Technische Universität Wien
Vienna University of Technology

DISSERTATION

Measurement and Modelling of Internet Traffic over 2.5 and 3G Cellular Core Networks

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

DI Philipp Svoboda
Wanriglgasse 1/5
A-1160 Wien
Österreich

geboren am 25. März 1978 in Wien
Matrikelnummer: 9825199

Wien, im November 2008

Begutachter:

Univ. Prof. Dr. Markus Rupp
Institut für Nachrichtentechnik und Hochfrequenztechnik
Technische Universität Wien
Österreich

Univ. Prof. Dr. Andreas Kassler
Computer Science Department
Karlstadt University
Sweden

Abstract

THE task of modeling data traffic in networks is as old as the first commercial telephony systems. In the recent past in mobile telephone networks the focus has moved from voice to packet-switched services. The new cellular mobile networks of the third generation (UMTS) and the evolved second generation (GPRS) offer the subscriber the possibility of staying online everywhere and at any time. The design and dimensioning is well known for circuit switched voice systems, but not for mobile packet-switched systems. The terms user expectation, grade of service and so on need to be defined. To find these parameters it is important to have an accurate traffic model that delivers good traffic estimates.

In this thesis we carried out measurements in a live 3G core network of an Austrian operator, in order to find appropriate models that can serve as a solid basis for traffic simulations.

A requirement for this work was a measurement system, which is able to capture and decode network traffic on various interfaces of the mobile core network. Such a system was established within the METAWIN system. Our results are based on this setup.

The work can be split into three parts. First, we studied the service usage on a per user and session base separately for UMTS and GPRS. This analysis revealed that the service usage for UMTS and GPRS is gradually becoming more and more similar and that the main application in terms of transfered bytes is HTTP. Nevertheless, the main application in terms of handsets is WAP. As seen in many network measurements a small fraction of the subscribers generates the majority of the traffic.

In a second step we derived traffic models at flow level for UDP/TCP, HTTP, WAP 1.x and WAP 2.0. The TCP and HTTP flows follow a heavy-tailed behavior as expected from wireline measurements. However, UDP and WAP flows showed no heavy-tail effects. Many properties of the mobile flows follow the same distributions as in wireline networks.

Third, we designed several source level traffic models. These models allow a better understanding of the user interaction and network settings. In the case of HTTP and FTP we updated the parameters of existing models in order to meet our requirements. The mobile Internet access features high RTTs, so we modified existing e-mail models to reproduce traffic at a better granularity. In order to provide models for upcoming applications we designed three models for online games and a push to talk implementation.

Kurzfassung

DER Wunsch nach einem Modell für den durch die Benutzer erzeugten Verkehr reicht zurück bis zu den Anfängen der kommerziellen Telefonie. Die Einführung der dritten Generation der Mobilfunktechnik (UMTS) beziehungsweise die Weiterentwicklung der zweiten Generation haben den Schwerpunkt der angebotenen Dienste von Sprache zu paketvermittelten Diensten verschoben. Im Gegensatz zu Sprachdiensten, wo seit Jahrzehnten bewährte Modelle verwendet werden, gibt es für paketvermittelte Dienste keine Verkehrsmodelle, welche auf ein mobiles Szenario zugeschnitten wären.

In dieser Arbeit haben wir die Messungen aus einem im operativen Betrieb befindlichen Mobilfunktnetz der dritten Generation herangezogen, um den Verkehr zu analysieren und in Folge passende Modelle zu entwerfen.

Die Basis unserer Arbeit bildet das METAWIN System. Dieses erlaubt es, auf verschiedenen Schnittstellen des Mobilfunktnetzes Paketdaten aufzuzeichnen und dann je Benutzer und Sitzung auszuwerten.

Wir haben die Arbeit in drei Bereiche gegliedert. Der erste Bereich befasst sich mit der Analyse des Benutzerverhaltens bezogen auf die verwendeten Dienste in UMTS und GPRS. Es hat sich gezeigt, dass die Hauptanwendung, bezogen auf das erzeugte Volumen, in beiden Netzen HTTP ist. Setzt man hingegen die Anzahl der involvierten Endgeräte als Massstab für die Wichtigkeit eines Dienstes, so sind die verschiedenen Versionen von WAP am wichtigsten. Ähnlich wie es sich in früheren Messungen an Festnetzkunden gezeigt hat, erzeugt ein kleiner Anteil der Kunden den Großteil der anfallenden Verkehrslasten.

