is in the middle of the cell which is modelled as a circle with radius  $r_1$ . The distance  $R$  from the base station to the terminal of a user has the following distribution:

$$P\{R \leq r\} = \frac{1}{\pi r_1^2} \int_0^r 2\pi s \, ds = \left(\frac{r}{r_1}\right)^2, \quad r \leq r_1. \quad (22)$$

For simplicity the cell radius,  $r_1$ , is in this thesis selected to be one so one cell can be represented with a unit circle. With this assumption Equation (22) can be simplified even more to the following form:

$$P\{R \leq r\} = \frac{1}{\pi} \int_0^r 2\pi s \, ds = r^2, \quad r \leq 1. \quad (23)$$

In cellular systems, the scheduler schedules all the transmissions of the users. In HSDPA-like systems transmissions of different users are differentiated by codes. This kind of a system is called CDMA. With CDMA it is possible that several users can send at the same time using the same frequency band without disturbing each other. It is assumed that there are  $K$  different codes in the system, and the scheduler will allocate these codes to the various users. User  $i$  can get  $k_i$  codes with an obvious limit of  $k_i \leq K$ .

### 5.2 Channel model

In modelling of the wireless channel in this thesis, only the path loss is taken into account and, for instance, slow fading and fast fading are left out of scope. It is assumed that the transmission rate depends only on the distance between the base station and the user terminal along with the path loss exponent  $\alpha$ . The relation

between the distance and the transmission rate can be presented as follows:

$$c(r) = \begin{cases} c_0, & r \leq r_0, \\ c_0 \left(\frac{r_0}{r}\right)^\alpha, & r > r_0, \end{cases} \quad (24)$$

where  $c_0$  is the maximum data rate,  $\alpha$  is the path loss exponent, and  $r_0$  is the radius of the inner circle where the maximum transfer rate is achieved. Equation (24) gives the total transmission rate of the system at distance  $r$  from the base station. The transmission rate per code at distance  $r$  is thus  $c(r)/K$ . Codes are assumed to be perfectly orthogonal so the transfer rate depends linearly on the number of codes. The service time of the flow is the size of the flow divided by the transmission rate.

The transmission rate function  $c(r)$  is represented in a graphical form in Figure 7 with  $r_0 = 1/7.94$  and path loss exponent values  $\alpha = 2$  and  $\alpha = 4$ . The value of  $r_0$  is selected based on a table from [5], and this same value is used through this thesis. The maximum data rate  $c_0$  has been set to one and the function is plotted over the unit circle. The case of  $\alpha = 2$  is drawn with a blue solid line and the case of  $\alpha = 4$  with a red dashed line.

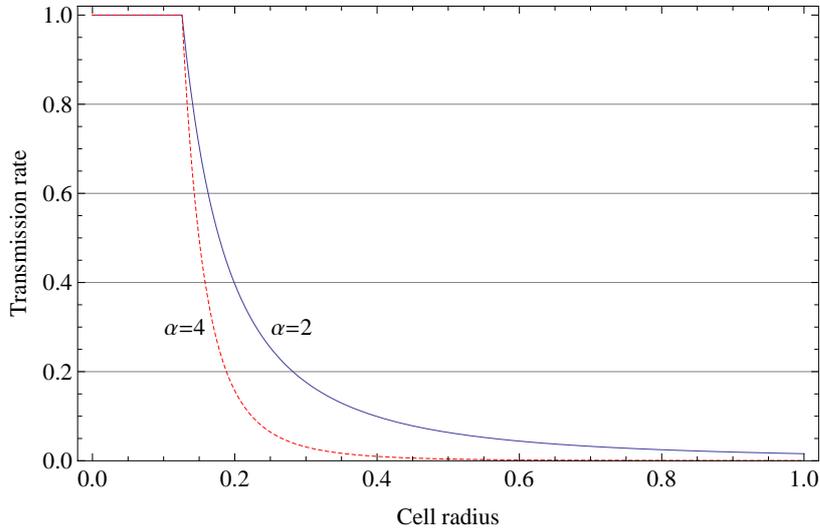


Figure 7: Transmission rate function  $c(r)$  as a function of cell radius with  $\alpha = 2$  and  $\alpha = 4$ .

### 5.3 Analysis of the mean service time

The transmission rate  $C = c(R)$  depends on the distance ( $R$ ) between the user and the base station as in Equation (24). The service time ( $S$ ) of the flow can be derived directly from the size of the flow ( $X$ ) and from the transmission rate function as follows:

$$S = \frac{X}{C}.$$

Assuming that the flow sizes, and the distances between the users and the base station are independent, the mean service time is

$$\frac{1}{\mu} = E[S] = E\left[\frac{X}{C}\right] = E[X]E\left[\frac{1}{C}\right] \quad (25)$$

Thus,

$$E[X] = \frac{\frac{1}{\mu}}{E\left[\frac{1}{C}\right]}. \quad (26)$$

The mean of the inverse of the transfer rate can be derived from  $c(r)$  with the information that users are uniformly distributed over the unit circle (cell area) by integrating over the unit circle as follows

$$E\left[\frac{1}{C}\right] = \int_0^1 \frac{1}{c(r)} 2r dr. \quad (27)$$

By combining Equations (26) and (27), the equation for the mean job size can be derived as follows

$$E[X] = \left(\mu \int_0^1 \frac{2r}{c(r)} dr\right)^{-1}. \quad (28)$$

Next the conditional mean service time conditioned on the distance and the flow size will be examined.

For the mathematical derivation, exponentially distributed flow sizes with the mean  $\mu$  are assumed. Based on the characteristics of the exponential distribution, a random number from the exponential distribution is larger than  $x$  with a probability of

$$P\{X > x\} = e^{-\mu x}.$$

The probability that the distance ( $R$ ) between the user and the base station is smaller than  $r$  is shown in Equation (22), and based on that the probability that the distance belongs to  $dr$  is

$$P\{R \in dr\} = \left(\frac{1}{r_1}\right)^2 2r.$$

Now

$$E[S|X \in A, R \in B] = E\left[\frac{X}{c(R)}|X \in A, R \in B\right] = E[X|X \in A] \cdot E\left[\frac{1}{c(R)}|R \in B\right]$$

where  $A$  and  $B$  are value ranges for both  $X$  and  $R$ .

Based on the previous formulas, equations for the conditional expectations for  $X$  and  $R$  in two cases ( $X \leq x$  and  $X > x$ , same for  $R$ ) will be derived next. By combining these formulas it would be possible to form the equations for the case  $x_1 < X < x_2$  too. In this thesis the division into two classes is enough and more general derivation is left out.

$$\begin{aligned}
1^\circ \quad E[X|X \leq x] &= \frac{1}{1 - e^{-\mu x}} \int_0^x y \cdot \mu e^{-\mu y} dy \\
&= \frac{1}{1 - e^{-\mu x}} \left( \int_0^x y \cdot (-e^{-\mu y}) + \frac{1}{\mu} \int_0^x \mu e^{-\mu y} dy \right) \\
&= \frac{1}{1 - e^{-\mu x}} \left( -xe^{-\mu x} + \frac{1}{\mu} (1 - e^{-\mu x}) \right) \\
&= \frac{1}{\mu} \cdot \frac{1 - e^{-\mu x} - \mu x e^{-\mu x}}{1 - e^{-\mu x}}
\end{aligned}$$

$$\begin{aligned}
2^\circ \quad E[X|X > x] &= \frac{1}{e^{-\mu x}} \int_x^\infty y \cdot \mu e^{-\mu y} dy \\
&= \frac{1}{e^{-\mu x}} \left( \int_x^\infty y (-e^{-\mu y}) + \frac{1}{\mu} \int_x^\infty \mu e^{-\mu y} dy \right) \\
&= \frac{1}{e^{-\mu x}} \left( xe^{-\mu x} + \frac{1}{\mu} e^{-\mu x} \right) \\
&= x + \frac{1}{\mu}
\end{aligned}$$

$$3^\circ \quad E\left[\frac{1}{c(R)} | R \leq r_0\right] = \frac{1}{r_0}$$

$$\begin{aligned}
4^\circ \quad E\left[\frac{1}{c(R)} | R > r_0\right] &= \frac{1}{1 - \left(\frac{r_0}{r_1}\right)^2} \int_{r_0}^{r_1} \frac{1}{c_0} \left(\frac{r}{r_0}\right)^\alpha \left(\frac{1}{r_1}\right)^2 2r dr \\
&= \frac{1}{1 - \left(\frac{r_0}{r_1}\right)^2} \cdot \frac{2}{c_0} \left(\frac{1}{r_0}\right)^\alpha \left(\frac{1}{r_1}\right)^2 \int_{r_0}^{r_1} r^{\alpha+1} dr \\
&= \frac{1}{1 - \left(\frac{r_0}{r_1}\right)^2} \cdot \frac{2}{c_0} \left(\frac{1}{r_0}\right)^\alpha \left(\frac{1}{r_1}\right)^2 \int_{r_0}^{r_1} \frac{1}{\alpha+2} r^{\alpha+2} \\
&= \frac{1}{1 - \left(\frac{r_0}{r_1}\right)^2} \cdot \frac{2}{c_0} \left(\frac{1}{r_0}\right)^\alpha \left(\frac{1}{r_1}\right)^2 \frac{1}{\alpha+2} (r_1^{\alpha+2} - r_0^{\alpha+2}) \\
&= \frac{1}{1 - \left(\frac{r_0}{r_1}\right)^2} \cdot \frac{1}{c_0} \frac{2}{\alpha+2} \left( \left(\frac{r_1}{r_0}\right)^\alpha - \left(\frac{r_0}{r_1}\right)^2 \right)
\end{aligned}$$

## 6 Homogeneous case

In this section, the so-called homogeneous case will be examined. The user population is assumed to be homogeneous which means that every user in the system has equal characteristics. At first, in Section 6.1, the assumptions made for this case will be introduced more exactly. Different policies used in the homogeneous case are clarified in Section 6.2. Section 6.3 is about the implementation of the simulator, and the verification of the simulator is introduced in Section 6.4. Results achieved in different cases under the homogeneous assumption are represented in Section 6.5. At the end, there is a short summary about the homogeneous case and the overall results.

### 6.1 Assumptions

It has been assumed that there are  $K$  codes in the system and all codes are similar. Each user device has equal characteristics with respect to the terminal constraints, i.e.,  $k_i = k$  for all users  $i$ . In practice this means that every user device in the network can handle the same amount of codes ( $k$ ) so there is not any order of priority among the users in this sense. With these assumptions made, the system can serve simultaneously  $n = K/k$  jobs and this value can be considered as the number of servers.

Arrivals are assumed to be independent and exponentially distributed, with the mean  $1/\lambda$ . The service rate  $\mu$  can be derived from Equation (11) with the given load value  $\rho$ , the arrival rate  $\lambda$ , and the number of servers  $n$  as follows

$$\mu = \frac{\lambda}{n\rho} .$$

The same transmission rate function  $c(r)$  (Equation (24)) is used for all users, and thus every user at the distance  $r$  from the base station will get an equal transmission rate. In this thesis  $r_0 = 1/7.94$  will be used [5] and the path loss exponent  $\alpha$  will be varying between two and four. Flow sizes are assumed to be independent and exponentially distributed. The mean size of the flows is calculated from Equation (28) with the given  $\mu$  and  $c(r)$ .

### 6.2 Policies

The Baseline and SRPT policies in the homogeneous case are introduced in Sections 6.2.1 and 6.2.2.

#### 6.2.1 Baseline policy

In the homogeneous case, the analytical results derived for the M/M/n-PS queueing model are used as a baseline when the performance improvement of SRPT policy is examined. Comparisons are done based on mean delays (same as queue length in case of  $\lambda=1$ ).

PS is used as a baseline policy and all comparisons with this policy are based on the analytical results and no simulations have been needed.

### 6.2.2 SRPT policy

In the homogeneous case, it is assumed that the SRPT policy takes  $n$  shortest (with respect to time) jobs into service where  $n = K/k$ . If an arriving job is shorter in the service time than the in-service job with the longest remaining service time, the latter job will be put on hold and the new job will get service immediately. Thus, the system will always serve the  $n$  shortest jobs.

When all users in the system have the same characteristics, the selection process is straightforward. One sorting based on the remaining processing times is enough and  $n$  first jobs from this ranking will get service at each point of time. After each arrival and departure the system will check which jobs will be served. For the multiserver SRPT policy, simulations are needed since no analytical results have been derived for this kind of system.

