Institut für Parallele und Verteilte Systeme
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Diplomarbeit Nr. 3711

Simulation and Evaluation of Replicated Workflow Executions

Stefan Schmidhäuser

Studiengang: Informatik

Prüfer: Prof. Dr. Rothermel

Betreuer: Dipl.-Inf. David Schäfer

begonnen am: 13. Januar 2015

beendet am: 15. Juli 2015

CR-Klassifikation: C.4, B.3.3

Abstract

Mobile devices are very common these days. It is rare to meet a person that does not own a mobile phone. In the past these devices were mainly used for voice calls. This changed however with the introduction of the so called smart-phones. Their powerful processors and large displays allowed a large number of new applications. Today these devices are used to check e-mails, surf the Internet or other applications that mainly use data-transmissions instead of voice calls. Today mobile standards like HSPA+ or LTE are therefore designed to provide very high data-rates. However these data-rates can only be achieved when the signal quality between mobile and base-station allows it. Since signal quality depends on a number of factors like distance to the base-station, shadowing and multipath fading, simulators for mobile communication are usually very complex. This complexity leads to long run times, which is not always acceptable.

This work will start by introducing a simulation model for mobile data-transmission, which is derived from existing work and specifications. This model will then be used to evaluate a number of heuristics, which will be introduced in the remainder of the document. These heuristics will simulate mobile communication over larger time scales, and will therefore perform faster.

Contents

List of Figures

List of Algorithms

1 Introduction

Today mobile phones are common devices. In fact it is rare to encounter a person who does not carry such a device on him or her. In the past early mobile phones offered little more functionality than voice calls. Data transfers were possible but transfer rates were low using second generation technologies like GPRS. With the introduction of smart-phones, a shift in the way these devices were used occurred. More powerful processors, high resolution touch-screens and an array of internal sensors like GPS opened up many new possibilities. Applications like e-mail clients, web-browser or mobile games became standard software on these devices. Instead of using these mobile devices for phone-calls, people more often tend to use them for other purposes. Commuters might for example use their smart-phone to surf the Internet to pass time while they use public transportation. Smart-phones equipped with GPS sensors may also be used as navigational devices by drivers. This shift from voice calls to data transfers was addressed in the third generation of mobile network technologies like UMTS, which offer higher transfer rates up to several megabits per second.

In order to achieve these high data rates, signal modulation techniques like quadruple amplitude modulation (QAM) are used. These techniques modulate the amplitudes of the signal to transfer multiple bits per symbol at once. Higher degrees of modulation result in better data rates but also increase the probability of bit transfer errors. Hence the degree of modulation that can be used while retaining an acceptable error rate is directly related to the quality of the signal between the mobile device and the Base-station it is subscribed to. Signal quality in return depends on the background noise level, interference caused by other senders and distance of the mobile device to the base-station.

Background Noise: This kind of signal disturbance is always present. It is caused by other electronic devices or natural sources. Compared to interference the amount of disturbance is usually rather small and often neglected. Nonetheless at a far enough distance a signal carrying information will have degraded to a level at which it can no longer be told apart from background noise.

Interference: While background noise cannot be influenced too much, network providers have some control over the amount of interference that will be produced when designing the infrastructure for a mobile network. Interference is caused by devices using the same frequency range that are sending at the same time. Every mobile device communicating with its base-station will increase the interference in that base-stations cell. While power control applied to the mobile devices by the base-station helps to minimize interference, this still results in the cell having a maximum capacity as to how many mobile devices it can serve until signal quality becomes so low that it can no longer be distinguished from interference.

Distance: Radio signals fade exponentially over distance. How much exactly a signal fades depends on several factors, such as the terrain. An obstacle rich environment, like a big city will lead to faster degrading signals than an open countryside, due to shadowing. Eventually the radio signal will become so weak that the receiver will not be able to differentiate it from the environmental background noise, at which point no information can be transmitted. Since the sender power of mobile devices is limited, signal quality degrades rapidly once a certain distance to the base-station is reached.

Taking these factors into account it becomes obvious that base-station placement is crucial for good mobile network connection quality. Placing only a small number of base-stations might lead to areas with no cell coverage at all if signal fading is to strong in the area. Additionally the created cells will be rather big, with potentially more subscribers per Base-station. Once such a big cell reaches its capacity, new communication requests of mobiles are rejected, leaving the mobile with no service at all. Mobile devices at the outer areas of a cell might suffer from bad signal quality and are only able to use low data rates. On the other hand placing a large number of base-stations will lead to very small cells with few subscribers. While increasing the systems overall capacity, this leads to an unnecessarily high number of hand-overs between cells, which produces additional overhead in the system. If cells are chosen too small, they might also cause additional inter-cell interference in neighboring cells.

In order to optimize Base-station placement, network providers can make use of very detailed simulators. With these 'low-level' simulators they are able to predict a systems behavior and choose a configuration with the desired characteristics. While these low-level simulators provide reliable results, they are also extremely costly in terms of computing time and resources. Running these expensive simulations on every possible configuration of base-stations is not feasible.

Simulations with limited resources or that want to produce fast results cannot afford to use such complex and costly simulators. Simulations that are only interested in the general behavior of wireless communication in an already existing system might not have the need to simulate details, like the exact propagation path of each signal. For many use case scenarios like the replicated workflow execution, described in 'Towards Ensuring High Availability in Collective Adaptive Systems'[SSB$^+$14] or routing algorithms as described in ' only general information like the number of failed messages or the average duration of a data transmission is actually useful. The workflow scheduling algorithm described in 'A Cost Eficient Scheduling Strategy to Guarantee Probabilistic Workflow Deadline' [BTKR15] for example might want to know how probable it is that a certain entity will answer in time, to decide if it needs to parallel schedule services at another entity. A low-level simulator would be unnecessarily costly for these kind of simulations. In order to keep run-times low, a more general heuristic of wireless data transfer behavior might be more useful, since it will provide results far faster than a low-level simulator.
These faster simulation models might also be helpful to reduce the time needed using low-level simulators by preemptively narrowing down the simulated configurations to those

that appear to be useful.

In this work we will introduce five simulation models that approximate the behavior of mobile data transfers in a previously defined system. The first model is derived from existing research and third generation mobile communication standards, that simulates data traffic on the bit level in intervals of milliseconds. This first heuristic will be used as a reference for the performance of our models. The other models are designed to be more general heuristics with lower run-times. They work on packet or message level in intervals of seconds. Finally we will evaluate these models using an implementation based on the NetLogo simulator.

Structure

The remainder of this Document is structured as follows:

Chapter 2 – Background Chapter 2 will familiarize the reader with the necessary background knowledge.

Chapter 3 – Main Model: In this chapter we describe the more detailed simulation model that we use to evaluate the other less accurate models.

Chapter 4 – Point in Time Heuristic This chapter introduces the Point in Time heuristic.

Chapter 5 – Position Extrapolation Heuristic This chapter introduces the Position Extrapolation heuristic.

Chapter 6 – Path Prediction Heuristic This chapter introduces the Path Prediction heuristic.

Chapter 7 – Interval Scheduling Heuristic This chapter introduces the Interval Scheduling heuristic.

Chapter 8 – Evaluation This chapter will present our results for the presented simulation models.

Chapter 9 – Related Work Here we will briefly discuss other works that are related to ours.

Chapter 10 – Conclusion Chapter 10 will summarize our and propose future works.

2 Background

Wireless communication using the air as transport medium behaves differently from wired systems. Signal propagation is influenced by a number of different factors unique to wireless communication, such as shadowing or multipath fading. In order to simulate signal propagation in a wireless environment it is therefore necessary to use a propagation model, to account for these factors. For this work we chose the propagation model introduced in [GJP+91] and [RM92], which we will describe in section 2.1.

The currently dominant Mobile Communication technology is Wideband Code Division Multiple Access (WCDMA) which is used in third generation UMTS systems. In these systems signals are spread over the entire available bandwidth, which makes them less vulnerable to interference that is limited to a small frequency range. On the other hand this means that every sender will cause interference over the whole bandwidth of the system whenever it is actively sending data. In order to minimize this interference base-stations apply power control to the mobile devices in the system, keeping their sender power as low as possible while still being received. Although this helps it does not completely negate the fact that every sender adds to the overall interference of the system and that at some point the amount of interference will reach a level at which it is no longer possible to differentiate it from an incoming data signal.
To prevent this from happening every cell, controlled by a base-station is assigned a capacity. base-stations will accept connections from mobile devices until a certain device or interference limit is reached. From this point on they will reject every new incoming connection request in order to keep interference in the cell at a level at which mobile communication is still possible. This process is called Call Admission Control (CAC). The method we chose to use in this work is taken from [KSL00] and will be discussed in section 2.2.

After a mobile device is accepted into a cell, data transmissions will be controlled by the base-station. The way this is done differs for mobile network technologies. Since it is the currently most commonly used technology we will focus on UMTS with High Speed Packet Access (HSPA). Over time the UMTS standard has evolved and been expanded to, in what is called Releases. These Releases update definitions or add new features. Specifically interesting for us are Releases 5 and 6, in which High Speed Packet Access was added to the standard. Release 5 first introduced High Speed Downlink Packet Access (HSDPA) , which would then be followed by High Speed Uplink Packet Access (HSUPA) in Release 6. In this work we will build on Release 7 [3GP11][3GP10], which expanded onto the previous Releases by adding higher level modulation codes and increasing possible bit-rates even further. UMTS after Release 7 is also known as HSPA+.

UMTS uses Frequency Division Duplex (FDD), which allows simultaneous transmission in Uplink and Downlink direction. In total a bandwidth of 10 MHz is used, with 5 MHz for Up- and Downlink respectively. Each of these 5 MHz blocks use their outer 0.58 MHz on both ends as guard band, leaving each direction with an effective bandwidth of 3.84 MHz.
Transmissions im UMTS are divided into intervals, each of which is either 80,40 ,20 or 10 milliseconds long, depending on the desired quality of service for the transmission. In HSPA+ a new interval size of 2 ms is introduced, which allows the base-station to adapt faster to changes in signal quality, which can make it necessary to change the signals modulation. Which mobiles are allowed to send during these intervals is decided by a proportional fair scheduler that takes into account each mobiles currently achievable data rate and average previous data rate. The scheduler is executed on the base-station and will be explained in more detail in section 2.3.

Achievable data rates are dependent on a mobiles signal quality in each direction. Better signal qualities allow the use of higher level signal modulation, which directly increases the number of transmitted bits per symbol. In turn this will also increase the required signal power per bit for the signal to be received correctly. Using high level signal modulation with low signal quality will result in in a high rate of transmission errors, which will lead to necessary retransmissions.
In order to maximize data rates HSPA+ uses different modulation techniques depending on the available signal quality. In the Uplink direction Quadrature Phase-Shift Keying (QPSK) and 16 times Quadrature Amplitude Modulation (QAM) are used. For the Downlink HSPA+ also allows the use of 32 times QAM. More on how data rates are calculated will be discussed in section 2.4.

Wireless communication is inherently more susceptible to transmission errors than its wired counterpart. Since the use of signal modulation further increases error rates, HSPA+ uses "Hybrid Automatic Repeat Request" (HARQ), which is a combination of Forward Error Correction (FEC) and Automated Repeat Request (ARQ).
FEC is realized by encoding the signal with a turbo code [BG96]. Turbo codes are powerful error correction codes that perform well with lower signal quality and allow the partial restoration of an erroneous transmission. Step two of HARQ consists of a retransmission of the previous data, and using the coding information of both original and retransmission to restore the correct data. More information on turbo codes and HARQ is given in section 2.5.

2.1 Propagation Model:

For our main heuristic we chose to use the well known Log-distance path loss model. The model consists of a path loss and a shadowing component and is often used to simulate radio signal propagation over longer distances of several hundreds up to thousands of meters.

In its original form the Log-distance path loss model is written as:

$$PL_{d_0 \to d}(dB) = PL(d_0) + 10n \log_{10}(\frac{d}{d_0}) + \chi$$

with $PL(d_0)$ being the path loss in decibel at a distance d_0, $PL_{d_0 \to d}$ the path loss in decibel at an arbitrary distance d, n as path loss exponent and χ being the shadowing component.
Alternatively the model can be written in its linear form, which in fact we used in our implementation. In this case power levels are calculated in Watt not decibel and the formula looks like:

$$P_{rec} = P_{sent} * d^{-n} * 10^{\xi/10}$$

with P_{rec} being the power at which the signal is received in Watt, P_{sent} as the power in Watt with which the signal was sent at its origin, n again as the path loss exponent and ξ being a normal distributed value with zero mean and standard deviation of σ given in decibels.
The second part of the right hand side term, describes the average signal degradation over distance, while the third part simulates influences like shadowing, which cause the usually observed log-normal distribution in received power levels.

The Log-distance path loss model describes an average signal power development in regards to distance. It does not take into account small scale signal degradation for individual users, caused by influences like multipath fading. This is acceptable for our purposes however since our goal is to provide a heuristic to simulate average mobile communications behavior on a cellular level with typical distances of several hundreds of meters.

Since the behavior of the model depends on how the values for path loss exponent n and the standard deviation σ of ξ are chosen, we refer to [RM92]. In this work values for n and σ are derived from power measurements in several German cities.
Observed values for n typically range from 2.5 up to 3.0, with one exceptionally high value of 3.8 in Frankfurt. Standard deviations for ξ are measured to range between 7 and 13 dB.

2.2 Call Admission Control:

Since WCDMA systems always use the whole available bandwidth when sending, each new signal will add to the systems overall interference. At high enough levels of interference, receivers will no longer be able to distinguish it from an incoming signal and the transmission is lost.