In einem weiteren Schritt haben wir Verkehrsmodelle auf Basis von Dienstflüssen erstellt. Wir haben dabei eine Auswahl der wichtigsten Dienste herausgegriffen: TCP/UDP, HTTP und WAP. Die Verteilungen für HTTP und TCP zeigen, ähnlich wie im Festnetz, eine sogenannte *heavy tail* Eigenschaft, während die anderen Dienste diese Eigenschaft nicht haben. Die Eigenschaften von UMTS und GPRS unterscheiden sich kaum, offenkundig reagieren diese Eigenschaften nicht auf die unterschiedlichen physikalischen Netzwerkparameter.

In dritten und letzten Schritt haben wir für einige der Topdienste separate Verkehrmodelle entworfen. Im Falle von HTTP und FTP haben wir die Parameter von existierenden Modellen an unsere Daten angepasst. Für den Maildienst haben wir gezeigt, dass man im Falle eines Internetzugangs mit hohen Latenzzeiten den Authentifizierungsvorgang detailiert nachbilden muss, um eine hohe Genauigkeit zu erhalten. Die drei Modelle für Dienste, aus dem Bereich der Onlinespiele dienen als Grundlage für zukünftige Dienste die mit steigender Leistungfähigkeit der Funkschnittstelle über kurz oder lang auch Einzug in der mobilen Welt finden werden. Ein Modell für *Push to Talk* Dienste rundet die Analyse ab.

Acknowledgements

"The shortest distance between two points is under construction"

N. Altito

I would like to thank some of the persons without whose support this work would not have been possible. First of all, Univ. Prof. Dr. Markus *Rupp*, whose support and encouragement helped me through the difficult times which such a thesis has. Second, Dr. Fabio *Ricciato* for the opportunity to work in the METAWIN research group at the *ftw.* in Vienna. Third, Univ. Prof. Dr. Andreas *Kassler*, whose support and review helped me to improve and polish my work. I am also very grateful for the many fruitful discussions I had with the METAWIN staff which helped me greatly in succeeding in my work. In this context I thank *mobilkom austria AG* for technical and financial support of this work[1] and the *Kapsch Carrier Com* as the second industrial partner in the METAWIN project. I would also like to express my thanks to my fellow colleagues of the *Mobilfunk Gruppe* at the Institute of Communications and Radio-Frequency Engineering in Vienna, Austria for an inspiring working environment and for many helpful comments and inputs. I would like to thank my father *Alfred*, my mother *Gabriele* and my brother *Markus* for supporting me in my work and providing me with a pleasant and stable background. Finally, I thank all my friends for their patience and the relaxing times we had.

Vienna, November 2008

Philipp Svoboda

Phone: (+43 1) 58801-38968
e-mail: psvbooda@nt.tuwien.ac.at

[1]The views expressed in this thesis are those of author and do not necessarily reflect the views within mobilkom austria AG.

Contents

Chapter 1

Introduction

Contents

1.1 Motivation

IN the beginning mobile networks were designed for one single service which was the voice telephony. The very first mobile radio telephone system was introduced in 1918 by the *Deutsche Reichsbahn*, which offered their first class passengers a radio based telephone link in the Berlin area [1, p. 872]. In Austria the first system test took place in 1930 [2]. However, the first large mobile network was established in 1958. It was the so called *A-Netz*[1]. The terminals were huge and their cost immense. Also the number of subscribers was limited due to the simple implementation.

The successor, the so called *B-Netz*, was introduced in 1972. An important new feature of the system allowed the subscribers to setup a call on their own. In the previous networks a central operator was involved in any call setup procedure.

The last analog technology was the *C-Netz*, which started in 1986 and was shut down in 2000. The terminals were still quite expensive, but with *only* 6.5 kg they were real lightweights compared to the previous generations.

Mobile telephony as we know it today started in the early 1990s. In 1990 the second generation of mobile communication technology, namely Global System for Mobile Communications (GSM), was introduced by the European Telecommunications Standards Institute (ETSI), supporting digital transmission of voice data. However, the data rates achieved for digital end to end communication were very small (9.6 kbit/s). The number of mobile terminals started to ramp up very fast and after 15 years they already exceeded the number of fixed telephone systems in Austria.