## 6.3 Implementation of the simulator

Simulations have been done using Wolfram Mathematica software (version 7.0.1.0) and no ready-made code is used. The simulator is event based, and jobs will arrive and depart from the system. The simulator takes four parameters which are represented in Table 2. The length of the simulation is the stopping condition of the simulator. The load of the system, together with the arrival rate and the number of servers, is used for defining the departure rate of the system.

Table 2: Parameters of the simulator in the homogeneous case

Parameter	Explanation
$T$	Length of the simulation
$\rho$	Load of the system
$\lambda$	Arrival rate of jobs
$K$	Number of servers

Besides the primary simulator function (*simulate()*) there are two other functions, one for drawing a random numbers from the exponential distribution (*expVar()*) and the other for defining the transmission rate (*dataRate()*).

### 6.3.1 Supporting functions

*expVar(a)* This is a simple function which returns a random number from an exponential distribution having the mean value of  $a$ . The random numbers needed by *expVar()* are generated using the *RandomReal()* function. *expVar()* should be changed if some other distribution will be simulated. In the current implementation, all randomized operations call this function.

*dataRate(r)* This function implements Equation (24) and returns the transmission rate at the given radius  $r$  from the base station. Parameters  $\alpha$ ,  $c_0$ , and  $r_0$  are defined inside the function.

### 6.3.2 Simulate() function

*simulate(T, rho, lambda, n)* This function does the actual simulation based on the given parameters. The parameters are described in Section 6.3 . At first the function calculates the departure rate as described in Section 5.3. Using calculated departure rate, the mean job size is calculated by Equation (28).

The simulator keeps track of which jobs are in service, how much each of those jobs have service time left, and when the jobs would be finished if no interruptions will occur. Mathematica lists are used as a storage format, and the lists are initialized at first. There is one list for the jobs in the queue and another for the jobs which have already been finished. Both lists have basically the same format, the list for the finished jobs has one extra field at the end. The format of the lists is as follows:

{ index of the job (starting from the value one),  
 arrival time of the job,  
 the size of the job at the beginning,  
 current size of the job,  
 transfer rate of the job,  
 remaining service time,  
 departure time of the job (only for the queue of the finished jobs) }

The arrival time of the first job is selected using the *expVar()* function and after that the main *while* loop is started. The *while* loop is divided into two parts which handle the arrivals and departures.

When a new job arrives, it is taken into the system and at first the radius and the job size are selected for the new job. The statuses of the other jobs are updated and the decision of which job will get service is done next. If there is at least one server free, the new job will get service immediately. Otherwise the function will test whether the new job is short enough to get service without queueing. Before continuing the *while* loop, the newest job is added into the list that keeps track of all jobs in the system. At the end of the arrival part, the arrival time for the next job is selected.

In departure cases, the function will search the job which is ending and updates the queue of finished jobs accordingly. Indexes of the internal structures are also updated and after that the servers are filled again with jobs based on the SRPT policy order.

When the simulation time is reached, the *simulate* function will update the status of each job in the service and then mark the rest of jobs to leave the system. The simulator does not know or care about the transient period at the beginning

and taking this into account is left for the handler of the results. A full list of served jobs during the simulation is returned at the end.

## 6.4 Verification of the simulator

The SRPT policy has been simulated and the results from the simulator have been compared with the analytical results. The simulator has been extended piece by piece during this thesis. New features have been implemented one by one and only after the base has been verified carefully.

**M/M/1-FIFO** At first a basic M/M/1 queueing model was implemented with a First In First Out (FIFO) policy. This is one of the simplest queueing models and thus a good starting point. The main structure of the final version of the simulator will also be similar so it is possible to start with a very simple model and extend it piece by piece. The first M/M/1-FIFO simulator has been verified comparing the results of the mean delay with analytical results derived for the particular queueing model. Simulated results (dots with error bars) are compared with analytical results (solid line) in Figure 8. The length of the simulation has been  $10^6$  arrivals and 50 % of that is assumed for a transient time at the beginning and has not been taken into account when the results have been calculated. The mean delay has been calculated from 500 000 samples for each value of the load ( $\rho$ ) and compared with the analytical result of the M/M/1 mean delay. The simulation has been repeated 10 times with the same parameters to define the confidence interval. In this thesis, the 95 % confidence interval will be used when error margins are represented. For the arrival rate, the value  $\lambda = 1$  has been used in the simulation. This means that the result can be assumed to be either the mean delay of the system or the mean queue length of the system. The simulator seems to be very accurate based on this testing.

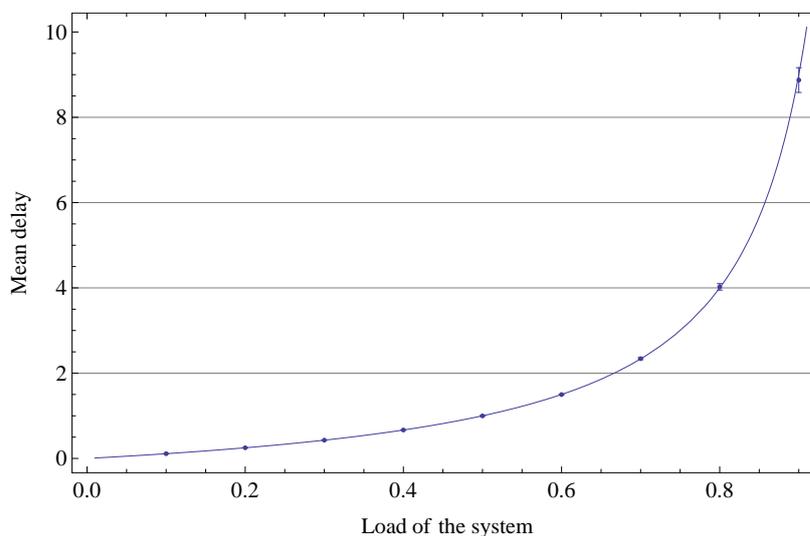


Figure 8: Verification of the results from the M/M/1-FIFO simulator.

**M/M/1-SRPT** Next the FIFO policy was replaced with the SRPT policy which means that now jobs are served in order where the shortest job (measured in time) gets served first. The results from this version of the simulator are compared with the analytical results (represented in Section 4.5) for the SRPT in the case of one server. This comparison is shown in Figure 9 where the analytical result is the solid line and the simulated results are the dots with error bars.

The length of the simulation has been 200 000 arrivals for each value of the load ( $\rho$ ) and 50 % of the samples are ignored because of the transient period at the beginning. The simulation is repeated ten times to determine the confidence interval. In the simulation the flow sizes are exponentially distributed with the mean of one. The simulator seems to be very accurate based on this testing.

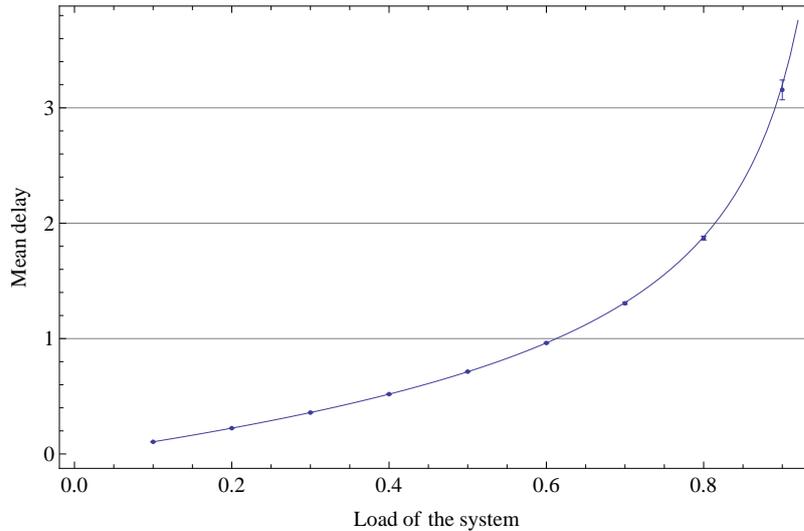


Figure 9: Verification of the results from the M/M/1-SRPT simulator.

**M/M/n-SRPT** The next extension to the simulator was the support for simulating the multiserver SRPT policy. Other assumptions made in the previous case remain and only the number of servers is changing. With  $n$  servers,  $n$  shortest (measured in time) jobs are taken into service. One server is serving only one job at a time.

The simulator does not have any limitations to  $n$ , the number of servers. The same testing procedure has been done like in the previous case with  $n = 1$ . With this assumption, the simulator must produce the same results as in the previous case, and it does. There are no analytical results for the SRPT policy in case of more than one server. Thus the numerical verification of the model is difficult. Through careful debugging the simulator looks like it is working just like it was designed and the results with  $n > 1$  make sense.

**M/M/n-SRPT with the distance information** In the previous case a constant transfer rate was assumed which is not a realistic assumption. Next the simulator

was extended with the distance-dependent transfer rate. The transfer rate will decrease as a function of the distance from the base station like represented in Equation (24). This extension has been tested by setting the  $c(r)$  to a constant value of one and the simulator produced the same results as before this extension. The test can also be done by setting the radius of the inner circle to one. Results do not differ from the previous case.

Based on these tests done, the simulator seems to work like planned and the results produced by it can be assumed to be accurate and correct.

## 6.5 Results

In this section, four different aspects are considered. At first in Section 6.5.1, the length of the transient is examined. In Section 6.5.2 it will be examined how SRPT improves the performance of the system compared with the fair PS policy. The results are given in absolute value form and also compared relatively with the fair policy. Section 6.5.3 slightly modifies the SRPT policy used in Section 6.5.2. In Section 6.5.4 the normal SRPT is discussed again. This time, the performance improvement is examined as a function of  $K$  with different values of the load of the system.

### 6.5.1 Transient period

At the beginning of the simulation results, the system oscillates until a steady state is reached. It is not reasonable to take those highly varying values into account when the steady state behaviour of the system is examined. The length of the transient period has been simulated using a parameter set presented in Table 3. The simulation has been repeated 10 times with the same values.

Table 3: Parameters for the transient period simulation in the homogeneous case

Name	Parameter	Value
Length of the simulation (time units)	$T$	300 000
Load of the system	$\rho$	0.9
Arrival rate of jobs	$\lambda$	1
Number of servers	$n$	3
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Individual simulation runs		10
Flow sizes		exponentially distributed

The window, 1 000 time units in length, has been slid through each of the 10 individual simulations and, the mean delay is calculated at each point of the window from these. From these 10 individual averages, one mean delay value for each window value has been calculated. The first window includes the first 1 000 time units of

the simulations, and after that the window is moved so that the next 1 000 time units are taken into account. These calculations of the evolution of the mean delay are represented in Figure 10. In the same figure the mean delay calculated over the latter  $\frac{2}{3}$  of the simulation is also shown.

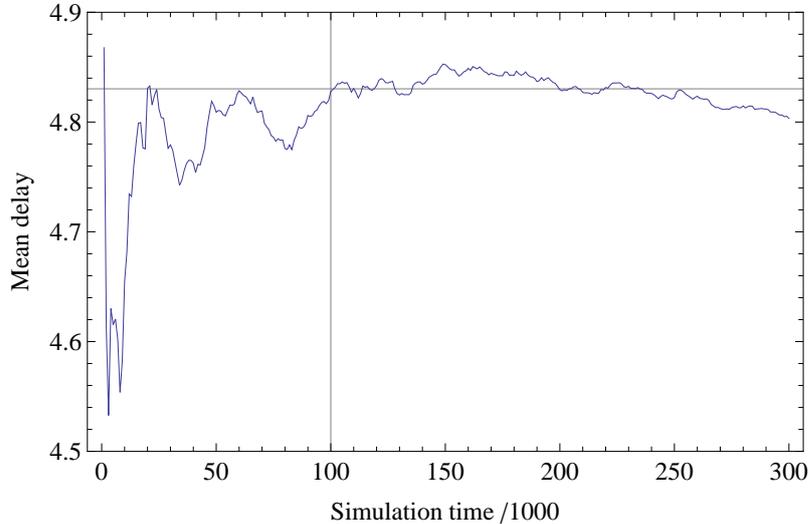


Figure 10: The evolution of the mean delay in the homogeneous case as a function of the simulation time. The horizontal line is the mean delay calculated from the later two thirds of the simulation.

The parameter set used to define the length of the transient period is one of the most challenging ones (high load,  $\rho = 0.9$ ) used in this homogeneous case. For that reason it is suitable to assume that this transient period is long enough to be used in all simulations done for the homogeneous case. For the rest of the simulations in the homogeneous case, the transient period of 100 000 time units will be used.