To prevent this, Call Admission Control (CAC) is used. Each cell is given a maximum capacity in form of a mobile device limit or a certain amount of interference that it may not surpass. Before allowing a mobile device to start sending data, each base-station

checks if the cell it controls is already at its capacity or not.
If the capacity is already reached the new device has to be rejected, or the cell might risk interference levels at which none of the subscribers would be able to receive signals any more. This case is called system outage and should occur as rarely as possible.

Interference in a cell can be divided into two components. 'Intra-cell interference' refers to the interference caused by other devices in the same cell as the receiver, while 'inter-cell interference' is caused by devices in neighboring cells. 'Inter-cell interference' usually occurs more often at the edges of a cell and contributes less to overall interference. The ratio between 'intra-cell interference' and overall interference:

$$F = \frac{I_{intra}}{I_{intra} + I_{inter}}$$

is often called the F-factor and can be used during the planning process of cell distribution [MT06].

In UMTS systems Frequency Divsion Duplex (FDD) is used to allow simultaneous uplink and downlink transmissions. This also means that base-stations will only interfere with other base-stations while mobile devices will also only interfere with each other. Since base-stations are stationary, downlink interference can be predicted more easily than uplink interference and be partially accounted for during the system planning phase. Therefore a cells capacity will depend on the uplink interference caused by mobile devices.

For this work we decided to use Call Admission based on Signal to Interference Ratio (SIR) levels, as described in 'SIR-Based Call Admission Control for DS-CDMA Cellular Systems' [LEZ94] and later in 'SIR-Based Call Admission Control by Intercell Interference Prediction for DS-CDMA Systems' [KSL00] which is based on the former work.
[LEZ94] introduces two algorithms for SIR-based CAC. The first of the algorithms only takes into account how a new sender will affect the cell it is added to, while the second one also looks at how the new addition will affect SIR levels of neighboring cells. The second algorithm is then improved upon in [KSL00]. In its original form the algorithm only takes into account average uplink interference measurements of cells, which assume an equal spacial distribution of mobile devices. In its improved form the base-stations exchange information about individual power levels of their subscriber, with which it is then possible to determine how much the addition of a new mobile would affect neighboring cells in more detail.

Signal to Interfrence Ratio (SIR) at a base-sation k is calculated in the following way:

$$\text{SIR}_k = \frac{S}{I(k) - S}$$

with SIR_k being the Signal to Interference Ratio at base-station k, S being the power of the desired signal the base-station wants to receive and $I(k)$ as the total power received at k.

Assuming all mobile devices in a cell are power controlled, so that their signal is received with the same power and using the log-distance path loss model we introduced in section 2.1, $I(k)$ can be expressed as:

$$I(k) = Sn_k + S \sum_{h \neq k} \sum_{i=1}^{n_h} (\frac{r_{ih}}{r_{ik}})^{\alpha} 10^{(\xi_{ik} - \xi_{ih})/10}$$

with n_k and n_h being the number of active senders in cell k or h respectively, r_{ik} and r_{ih} the distance of mobile i to base-station k or h, α and ξ as path-loss exponent and shadowing parameter.

According to [KSL00], the additional inter-cell interference $L_m(h,k)$ a mobile m would cause in base-station h, if it would be accepted by base-station k can be calculated by:

$$L_m(h,k) = (\frac{r_{mk}}{r_{mh}})^{\alpha} 10^{(\xi_{mh} - \xi_{mk})/10}$$

To decide if new connections should be accepted, [LEZ94] defines the term of residual capacity R_k. It is defined as the additional number of connections a base-station can accept until an acceptable transmission quality can no longer be guaranteed.
To this end a SIR threshold SIR_{th} is chosen. SIR_{th} is a design parameter and is chosen depending on the desired minimal transmission quality.

With this the resulting residual capacity for each cell j, in case a mobile m should be accepted by base-station k is given by:

$$R_k^{(j)} = \begin{cases} \left\lfloor \frac{1}{\text{SIR}_{th}} - \frac{1}{\text{SIR}_k} \right\rfloor, & \text{if } j = k \\ \left\lfloor \frac{1}{\text{SIR}_{th}} - \frac{1}{\text{SIR}_k} - L_{m(j,k)} \right\rfloor, & \text{if } j \text{ close to } k \end{cases}$$

If the resulting residual capacity for each impacted cell is larger than zero, m is allowed to transmit data. Otherwise it would have been rejected since the resulting interference level might block transmissions that are already in progress.

2.3 Scheduler:

Scheduling in UMTS is performed locally by the base-station for each cell. Additionally HSPA+ defines data transmission intervals of 2 milliseconds in downlink as well as uplink direction. This way scheduling can react fast to changes in a mobiles signal quality.

To take advantage of the ability to quickly adapt to these changes, a proportional

fair scheduler is used in these systems. Proportional fair schedulers give preference to mobiles with better signals, while still providing a certain degree of fairness and preventing starvation of individual mobiles with worse connections.
This seems especially useful when taking into account the fluctuation of signal quality, as seen in the log-normal distribution observed during measurements.

Since each cell is controlled by only one base-station, downlink transmissions are only limited by the available sender power of that base-station. In the uplink however, it is often the case that several mobiles with limited sender power compete for transmission slots. If all of these mobiles would start sending at the same time the amount of interference would most likely make it impossible for the base-station to identify any useful signal. This environment can be described as interference limited. To prevent too much interference, the base-station has to be careful when scheduling uplink transmission. It is only allowed to assign certain amounts of sender power to mobiles to ensure the increase in interference does not exceed a previously specified amount.
In 'HSPA Performance and Evolution: A practical perspective' [TLKF09] a maximum uplink noise rise of 6 decibels is given as reasonable. This still allows more than one mobile to transmit in uplink direction simultaneously, but typically less than 4.

The particular scheduling algorithm used in UMTS systems is discussed in 'Data Throughput of CDMA-HDR a High Efficiency-High Data Rate Personal Communication Wireless System' [JPP00]
In order to decide which devices will send or receive data, the base-station keeps a record of all its subscribers recent average data rates. These averages are then compared to the estimated maximal currently achievable data-rate, to determine if an individual mobile is currently experiencing a good signal period or not. In detail this is done by calculating the following ratio for every subscriber:

(2.3.I) $DRC(t)/R(t)$

where $DRC(t)$ is the currently achievable data rate and $R(t)$ the average rate base-station remembers.
After assigning transmission slots to the mobiles with the highest calculated ratios, the base-station updates its remembered average for each mobile in the following way:

(2.3.II) $R_i(t+1) = (1 - \frac{1}{t_c})R_i(t) + \frac{1}{t_c} * R_{assigned}$

$R_{assigned}$ is the data-rate the according mobile was assigned during the last scheduling interval. So if the mobile was not assigned any slots $R_{assigned}$ equals zero.
The parameter t_c reflects how long the base-station remembers. If the connection quality of a mobile decreases abruptly, this change will be interpreted as a temporary abnormality for a duration depending t_c. During this time the mobile will not be assigned any slots, since the base-station expects the channel quality to increase again at any time.
Higher values for tc will in general improve overall throughput, but at the same time increase the risk of starvation for individual mobiles.

2.4 Data Rates in HSPA+:

When it was first introduced in 1999 UMTS in its original form called Release'99 supported data rates of up to 384kbits/s. Over the following years, these were significantly improved upon in later Releases by introducing new transmission techniques and adapting the standard.

The most important milestones in this development are probably Releases 5 through 7. Release 5 drastically increased downlink data rates by introducing High Speed Downlink Packet Access. This was realized by creating a new dedicated downlink channel type, decreasing the time interval between transmissions to 2 milliseconds and using more advanced signal modulation techniques like Quadrature Amplitude Modulation if the signal quality allows it.
Release 6 introduced High Speed Uplink Packet Access, which is very similar to HSDPA in regards that it introduced a new uplink channel type and reduced the transmission time interval to 2 milliseconds, but differs in small areas like the number of codes and the degree of signal modulation that are allowed.
Starting with Release 7 the new standard is referred to as HSPA+ and added another degree of signal modulation to downlink and uplink. It was also the first Release that supported Multiple Input Multiple Output, a transmission technique that receives a signal over multiple antennas to achieve higher data rates.

With Release 7 the supported signal modulation techniques for downlink transmissions included QPSK, 16QAM and 64QAM. The theoretical maximum achievable data-rate using these modulation techniques can be calculated using the following specification parameters that can be obtained from [3GP09]:
Each 10 millisecond frame contains 15 transmission slots. With the transmission time interval reduced to 2 milliseconds, that is 3 slots per interval. Each slot can transmit 2560 chips or symbols. The spreading factor of the signal is fixed to 16. In downlink direction a maximum of 15 codes are allowed. This leaves us with:

$$(2560 * 3 * 500 * 15)/16 = 3600000 \text{ chips per second}$$

Applying signal modulation with QPSK having 2 bits per symbol, 16 QAM having 4 bits per symbol and 64 QAM having 6 bits per symbol, we get theoretical data rates of 7.2 Mb/s , 14.4 Mb/s and 21.6 Mb/s.

Data rate calculation in uplink direction is done analogous. However signal modulation is limited to QPSK and 16QAM. Also the maximum number of codes that is allowed to be used is 4, in which case 2 codes with spreading factor of 4 and two codes with a spreading factor of 2 have to be used. This calculates to theoretical uplink data-rates of 5.76 Mb/s and 11.52 Mb/s.

It is important to keep in mind that these data-rates are only theoretically achievable in a single user system with no additional load at all and without using any form of forward error correction.

Which signal modulation technique can be used for data transmissions depends on the quality of the signal. Higher level modulation techniques have higher Bit Error Rates (BER), since the Energy per Bit $\frac{E_b}{N_0}$ is lower.

$\frac{E_b}{N_0}$ is a function of the Signal to Interference Ratio and can be calculated as

$$\frac{E_b}{N_0} = SIR * \frac{B}{R}$$

with B being the systems Bandwidth (3.84 MHz for UMTS) and R as the gross bit-rate.

For this reason every signal modulation technique is assigned a threshold signal quality measure in SIR ir EbNo, at which point the Bit Error Rate surpasses a limit that is deemed tolerable. The BER threshholds for HSPA+ are usually chosen in the range of 10^{-7} to 10^{-8}. Since Packets have a size of up to 12000 Bit this will keep the Packet Error Rate at around 10^{-4}.

2.5 Hybrid Automated Repeat Request (HARQ):

While the use of signal modulation techniques increases achievable data-rates, it also makes signals more susceptible towards transmission errors. For this reason the HARQ protocol is used in HSPA+. In addition to the traditional Automated Repeat Request protocol, HARQ also uses forward error correction to minimize the number of necessary retransmits.

Forward error correction is realized by encoding the transmitted data with turbo codes. Turbo codes are error correcting codes that perform very well at lower signal qualities. Using turbo codes involves the use of an encoder on sender size and a decoder at the receiver.
A traditional turbo encoder consists of two encoders, which may be different from each other. The sender first encodes the data stream it wants to send using the first encoder. Afterwards the result of the first encoder is fed to an interleaver, which rearranges the bit sequence following certain rules. The permutation of the bit stream is then given to the next encoder, which in the traditional case produces the final bit sequence that will be sent.
At this point the bit sequence has a code rate of 1/3, meaning that for every 1 data bit 2 redundancy bits will be sent and therefore the actual data-rate is also only a third of the theoretical data-rate. To increase the code rate the sequence can be punctured. This means that certain bits are removed from the sequence. Which bits are removed is defined by a specific puncture pattern. This reduces the amount of redundancy data and thereby increases the actual data-rate, but also reduces the error correction ability.
On the receiver side the incoming data stream is decoded by a number of decoders, which is equal to the number of encoders.

If a transmission error should occur despite forward error correction, a retransmit will be necessary. In this case the decoders will be able to use the redundancy information of the original transmission together with that of the retransmission to restore the original data with a higher probability.

More detailed information on turbo codes can be found in [BG96].

Simulating the whole turbo en- and decoding process would be very complex and costly. Instead we decided to determine transmission Bit Error Rates by using a lookup table, which maps EbNo values to a resulting Bit Error Rate. We obtained this table performing Matlab simulations using the Coded Modulation Library (CML) [CML09]. Unfortunately the library only supports up to 16QAM modulation natively, so we had to approximate the table for 64QAM Bit Error Rates. We did so by comparing the performance of the natively supported UMTS turbo code tables for QPSK, 16QAM and 64QAM with the equivalent QPSK and 16QAM results obtained using the HSPA simulations. Since the turbo codes used are identical we approximated the 64QAM table by applying the same relative rise in EbNo that is observed between the UMTS results.

3 Main Model

In this chapter we will introduce our main model, which is based on the contents of the previous sections in this chapter.

In contrast to the other heuristics we will introduce in the following chapters, our main model will always be executed in timesteps of 2 milliseconds. We do this because we chose to adhere to the HSPA+ standard when simulating mobile data transfers and HSDPA as well as HSUPA transmission time intervals are set to 2 milliseconds.

The execution of the model can be divided into three basic steps, performed by the components 'Mobile' and 'Base-station' during each timestep.

Mobiles simulate the moving clients, subscribing to Base-stations as for example mobile phones do. Each Mobile is assigned a sender power and keeps track of its position. Together these will be used to determine its uplink range using the log-distance path loss model. Each mobile also keeps track of its own average data-rate which will be used for scheduling purposes.
Since our goal is to simulate data transmissions every Mobile will maintain a queue of messages that will have to be processed in order. Messages are generated in pairs of uplink request and downlink response while a Mobile is currently active and the previous exchange has been finished. Message sizes are chosen using a minimal message size to which an additional number of bits is added. The added number of bits is determined by a Poisson Distribution, which is slightly shifted to the right. This way the created messages largely tend to have a size around the mean of the Poisson distribution, while still creating few uncommonly large messages.
Every message is created with a starting and an expiration time. The starting time is chosen using a gaussian distribution to determine the time until the next request should be sent. It represents the earliest time at which the message may be sent. Additionally an expiration time is chosen for each up- and downlink message, based on the minimal available data-rate. We chose to use a gaussian distribution with its mean at five times the expected transmission time at the lowest availabale data-rate. In case a message is not completely sent until this expiration time is reached, the message is no longer considered relevant and will be aborted and removed from the queue. Should this happen to an uplink message, the according downlink message will also be removed since in theory the base-station will never have received the request in the first place.

Base-stations are stationary components in our system. For our simulations we expect a previously generated distribution of base-stations.

Every base-station will maintain a list of neighboring base-stations and a list of its subscribed mobiles. These will be used for cell capacity calculations and scheduling purposes.
Base-station have a higher sender power than mobiles, giving them a superior range. However sending at high power levels will result in additional downlink interference in neighboring cells. This is why each base-station is assigned a high and low sender power. If all neighboring base-station are currently inactive the base-station may send using its higher power setting, since it will not cause any interference. If a neghboring cell is active however the base-station has to resort to the lower power setting in order to minimize downlink interference in the neighboring cell, and minimize overlapping cell areas.

Base-stations are also responsible for the scheduling process. During each timestep the devices with the highest priority are chosen and allowed to send or receive an amount of data, depending on the achievable data-rate.

The whole communication process for our main model in each timestep can be broken down into the following three steps:

- Call Admission Control.
- Scheduling.
- Message Transmission.

In the Rest of this section we will discuss these three steps in detail.

3.1 CAC:

At the beginning of every interval each mobile will check if it wants to send or receive data. If so, the simulated mobile device will continue as shown in 3.1.
First the device needs to determine its signaling range. In order to do so all mobile devices are given a maximum sender power P_{mobile}. To use the log-distance path loss model we will also have to determine a shadow fading parameter ξ_t for the current timestep which we do using an according gaussian distribution.
To limit the signals range we have to set a noise level against which we can compare the received signal strength, and determine whether it is strong enough to transmit data or not. We will refer to this as *Noise*.
By rearranging the log-distance path loss model we can then calculate a mobiles maximum range R using:

$$R = \sqrt[\alpha]{\frac{P_{mobile}}{Noise} * 10^{\xi_t/10}}$$

with α being the chosen path loss exponent for the simulation environment.

If the mobile device is already subscribed to a base-station, it will check if this base-station is still in signaling range. If so no further steps have to be taken at this point, since the device is already being served by that base-station.
Should the base-station be out of range or in case the mobile device just started a new transmission, it will check for base-stations that are in range. If at least one base-station is found in range, the mobile will start trying to subscribe to a base-station, preferring the closest one.

The CAC process is performed as described in section 2.2. Target base-station k and its neighbors calculate their residual capacity, for the case that the mobile were to subscribe to base-station k using:

$$R_k^{(j)} = \begin{cases} \left\lfloor \frac{1}{\mathrm{SIR}_{th}} - \frac{1}{\mathrm{SIR}_k} \right\rfloor, & \text{if } j = k \\ \left\lfloor \frac{1}{\mathrm{SIR}_{th}} - \frac{1}{\mathrm{SIR}_k} - L_{m(j,k)} \right\rfloor, & \text{if } j \text{ neighbor of } k \end{cases}$$

When calculating the additional inter-cell interference L_m the same random variable ξ_t is used as shadowing parameter ξ_{mk} between base-station k, which we want to subscribe to, and the mobile. Otherwise this would mean that channel conditions would have changed. The random parameters ξ_{mh} however will be determined randomly by a gaussian distribution.
In case all residual capacities return positive values larger than zero, the mobile will be accepted by base-station k and will be considered for data transfer during the scheduling step until it is either finished or out of range.

If no base-station is in range or the residual capacities were too low the mobile will not be able to establish a connection during this interval. This also means that it will not be able to receive or send any data.

Algorithm 3.1 illustrates the necessary steps for the CAC process in form of Pseudocode.

3.2 Scheduling:

At this point all mobile devices have had a chance to check if their base-station was still in range or could try to subscribe to a new base-station. Base-stations should now know which mobile devices are in their cell and want to transmit or receive data.

As a first step each base-station will check if any of its neighbors will be sending data this timestep. If so the base-station may only send with its lower power setting in order to minimize downlink interference, otherwise it is able to use its high power setting to achieve better downlink signal quality in its own cell.
Once the power setting is chosen, each base-station will start the scheduling process. Scheduling is performed using the proportional fair scheduler described in section 2.3.

Algorithmus 3.1 Call Admission Main

```
procedure CAC
    ξ = Rnd.Gauss(mean, standarddeviation)
    Range = (P_mobile / Noise)^(1/α) * 10^(ξ/10)
    if (AlreadySubscribed?) and (Range > DistanceToBaseStation) then
        EXIT
    else
        Candidates = getBaseStations(Range)
        if (Candidates == null) then
            EXIT # no BaseStations in range
        else
            for all BaseStations ∈ Candidates do
                CALCULATE RESIDUAL CAPACITIES(BaseStation + Neighbors)
                if (AllResidualCapacities > 0) then
                    SUBSCRIBETO(BaseStation)
                    EXIT
                end if
            end for
        end if
    end if
end procedure
```

To determine each mobiles estimated maximum data-rate, we first need to calculate each mobiles Signal to Interference Ratio. From there we can go on to calculate the Energy per Bit and select the highest degree of signal modulation, that provides a low enough Bit Error Rate.

Downlink Interference is only caused by other base-stations. Hence we can calculate the downlink SIR of a mobile using:

$$\text{(3.2.I)} \quad \text{SIR}_{downlink} = \frac{S_k}{I - S_k}$$

where S_k is the downlink signal power received from base-station k, and I is the sum of all downlink signals from all base-stations in range, received by the mobile.
Individual received signal powers are calculated using the log-distance path loss model:

$$S_k = P_k * d_{mk}^{-\alpha} * 10^{\xi/10}$$

Uplink intererence is caused by other mobiles sending at the same time. We assume all mobiles to be power controlled, and not to send without permission of the base-station. This way uplink interference will be kept minimal and at a constant level. The only mobiles causing additional interference will be the ones that are sending and those are, as mentioned in section 2.3, also limited to a constant noise rise. We are therefore able to estimate uplink SIR by simply calculating the received signal power at the base-station using:

$$\text{(3.2.II)} \quad \text{SIR}_{\,uplink} = \frac{S_m}{Noise + const}$$

where s_m is the mobiles received signal at the base-station, which can again be calculated using the log-distance path loss model.

To get from SIR to Energy per Bit, we need to calculate the 'coding gain' of each signal modulation technique. 'Coding gain' is defined as the quotient of bandwidth and gross data-rate $\frac{B}{R}$.
Bandwidth in a HSPA system is 3.84 MHz in down as well as uplink direction. The gross data-rate depends on signal modulation technique and the coding rate of the FEC. Using for example a 16QAM modulation with a theoretical data rate of 14.4 Mb/s encoded with a turbo code with coding rate 1/3, will result in a gross data-rate of 4.8 Mb/s.
The resulting coding gain is 3.84/4.8, which we can use to calculate the Energy per Bit for that particular signal modulation technique using

$$\frac{E_b}{N_0} = SIR * \frac{B}{R}$$

The larger the gross-data rate of a signal modulation technique, the lower its Energy per Bit will get. The lower EbNo gets the higher the probability of transmission errors will be. Since the BER should not fall below a certain value, we can find the highest achievable data-rate by selecting the signal modulation technique with the highest gross data-rate, whose EbNo still translates to a BER that is within the bounds.
To find that modulation technique we will start by calculating EbNo for the modulation technique with the highest gross data rate and look up the matching BER rate in our lookup table. If the BER is below our threshold we choose that modulation technique. Otherwise we will continue with the next modulation technique, which results in the next highest gross data-rate, until we arrive at a BER that is within bounds.

Now that the data-rate is known the base-station can continue the scheduling process. According to the proportional fair scheduling algorithm, each mobiles priority is calculated.
In downlink direction the mobile device with the highest priority is chosen to receive data from the base-station this turn. In uplink direction the scheduler is given a 'budget' of 6 dB of additional interference, which it can spend amongst mobiles that want to send to the base-station. The first mobile that is allowed to send data to the base-station is again determined by calculating priorities. In order to serve as many mobiles as possible, the base-station then determines the lowest threshold SIR at which the mobiles selected modulation technique is below the acceptable BER limit. The base-station then calculates the uplink interference caused by that mobile if it were to send with just enough power to achieve that threshold SIR. If the device would cause

less than 6dB of interference the base-station selects the mobile with the next highest priority and performs the same calculations and checks if the sum of interference is still below 6 dB. This process is repeated until the budget of 6 dB is used or no more additional devices can be found that fit in the budget. Finally the base-station divides the whole 6 dB budget among the selected devices and tells each of them at which power to send.

Finally after all scheduling decisions are made by the base-station, each device needs to update its average data-rate. This is done as described in section 2.3 using the formula:

$$R_i(t+1) = (1 - \frac{1}{t_c})R_i(t) + \frac{1}{t_c} * R_{assigned}$$

where $R_{assigned}$ is the gross data-rate of the selected signal modulation technique for mobiles that where chosen by the scheduler this turn, and zero for others.

Algorithm 3.2 outlines the general flow of the scheduling process.

Algorithmus 3.2 Scheduler Main

```
procedure SCHEDULER
    for all Mobiles ∈ Subscriber do
        CALCULATESIR
        Eb/N0 = SIR * B/R
        DRC = SELECTBESTMODULATIONSCHEME(Eb/N0)
        Priority = DRC_mobile / R_mobile(t)
    end for
    DETERMINEHIGHESTPRIORITY(downlink)
    for all  Mobiles ∈ Uplink do
        while Budget > 6dB do
            DETERMINEHIGHESTPRIORITY(uplink)
            Budget = Budget − Sir
        end while
    end for
    for all Mobiles ∈ Subscriber do
        UPDATEAVERAGEDATARATE(
    end for)
end procedure
```

3.3 Message Transmission:

During the third phase of the model the devices which were chosen by the scheduler will send or receive data.

The amount of data each device will send or receive this interval is given by

$$Data = Rate * Interval = Rate * 0.002s$$

Retransmits are performed on packet level. In HSPA the maximum allowed packet size is set to 12000 bits, so if *Data* is less or equal to 12000 only one packet will be sent.
To determine if a retransmit is necessary each packet will be checked for bit errors. To do so we determine the appropriate BER by looking it up in our lookup table. Then we create a uniformly distributed random number between 0 and 1 for each bit in the current packet. If one of these random numbers is below the BER, an error has occurred and the packet data needs to be retransmitted. If a packet fails 3 times in a row the whole message is considered to be failed and needs to be retransmitted completely.
In case no transmission errors occur the remaining data volume is reduced by the amount of the packet.
Once all packets of an interval are sent and the message is not completely transmitted, the mobile checks if the messages expiration time is reached or not. If so the message will be marked as expired and removed from the queue. If the message was an uplink request the next message in the queue will be the according downlink response. Since the base-station would never have received the request, the response should also be removed from the queue. If the expiration time is not yet reached, the message will continue to be processed.

Once the whole message is sent, it will be marked as completed and if more messages exist, replaced by the next message from the queue. If there are no more messages in the queue, but the device is still supposed to be active, a new pair of uplink and downlink messages will be created and added to the queue.

Algorithm 3.3 illustrates the general implementation of the transmission process.

Algorithmus 3.3 Message Transmission Main

```
procedure MESSAGE TRANSMISSION
    DETERMINEPACKETSPERINTERVAL
    for all Packets do
        for all Bits ∈ Packet do
            CHECKFORERROR(Bit)
            if Error then
                ErrorCount++
            end if
        end for
        if ErrorCount == 0 then
            REDUCEREMAININGSIZE(Packet)
        else if ErrorCount == 3 then
            RETRANSMITMESSAGE
        end if
    end for
    if MessageExpired then
        ABORT
    else if MessageCompleted then
        FINISH
    end if
end procedure
```

4 Point in Time Heuristic

While the previous heuristic was based on existing theoretical work and specification, the other heuristics in the following chapters are more abstract.
The Main heuristic tried to be close to the source material and hence worked with 2 millisecond intervals and randomly generated parameters each interval. This made sense for these small intervals but is also rather expensive to execute.
Therefore the following heuristic works on averages over larger time intervals in the dimension of seconds and up to minutes. The main idea for this Point in Time heuristic is to collect all the information that is available at the current time and use it to perform the simulation for the whole next extended interval.

4.1 Heuristic

Just as the main heuristic we will use mobiles and base-stations. The tasks they perform will differ slightly however, due to the fact that we will not perform detailed scheduling, but will instead approximate average data-rates for data transmissions.
Despite these changes the basic three step structure introduced in the main heuristic remains.