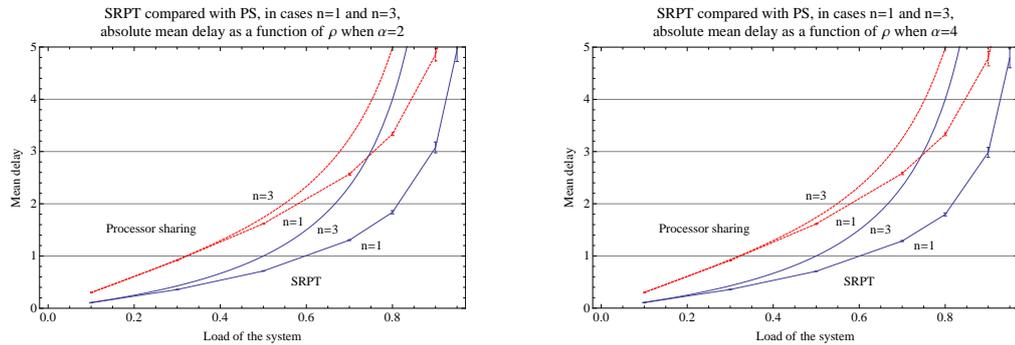
### 6.5.2 SRPT compared with PS

In this section, the performance improvement of the SRPT policy is examined. Simulations have been done with a parameter set presented in Table 4. The same parameter set is simulated 10 times to determine the confidence interval.

The results of the simulations and the calculated absolute values are represented in Figure 11. Different values of the load of the system ( $\rho$ ) are drawn as their own lines. Cases with  $n = 3$  are shown with dashed lines in the figures. It can be seen from the figure that the performance improvement of the SRPT policy is the most notable with high values of  $\rho$ . With low values of  $\rho$  there is not much difference between PS and SRPT. However even in that case there is a distinguishable difference between policies in favour of SRPT. The effect of path loss exponent  $\alpha$  is surprisingly low and it does not even distinguish from the error margins of Figure 11. Path loss exponent affects the transmission rate function but the effect seems to be

Table 4: Parameters of the simulator, SRPT compared with PS

Name	Parameter	Value
Length of the simulation (time units)	$T$	200 000
Load of the system	$\rho$	0.1 ... 0.95
Arrival rate of jobs	$\lambda$	1
Number of servers	$n$	1, 3
Path loss exponent	$\alpha$	2, 4
Centre area radius	$r_0$	1/7.94
Transient period	<i>transient</i>	100 000
Individual simulation runs		10
Flow sizes		exponentially distributed

Figure 11: The absolute values of the mean delay for SRPT and PS with  $\alpha = 2$  (left) and  $\alpha = 4$  (right).

minimal at least in the homogeneous case. This means that the results would apply for urban area as well as for rural areas.

The relative improvement of the SRPT policy compared with the PS policy is shown in Figure 12. From this figure it is easy to see the effect of  $\rho$  in the performance improvement. With one server SRPT improves the performance with all values of  $\rho$ . Already with  $n = 3$ , SRPT does not make much difference with the load values  $\rho < 0.5$  compared with PS. It seems that the performance improvement of SRPT is decreasing when there are more servers in the system. This behaviour will be examined in more detail in Section 6.5.4.

### 6.5.3 Modified SRPT with selections based on bit sizes

In the previous section, the basic SRPT policy was examined. Now the policy is changed so that it will prioritize the flows for which the bit sizes are the lowest. With this modification it is not needed to know the transmission rate of each flow which was assumed the previous section. Simulations have been done using the parameter set which is given in Table 5. Parameters are the same as in the previous section, but now only the case  $\alpha = 2$  has been simulated. In addition the simulations have been done with  $n = 3$ .

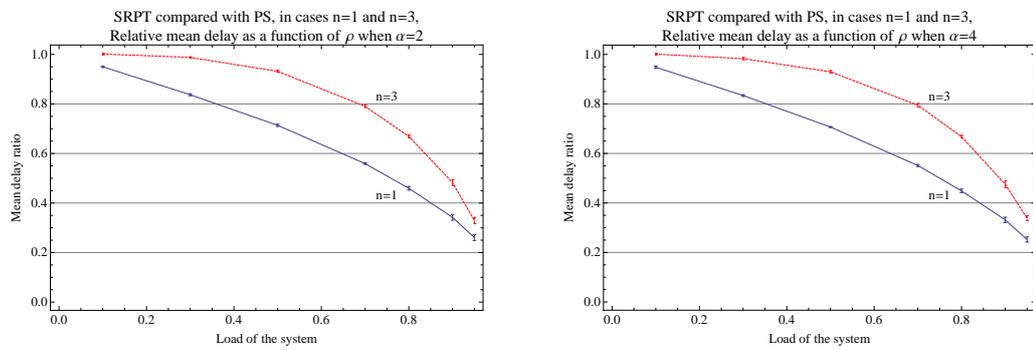


Figure 12: The relative performance improvement of SRPT with  $\alpha = 2$  (left) and  $\alpha = 4$  (right).

Table 5: Parameters of the simulator, modified SRPT compared with PS

Name	Parameter	Value
Length of the simulation (time units)	$T$	200 000
Load of the system	$\rho$	0.1 ... 0.95
Arrival rate of jobs	$\lambda$	1
Number of servers	$n$	1, 2, 3
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Transient period	<i>transient</i>	100 000
Individual simulation runs		10
Flow sizes		exponentially distributed

The simulated results are compared with the analytical results of the PS policy with the same assumptions for  $\rho$  and  $n$ . Figure 13 shows the results of the modified SRPT and also the basic SRPT policy with the same parameters. Three curves, one for each value of  $n$ , are shown.

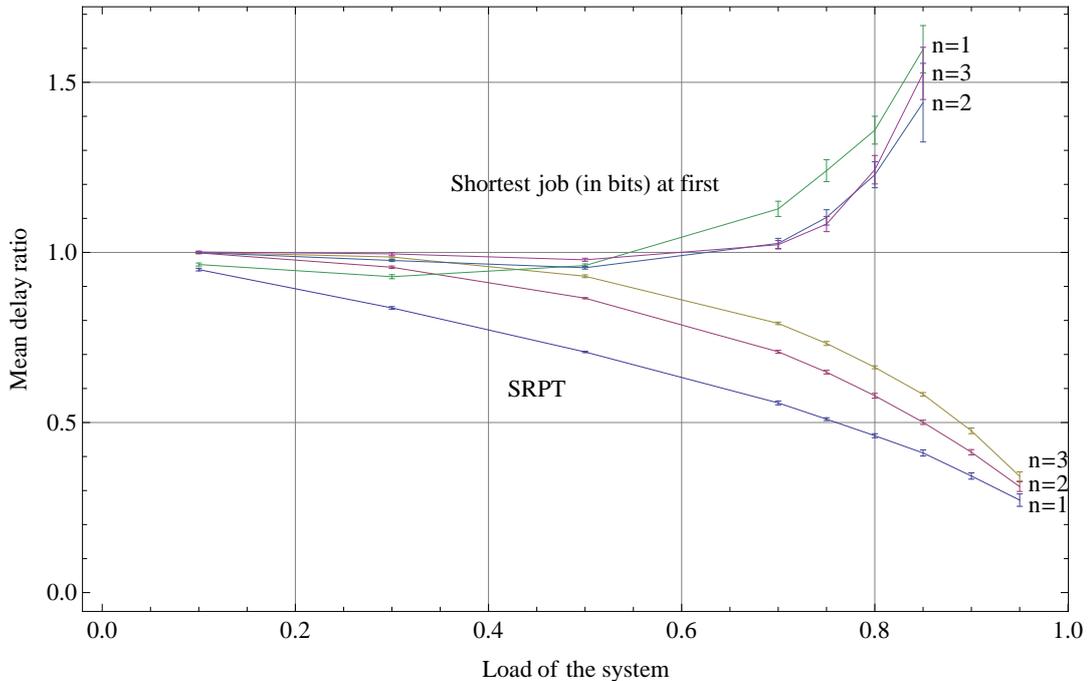


Figure 13: Mean delay of SRPT and modified SRPT compared with PS as a function of the load of the system with 95 % confidence interwalls.

The performance of the modified SRPT policy seems to be much worse than the basic SRPT. With load values more than 0.5 the modified version is even worse than the PS policy. The variation in different policies is smaller with low values (below 0.5) of  $\rho$ . With higher values of  $\rho$  the basic SRPT outperforms the modified version systematically. It is not worth selecting the flows by the bit size. If the transmission

rate is not known and it is hard to measure or determine, it is better to use PS than this kind of a modified version of SRPT.

#### 6.5.4 Effect of $K$ for the performance improvement

Based on Figure 13 the performance improvement of the SRPT policy seems to be decreasing when the number of servers, or codes, increases. This aspect will be examined further in this section.

The case has been simulated with different values of  $\rho$  and  $n = K/k$  is increased from one to 25. The other simulation parameters are shown in Table 6. The simulation results have been compared with the analytical results of the PS policy with the same values of  $\rho$  and  $n$ .

Table 6: Parameters of the simulator, effect of  $K$  for the performance improvement

Name	Parameter	Value
Length of the simulation (time units)	$T$	200 000
Load of the system	$\rho$	0.5, 0.8, 0.9, 0.95
Arrival rate of jobs	$\lambda$	1
Number of servers	$n$	1 ... 25
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Transient period	<i>transient</i>	$0.5T$
Individual simulation runs		10
Flow sizes		exponentially distributed

The simulated results are shown in Figure 14. From the figure it can be seen that the performance gain which can be achieved with the SRPT policy is largest when only a few servers are used and  $\rho$  is relatively high. With smaller values of  $\rho$ , the advantage of SRPT decreased to almost zero already with values of  $n$  less than ten. With high loads, there is still a remarkable improvement even if the value of  $n$  is more than 20.

When the number of servers is increased, it is more and more probable that there are not enough jobs in the system to feed all the servers. For that reason, some servers will be idle which of course reduces the efficiency of the system. With high loads there are on average more jobs in the system, so the probability of an idle server is smaller.

Based on Figure 14, the results of every value of the load seem to asymptotically reach the value of the PS policy. The case has been simulated with values of  $n \leq 25$  to keep the simulation times reasonable. The same reason applies for the number of different values of  $\rho$ .

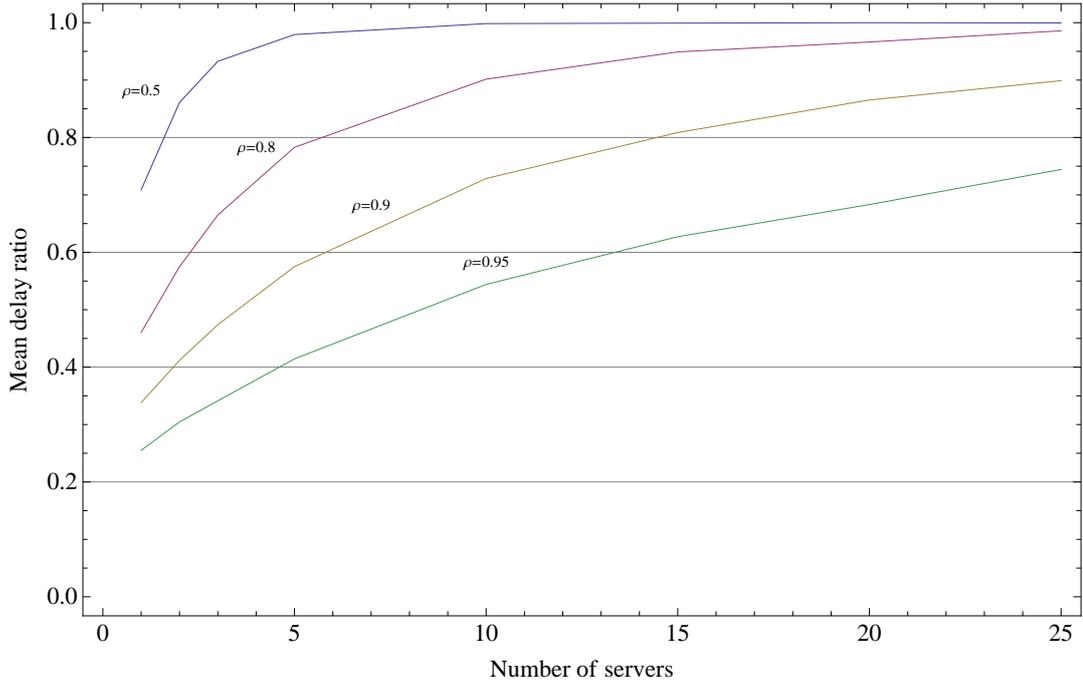


Figure 14: The effect of  $n = K/k$  on the performance improvement of SRPT.

## 6.6 Summary

In this section, the so-called homogeneous case has been examined. It is assumed that every user in the system has equal characteristics.