4.1.1 Call Admission Control

If mobile is currently active and has a message in its queue at a given timestep, it checks if the messages starting time is during the next interval. If so the mobile will try to process this message during the interval. As with the Main heuristic, the first step to do so, is to check if the mobile is in range of a base-station and if so is accepted as a subscriber.
The main difference in this step are the larger time intervals. During a single step with a duration of 1 second the Main Heuristic would have generated 500 different shadowing parameters ξ, which would all translate into different range calculations for the mobile. ξ follows a Gaussian distribution with a mean of zero. We decided to calculate the range for the larger time intervals using that mean, which results in the disappearance of the shadowing component:

(4.1.I) $R = \sqrt[\alpha]{\frac{P_{mobile}}{Noise} * 10^{0/10}} = \sqrt[\alpha]{\frac{P_{mobile}}{Noise}}$

This enables us to determine which base-stations are in range of the mobile. The mobile then tries to subscribe to a base-station, preferring the closest one. Instead of calculating the residual capacity for each cell and its neighbors, we introduce a hard capacity for each cell. This hard capacity directly tells us the maximum allowed number of subscribers.
For this hard capacity we only consider intra-cell interference. Since we assume power control that means that:

$$\mathrm{SIR}_k = \frac{S}{I(k) - S} = \frac{S}{Sn_k - S} = \frac{1}{n_k - 1}$$

and since the residual capacity is defined as :

(4.1.II) $R_k = \left\lfloor \frac{1}{\mathrm{SIR}_{th}} - \frac{1}{\mathrm{SIR}_k} \right\rfloor = \left\lfloor \frac{1}{\mathrm{SIR}_{th}} - \frac{n_k - 1}{1} \right\rfloor$

we can see that SIR_{th} correlates directly with the hard capacity. For example an SIR_{th} of 0.02 directly translates to a hard capacity of 50.

If the target base-station of the mobile has not yet reached that hard-capacity, the mobile is allowed to join.

4.1.2 Determining Data-Rates

While the main heuristic could individually schedule which mobiles are allowed to send or receive data at each timestep, we cannot do that with intervals that are several seconds long. If we were to use the same scheduling algorithm on a time interval of 5 seconds, only one mobile would be able to receive data from the base-station, while other mobiles would starve and their messages would expire with a high probability.
For this reason we will instead approximate the achievable data-rates for up- and downlink based on currently available information. We start by determining the maximum data-rates for each direction. This is done in the same way as in the Main heuristic.

First we calculate the up- and downlink SIR as before, with the small change that when calculating received signal powers, we will replace ξ by its mean 0 since we are working with larger time intervals. This leaves us with the same general formulas for up- and downlink SIR that are used in 3 section 2.6.2, with the difference that we calculate the received power as follows:

(4.1.III) $S_k = P_k * d_{mk}^{-\alpha} * 10^{0/10} = P_k * d_{mk}^{-\alpha}$

We then determine the signal modulation technique with the highest gross-data rate that still fulfills our BER limitations by calculating the respective EbNo values from the SIR and consulting our lookup table.

The acquired data-rates represent a maximum, which is only achievable if the mobile is the only entity competing for data transmissions. Since there are usually multiple mobiles subscribed at a base-station this is rarely the case.

To compensate for this, the base-station separates its subscribers in those who currently want to use the uplink and those who want to use the downlink. Each group will potentially compete for transmission slots among each other. To determine which devices will actually compete the base-station then compares the starting time of the current message of each mobile. If two messages start during the same second they are considered as competing.

For devices in the downlink, having competing mobiles means that they would theoretically have to share the the total number of slots that are available for the downlink, meaning that they would only be able to send a fraction of the time depending on the number of competitors. To simulate this we divide the downlink data rate of each mobile by the number of competitors for the second their current message starts.
If no other mobiles want to receive a message during the same second, the number of competitors is only the mobile itself, which means it can receive at its maximum rate. If one other mobile wants to send the number of competitors increases to 2 and the data-rate is halved. For 3 competitors each mobile can only use a third of the data-rate and so forth.

In uplink direction a base-station is able schedule multiple mobiles at the same time, depending on the 'budget' and the cost of a connection. According to [TLKF09] this number is usually less than four. Since uplink modulation in HSUPA only supports up to 16QAM, which achieves low enough BER rates at relatively low power output, 3 parallel transmissions seem reasonable.
Therefore when it comes to the uplink direction we assume that slots will only have to be divided for a number of at least 4 competitors or more and the approximated data-rate calculates is

(4.1.IV) $Rate_{approx} = Rate_{max} / \max(1, \frac{competitors}{3})$

since the rate cannot exceed the maximum.

4.1.3 Message Transmission

Once a data-rate is estimated for each communication direction, the Point in Time heuristic begins sending messages. Messages are always sent as a whole. This means that for messages whose starting time lies close to the end of the interval, we could end up already sending into the next interval.

To determine the time needed to send a message, the total amount of transmitted bits is calculated by adding an approximated number of failed packets to the messages original size.
First we look up the BER and calculate the packet size for the current transmission, as it was described for the Main heuristic. The average number of failed packets can be calculated by multiplying the Packet Error Rate with the messages original number of packets. PER can be obtained by multiplying BER with packet size and dividing the messages size by the packet size provides the original number of packages. To add a certain degree of fluctuation the actual number of failed packets is then given by a normal distributed random value, with the distributions mean being the average number of failed packets.

Message failures occur if a packet fails 3 times in a row, which happens with a probability that is the third power of the PER. To account for these failures we generate a uniformly distributed random value between 0 and 1 for each sequence of three in the number of failed packets. If one of these random values is below the probability for message failure, we assume that such a failure occurred. In this case an additional number of bits is added to the messages final size.
If we know which sequence of packets caused the failure, we can determine where it occurred in regards to the other packet failures. If we assume packet failures to be uniformly distributed in general, we can also tell where in the message the failure occurred. Since everything from before that point will have to be retransmitted, we add an amount of bits equal to the part of the message before the failure to the final message size.

After adding all the failed packets and if necessary the retransmitted part of the message to the final size, we can calculate the messages completion time $TEnd$, at which point the message will be completely transmitted by using the data-rate we estimated in the previous step.

$TEnd = TStart + \frac{MessageSize}{DataRate}$

To determine if the message was successfully sent, we then need to check if $TEnd$ exceeds the expiration time of the message. If not the message was successfully sent and the mobile will continue processing the next message in the queue, if that messages starting time is still within the current time interval. If no messages are left in the queue but the mobile is still marked as active, a new a pair of uplink and downlink messages will be added to the queue.
If $TEnd$ exceeds the expiration time, the message will me marked as expired and removed

from the queue. In case the expired message was an uplink request we also need to to remove the next message from the queue, since it would be a downlink response to a request which was never received.

5 Position Extrapolation Heuristic

The previous Point in Time Heuristic departed from the small 2 millisecond intervals that were previously used. Instead the heuristic is supposed to run on intervals of several seconds. During such a long interval the position of a mobile device is probable to change by an amount that will also influence signal quality.
Since the Point in Time heuristic only uses the conditions encountered at the beginning of an interval, it cannot account for such changes, which might lead to growing inaccuracies for larger time intervals. Extreme condition changes between intervals, are an indicator that parts of the last interval were sent using over- or underestimated conditions.

In order to smoothen transitions between intervals and account for probable changes in conditions during the interval we define the Position Extrapolation Heuristic. By trying to predict where the mobile device will be located at the end of the current interval, we are able to anticipate probable changes in signal quality and other conditions.
In this heuristic the expected next Position of a mobile device is extrapolated by calculating the change between its current and last known actual position. This change tells us in which general direction the device travels and at what speed. To determine the next expected position of the mobile we assume that the mobile will continue that same movement for the next interval. Although this may not be exactly the case, this approximation should still be helpful due to the restrictions of natural movement.
If a device travels at low speeds, as a pedestrian does for example, it will not travel large distances during an interval of seconds. This means that even if the direction changes abruptly during the next interval, the false predictions will not have a large negative effect on the result since communication conditions should not drastically change for smaller distances.
In case a mobile moves at high speeds, distances traveled during an interval might be large enough to observe clear changes in communication conditions. When traveling at such high speeds however it is unlikely to perform sharp turns and change directions by a large amount. This can be observed for cars on a highway, or trains on a railway.

The path between the mobiles current position and the extrapolated next position is then divided into segments of equal length. Conditions for mobile communication, such as SIR values, are then calculated at the starting point of each segment and their overall average is used to simulate communications for the current time interval.
Contrary to the Point in Time heuristic we will not send messages as a whole. Instead we will stop sending at the end of an interval and continue the current message with the updated conditions of the next interval.

5.1 Implementation

As before we use mobiles and base-stations for the simulation. The simulation process will again be described in the same three steps as the previous Point in Time heuristic.

5.1.1 Call Admission Control

Call Admission Control is performed in the same way as for the Point in Time Heuristic.
Since we are again working with time intervals in the dimension of multiple seconds, we determine the uplink range of a mobile by using equation (4.1.I), averaging over the shadowing component. With this we determine all base-stations that the device can potentially subscribe to. Starting with the closest one, the mobile tries to subscribe to each base-station, until one of them accepts it.
Base-stations are given a hard-capacity, which is again determined by equation (4.1.II). A mobile can only be accepted if the current number of subscribers at a base-station is below that hard-capacity limit.

5.1.2 Determining Data-Rates

While the Point in Time heuristic calculated data-rates only for its current position, we will now do so for an additional set of positions, in order to obtain an average data-rate over the mobile devices expected path for the next interval.

The amount of additional positions at which we determine data-rates is given directly by the number of segments we chose to divide the extrapolated path to the next position into. If we decide not to segment the path, we will only have to use the next position. For 2 segments we will use one additional position, the midpoint between current and next position. 3 segments will result in 2 additional positions along the extrapolated path and so forth. Using more segments will improve the accuracy of the average, if the extrapolation is not too far off the actual path. This improvement will reduce with the number of segments and each additional segment will increase the simulation cost.

To determine the expected next position of the mobile device, we first calculate the actual change of position during the last interval. For this purpose we assume $dif(position1, position2)$ to be a function that returns the difference between 2 coordinates. The expected next position is then calculated by

$$nextPosition = currentPosition + dif(lastPosition, currentPosition)$$

Accordingly positions of additional points on the path will be calculated by

$$iPosition = currentPosition + i * \frac{dif(lastPosition, currentPosition)}{n}$$

where n is the total number of segments and i refers to the ith extrapolated point on the path. Since we already calculated the next position separately, i ranges from 1 to $n-1$.

Since the mobile is moving during the time interval, it can happen that it moves out of the range of its original base-station and into a neighboring cell. Because of this we determine the closest base-station for the mobile at each additional position. When calculating SIR values, we will then assume that it is connected to that base-station. We will not perform a complete CAC step, because we assume that on average the probability of a mobile entering the cell is as high as that of a mobile leaving the cell, in which case the overall cell load would not change.

We now know all extrapolated positions and have determined to which base-station the mobile would be connected to at each of them. With this information we calculate the up- and downlink SIR as previously described for each position.
Calculation of up- and downlink SIR for a mobile at a certain position is done the same way as for the Point in Time heuristic however. First received power levels for both directions are calculated using equation (4.1.III). These power levels are then used in equations (3.2.I) and (3.2.II) to get the according SIR values.
These SIR values are then used to look up the signal modulation techniques with the highest gross data-rate, that fulfill the BER limitations for the system, at each point. The resulting data-rates for each direction are then added up and divided by the number of positions we used to get the average maximum data-rate during the interval.

To account for multiple mobiles competing for transmission slots at the same time we use the same method for data-rate adjustment as with the Point in Time heuristic. At the beginning of a time interval the base-station checks the direction of a mobiles current message and divides the devices by up- and downlink. It then counts for each mobile, how many other devices from the same group, will start sending or receiving during the same second. These are the devices competitors and we will adjust the average data-rate to account for them.
Downlink slots can only be used for one mobile device, so the data-rate will be directly reduced for every other mobile that wants to receive data simultaneously. Therefore the actual downlink data-rate for a mobile during the next interval will be the unadjusted data-rate divided by the number of competitors, including the mobile itself.
Multiple mobiles can send data in uplink direction simultaneously, as long as the 'budget' of 6 dB additional interference is not surpassed. This usually allows 3 senders to communicate simultaneously with the base-station. Therefore we again use equation (4.1.IV) to adjust the uplink data-rate.

5.1.3 Message Transmission

In the Postition Extrapolation heuristic messages are sent until the end of the time interval. If a message is not completed by then, it will be continued at the start of the next time interval using the new data-rate.

In general messages are sent per Transmission Time Interval (TTI). This means that the heuristic first calculates how many bits can be transmitted per interval of 2 milliseconds by using the average data-rates calculated in the previous step. The number of bits per TTI is then divided into packets, with the maximum size of a HSPA packet being 12000 bits.

In order to check for transmission errors, we calculate the Packet Error Rate (PER). To do so we first look up the BER for our currently used signal modulation technique. The PER is then obtained by multiplying the packet size with that BER.

The message is then sent in steps, that each represent one TTI.
Each step we try transmit the amount of data per TTI we previously calculated in the form of packets. For each packet we generate a random value that is equally distributed between 0 and 1 to check if the transmission was successful or not. If the random number is below the PER we assume a transmission failure for the packet and it will have to be retransmitted. If the same packet fails 3 times in a row, the message is considered to have failed and will have to be retransmitted completely, including everything that has been sent up to this point. In case no transmission errors occur the remaining amount of data that has to be sent is reduced by the size of the packet.

Once the maximum amount of data per TTI has been transmitted, the messages duration is increased by 2 milliseconds. To conclude the step, the transmitting mobile device checks if the starting time of the current message plus its duration equals the end of the time interval, or the expiration time of the message.
In the first case the transmission is interrupted, until new data-rates are calculated for the next time interval. In the second case the message is marked as expired and is removed from the queue. As always if the message was an uplink request, we also need to remove the following downlink response from the queue, since it would never have been created.

If the complete amount of data has been transmitted and the expiration time has not been reached, the message is marked as completed and removed from the queue so the next message can be processed.

6 Path Prediction Heuristic

The Path Prediction Heuristic is motivated in the same way as the previous Position Extrapolation Heuristic. By predicting where the mobile device will be at the end of the time interval we can predict how the conditions for mobile communication will change.

The main difference between the two heuristics is the way in which the next position is determined. The Position Extrapolation Heuristic uses a fairly simple linear extrapolation to determine the next position of a mobile device. For the Path Prediction Heuristic we assume the use of a path prediction algorithm.
Path prediction is a well researched topic due to its applicability to Location Based Services. There exist a large number of algorithms, some of which will for example use GPS data of mobile phones together with historical trip data to predict future positions. an example for such an algorithm can be found in [PMBL+08].

We will not discuss path prediction in any more detail, since it would exceed the scope of this work. Instead we assume that the chosen path prediction algorithm will produce very accurate results. To simulate this, we will assume the mobiles current position to be predicted by the path prediction algorithm, and then process messages whose starting times are in the last time interval. Due to this the Path Prediction Heuristic is always one time interval behind the other heuristics.

Calculating the average data-rates for an interval is then done identical to the Postion Extrapolation Heuristic. We interpolate the path between the mobiles last position and its 'predicted' current position with a straight line. This path is then divided into a number of segments at the end of which an additional intermediate position will be calculated. For each position, up- and downlink data-rates are calculated. The average of these data-rates is then used to simulate message transmission during the last time interval.

6.1 Implementation

We again use mobiles and base-stations as components of the system. The usual three step structure also applies.

6.1.1 Call Admission Control

Due to time intervals of several seconds, we again use equation (4.1.I) to determine the range of the mobile device, averaging over shadowing effects. After finding all base-stations in range, the mobile tries to connect to one of them, preferring closer base-stations over more distant ones.
Base-stations are assigned a hard-capacity determined by equation (4.1.II). If the base-station currently has less subscribers than the hard-capacity, a mobile is allowed to subscribe and will be serviced.

6.1.2 Determining Data-Rates

The process of determining the data-rates used for transmissions is largely equivalent to the one performed for the Position Extrapolation Heuristic, which is described in more detail in 5.1.1. The only difference being that we will not have to determine the mobiles next position, since we assume that the current position is already the result of a path prediction algorithm. In order to approximate the changes in communication conditions during the interval, we divide the path between current and last position in a specified number of segments. At the end point of each segment we determine the maximum data-rate the mobile could achieve. The positions of these end points is calculated by

$$iPosition = lastPosition + i * \frac{dif(lastPosition, currentPosition)}{n}$$

with *iPosition* being the position of the end point of the *i*th segment.

To account for possible handovers between base-stations during the movement, we determine the the closest base-station for each position, and assume that at that point the mobile is subscribed to said base-station.
We then continue to calculate up- and downlink SIR values for the last, current and all intermediate positions. Like the other heuristics using intervals of several seconds we first use equation (4.1.III) to calculate received power levels, which are then used in equations (3.2.II) and (3.2.I) to obtain the desired SIR values.
These SIR values are then used to determine which signal modulation technique can be used at each point. We always want to select the modulation technique with the highest gross data-rate, that still satisfies our BER limitations.

Knowing the maximum data-rate at each position, we are now able to calculate the average of these data-rates, which will be uses for simulating transmissions during the time interval. This is done by summing up all maximum data-rates and afterwards dividing that sum by the number of positions that were used.

To account for multiple devices transmitting at the same time, we determine a devices number of competitors. Each other mobile whose current message is transmitted in the same direction and starts during the same second is considered as competing.

Downlink slots can only serve one mobile each TTI. Competitors directly reduce the amount of slots for each mobile and therefore the data-rate. Hence we adjust the downlink data-rate by dividing it by the number of competitors, including the transmitting mobile itself.
During one TTI on average 3 mobiles are allowed to transmit in uplink direction simultaneously. Each additional competitor will reduce the data-rate. Therefore we again use equation (4.1.IV) to adjust uplink data-rates.

6.1.3 Message Transmission

Like the Position Extrapolation Heuristic, message transmission is done in steps, each representing one TTI of 2 milliseconds.

First we use the average data-rates for up- and downlink to determine the amount of data, that can be sent every TTI. That amount is then divided into packets of maximal 12000 bits each.
To simulate transmission errors we first determine the BER for the current transmission, and then convert it to the PER by multiplying the BER with the packet size.

During each step the previously determined amount of data is transmitted. Packets are each checked for transmission errors by generating a random number and comparing it to the PER. The random numbers are uniformly distributed between 0 and 1, and an error is assumed if a random number is smaller than the PER.
In case of an error the packet will have to be retransmitted. Should the same packet experience 3 transmission errors in a row we consider the message to be failed. In this case all transmission progress is reverted and the whole message will have to be retransmitted from the start.
If a packet is transmitted without error, the remaining amount of data is reduced by accordingly.

At the end of each step, a messages duration is increased by the duration of one TTI, in our case 2 milliseconds. At this point the mobile checks if the time interval is over or if the message is expired, by comparing the messages starting time plus its duration to the end of the time interval and the messages expiration time.
In case the interval is over, the transmission is paused until a new data-rate is determined, with which the transmission will continue. Exceeding the expiration time on the other hand, will cause the message to be aborted and removed from the queue. If the message was an uplink request, the according downlink response will have to be removed from the queue as well.

If all data of the message is sent successfully ind in time, the message will be marked as completed and the next message will be processed.

7 Interval Scheduling Heuristic

All previous heuristics treat mobile devices equally, regardless of how good or bad their conditions for mobile communication are. When looking at the proportional fair scheduler which is used in real systems however, we see that mobiles with good conditions are treated favorably and assigned more slots than mobiles with worse conditions.

We want to emulate this behavior with the Interval Scheduling Heuristic. This heuristic will also operate on time intervals of several seconds. Therefore the initial maximum data-rates will again be obtained by averaging over short term influences like shadowing. This is done in an equivalent way to the Point in Time Heuristic.
Once the initial possible data-rates are determined, we calculate a weight for each mobile. This weight is then used to adjust the data-rates and indicates the priority given to each mobile device by the scheduler. Mobiles with a bigger weight would be assigned more slots and will effectively transmit data faster than other mobiles competing for the same slots.

In order to determine the weights for all mobiles we will use a proportional fair scheduler. Since the average data-rates used for scheduling are updated after each assigned slot we will have to determine how many slots have to be assigned during the interval. This can easily be done by dividing the interval length by 2 milliseconds.
At the beginning of each time interval the base station will perform scheduling for its whole duration. The initial data-rates that were calculated for the mobiles current position are used as expected data-rates for each step, while average data-rates for all subscribed mobiles are updated for every scheduled TTI.

At the end of this process the base-station possesses a list of assigned up- and downlink slots per mobile. To determine the final weight of a mobile in up- or downlink direction, the actual number of assigned slots for that mobile are compared to the median of assigned slots in each direction.

7.1 Implementation

As for the other heuristics, we divide the detailed description of the heuristic in 3 steps. The first two steps of the implementation of the Interval Scheduler Heuristic will be close to the Point in Time Heuristic. During the step in which data-rates are determined we will additionally calculate the weights for each mobile. The third step during which messages are sent is closer to the Position Extrapolation and Path Prediction Heuristic, in that messages are sent in small chunks of data, representing individual TTIs.

7.1.1 Call Admission Control

CAC is performed in the same way as for the other heuristics. We use equation (4.1.I) to determine each mobiles range. Afterwards each mobile tries to subscribe to one of the base-stations in range, starting with the closest one.
If the mobile is accepted by a base-station is defined by a hard-capacity, which is given by equation (4.1.II). Is the number of mobiles that are already subscribed at the base-station less than that hard-capacity, the mobile is accepted. Otherwise the mobile is rejected and will not be served by this particular base-station.

7.1.2 Determining Data Rates

The first thing we have to do in this step is to determine each mobiles achievable up- and downlink data-rates. These data rates are then used during the scheduling, and adjusted by the resulting weights.
We will determine these initial data-rates in the same way as described in 4.1.2. First equation (4.1.III) is used to determine received power levels for up- and downlink for each mobile at its current position. We then calculate the according SIR values using equations (3.2.II) and (3.2.I).
By means of these SIR values we are able to chose which signal modulation technique is used for the next interval, and with that what data-rates are achievable. As with the other heuristics we choose the signal modulation technique with the highest gross data-rate, whose BER resulting from the current SIR is below the minimal BER allowed by the system.

To perform the scheduling at the beginning of the time interval, each mobile will also have to remember its average gross data-rate $R(T)$ up to this point. The number of scheduling steps that will be performed for this time interval is given by

$$Duration_{interval}/TTI$$

Since the TTI for HSPA is set to 2 milliseconds, that means we will perform 500 scheduler steps for each second.
Each scheduler step is performed as described in 2.3. First the base-station calculates each mobiles priority using equation (2.3.I), with $DRC(t)$ being the previously determined achievable data-rates for up- and downlink. The same $DRC(t)$ is used for every scheduler step performed for this time interval.
Now that the base-station knows all the priorities, it assigns an up- and downlink slot to the mobile whose corresponding priority is the highest.
Finally before executing the next scheduling step each mobiles remembered gross data-rate is updated, according to equation (2.3.II). $R_{assigned}$ is the achievable data-rate of a mobile in up- or downlink direction, depending on what kind of slot it was assigned and zero for all mobiles who did not get assigned a slot this turn.

Once all slots available during the interval have been assigned, the base station calculates the median of the number of assigned slots per mobile in each transmission direction.

The median will be used as a baseline to determine the weight distribution of the mobiles. Let the median represent a weight of 1. A mobile that is assigned less slots than the median, will then be given a weight < 1 and its data-rate throughout the interval will be reduced. The actual difference between median and assigned slots for a mobile is an indicator on how strong a mobile is preferred or discriminated during this time interval.

To prevent extreme and unrealistic weights, as for example a zero weight for a mobile that gets not assigned any slots at all, an upper and lower bound is introduced. The following formula is then used to determine a mobiles final weight.

$$weight = \begin{cases} 1.75 & \text{if } slots/median > 1.75 \\ slots/median & \\ 0.25 & \text{if } slots/median < 1.75 \end{cases}$$

Before applying the weight to the data-rates we, check which mobiles are competing with each other. Mobiles are considered as competing if their current message to begin of the interval has the same direction, and that messages starting time is during the same second. In this case the mobiles are competing for slots and will reduce each others effective data-rate.
Since downlink slots can only be assigned to one of the competitors, the downlink data-rate of a mobile has to be divided by the number of competitors, including itself.
In uplink direction, on average 3 mobiles are allowed to send at the same time. Therefore data-rates only have to be adjusted for 4 or more competitors. This can be accomplished by using (4.1.IV).

Finally each mobiles data-rates for up- and downlink are calculated using

$$FinalRate = Min(weight * rate_{competitors}, rate_{max})$$

where $rate_{competitors}$ is the data-rate adjusted for competitors, and $rate_{max}$ is the maximum achievable data-rate without competitors. The upper bound is necessary since the data-rate is limited by the signal modulation technique and the maximum amount of slots. If we would increase it, we would have to use a higher level modulation technique. However since we already chose the best modulation technique that satisfies our BER restrictions, this is not possible.

7.1.3 Message Transmission

Messages are sent similarly to the Position Extrapolation and Path Prediction Heuristic. Data is sent in steps, each representing one TTI.
The amount of data, that can be sent during one TTI can be obtained directly from the data-rate. The data per TTI is then divided into packets, of maximal 12000 bits.
Each step the amount of data in packets that we just determined will be transmitted. To simulate transmission errors, we will generate a uniformly distributed random number for

every bit in a packet. If one of the generated numbers is lower than the BER, the packet is assumed to have suffered a transmission error. In this case the packet will have to be retransmitted.
Should the same packet fail 3 times in a row, we assume the whole message to be failed. This means that all previously transmitted data will have to be retransmitted as well.

At the end of each TTI step the duration of the message will be increased by 2 milliseconds. If the starting time of the message plus its duration exceed the expiration time, the message will be marked as expired and removed from the queue. In case the message was an uplink request the follow up downlink response will also ave to be removed from the queue.
In case the starting time plus duration reach the end of the current time interval and the message is not completely processed, the transmission will be paused until new data-rates are calculated.

Once all data is successfully transmitted and the message is not expired, it is marked as completed and the next message in the queue is processed. In case the queue is empty, but the mobile is still marked as active, a new set of messages will be added to the queue.

8 Evaluation

In the first section of this chapter we will describe the simulation environment and the system parameters that were used in detail. In the following section we will then present and discuss our evaluation results.

8.1 Simulation Setup

To evaluate our main model and the heuristics, we performed simulations using NetLogo [CCL], a small multi-agent modelling environment, that is developed at 'The Center for Connected Learning (CCL) and Computer-Based Modeling'.