It was not possible to do long simulation runs since, for example, one round of a single parameter set (one of the ten rounds done with the same parameter set) takes with the hardware used (Intel®Core™2 Quad CPU Q8200 2.33 GHz, 2.96 GB RAM) about one hour. However, the license of the Mathematica software allowed using only one core which should be taken into account when comparing simulation times with other hardware. All simulations have been done ten times with identical parameters and there have been 7 to 9 different load values in each case.

SRPT seems to improve the performance of the system compared to PS in all cases which have been examined. The improvement seems to be highest with higher values of  $\rho$ . When the number of servers (or codes in the system) is increasing the improvement is decreasing. This kind of behaviour is well understandable and these results have confirmed these expectations. An analytical proving of this lies beyond the scope of this thesis and thus is left out.

## 7 Heterogeneous case

In this section, the so-called heterogeneous case will be examined. In contrast to the homogeneous case, it is no longer assumed that all user devices have the same characteristics. It is possible that the number of codes, which each user is able to use, varies between the user classes. After this modification it is not easy to predict how different policies will behave. There are no suitable analytical results for PS in this kind of scenario and also the PS policy must be simulated.

In Section 7.1 assumptions made for this case are introduced. The PS policy in the heterogeneous case will be introduced in Section 7.2.1 and similarly SRPT is discussed in Section 7.2.2. Section 7.3 is about the implementation of the simulator, and the verification of the simulator is presented in Section 7.4. The results achieved in the different cases under the heterogeneous assumptions are given in Section 7.5. At the end, there is a short summary about the heterogeneous case and the overall results.

### 7.1 Assumptions

It is assumed, just like in the homogeneous case, that there are altogether  $K$  codes in the system and all codes are similar. Users are now divided into  $J$  classes. Users in a certain class have the same characteristics related to the number of codes which they are able to use. In class  $j$ , every user is able to use  $k_j$  codes. Other assumptions remain as in the homogeneous case.

The system may give fewer than  $k_j$  codes for a user belonging to class  $j$ . However, this is not always possible and in that case the system may give less than  $k_j$  codes for the user. This can happen when it is not possible to divide  $K$  equally among the classes. For example, when  $K = 17$  and there is already one user in the system from each of the two classes with  $k_1 = 5$  and  $k_2 = 10$ . There are two codes left and it would not be reasonable to leave these codes unused so the system will allocate these codes to some user. The transfer rate of a particular user depends on the number of allocated codes and, just like in the homogeneous case, from the distance between the user and the base station.

### 7.2 Policies

This section is about the studied policies in the heterogeneous case. The PS and SRPT policies in this case are introduced in Sections 7.2.1 and 7.2.2.

#### 7.2.1 Baseline policy

With user classes, it is no longer enough to share the transmission time equally among the users one by one. The number of codes used by each user must be taken into account. However, the fair sharing is still based on time and the distance does not have an effect on the scheduling.

Next an equation for the PS policy will be derived. The derivation can be divided into two cases, there can be more or fewer than  $K$  codes in the system. The equation

is derived for the case of two classes, since with more than two classes the procedure would be similar. The transmission rate is given by  $c(r)$  which is the transmission rate with all  $K$  codes. The transmission rate per single code is then  $c(r)/K$ .

It is assumed that the users are divided into two classes and the users in the first class can use  $k_1$  codes and the users in the second class can use a maximum of  $k_2$  codes. There are  $n_1$  users in the first class and  $n_2$  users in the second one. In the first case, the number of codes in use in the system is fewer than or equal to  $K$ ,  $n_1k_1 + n_2k_2 \leq K$ . In this case, each user gets served with the amount of codes that the user is able to use ( $k_j$  for each user class). The transmission rate which a class- $j$  user will get is then  $(k_j/K) c(r)$ .

In a case where  $n_1k_1 + n_2k_2 > K$ , there are not enough codes in the system to give the requested number of codes to every user all the time. The situation can be illustrated with Figure 15. It can be thought that the window with the length  $K$  moves through all requested codes (total  $n_1k_1 + n_2k_2$ ) and thus every job in the class  $j$  will get served by zero to  $k_j$  codes at each step depending on the state of the window.

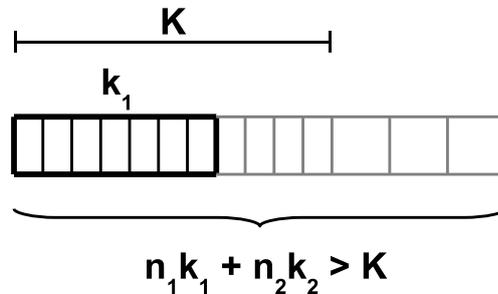


Figure 15: Round-robin in the heterogeneous case.

The movement of the window can be visualized like in Figure 16 which presents how one job of the class one will get served when the window is moving. The service rate of a single job is cyclic and it will repeat as long as  $n_1k_1 + n_2k_2$  stays the same. In Figure 16, one period of a class one job is shown. For every job in the same class, the figure would be the same, and between the classes there are differences at the scaling of the y-axis. The grey area represents the service received by the job. From the figure it is possible to calculate the number of codes that one job will get during one windowing period,  $(Kk_1)/(n_1k_1 + n_2k_2)$  (the grey area). The result has the same form for the second class as well, only the value of  $k$  in the numerator will change. By multiplying this by the transmission rate per one code,  $c(r)/K$ , we get the transmission rate  $(k_j/(n_1k_1 + n_2k_2)) c(r)$  for the job in the class  $j$ .

By summarizing these two cases we can form an equation for a transmission rate of a class- $j$  user at distance  $r$  from the base station, in case of two classes and with the transmission rate function  $c(r)$ , as follows

$$\frac{k_j}{\max(K, n_1k_1 + n_2k_2)} c(r). \quad (29)$$

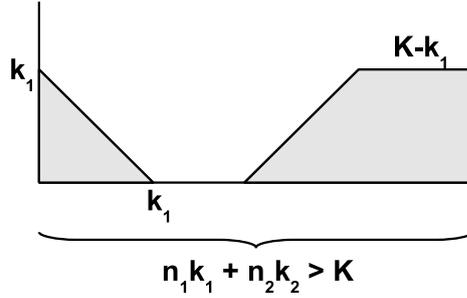


Figure 16: Defining the transmission rate for a class 1 user.

The Equation (29) can be interpreted as a Discriminatory Processor Sharing (DPS) system. In this thesis, term PS will be used when this kind of system is discussed about.

### 7.2.2 SRPT policy

SRPT selects jobs with the shortest remaining processing time at the moment of selection, including the information about the number of available codes, into the service. In the homogeneous case, it was easy to select certain jobs into service since there was always room for an entire job, or the system was already full. In the heterogeneous case it is possible that the last job selected into service will not get the requested number of codes and instead it must be satisfied with fewer.

In SRPT cases, the ranking of the jobs is dynamic because it is not possible to rank the jobs in the system only once and after that do the selection process. As long as every job would fit into service with all codes the job is able to use, the PS-kind static ranking is enough. Immediately, when it is possible that some job would not fit into service with full codes, all jobs must be ranked again. At this time, the maximum number of codes each job would get will be the number of available codes in the system.

This can be clarified with a short example where the total number of codes  $K = 25$  and there are two user classes: class 1 with 5 and class 2 with 15 codes. It is assumed that there are two users in each class, a total four users in the system. The first selection is simple since each class would get service at full rate of their own class. Assume a class-2 user is selected into service at first. After this, there are 10 codes available. The class-1 users would get service at full rate if they would be selected into service next. If the class-2 user would be selected into service, the system can not offer more than 10 codes for that user. At this point, the ranking of jobs is done based on the number of available codes and the service time of the class 2 user is calculated with 10 codes. The class-1 users are treated just as before.

With a notation of the total number of codes in the system ( $K$ ) and the number of already reserved codes ( $L$ ), the service rate of a class  $j$  user at the distance  $r$  from the base station, in case of transmission rate function  $c(r)$ , can be formulated

as follows

$$\frac{\min(K - L, k_j)}{K} c(r). \quad (30)$$

### 7.3 Implementation of the simulator

The basic structure of the simulator is the same as in the homogeneous case, introduced in Section 6.3. In the heterogeneous case the simulator has been extended to support also user classes. In this thesis it has been enough to assume equal probabilities between the classes. No weights between classes are supported even though it would be straightforward to add this into the code of the simulator.

The simulator takes five parameters which are presented in Table 7. Most of the parameters are exactly the same as in the homogeneous case. The parameter *classes* defines the number of codes which a user belonging to a particular class is able to use at maximum.

Table 7: Parameters of the simulator in the heterogeneous case

Parameter	Explanation
$T$	Length of the simulation
$\rho$	Load of the system
$\lambda$	Arrival rate of jobs
$K$	Total number of codes in the system
$classes\{k_1, k_2, \dots, k_j\}$	List of codes which users in each class are able to use

In the simulator it is not possible to sort jobs once (as in the homogeneous case) and then select as many jobs into service as there are codes available. After every new selection it must be checked whether there is still room for any of those jobs in the system. If it is possible to take any of the jobs in the system into service with full amount of codes, there is no need to update the ranking. In the simulator, this has been implemented for simplicity so that the selection is based on the current number of available codes at every point in time.

In the heterogeneous case, also the PS simulator has been done and the parameters needed are the same as shown in Table 7. The source codes for both SRPT and PS simulators are given in Appendixes A and B.

### 7.4 Verification of the simulator

The new version of the simulator has been verified by extensive testing procedures. The results from the new version of the simulator have been compared with the results produced by the simulator of the homogeneous case. The homogeneous case is a special case of the heterogeneous case, the case of one user class. The simulator of the heterogeneous case can be tested against the results of homogeneous case whit

suitable parameters. The number of servers must be equal in both cases, thus it must hold that

$$n_{homogeneous\ case} = \frac{K_{heterogeneous\ case}}{k_{1heterogeneous\ case}}$$

where in the heterogeneous case only one (or several equal) user class exists and users in that class can use at maximum  $k_1$  codes.

The handling of various user classes has been tested by simulating the homogeneous case by giving two equal user classes to the heterogeneous simulator. The results are exactly the same as in the previous test when the other parameters are the same. When testing other combinations of user classes with different values of  $k_j$ , the results seem to be well in line and it is reasonable to assume that the simulator works like planned. A similar verification process has also been done for PS simulator when simulated results have been compared with analytical results in case of one server. The handling of various classes has been tested like with the SRPT simulator.

## 7.5 Results

This section considers three different aspects of the heterogeneous case. Section 7.5.1 is about defining the length of the transient period similarly as in the homogeneous case. In Section 7.5.2, the performance improvement of the SRPT with terminal constraints is compared with the PS. The conditional mean delay with respect to the size and distance in case of the heterogeneous case is examined in Section 7.5.3.

### 7.5.1 Transient period

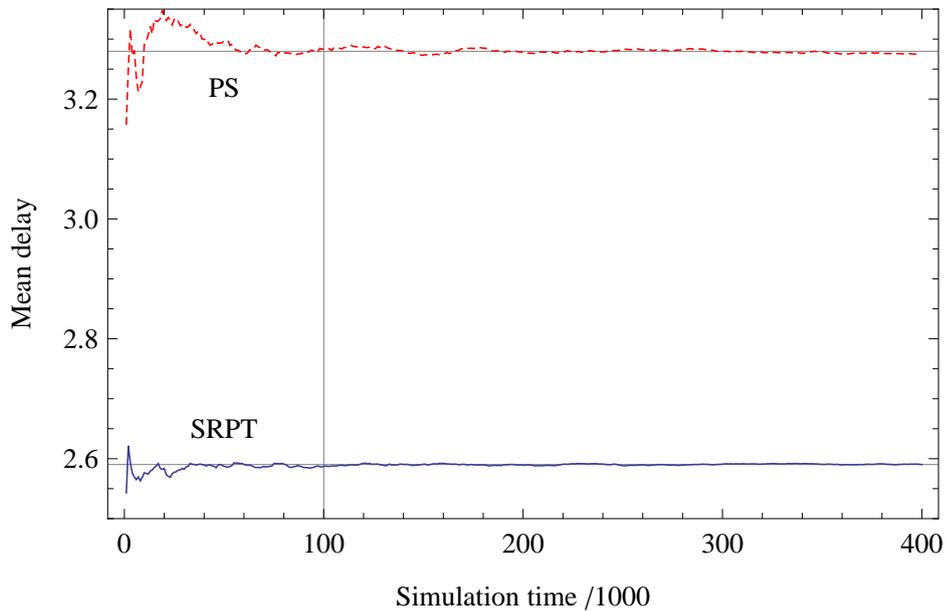
The length of the transient period in the homogeneous case was examined in Section 6.5.1. A similar examination has been done for the heterogeneous case. At this time, a couple of simulation parameter sets, which are also used later, are examined in a sense of the length of the transient period. In the heterogeneous case, the PS policy will be simulated and the transient behaviour will be examined also in that case. Simulation parameters are shown in Table 8.