The main model and the heuristics were implemented on top of another simulator programed for NetLogo, which simulates movement patterns of the population of a small city. Whenever a person started moving, that person was also considered as an active mobile device. Whenever a person arrived at its destination, that person was considered to have become inactive.
As soon as mobile device became active, a new set of up- and downlink messages was added to its queue. New messages were generated as is described in **??**. The minimum size of an uplink message was set to 250 kilobit to which an additional number of 250 kilobit blocks were added. The amount of blocks that were added to each uplink message was determined by a Poisson distribution with mean of 5 and a left shift of 4. Downlink messages were generated with a minimum size of 500 kilobit and an additional number of 500 kilobit blocks, which was determined by a Poisson distribution with mean 10 and left shift of 8.
The uplink messages starting time was generated by using a Gaussian distribution with a mean of 3 and a standard deviation of 2. Downlink messages are considered to be responses to the uplink message, so their starting time will always be the time at which the uplink message was completed.
To determine expiration times, the time a message would need to be transmitted using the minimal possible data-rate was calculated. That time was then multiplied by a random value determined by a Gaussian distribution with a mean of 5 and a standard deviation of 1. However the expiration time could not be less than 2 times the transmission time at the minimum data-rate.

Base-stations were distributed evenly to form hexagonal cells with a coverage that is equal to the range of the mobile devices. Sender powers for base-stations were set to a maximum of 23 dBml and a minimum of 15 dBm. Mobile devices were always set to a

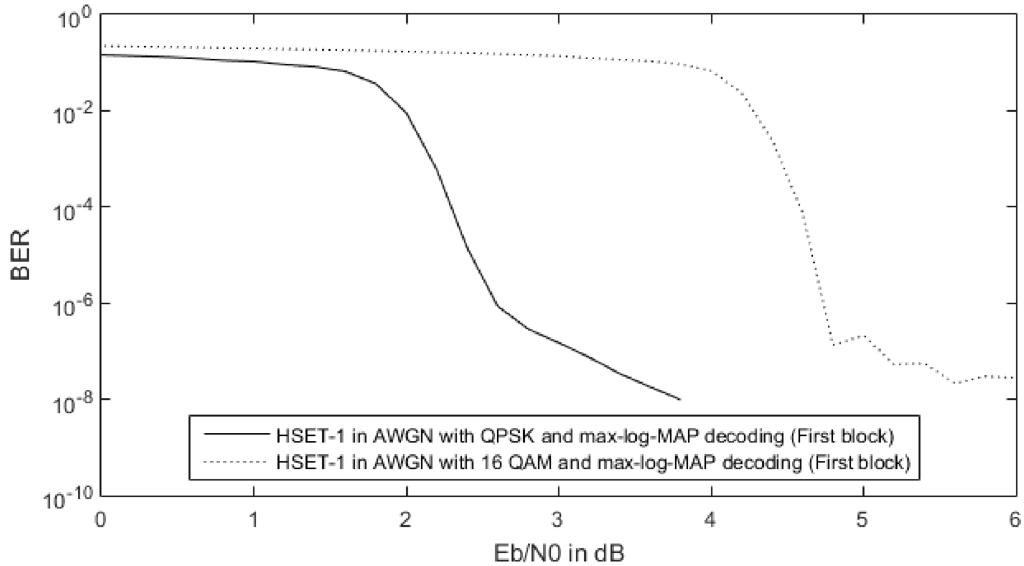

Figure 8.1: Simulated Bit Error Rates according to Eb/No values

maximum sender power of 15 dBm, which is equal to 0.32 Watt.
After consulting [RM92], the path-loss exponent for the log distance path loss model was set to 3 and the shadowing parameter ξ was generated by a Gaussian distribution with zero mean and a standard deviation of 8.
The scheduler memory parameter t_c, which is used in equation (2.3.II) by base-stations was set to 1000.
The Interference for uplink Sir calculation was set to -80 dBm. We chose this value according to [LJL$^+$], which gives measurements fo the noise floor around -100 dBM, to which we added an additional noise rise of 20dBm, which can be controlled by applying power control to mobile devices.
As in [LEZ94] the threshold SIR_{th} was set to 0.01 which results in a hard-capacity of 100 devices per base-station.
As mentioned in 2.5 Bit Error rates were obtained from Matlab simulations using the CML library [CML09]. The resulting BER values for QPSK and 16 QAM signal modulation are shown in figure 8.1. The signal modulation techniques used in the simulation were QPSK, 16 QAM or 64 QAM in downlink direction and QPSK or 16QAM in uplink direction. Each modulation technique was assumed to have a code rate of 1/3 as defined in the HSPA+ specifications. We also limited the number of codes per transmission to 12 instead of a maximum of 15. This means the available gross-data rates were 1.92Mbit/s, 3.84Mbit/s and 5.76 MBit/s for downlink as well as 1.536MBit/s and 3.072MBit/sin uplink direction. In reality higher coding rates can be achieved by puncturing.

All simulations were performed on a Intel Core i5-2500K CPU running at 3.3GHz with 8 GB of Ram. The Operating system used was Ubuntu Desktop 14.04.

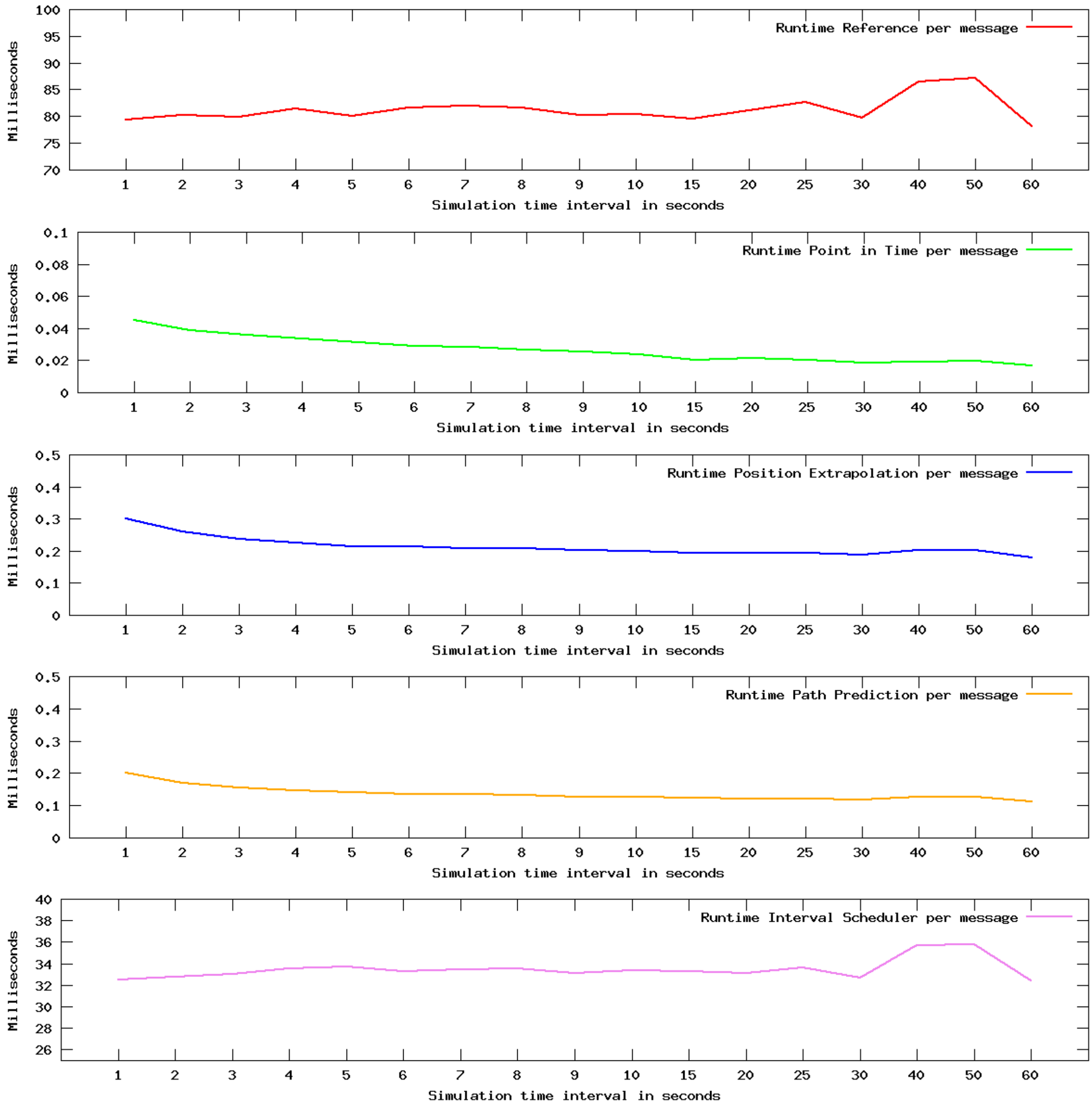

Figure 8.2: Runtimes per message in milliseconds for the main model and each heuristic as measured for different simulation time intervals.

8.2 Results

8.2.1 Runtimes

In this subsection we compare the runtimes of our main model and the heuristics. for different simulation time intervals. In order to make the figures more readable we calculated the runtime per message in milliseconds for each model. The results can be seen in figure 8.2.

As expected the runtime for the main model far exceeds that of the other heuristics. This is easily explained by the fact that the model always operates in intervals of 2 milliseconds and performs detailed scheduling and accounts for shadowing each time the SIR values and ranges are calculated.
This also explains the high cost of the Interval Scheduler heuristic. Although the heuristic only performs scheduling at the beginning of each interval, scheduling itself is still performed for every TTI of 2 milliseconds. This would also explain the obvious similarity between the two figures.
We can also observe that the runtime of every other heuristic decreases with longer time intervals. This can be accredited to the fact that each of these heuristics calculates data-rates only once for each time-interval, regardless of the number of TTIs. Since calculations have to be performed less frequently for longer time intervals, the additional computation cost decreases.

8.2.2 Message Durations

Figures 8.3 and 8.4 show the median of the differences between simulated message duration in down- and respectively uplink direction.
We see that although we increase the time interval, the median is relatively stable. This can be accounted to the fact that, while a heuristic will overestimate data-rates for some intervals, it will also underestimate them for others. This will result on average in an equal number of messages that are sent too fast and those that are sent too slow. Since the median is taken from in between the two extremes, it makes sense that it fluctuates slightly around zero.

One might however expect that with increasing time intervals during which data-rates are over- or underestimated, the standard deviation from the median might increase. The standard deviation from the median is is given in the figures by ways of error bars in y-direction. Contrary to the expectation, the standard deviation is also relatively constant regardless of time interval, except for a few extreme values for the Point in Time heuristic and the Position Extrapolation heuristic.
This behavior can probably be accredited to the fact that the available data-rates are limited. Since we only used signal modulation techniques up to 64QAM with a code rate of 1/3 the difference between maximum and minimal achievable data-rate is not that big. Even less so for uplink transmissions where we have only 2 different data-rates available. Therefore it is impossible to over- or underestimate a messages duration by an amount that would

result in significantly higher standard deviations. We can however observe that standard deviations for downlink transmissions are in fact higher, since the difference between minimum and maximum data-rate is double as high as the difference between the respective uplink data-rates.

8.2.3 Expired Messages

Figure 8.5 shows us the difference between the number of expired messages that were observed for the main model and each heuristic. We clearly see that all heuristics except for Path Prediction tend to overestimate the number of expired messages. The three figures even look extremely similar.

The Point in Time heuristic starts out rather close to the main model which is used as a reference, but gradually worsens for increasing time intervals. One reason for this could be the fact that data-rates are only determined once per interval. In case a mobile is positioned at the edge of a cell at the beginning of an interval, its estimated up- and downlink data-rates will be low. These data-rates are used over the course of the whole interval, so the longer the interval the longer the time during which the mobile will have to send using the low data-rate. Using these low data rates over a long time will increase the probability of an expired message.

While the Position Extrapolation heuristic starts of underestimating the number of expired messages, we see the same behavior as for the Point in Time heuristic. The fact that the number of expired messages during smaller time intervals is underestimated could be attributed to the following factors. For smaller time intervals the two point extrapolation will be rather accurate since the mobile will not have been able to move great distances. Also while the main model experiences random intervals of bad and good connections due to the shadowing parameter, the data-rate for the Position Extrapolation is fixed during an interval. This means that if the heuristic determines it can use the highest available data-rate it will be able to do so over the whole interval. The random component in the main model however will lead to lower data-rates from time-to time. With increasing time intervals the extrapolation will become less accurate due to the larger changes in position in between intervals. It could also be possible that since we again only calculate data-rates at the beginning of an interval, the effect of intervals with a low data-rate will become more dominant with increasing time steps.

Contrary to the other heuristics, the Path Prediction heuristic consistently underestimates the percentage of expired messages by about 0.3%. Instead of increasing with larger time interval, the number of expired messages actually decreases.

Like the Point in Time heuristic and even more so, the Interval Scheduler overestimates the amount of expired messages. This might have similar reasons as we expected for the Point in Time Heuristic. Since data-rates are only calculated once at the beginning of each interval, a mobile might be stuck with a low transmission rate, which increases the probability of expired messages. A problem unique to this heuristic might be a property of the proportional fair scheduler. The proportional fair scheduler tries to take advantage of periods in which a mobile has a strong signal. Therefore mobiles with stronger signals are preferred. To provide fairness however, the scheduler also takes into account a mobiles gross data-rate in the near past. How long in the past the scheduler remembers is determined by the parameter t_c, which in our simulation was set to 1000, which relates to a memory of one second. If a mobile had a very strong connection during the last interval, its gross data-rate will be relatively high. If that mobile would only be able to send at a very low data-rate during the next interval, then it would hardly be assigned any slots at all, due to its high gross data-rate in the past. This in turn would result in very low weight and probably an even lower data-rate, largely increasing the probability of a timeout.

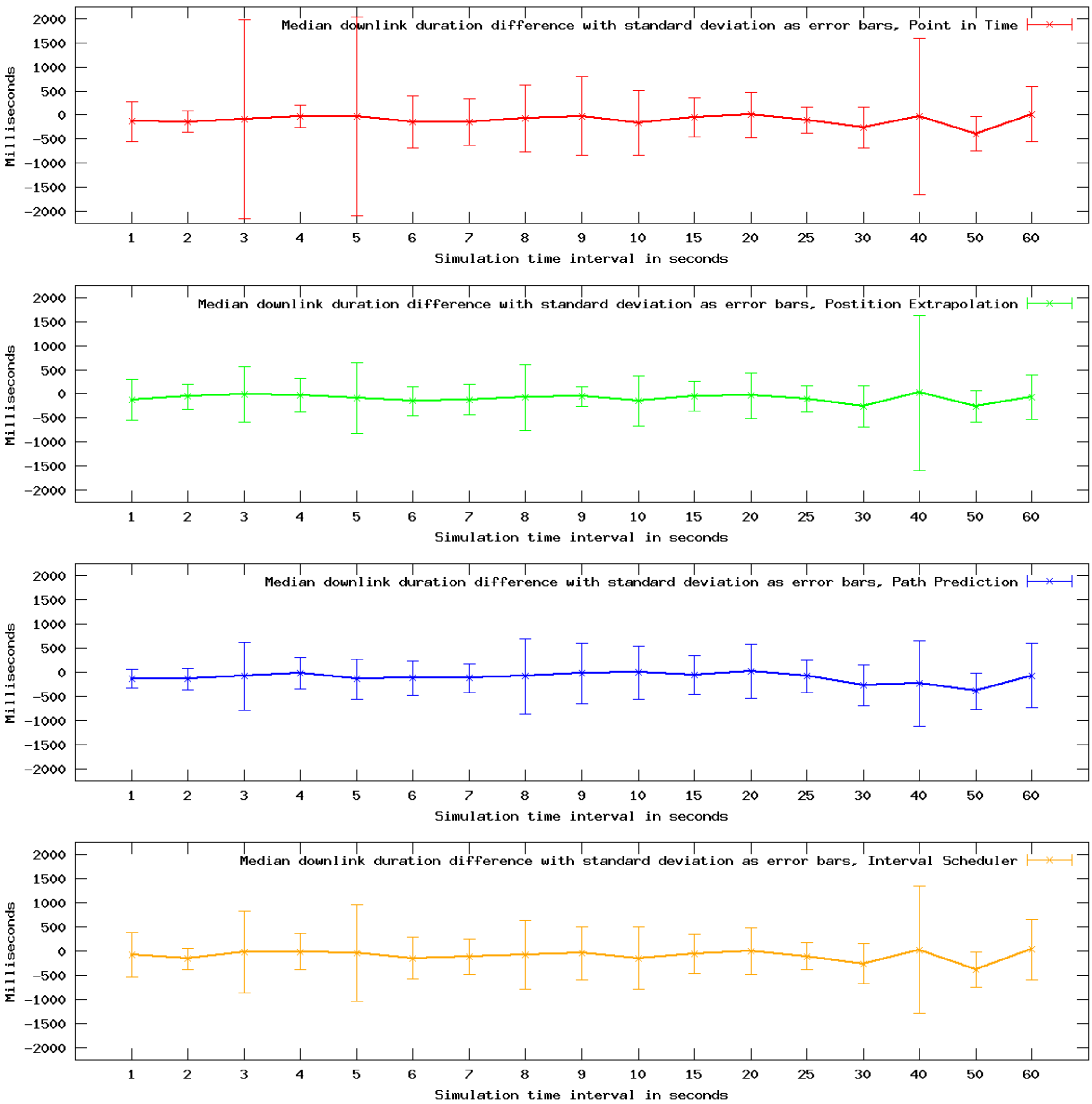

Figure 8.3: Difference between message durations in downlink direction. The figures show the Median of the difference for each time interval, and the standard deviation as error bars.

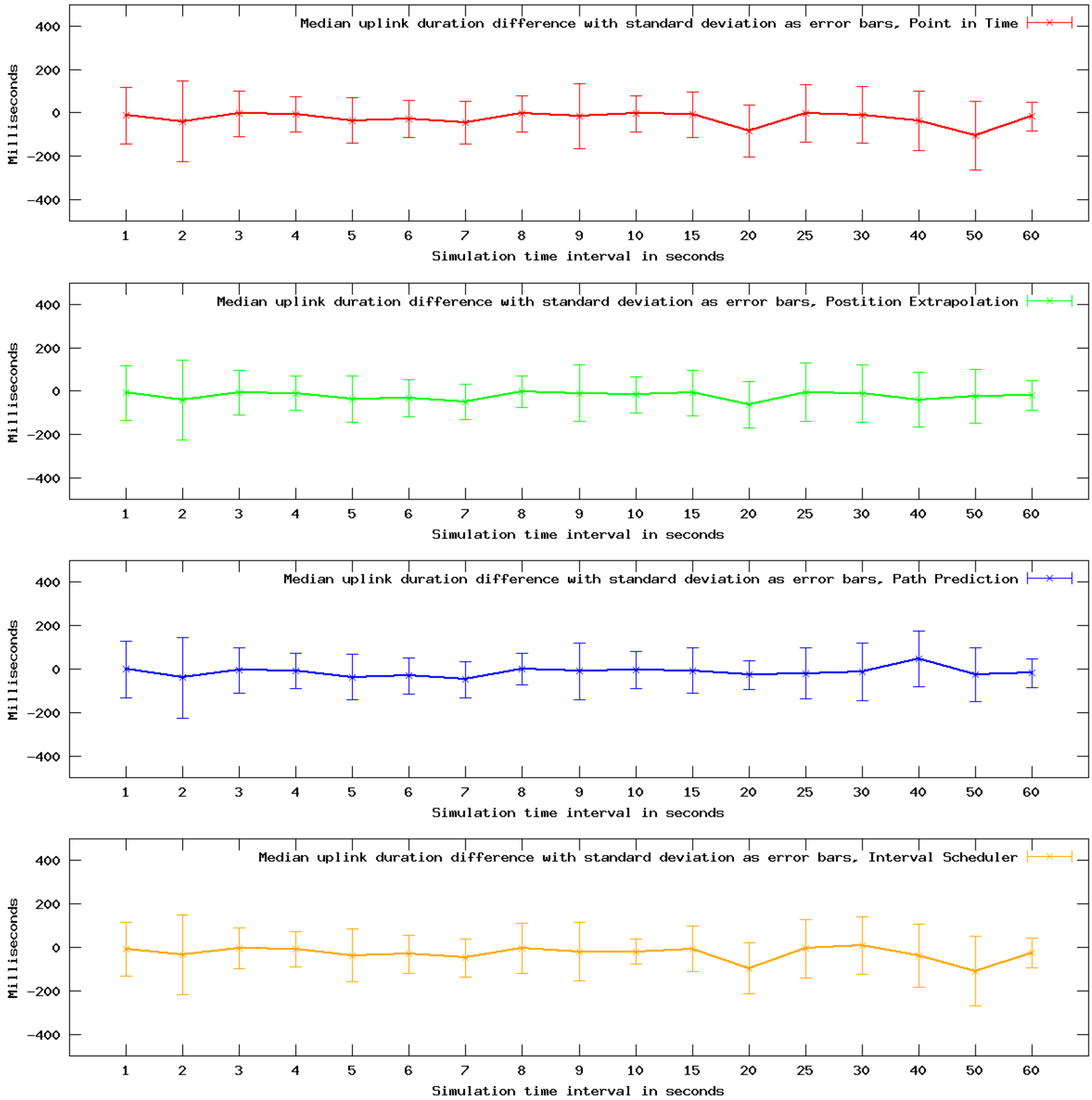

Figure 8.4: Difference between message durations in uplink direction. The figures show the Median of the difference for each time interval, and the standard deviation as error bars.

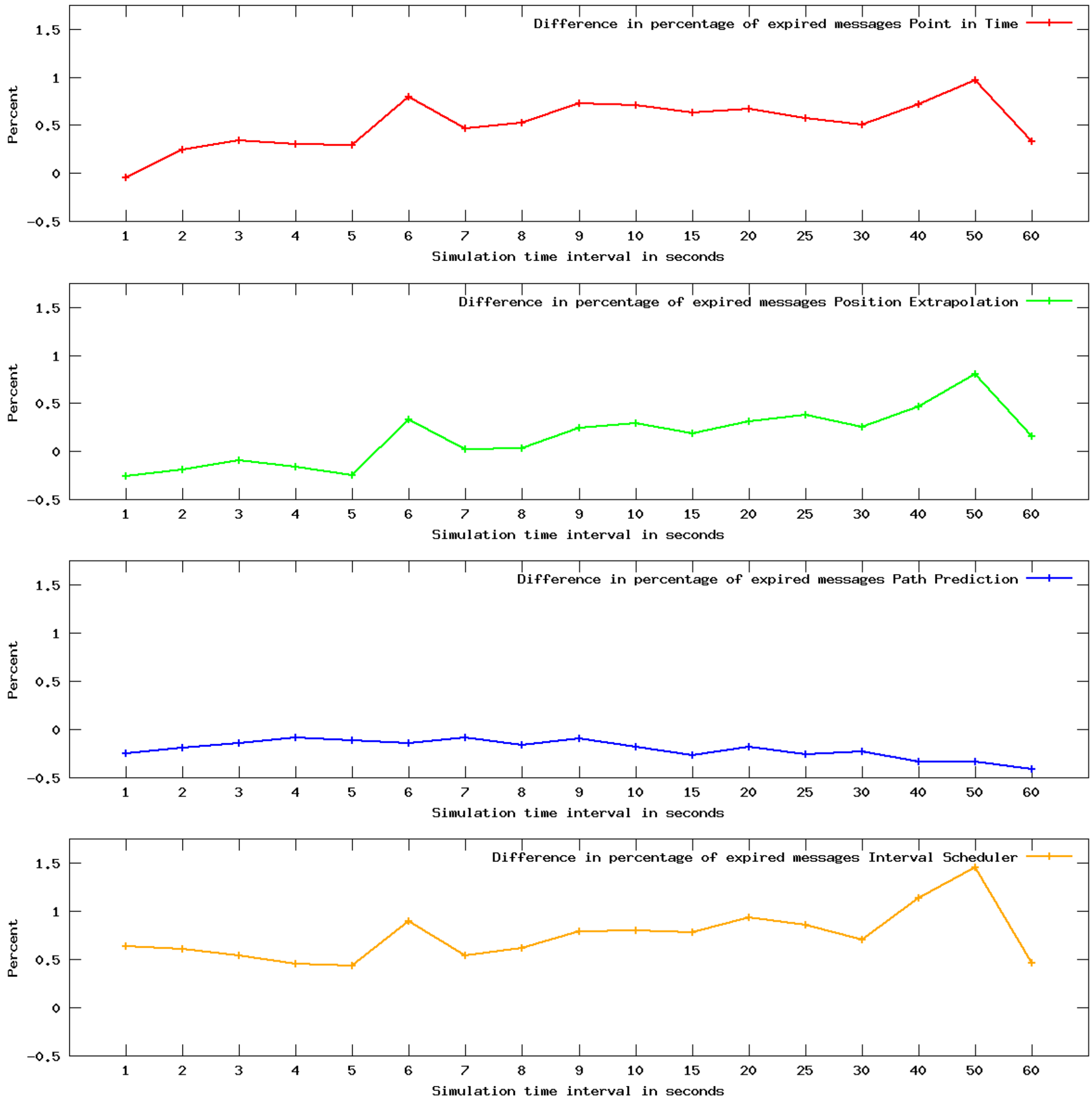

Figure 8.5: Difference between the percentage of expired messages of the main model and the heuristics for different time intervals. Percentage was calculated in regard to all transmitted messages and the difference was determined by subtracting the main models percentage from that of the heuristic.

9 Related Work

Since UMTS was first introduced in 1999, it has been the topic of many research papers that propose improvements and additions to the standard. In 'Traffic Characterization for a UMTS Radio Access Network' [EAH02] for example, a UMTS traffic model is introduced to evaluate Measurement Based Admission Control (MBAC). MBAC is a class of admission control algorithms that use on-line measurement of network properties to decide if new data flows are accepted or not. Examples for this type of Admission Control algorithms can be found in [GT03] and [QK98].

However we are more interested in the UMTS traffic model. The model itself is divided into several parts. In the first part connections are modeled. New connections are assumed to arrive following a Poisson process and have a holding time that is exponentially distributed. The model also differentiates between voice and data connections, since both can use different types of channels.
These channels are then modeled in the next part. Since the document refers to an earlier UMTS release than our work, only dedicated and shared channels are considered. Voice connections will always use a dedicated channel, while data connections have the option to use a shared channel, which is divided between a number of devices in the same cell.
Part 3 introduces an activity model for the channels, based on a two state Markov process. Inactive and active periods for a voice channel indicate for example periods of silence and talking during a conversation. The Markov process will alternate between active and inactive with both states being assumed to be exponentially distributed.
The following part of the model determines packet sizes for each channel. Packet sizes differ for voice and data connections and depend on Quality of Service (QoS) demands.
In the final part of the model all the previous information is aggregated to calculate the batch size contributed by voice and data connections and the mean traffic rate of the system.