Figure 17 shows the evolution of the mean delay of the SRPT and PS policy with  $\rho = 0.7$ . The curve of the PS policy is at the top (dashed line) of the figure and the curve of SRPT at the bottom. The y-axis values are absolute values of the mean delay of the system. The x-axis values show the number of the window,  $[0, T/1000]$ . The mean delay calculated from the latter 75 % of the simulation has also been plotted in the same figure. The transient period has been bounded with a vertical line. The case with  $\rho = 0.9$  is presented in Figure 18 and the other parameters are the same as in the case with  $\rho = 0.7$ .

From Figures 17 and 18 it can be seen that the transient period will take approximately 100 000 time units. With  $\rho = 0.7$  the length of the transient period is almost half of the length of the transient period of the  $\rho = 0.9$  case. In a SRPT case, even

Table 8: Parameters for the transient period simulation in the heterogeneous case

Name	Parameter	Value
Length of the simulation (time units)	$T$	400 000
Load of the system	$\rho$	0.7, 0.9
Arrival rate of jobs	$\lambda$	1
Total number of codes in the system	$K$	20
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Classes		{5,15}
Individual simulation runs		10
Flow sizes		exponentially distributed
Window size		1 000 time units

Figure 17: The evolution of the mean delay in the heterogeneous case when the load of the system is  $\rho = 0.7$  as a function of the simulation time.

a shorter period would be sufficient with both load values. With the PS policy, the system needs more time to reach a stable state and it would be appropriate to use a longer transient period than 100 000 time units at least if the load of the system is more than  $\rho = 0.9$ . For the rest of the simulations in the heterogeneous case, the transient period of 100 000 time units will be used.

### 7.5.2 SRPT compared with PS

For the heterogeneous case, similar simulations have been done as in the homogeneous case in Section 6.5.2, but this time the situation of the two classes will be examined. The simulations have been done with a parameter set which is pre-

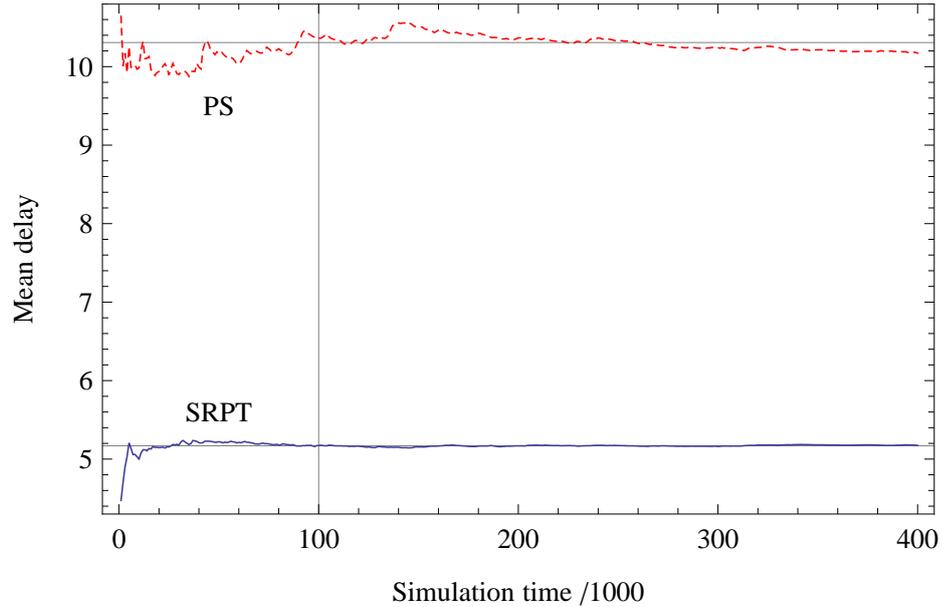


Figure 18: The evolution of the mean delay in the heterogeneous case when the load of the system is  $\rho = 0.9$  as a function of the simulation time.

sented in Table 9. The same parameter set is simulated ten times to determine the confidence interval.

Table 9: Parameters of the simulator, SRPT compared with PS

Name	Parameter	Value
Length of the simulation (time units)	$T$	200 000
Load of the system	$\rho$	0.1 ... 0.9
Arrival rate of jobs	$\lambda$	1
Total number of codes in the system	$K$	20, 75
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Classes		{5,15}
Transient period	<i>transient</i>	100 000
Individual simulation runs		10
Flow sizes		exponentially distributed

Two different cases have been examined. In the first one, the number of codes is  $K = 20$  and in the later one  $K = 75$ . The simulated results are represented in Figures 19 and 20. It can be seen from the figures that in both cases SRPT improves the performance compared to PS. The figures also present how different classes behave in both situations.

With higher values of  $K$ , the choice of the policy does not seem to have much effect with small values of  $\rho$ . With  $K = 75$  the effect of SRPT is distinguishable only when  $\rho > 0.5$ . This behaviour is well in line with the results obtained in the homogeneous case in Section 6.5.4 where the effect of  $K$  was examined. Overall, Figures 19 and 20 are very similar to the figures shown in the homogeneous case (Section 6.5.2) and thus the heterogeneity of the terminal constraints do not seem to have a great impact on the results.

The ratio between the classes seems to remain almost the same when  $\rho$  is changed. Class 2 with more codes can achieve shorter delays and it earns more compared with PS than class 1. The total absolute delay of the system settles between the curves of the two classes as expected.

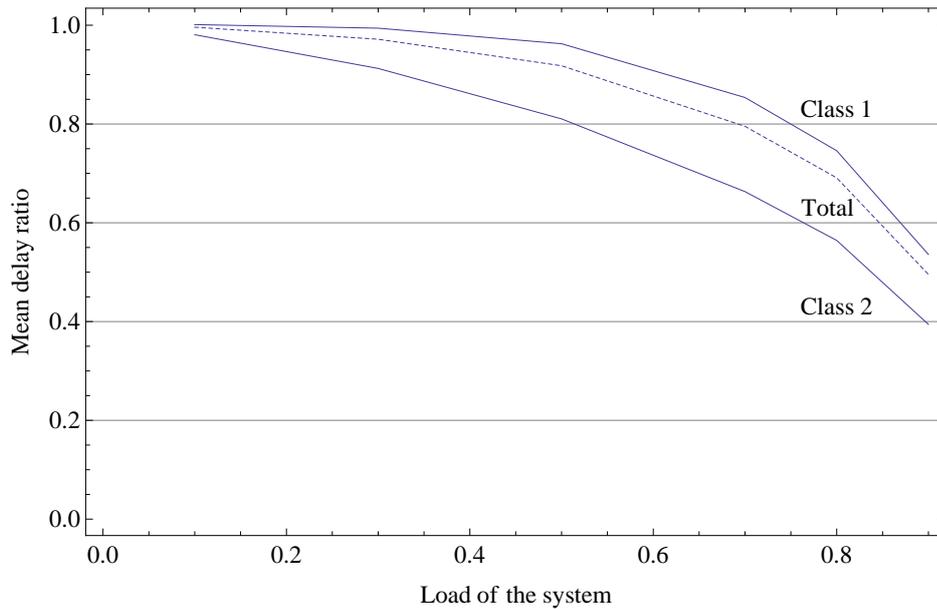


Figure 19: Mean delay of SRPT compared with PS with  $K = 20$ .

Simulations have been done also with three classes. This time, users in class 1 can use at most 5 codes, users in class 2 at most 10 codes, and users in the third class are able to use at most 15 codes. The parameters of this simulation are shown in Table 10. The other parameters are exactly the same as in previous simulations where two classes were examined.

The results of the three class simulation are shown in Figure 21 which shows that the total delay of the system settles between the curves of the different classes. Behaviour does not change much from the case with two classes with the same value of  $K$ . Class 1, with the least amount of codes, gets the most unfavourable service when the delays of the SRPT policy have been compared with the results of the PS policy. Like in the previous cases, the ratio between the mean delays of the different classes does not vary much as a function of  $\rho$ . The small variation can be explained

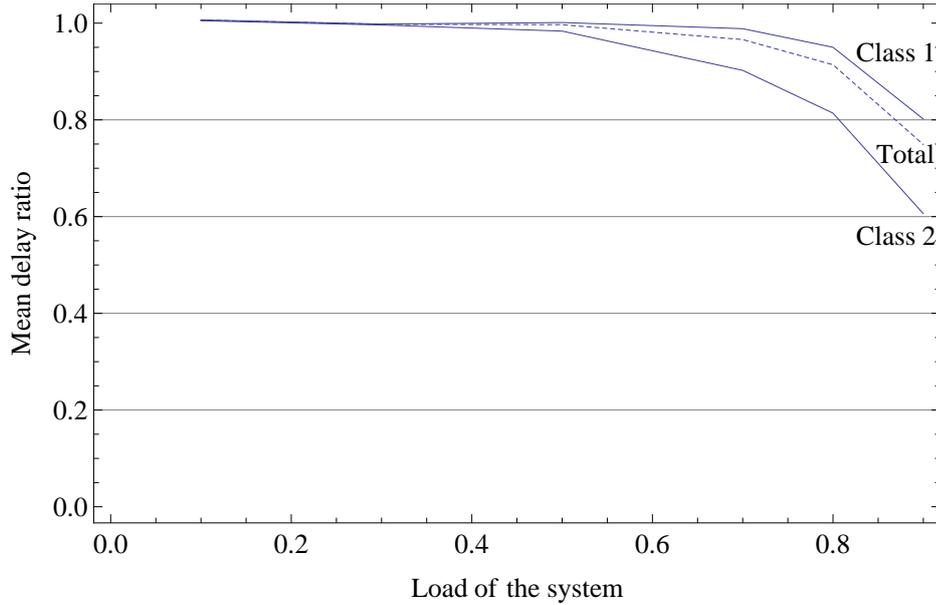


Figure 20: Mean delay of SRPT compared with PS with  $K = 75$ .

Table 10: Parameters of the simulator, SRPT compared with PS with three classes

Name	Parameter	Value
Length of the simulation (time units)	$T$	200 000
Load of the system	$\rho$	0.1 ... 0.9
Arrival rate of jobs	$\lambda$	1
Total number of codes in the system	$K$	20
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Classes		{5,10,15}
Transient period	<i>transient</i>	100 000
Individual simulation runs		10
Flow sizes		exponentially distributed

with the indiscriminateness of the simulation results. Absolute values in the case with three classes are almost identical to the case of two classes with the same value of  $K$ . From this point of view, it does not seem to make much difference whether the number of classes is changing. Based on these results it can well be said that terminal constraints do not change the case notably in either direction.

### 7.5.3 The conditional mean delay with respect to the size and distance

In this section, the results from the simulations are classified based on the distance from the base station and the flow size. Both variables are divided into two classes which lead to a total of four cases: small flows near the base station, small flows far from the base station, large flows near the base station, and large flows far from the

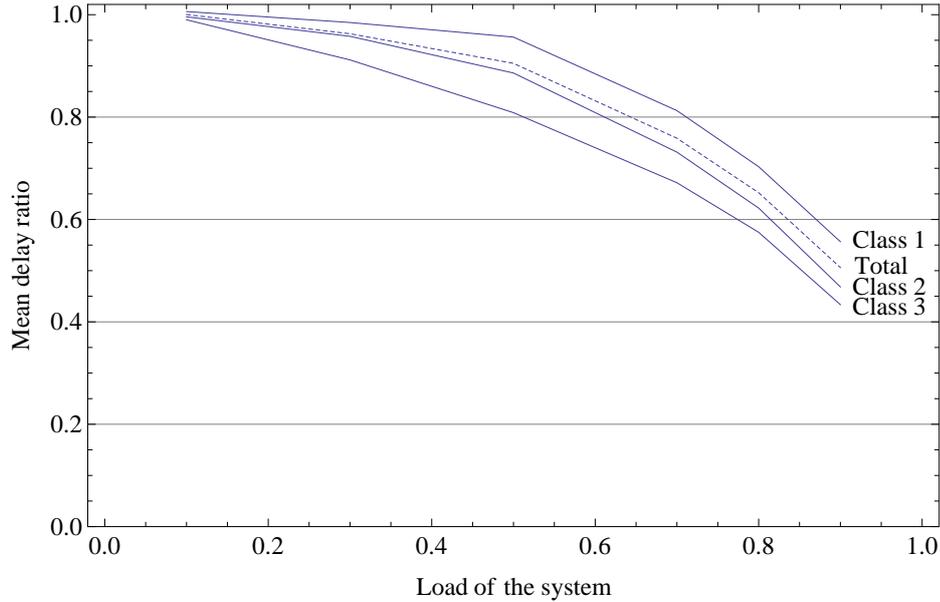


Figure 21: Mean delay of SRPT compared with PS with three classes when  $K = 20$ .

base station. As a threshold between small and large flows, the mean flow size and twice the mean flow size have been used. As a threshold between the distances far and near, the radiuses  $r_0$  and  $2r_0$  have been used. These two classification parameter sets are presented in Tables 11 and 12.