Another analytical model for a mobile network is presented in 'LTE Radio Schedulers Analytical Modeling using Continuous Time Markov Chains' [ZWLG13]. As the name implies this work focuses on the current LTE standard, which offers higher data-rates than HSPA+, but is not yet fully deployed.
This work simulates the performance of a number of different scheduler algorithms. To do so the model first determines the number of currently communicating devices using a Continuous Time Markov Chain. The chain has a total of N+1 states, with N being the maximum amount of possible senders. Each state represents how many devices are currently active. Therefore a transition from state n to state n-1 means that a device has finished sending and has become inactive for the time being. This happens with the generic rate $\mu(n)$. On the other hand a transition from state n to n+1, will occur with the rate $(N - n)\lambda$

and means that an inactive device has just become active again.
Like HSPA+ and other 3GPP standards, LTE supports the use of various signal modulation schemes, which as we know directly influence the achievable data-rate. Instead of calculating SIR values, this model assigns every signal modulation scheme a probability with which the active devices are able to use that modulation technique. Each active device is then assigned a signal modulation technique based on these probabilities.
To simulate the systems performance for different scheduling algorithms, selection algorithms are introduced that select a set of active devices based on their achievable data-rates and other criteria relevant for the individual scheduler. A MaxThroughput scheduler for example will always select a set of devices with the highest achievable data-rates.
Based on the selections for each scheduler the systems throughput is then evaluated.

The previously mentioned models focus on estimating a systems throughput, since it is a very popular topic in the area of mobile communications. This seems especially the case since Bit Error Rates for wireless communication have been reduced due to the introduction of new FEC methods like turbo codes.

'Multicast-Based Interference of Network-Internal Loss Character' [CDHT99] introduces a model to determine loss rates for individual links in a large scale network, using multicasts.
In large scale networks it is impractical to monitor every link on an end-to end path. Therefore multicast traffic measurements are used in this work to determine network characteristics. In order for the model to function, the network paths first have to be projected onto logical trees. As to how this is done can be taken from the document.
Packet loss is modeled as a sequence of independent Bernoulli processes. Each logical link between two nodes is assigned a loss probability of α_k, where k is the node at which the link terminates. Each node is assigned a value $X_k \in \{0,1\}$, signaling if the packet was received by that node or not.
X_{root} is always 1 since it is the origin of the packets. From there the probability that a child node receives the packet and $X_{child} = 1$, is given by α_{child}. Accordingly the probability of $X_{child} = 0$ is given by $1 - \alpha_{child}$. If a node k does not receive the packet and $X_k = 0$, then X_j for all children j of k is also 0.

In a real system α values can be determined by sending a series of packet probes via multicast. Knowing the structure of the multicast tree and how counting how many probes are returned by the individual devices, then allows calculation of α values for each link.

All of the previously mentioned works modeled only one aspect, be it throughput or loss rate. In 'End-to-end TCP Performance in W-CDMA / UMTS' [CCE03] a model for a TCP channel consisting of both wired and wireless components is described and its throughput is calculated as function of packet loss probability and Bit Error Rate.
Since the document was released in 2003, an older Release of UMTS is used. This is important to mention since the model assumes the use of a simple Automatic Repeat

reQuest (ARQ) as the only way to of error correction.

The model separates the end-to-end path into two parts. The wired part is assumed to be governed by packet loss and retransmissions, while the wireless part is assumed to be dominated solely by error.
TCP throughput is then calculated as a function of packet loss and Frame Error Rate (FER)

$$Th(p, FER)^{-1} = T_0 min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2) + \sqrt{\frac{2bp}{3}}$$
$$(RTT_{wire+nbD_{ARQ}} + RTT_{wireless} + ND_{ARQ}\frac{FER(nb-1)}{1-FER})$$

with RTT_{wire} and $RTT_{wireless}$ being the round trip times in the wired and wireless section, T_0 as the TCP time-out, b the number TCP segments sent back-to-back for which only one cumulative ACK is generated, D_{ARQ} denotes the fixed component of the ARQ delay and n is the number of ARQ frames transmitted over the radio link per one TCP segment.

10 Conclusion

Data-transmission over mobile communication networks will become even more prominent in the future. New standards like Long Term Evolution (LTE) will further increase available data-rates. The inherent problems with wireless communication will however remain. Factors like limited sender power on mobile devices or interference will still have to be accounted for. In such cases good simulation models for mobile communication can help with design questions and be used to predict a systems behavior.

In this work we provided a simulation model for mobile communication, that is derived from previous analytical work and current specifications. The model adheres to the currently dominating standard HSPA+ and is analytically correct. We felt however that the performance of the model could still be improved. Therefore we introduced a number of more abstract heuristics to estimate the behavior of mobile communication in a predefined system. In conclusion we implemented our model and the heuristics in a simulator and evaluated their performance.

10.1 Future Work

Our model is able to estimate the behavior of a mobile communication system during a simulation. For real time applications however this might still be unacceptable. It might be beneficial to find a method with which a mobile can estimate the result of a transmission on the fly. If the probability with which the message would expire is too high, it might be better to delay it. This will help reduce cell load and therefore enable other transmissions to be finished faster, so that the mobile will be able to transmit with a higher gross data-rate at a later point. This way the mobile would also conserve power since sending data consumes a large amount of power. Such a model might for example be interesting for the drone delivery system described in 'Increasing Availability of Workflows Executing in a Pervasive Environment' [SBTR14] If a drone would be able to determine if it will be without connection in real time, precautions could be taken.

Since the mobile communication process is controlled by the base-station however, mobiles have little information about the state of the system. Since system wide properties like cell load are only known to the base-station, the mobile will have to decide if it should try to send a message based on local information.
Cell load for example might be derived from the number of slots a mobile is assigned in relation to its signal quality. If the mobile has a very good signal quality and is still only

assigned few slots, then chances are high that cell load is rather high at the moment.

A mobile also regularly measures its signal quality towards the base-station. If the mobile moves through disadvantageous terrain or is at the edge of a cell, signal quality might fluctuate strongly or it might be hard to keep up a connection at all.
In these cases it might be more sensible to delay the transmission and wait for better conditions. However there is of course no guarantee that conditions might actually improve in the near future.

Bibliography

[3GP09] 3GPP: *Physical channels and mapping of transport channel onto physical channels (FDD)*. http://www.etsi.org/deliver/etsi_ts/125200_125299/125211/07.10.00_60/ts_125211v071000p.pdf. Version: 2009. – [Online; accessed 2-July-2015] (Zitiert auf Seite 19)

[3GP10] 3GPP: *Enhanced uplink; Overall description*. http://www.etsi.org/deliver/etsi_ts/125300_125399/125319/07.08.00_60/ts_125319v070800p.pdf. Version: 2010. – [Online; accessed 2-July-2015] (Zitiert auf Seite 13)

[3GP11] 3GPP: *High Speed Downlink Packet Access (HSDPA); Overall description*. http://www.etsi.org/deliver/etsi_ts/125300_125399/125308/07.12.00_60/ts_125308v071200p.pdf. Version: 2011. – [Online; accessed 2-July-2015] (Zitiert auf Seite 13)

[BG96] Berrou, Claude ; Glavieux, Alain: Near optimum error correcting coding and decoding: Turbo-codes. In: *Communications, IEEE Transactions on* 44 (1996), Nr. 10, S. 1261–1271 (Zitiert auf den Seiten 14 und 21)

[BTKR15] Bach, Thomas ; Tariq, Muhammad A. ; Koldehofe, Boris ; Rothermel, Kurt: A Cost Efficient Scheduling Strategy to Guarantee Probabilistic Workflow Deadlines. In: *Proceedings of the International Conference on Networked Systems*. Cottbus, Germany : IEEE Computer Society, März 2015, 1–8 (Zitiert auf Seite 10)

[CCE03] Chahed, Tijani ; Canton, A-F ; Elayoubi, Salah-Eddine: End-to-end TCP performance in W-CDMA/UMTS. In: *Communications, 2003. ICC'03. IEEE International Conference on* Bd. 1 IEEE, 2003, S. 71–75 (Zitiert auf Seite 60)

[CCL] CCL: *NetLogo Homepage*. https://ccl.northwestern.edu/netlogo/. – [Online; accessed 14-July-2015] (Zitiert auf Seite 49)

[CDHT99] Cáceres, Ramón ; Duffield, Nick G. ; Horowitz, Joseph ; Towsley, Donald F.: Multicast-based inference of network-internal loss characteristics. In: *Information Theory, IEEE Transactions on* 45 (1999), Nr. 7, S. 2462–2480 (Zitiert auf Seite 60)

[CML09] CML: *The Coded Modulation Library*. http://www.iterativesolutions.com/Matlab.htm. Version: 2009. – [Online; accessed 2-July-2015] (Zitiert auf den Seiten 21 und 50)

[EAH02] El Allali, Hommad ; Heijenk, Geert: Traffic characterization for a UMTS radio access network. In: *Mobile and Wireless Communications Network, 2002. 4th International Workshop on* IEEE, 2002, S. 497–501 (Zitiert auf Seite 59)

[GJP+91] Gilhousen, Klein S. ; Jacobs, Irwin M. ; Padovani, Roberto ; Viterbi, Andrew J. ; Weaver, Lindsay ; Wheatley III, Charles E. u. a.: On the capacity of a cellular CDMA system. In: *Vehicular Technology, IEEE Transactions on* 40 (1991), Nr. 2, S. 303–312 (Zitiert auf Seite 13)

[GT03] Grossglauser, Matthias ; Tse, David N.: A time-scale decomposition approach to measurement-based admission control. In: *IEEE/ACM Transactions on Networking (TON)* 11 (2003), Nr. 4, S. 550–563 (Zitiert auf Seite 59)

[JPP00] Jalali, A ; Padovani, R ; Pankaj, R: Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In: *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st* Bd. 3 IEEE, 2000, S. 1854–1858 (Zitiert auf Seite 18)

[KSL00] Kim, Il-Min ; Shin, Byung-Cheol ; Lee, Dong-Jun: SIR-based call admission control by intercell interference prediction for DS-CDMA systems. In: *Communications Letters, IEEE* 4 (2000), Nr. 1, S. 29–31 (Zitiert auf den Seiten 13, 16 und 17)

[LEZ94] Liu, Zhao ; El Zarki, Magda: SIR-based call admission control for DS-CDMA cellular systems. In: *Selected Areas in Communications, IEEE Journal on* 12 (1994), Nr. 4, S. 638–644 (Zitiert auf den Seiten 16, 17 und 50)

[LJL+] Lin, Hsin-Piao ; Juang, Rong-Terng ; Lin, Ding-Bing ; Ko, Cheng-Yi ; Wang, Yi: Background Noise Floor Measurements and Cells Planning for WCDMA System. In: *Proc. of the GlobeCom 2002 Conference (paper# 811)* (Zitiert auf Seite 50)

[MT06] Masmoudi, Anis ; Tabbane, Sami: Other-cell-interference factor distribution model in downlink WCDMA systems. In: *Wireless Personal Communications* 36 (2006), Nr. 3, S. 245–275 (Zitiert auf Seite 16)

[PMBL+08] Persad-Maharaj, Narin ; Barbeau, Sean J. ; Labrador, Miguel A. ; Winters, Philip L. ; Pérez, Rafael ; Georggi, Nevine L.: Real-time travel path prediction using GPS-enabled mobile phones. In: *Proc. 15th World Congress on Intelligent Transportation Systems* Citeseer, 2008 (Zitiert auf Seite 41)

[QK98] Qiu, Jingyu ; Knightly, Edward W.: QoS control via robust envelope-based MBAC. In: *Quality of Service, 1998.(IWQoS 98) 1998 Sixth International Workshop on* IEEE, 1998, S. 62–64 (Zitiert auf Seite 59)

[RM92] Rappaport, Theodore S. ; Milstein, Laurence B.: Effects of radio propagation path loss on DS-CDMA cellular frequency reuse efficiency for the reverse channel. In: *Vehicular Technology, IEEE Transactions on* 41 (1992), Nr. 3, S. 231–242 (Zitiert auf den Seiten 13, 15 und 50)

[SBTR14] Schäfer, David R. ; Bach, Thomas ; Tariq, Muhammad A. ; Rothermel, Kurt: Increasing Availability of Workflows Executing in a Pervasive Environment. In: *Proceedings of the 2014 IEEE International Conference on Services Computing*, IEEE Computer Society, June 2014. – ISBN 978–1–4799–5066–9/14, S. 717–724 (Zitiert auf Seite 63)

[SSB+14] Schäfer, David R. ; Sáez, Santiago G. ; Bach, Thomas ; Andrikopoulos, Vasilios ; Tariq, Muhammad A.: Towards Ensuring High Availability in Collective Adaptive Systems. In: *Proceedings of the First International Workshop of Business Processes in Collective Adaptive Systems: BPCAS'14; Eindhoven, Netherlands, September 8, 2014*, Springer, September 2014 (Zitiert auf Seite 10)

[TLKF09] Tapia, P. ; Liu, J. ; Karimli, Y. ; Feuerstein, M.: *HSPA Performance and Evolution: A practical perspective*. Wiley, 2009 `https://books.google.de/books?id=PEJtezu6BMsC`. – ISBN 9780470742051 (Zitiert auf den Seiten 18 und 33)

[ZWLG13] Zaki, Yasir ; Weerawardane, Thushara ; Li, Xi ; Gorg, Carmelita: LTE Radio schedulers analytical modeling using continuous time Markov chains. In: *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP* IEEE, 2013, S. 1–10 (Zitiert auf Seite 59)

Declaration of Authorship

I hereby certify that the diploma thesis entitled " Simulation and Evaluation of Replicated Workflow Execution" is entirely my own work except where otherwise indicated. Passages and ideas from other sources have been clearly indicated.

(Stefan Schmidhäuser)