Table 11: Classification parameter set 1

Name of the class	Value range
Small	$X \in [0, E[X]]$
Large	$X > E[X]$
Near	$R \in [0, r_0]$
Far	$R > [r_0, 1]$

Table 12: Classification parameter set 2

Name of the class	Value range
Small	$X \in [0, 2E[X]]$
Large	$X > E[X]$
Near	$R \in [0, 2r_0]$
Far	$R > [2r_0, 1]$

The case with  $K = 20$  will be examined first. The simulations have been done with the parameter set presented in Table 13. The same parameter set is simulated

ten times to determine the confidence interval. Longer simulation runs than in previous simulations in the heterogeneous case have been used to make sure that there would be enough flows in each classification group. Again both PS and SRPT have been simulated since no analytical results are available even for PS. The same transient period (100 000 time units) has been used as previously and the increase in the length of the simulation run has directly increased the amount of the simulation data.

Table 13: Parameters of the simulator, the mean delay with respect to the size and distance

Name	Parameter	Value
Length of the simulation (time units)	$T$	400 000
Load of the system	$\rho$	0.7, 0.8, 0.9
Arrival rate of jobs	$\lambda$	1
Total number of codes in the system	$K$	20, 75
Path loss exponent	$\alpha$	2
Centre area radius	$r_0$	1/7.94
Classes		{5,15}
Transient period	<i>transient</i>	100 000
Individual simulation runs		10
Flow sizes		exponentially distributed

The simulation results have been classified in two parameter sets shown in Tables 11 and 12. The same simulation has been classified in all following cases, only classification parameter sets have been changed. It is notable that in this case the classification has been done based on the *bit sizes* of flows and the transmission rate has not been included to classifications. The transmission rate will of course have an impact on the actual simulated results but it will not have any effect on the classification. The classified results from the simulations, with  $\rho = 0.7$  and Classification parameter sets 1 and 2, are shown in Tables 14–17 for both SRPT and PS. Similar classifications have been done also for the case of  $\rho = 0.8$  and  $\rho = 0.9$  but to save space these results will not be shown here.

It can be seen from these results that the mean delay ratio between the classes 1 and 2 (the third column) does not depend much on the classification of the data. The ratio of the delay of the classes seems to be very near the ratio of the number of codes for the classes. Similar behaviour can be seen also from the results of the PS simulations. Ratios between classes are smaller for PS compared to SRPT. It may be that the fair way of sharing the resources fades out a part of the differences

Table 14: SRPT ( $\rho = 0.7, K = 20$ ) with Classification parameters 1

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.0654 \pm 0.0017$	
Class 2	$0.0219 \pm 0.0003$	
Small near, total	$0.0436 \pm 0.0009$	2.9879
Class 1	$2.3465 \pm 0.0082$	
Class 2	$0.7872 \pm 0.0019$	
Small far, total	$1.5668 \pm 0.0047$	2.9807
Class 1	$0.1003 \pm 0.0024$	
Class 2	$0.0344 \pm 0.0005$	
Large near, total	$0.0673 \pm 0.0010$	2.9154
Class 1	$4.7958 \pm 0.0304$	
Class 2	$1.6587 \pm 0.0069$	
Large far, total	$3.2273 \pm 0.0182$	2.8912
Total	$2.5911 \pm 0.0120$	

Table 15: PS ( $\rho = 0.7, K = 20$ ) with Classification parameters 1

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.1121 \pm 0.0036$	
Class 2	$0.0458 \pm 0.0005$	
Small near, total	$0.0790 \pm 0.0017$	2.4466
Class 1	$3.5046 \pm 0.0133$	
Class 2	$1.4717 \pm 0.0088$	
Small far, total	$2.4881 \pm 0.0081$	2.3812
Class 1	$0.1718 \pm 0.0056$	
Class 2	$0.0682 \pm 0.0026$	
Large near, total	$0.1200 \pm 0.0032$	2.5169
Class 1	$5.2384 \pm 0.0322$	
Class 2	$2.3405 \pm 0.0191$	
Large far, total	$3.7895 \pm 0.0246$	2.2381
Total	$3.2719 \pm 0.0174$	

between the user classes. With these simulations no definite conclusions of this behaviour can be done, and more simulations with extensive parameter sets will be needed to identify reasons behind this kind of behaviour.

The performance improvement of the SRPT policy in this case can be visualized like in Figure 22 where the classification has been done based on the Classification parameter set 1 (Table 11). The mean delay of the SRPT in each class (small flows near, small flows far, large flows near, and large flows far) have been compared with the PS policy. It can be seen from Figure 22 that the performance improvement is the lowest in the large flows far from the base station. The flows near the base station, regardless of the size of the flow, seem to benefit the most from the SRPT. Small flows far from the base station can benefit almost as much. The performance

Table 16: SRPT ( $\rho = 0.7$ ,  $K = 20$ ) with Classification parameters 2

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.0478 \pm 0.0006$	
Class 2	$0.0161 \pm 0.0001$	
Small near, total	$0.0319 \pm 0.0003$	2.9734
Class 1	$4.6566 \pm 0.0194$	
Class 2	$1.5543 \pm 0.0033$	
Small far, total	$3.1054 \pm 0.0105$	2.9958
Class 1	$0.0475 \pm 0.0007$	
Class 2	$0.0161 \pm 0.0005$	
Large near, total	$0.0318 \pm 0.0005$	2.9411
Class 1	$5.8616 \pm 0.0475$	
Class 2	$2.0801 \pm 0.0134$	
Large far, total	$3.9708 \pm 0.0290$	2.8179
Total	$2.5911 \pm 0.0120$	

Table 17: PS ( $\rho = 0.7$ ,  $K = 20$ ) with Classification parameters 2

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.0830 \pm 0.0007$	
Class 2	$0.0329 \pm 0.0003$	
Small near, total	$0.0579 \pm 0.0004$	2.5229
Class 1	$6.2082 \pm 0.0287$	
Class 2	$2.6576 \pm 0.0173$	
Small far, total	$4.4329 \pm 0.0205$	2.3360
Class 1	$0.0804 \pm 0.0020$	
Class 2	$0.0325 \pm 0.0012$	
Large near, total	$0.0565 \pm 0.0011$	2.4726
Class 1	$5.8580 \pm 0.0447$	
Class 2	$2.6784 \pm 0.0300$	
Large far, total	$4.2682 \pm 0.0344$	2.1871
Total	$3.2719 \pm 0.0174$	

improvements in those flows are much more significant than the improvements in the bigger flows in the same distance range. These results are well in line with the fact that SRPT favours small jobs and larger ones will suffer relatively. However, Figure 22 shows that with high loads even large flows far from the base station can benefit from the SRPT and there is also improvement in the mean delay. It seems that every group will perform better under SRPT than under PS.

In Figure 23, the same simulation result set has been classified based on the Classification parameter set 2 (Table 12). This difference in classification parameters increases the number of flows belonging to groups of small flows and flows near the base station. With this classification the group of large flows will include flows which are really the largest in the system. This way it can be seen how SRPT treats large

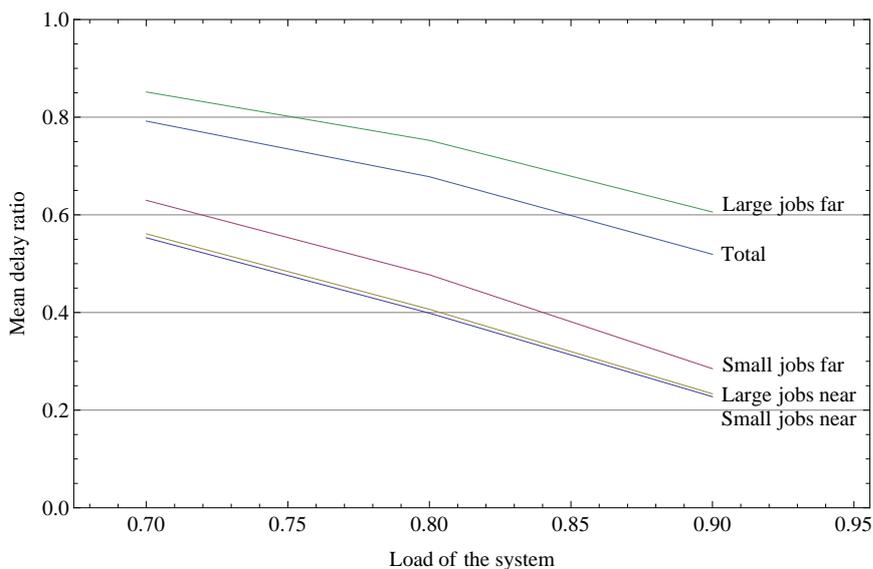


Figure 22: Relative performance improvement of SRPT compared to PS with Classification parameter set 1.

flows.

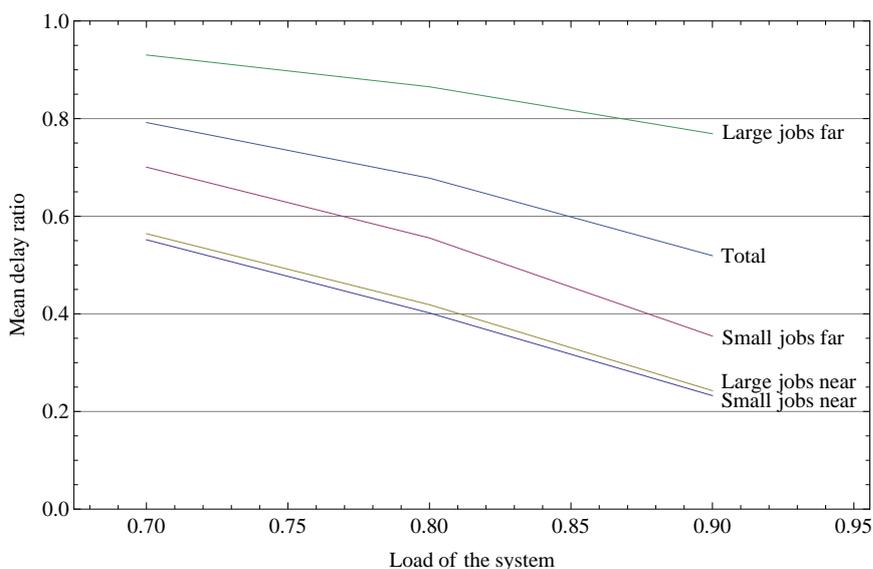


Figure 23: Relative performance improvement of SRPT compared with PS with Classification parameter set 2.

Classification does not seem to have much effect on the results and the order of the curves of the classes in the figure remains the same as in the first case. The order of the classes remains the same and absolute values are close to each other when compared with the results in Figures 22 and 23.

The performance improvement in the large flows far from the base station seems to be much lower than in the first case where Classification parameter set 1 was used. This behaviour is well understandable since now those large flows are really the largest ones. Also the distance has some effect but it does not seem to have as large effect on the results as the flow size limit.

Next the case of the mean delay with respect to the size and distance will be examined in case of  $K = 75$ . Both PS and SRPT cases have been simulated with three load values as in the previous case of  $K = 20$ . The classification parameter sets are exactly the same as previously and the classified results in case of  $\rho = 0.7$  are shown in Tables 18–21 as the previous case of  $K = 20$ .

Table 18: SRPT ( $\rho = 0.7$ ,  $K = 75$ ) classified with the mean flow size and  $r_0$

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.2473 \pm 0.0049$	
Class 2	$0.0826 \pm 0.0012$	
Small near, total	$0.1650 \pm 0.0022$	2.9913
Class 1	$7.9690 \pm 0.0114$	
Class 2	$2.6259 \pm 0.0090$	
Small far, total	$5.2975 \pm 0.0058$	3.0347
Class 1	$0.3800 \pm 0.0082$	
Class 2	$0.1299 \pm 0.0026$	
Large near, total	$0.2549 \pm 0.0035$	2.9251
<b>Class 1</b>	<b><math>13.5500 \pm 0.0468</math></b>	
Class 2	$4.3281 \pm 0.0169$	
Large far, total	$8.9390 \pm 0.0248$	3.1307
Total	$7.5158 \pm 0.0151$	

From the simulation result it can be seen that the ratio between the classes seems to be almost the same through all classification groups. This behaviour looks similar as with  $K = 20$  and it is reasonable to think that  $K$  does not have a major effect on the interneccine relations of the user classes. This time, the ratios of SRPT and PS are closer to each other than in case of  $K = 20$ . In Section 6.5.4 it was shown that the performance improvement of SRPT decreases when the number of codes or servers in the system increases. The convergence of these results fits well in the previous result achieved in this thesis. With large values of  $K$ , SRPT starts to behave more and more like PS and higher and higher values of  $\rho$  are needed to see the difference.

It can be seen from Tables 18–21, that for large class-1 flows far from the base station, the PS policy is in fact a better choice than SRPT. These four lines have been bolded in the tables. When the data has been classified with the Classification

Table 19: PS ( $\rho = 0.7$ ,  $K = 75$ ) classified with the mean flow size and  $r_0$ 

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.2722 \pm 0.0046$	
Class 2	$0.0933 \pm 0.0014$	
Small near, total	$0.1828 \pm 0.0023$	2.9161
Class 1	$8.6035 \pm 0.0353$	
Class 2	$3.0063 \pm 0.0132$	
Small far, total	$5.8049 \pm 0.0207$	2.8618
Class 1	$0.4214 \pm 0.0091$	
Class 2	$0.1428 \pm 0.0035$	
Large near, total	$0.2821 \pm 0.0042$	2.9507
<b>Class 1</b>	<b><math>13.3404 \pm 0.0605</math></b>	
Class 2	$4.6952 \pm 0.0215$	
Large far, total	$9.0178 \pm 0.0391$	2.8412
Total	$7.7449 \pm 0.0300$	

Table 20: SRPT ( $\rho = 0.7$ ,  $K = 75$ ) classified with twice the mean flow size and  $2r_0$ 

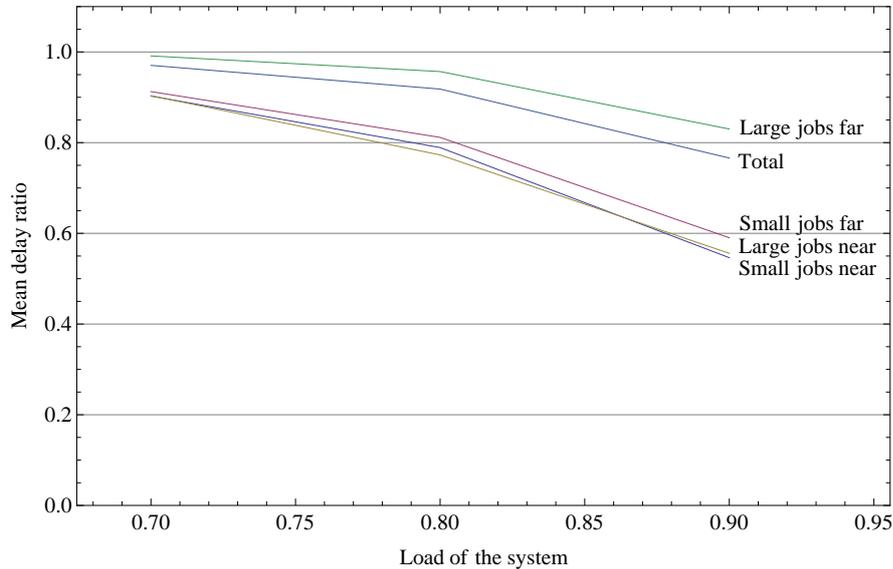
	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.1810 \pm 0.0020$	
Class 2	$0.0598 \pm 0.0004$	
Small near, total	$0.1204 \pm 0.0009$	3.0261
Class 1	$14.7351 \pm 0.0347$	
Class 2	$4.7791 \pm 0.0111$	
Small far, total	$9.7571 \pm 0.0168$	3.0832
Class 1	$0.1725 \pm 0.0037$	
Class 2	$0.0581 \pm 0.0012$	
Large near, total	$0.1153 \pm 0.0018$	2.9694
<b>Class 1</b>	<b><math>15.8117 \pm 0.0714</math></b>	
Class 2	$5.0376 \pm 0.0318$	
Large far, total	$10.4247 \pm 0.0448$	3.1387
Total	$7.5158 \pm 0.0151$	

parameter set 1 also the group of large flows far from the base station seems to perform better under SRPT than PS even though the class 1 of the same group suffers from SRPT. With Classification parameter set 2, the whole group of large flows far from the base station suffers from SRPT with load values  $\rho = 0.7$  and  $\rho = 0.8$ . In case of  $\rho = 0.9$  also this group does better under SRPT. This can be seen from Figures 24 and 25 where the mean delay of each classification group in case of SRPT have been compared with the mean delay of the same group in case of PS.

The order of the classification groups in Figures 24 and 25 is the same as in the case of  $K = 20$  shown earlier. Also the order has not changed when the classification parameter sets have been changed. In this case the flows near the base station seem

Table 21: PS ( $\rho = 0.7$ ,  $K = 75$ ) classified with twice the mean flow size and  $2r_0$ 

	Mean delay	Mean delay of Class 1/Class 2
Class 1	$0.1969 \pm 0.0018$	
Class 2	$0.0679 \pm 0.0006$	
Small near, total	$0.1324 \pm 0.0011$	2.8986
Class 1	$15.4614 \pm 0.0604$	
Class 2	$5.4126 \pm 0.0151$	
Small far, total	$10.4370 \pm 0.0363$	2.8565
Class 1	$0.1934 \pm 0.0042$	
Class 2	$0.0669 \pm 0.0010$	
Large near, total	$0.1302 \pm 0.0021$	2.8903
<b>Class 1</b>	<b><math>15.0580 \pm 0.0771</math></b>	
Class 2	$5.3170 \pm 0.0374$	
Large far, total	$10.1876 \pm 0.0521$	2.8320
Total	$7.7449 \pm 0.0300$	

Figure 24: Relative performance improvement of SRPT compared with PS with classification of the mean flow size and  $r_0$  when  $K = 75$ .

to benefit more than the other flows.

## 7.6 Summary

In this section, the so-called heterogeneous case in which the user population has been divided into user classes with different terminal constraints, has been examined. No major differences came up when the heterogeneous case was compared with the homogeneous case. When the total number of codes was increased in the system, similar behaviour was found as in the homogeneous case.

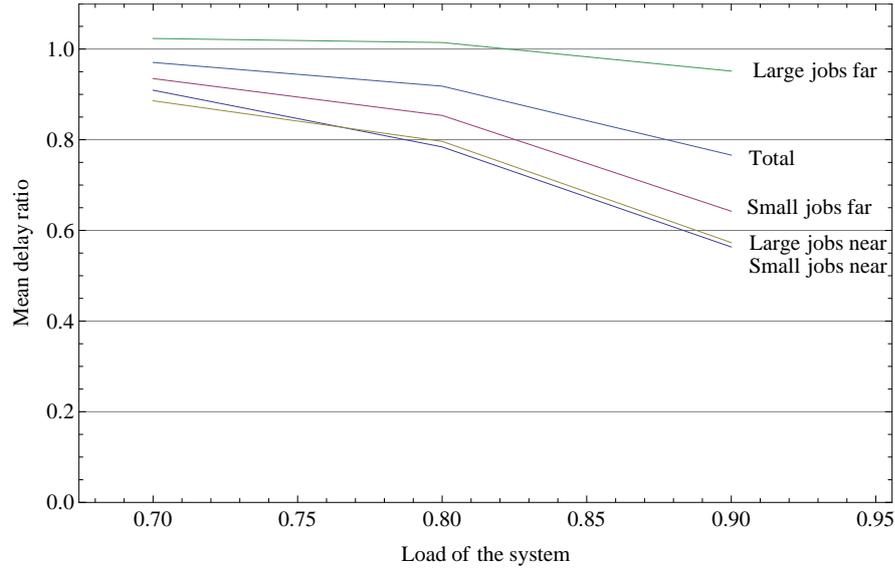


Figure 25: Relative performance improvement of SRPT compared with PS with classification of twice the mean flow size and  $2r_0$  when  $K = 75$ .

Compared to the homogeneous case, the simulation times have been increased for the heterogeneous case. When the mean delay with respect to the size and the distance were examined 400 000 time units were used as the length of the simulation. With the same hardware mentioned in Section 6.6, one round of the simulation took about three hours to run.

It can be seen from the results of Section 7.5.3 that under high load even the largest flows will perform better with the SRPT than with PS. The performance improvement in smaller flows is remarkable even under a lighter system load. Very large flows far away from the base station do not benefit much from SRPT but with high loads SRPT outperforms PS even in this case.

## 8 Summary

In this thesis, the performance improvement of a single cell setup in a cellular mobile network has been examined when size-based scheduling is used. An HSDPA-like CDMA system has been under examination and various combinations of user categories and terminal constraints have been examined. As a performance measure the mean flow level delay has been used when comparing the SRPT policy and the fair PS policy.

### 8.1 Conclusions

Generally, SRPT seems to be a much better policy than the PS policy when the goal is to minimize the mean delay of the system. The highest performance improvement is achieved when the load of the system is high. When the system load  $\rho$  is below 0.5, there is not much difference between the SRPT and PS policies. The performance improvement of SRPT with low loads is noticeable but not significant. Increasing the total number of codes ( $K$ ) in the system seems to decrease the performance improvement of SRPT. Combining these two observations together, the highest performance improvement with the SRPT policy can be achieved in high load networks with a low number of  $K$ .

We also analyzed the conditional mean delays with respect to the size and distance. The performance improvement of flows near the base station is higher than flows which are farther. There is not large difference between small and large flows which are near the base station. When considering only the flows far from the base station, the size has a significant effect and very large flows are those who will benefit from the SRPT policy least.

The effect of the size and the distance on the conditional mean delay was an interesting case and there is still room for future research. The data was classified in two parts in both the size and the distance sense. Changing classification parameters does not seem to have a radical impact on the results and the order of groups remains the same regardless of the classification. In this case, similar behaviour as in the homogeneous case was observed: increasing the number of codes in the system will decrease the performance improvement of SRPT.

In the heterogeneous case, where user classes differ with respect to the maximum number of codes they can use, it is also important to consider the performance between the user classes. With the used data classification, each group as a whole performed better with SRPT. However, even if the whole group of large flows far from the base station benefited from SRPT, flows in the class with less codes suffered when using SRPT. It is possible that some very large flows might get starved under SRPT. This behaviour becomes apparent with large values of  $K$ . With smaller values of  $K$ , every group and class of flows outperformed the PS policy. In this case the number of codes in the system, together with the load of the system, defines the possible performance improvement which can be achieved with SRPT when compared to PS.

In the homogeneous case, the effect of the parameters of the transmission rate

function was examined by changing the path loss parameter  $\alpha$  from 2 to 4. This does not seem to have much effect on absolute or relative results for the mean delay. Other parameters in the path loss model are just coefficients and the effect of  $c_0$  and  $r_0$  have not been analysed since it did not look very interesting when examining the effect of the terminal constraints.

The modified version of SRPT, where the smallest flows measured in bits are taken into service at first, looked unusable and therefore it is not recommended to use in any case. The performance was worse with almost all values of the system load compared with PS.

## 8.2 Future research

In this thesis, many interesting cases have been simulated. Because of long simulation runs, it was not possible to simulate and analyse all possible combinations of the parameters. Simulations have been used to obtain a general overview of each case and the remaining combinations have just been left outside this thesis. It would be interesting to get more results by running the simulator using also other values of  $K$  and examine what would happen when the number of classes is increased.

In this thesis, all random numbers have been drawn from the exponential distribution. Changing the flow size distribution, for example, to some more heavy-tailed distribution, might affect the result.

Only one transmission rate function has been examined and only the effect of changing the path loss exponent has been simulated. It might be that with a more accurate wireless channel model, more reliable results would be obtained.

The classification based on the size and the distance would be more informative if the classification had been done for more than four groups. With nine (three groups for the size and three groups for the distance) the results would be more accurate. However, this would require much longer simulation runs to guarantee that there would be enough data also for the most uncommon groups. With the simulation runs done in this thesis, 2 times 2 classification was the maximum which could be executed while preserving the accuracy of the calculations.

## References

- [1] 3GPP Technical Report 25.306. *UE Radio Access Capabilities, version 5.1.0* 3GPP, 2002
- [2] Aalto, S., and Lassila, P., Impact of size-based scheduling on flow level performance in wireless downlink data channels, in Proceeding of ITC-20, 2007.
- [3] Bansal, N., and Harchol-Balter, M., Analysis of SRPT scheduling: investigating unfairness, in Proceeding of ACM SIGMETRICS, 2001.
- [4] Belloni, F., *Fading Models*, Article, S-88 Signal Processing Laboratory, Helsinki University of Technology, 2004.
- [5] Bonald, T., and Proutire, A., Wireless downlink data channels: user performance and cell dimensioning, in Proceeding of ACM Mobicom, 2003.
- [6] Goldsmith, A. *Wireless Communications*, Cambridge University Press, New York, 2005.
- [7] GSM Technical Specification. *GSM 02.34, High Speed Circuit Switched Data (HSCSD), Stage 1, Version 5.2.0*. ETSI, 1997.
- [8] GSM Technical Specification. *GSM 03.34, High Speed Circuit Switched Data (HSCSD), Stage 2+, Version 5.2.0*. ETSI, 1999.
- [9] Kleinrock, L., *Queueing Systems, Volume I: Theory*, Wiley, New York, 1975.
- [10] Kleinrock, L., *Queueing Systems, Volume II: Computer Applications*, Wiley, New York, 1976.
- [11] Korhonen, J., *Introduction to 3G Mobile Communications, 2nd edition*, Artech House, Norwood, 2003.
- [12] Pinedo, M., *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [13] Schrage, L.E., A proof of the optimality of the shortest processing remaining time discipline, *Operations Research* 16, 678-690, 1968.
- [14] Schrage, L.E., and Miller, L.W., The queue M/G/1 with the shortest remaining processing time discipline, *Operations Research* 14, 670-684, 1966.

## Appendix A: Source code of the simulateSRPT() function

```

simulateSRPT[T_, rho_, lambda_ , K_, classes_] :=
Module[r, c, t,  $\mu$ , meanJobSize, tprev, idx, jobSize, queue,
  queueResult, inService, codesInUse, currentJobSize, departures,
  arrival, numOfCodes, currentServer, jobsInService, codes,
  serviceTimes, swej, i, j,

  (* Calculate the mean size of an arriving job *)
   $\mu$  = lambda/(K*rho);

  (* Mean workload is defined with the given  $\mu$  and the data rate function *)
  meanJobSize = rho/(lambda*NIntegrate[2*r*1/dataRate[r], r, 0, 1]);

  t = 0; (* Start time *)
  tprev = 0; (* Previous time *)

  idx = 0; (* Id number for jobs *)
  jobSize = 0; (* Workload of the next job (bits) *)

  (* List for unfinished jobs, idx, arrival time, original job size,
    distance, current size, num of codes, service rate per code *)
  queue = ;

  queueResult = ; (* List for finished jobs *)

  (* Initializing some lists, length == K *)
  inService = codesInUse = currentJobSize = Table[0, K];
  departures = Table[2*T, K];

  arrival = t + expVar[lambda]; (* The arrival time for the next job *)

  While[t < T,
    tprev = t;

    If[arrival < Min[departures],
      t = arrival;
      r = Sqrt[RandomReal[]];
      jobSize = expVar[meanJobSize];
      numOfCodes = RandomChoice[classes]; (* Class selection *)

      (* Add the new job to the end of the job list *)

```

```

idx++;
AppendTo[queue, idx,t,jobSize,r,jobSize,numOfCodes,dataRate[r]/K];

(* Udate the status of jobs currently in service *)
For[i = 1, i <= K, i++,
  If[inService[[i]] > 0,
    currentJobSize[[i]] = currentJobSize[[i]] - (t - tprev) *
      codesInUse[[i]] * queue[[inService[[i]], 7]]
  ];
];

(* If there are enough codes to serve the new job at full rate *)
If[Total[codesInUse] + numOfCodes <= K,
  (* Find the first free 'server' *)
  currentServer = First[Position[inService, 0]][[1]];
  inService[[currentServer]] = Length[queue];
  codesInUse[[currentServer]] = numOfCodes;
  currentJobSize[[currentServer]] = jobSize;
  departures[[currentServer]] = t + jobSize/(numOfCodes*dataRate[r]/K);
,

  (* Update current job sizes to the queue *)
  For[i = 1, i <= K, i++, If[inService[[i]] > 0,
    queue[[inService[[i]], 5]] = currentJobSize[[i]]];

  (* Initialize the lists *)
  inService = codesInUse = currentJobSize = Table[0, K];
  departures = Table[2*T, K];

  (* and select new jobs into service *)
  jobsInService = ;
  i = 1; (* counter for number of jobs in service *)

  While[K - Total[codesInUse] > 0 &&
    Length[Delete[queue, jobsInService]] > 0 && i <= Length[queue],
    (* Available codes for each job *)
    codes = (Min[K - Total[codesInUse], #]) & /@ queue[[All, 6]];

    (* Service time for each job *)
    serviceTimes = queue[[All, 5]]/(codes*queue[[All, 7]]);

    (* The shortest job will be selected *)
    inService[[i]] = First[Position[serviceTimes,
      Min[Delete[serviceTimes, jobsInService]]]][[1]];

```

```

codesInUse[[i]] = codes[[inService[[i]]]];
currentJobSize[[i]] = queue[[inService[[i]], 5]];

(* The departure time for each job *)
departures[[i]] = t + currentJobSize[[i]]/(codesInUse[[i]] *
  queue[[inService[[i]], 7]]);
AppendTo[jobsInService, inService[[i]]];
i++;
];

];

(* An arrival time for the next job *)
arrival = t + expVar[lambda];

,

(* nextdeparture < nextarrival *)
t = Min[departures];

(* Case when the queue is empty and the next arrival is after T *)
If[t > T, Break[]];

(* Server with ending job *)
swej = First[Position[departures, Min[departures]]][[1]];

(* Add the ended job to the list of finished jobs
  with the ending time t *)
AppendTo[queueResult, queue[[inService[[swej]]]]];

(* Job completed, current job size is zero *)
queueResult[[Length[queueResult], 5]] = 0;

(* Departure time *)
AppendTo[queueResult[[Length[queueResult]]], t];

(* Delete the ended job from the queue *)
queue = Delete[queue, inService[[swej]]];

(* Update inService queue index according to the delete operation
  Del (3) -> idx (4)->idx (3), idx (5)->idx (4),... *)
For[i = 1, i <= Length[inService], i++,
  If[inService[[i]] > inService[[swej]], inService[[i]]--];
];

```

```

(* Initialize inService, departures, codes in use,
   and current job size values *)
inService[[swej]] = codesInUse[[swej]] = currentJobSize[[swej]] = 0;
departures[[swej]] = T*2;

(* Udate the status of jobs currently in service to the queue *)
For[i = 1, i <= K, i++,
  If[inService[[i]] > 0,
    currentJobSize[[i]] = currentJobSize[[i]] - (t - tprev) *
      codesInUse[[i]]*queue[[inService[[i]], 7]];
    queue[[inService[[i]], 5]] = currentJobSize[[i]];
  ];
];

(* If there are jobs without the service *)
If[Total[codesInUse] < Total[queue[[All, 6]]],
  inService = codesInUse = currentJobSize = Table[0, K];
  departures = Table[2*T, K];

  (* and select new jobs into service *)
  jobsInService = ;
  i = 1; (* counter for number of jobs in service *)

  While[K - Total[codesInUse] > 0 &&
    Length[Delete[queue, jobsInService]] > 0 && i <= Length[queue],
    codes = (Min[K - Total[codesInUse], #]) & /@ queue[[All, 6]];
    serviceTimes = queue[[All, 5]]/(codes*queue[[All, 7]]);
    inService[[i]] = First[Position[serviceTimes,
      Min[Delete[serviceTimes, jobsInService]]][[1]];
    codesInUse[[i]] = codes[[inService[[i]]]];
    currentJobSize[[i]] = queue[[inService[[i]], 5]];
    departures[[i]] = t + currentJobSize[[i]]/(codesInUse[[i]] *
      queue[[inService[[i]], 7]]);
    AppendTo[jobsInService, inService[[i]]];
    i++;
  ] (* End While *)

];
;
];
]; (* End While *)

(* Update the remaining job sizes for the jobs in inService *)
For[i = 1, i <= K && Length[queue] > 0, i++,

```

```
If[inService[[i]] > 0,  
  queue[[inService[[i]], 5]] = currentJobSize[[i]];  
  ];  
];  
  
(* Remaining jobs will be marked as leaving  
  at the end of the simulation *)  
For[j = 1, j <= Length[queue], j++,  
  AppendTo[queueResult, queue[[j]]];  
  AppendTo[queueResult[[Length[queueResult]]], T];  
  ];  
  
(* Return list *)  
queueResult  
]
```

## Appendix B: Source code of the simulatePS() function

```

simulatePS[T_, rho_, lambda_ , K_, classes_] :=
Module[r, c, t,  $\mu$ , meanJobSize, tprev, idx, jobSize, queue,
  queueResult, arrival, departure, totNumOfCodes, numOfCodes, ending,,

  (* Calculate the mean size of an arriving job *)
   $\mu$  = lambda/(K*rho);

  (* Mean workload is defined with the given  $\mu$  and the data rate function *)
  meanJobSize = rho/(lambda*NIntegrate[2*r*1/dataRate[r], r, 0, 1]);

  t = 0; (* Start time *)
  tprev = 0; (* Previous time *)

  idx = 0; (* Id number for jobs *)
  jobSize = 0; (* Workload of the next job (bits) *)

  (* List for unfinished jobs, idx,arrival time,original job size,
    current size,num of codes,service rate per code,departure *)
  queue = ;

  queueResult = ; (* List for finished jobs *)

  arrival = t + expVar[lambda]; (* The arrival time for the next job *)
  departure = 2*T;
  totNumOfCodes = 0;

  While[t < T,
    tprev = t;

    If[Length[queue] > 0, departure = Min[queue[[All, 8]]],
      departure = 2*T];

    If[arrival < departure,
      t = arrival;
      r = Sqrt[RandomReal[]];
      jobSize = expVar[meanJobSize];
      numOfCodes = RandomChoice[classes]; (* Class selection *)

      If[Length[queue] > 0,
        (* Upate the status of jobs currently in service *)
        queue[[All, 5]] -= (t - tprev)*(queue[[All, 6]] /

```

```

    Max[K, totNumOfCodes]) * queue[[All, 7]]*K;
];

(* Add the new job to the end of the job list *)
idx++;
AppendTo[queue, idx, t, jobSize, r, jobSize, numOfCodes,
  dataRate[r]/K, t + jobSize/(numOfCodes*dataRate[r]/K)];

(* Update the number of codes in queue (k1*n1 + k2*n2 + ...) *)
totNumOfCodes += numOfCodes;

(* Calculate the departure times for jobs in queue *)
queue[[All, 8]] = t + queue[[All, 5]]/(queue[[All, 7]] * K *
  queue[[All, 6]]/Max[K, totNumOfCodes]);

(* An arrival time for the next job *)
arrival = t + expVar[lambda];

,

(* nextdeparture < nextarrival *)
t = departure;

(* Index of ending job *)
ending = First[Position[queue[[All, 8]], departure]][[1]];

(* Add the ended job to the list of finished jobs
  with the ending time t *)
AppendTo[queueResult, queue[[ending]]];

(* Job completed, current job size is zero *)
queueResult[[Length[queueResult], 5]] = 0;

(* Departure time *)
queueResult[[Length[queueResult], 8]] = t;

If[Length[queue] > 0,
  (* Udate the status of jobs currently in service *)
  queue[[All, 5]] -= (t - tprev)*(queue[[All, 6]] /
    Max[K, totNumOfCodes])*queue[[All, 7]]*K;

  (* Update the number of codes in queue *)
  totNumOfCodes -= queue[[ending, 6]];

  (* Delete the ending job *)

```

```

queue = Delete[queue, ending];

(* Calculate the departure times for jobs in queue *)
queue[[All, 8]] = t + queue[[All, 5]]/(queue[[All, 7]] * K *
  queue[[All, 6]]/Max[K, totNumOfCodes]);
];
;
];
]; (* End While *)

(* Update the status of jobs currently in service *)
queue[[All, 5]] -= (t - tprev)*(queue[[All, 6]] /
  Max[K, totNumOfCodes])*queue[[All, 7]]*K;

(* Remaining jobs will be marked as leaving
  at the end of the simulation *)
For[i = 1, i <= Length[queue], i++,
  AppendTo[queueResult, queue[[i]]];
  queueResult[[Length[queueResult], 8]] = t;
];

(* Return list *)
queueResult
]

```