Meanwhile another technology began to emerge: the Internet. The number of Internet hosts also started to grow rapidly. However, in the beginning this technology was driven by wired Internet access technologies such as cable and Asymmetric Digital Subscriber Line (ADSL) modems. With networks voice traffic soon became less important as the number of subscribers of Internet services greatly increased. This evolution also had an impact on mobile communication networks. A new technology was standardized for packet-switched traffic only: General Packet Radio Service (GPRS) [3]. From there on the Internet and the mobile networks went into a merging process, which today, in 2007, results in mobile flat rate contracts being cheaper than rates for fixed-lines access.

At the end of the last century the standardization process of the third generation of mobile communication technologies, Universal Mobile Telecommunications System (UMTS), was finalized [4]. This new technology was a leap forward to the replacement of fixed Internet access technologies. While GPRS supports only data rates in the order of fixed analog modems (e.g. 10-60 kbit/s). UMTS, in Dedicated Channel (DCH) mode, can support up to 386 kbit/s and beyond. With the increase of data rate came a reduction of the Round Trip Time (RTT) from 1000 ms to 140 ms. These parameters are already close to the performance of an ADSL line. Just six years later UMTS was further improved by the introduction of High-Speed Downlink Packet Access (HSDPA). Currently (2007), it enables user download rates of up to 7.2 Mbit/s per host.

The last mile, providing connectivity from a communications provider to a customer, is in a wireless cellular network as a shared resource with strong limitations [5, 6]. In addition to this the spectrum which is used to transmit data over the air interface is very expensive. Many countries in the European

[1] Translation from `http://www.compuseite.de/handys/a_netz_e_netz.htm`

Union arranged auctions for the slots of the spectrum with the introduction of UMTS which achieved amazingly high revenues. Given this, every operator is keen to limit the data rate each user consumes. However, the expectations of the users migrating to mobile packet-switched networks is based upon the experience gathered in fixed line environments. Thus, even if the new technologies deliver more and more performance to the end terminal, it will be difficult to fulfill the needs of the market. In order to offer a good user experience, a mobile network will have to be well planned and optimized. For both, planning and optimization, it is important that the mobile operator understands the resources deployed in the field and the characteristics of the traffic generating the load in the network. With these two aspects he can calculate and define a Quality of Service (QoS) parameter. In most cases the actual values are well known. However, in order to plan the ongoing increase of resources in the network we need to be able to forecast the traffic load. It is essential in network planning and optimization to have a good estimation of traffic load.

In voice telephony the communication industry relied for years upon the Erlang models, with some adaptations (A,B and C). It assumes an exponential arrival process leading to Poissonian traffic models. The easy way these models can be handled made them favorable also in the beginning of packet-switched traffic. They were then extended to the class of models based on the Markovian arrival process, which are also easy to handle analytically. However, it soon turned out that in case of packet-switched traffic these models underestimated the buffer and link capacities [7, 8, 9, 10, 11]. Researchers discovered that packet-switched traffic has a more complex nature resulting in heavy-tails and self-similarity [12, 13]. The provisioning of links could not rely on numbers such as mean data rates. If implemented like this, there would be the risk of packet loss and higher delays caused by queuing.

The first step in network planning requires a detailed picture of the service mix generated by the customers. As there were no real world traces available, engineers began with traces from fixed wireline Internet customers and extrapolated them for their needs [14, 15]. However, a mobile terminal, offers different capabilities from that of a personal home computer. This results in a different services usage where services may be substituted with others, e.g., Wireless Application Protocol (WAP) [16] replacing Hyper Text Transfer Protocol (HTTP) [17].

It is therefore crucial to analyze the way mobile networks are loaded in the real world and to try to model the services for putting the highest load onto the system interfaces. The motivation for this thesis was caused by the necessity of being able to answer the questions which arose in making such an analysis.

1.2 Outline and Contributions

MOBILE networks differ not only in high level parameters like data rate, delay and jitter but such networks also use a different protocol stack, e.g., to hide terminal mobility. Therefore, we had to start with issues regarding the choice of the interface to monitor and so on.

The following itemization gives an outline of the main contributions to this thesis:

- A framework to extract UMTS and GPRS related information for a measurement system, as well as cumulative statistics.
- Analysis of the user traffic patterns separately for UMTS and GPRS.
- Models for TCP and UDP flows on a per service basis.
- Adaptation of parameters for existing models to fit UMTS and GPRS scenarios.
- Defining new models for emerging services in the area of online gaming.

Chapter 1: Motivation In this first chapter we present the motivation of this thesis. In addition we highlight the main issues that are addressed in this work. The drawing in Fig. 1.1 explains how the different topics of this thesis are split into the different chapters.

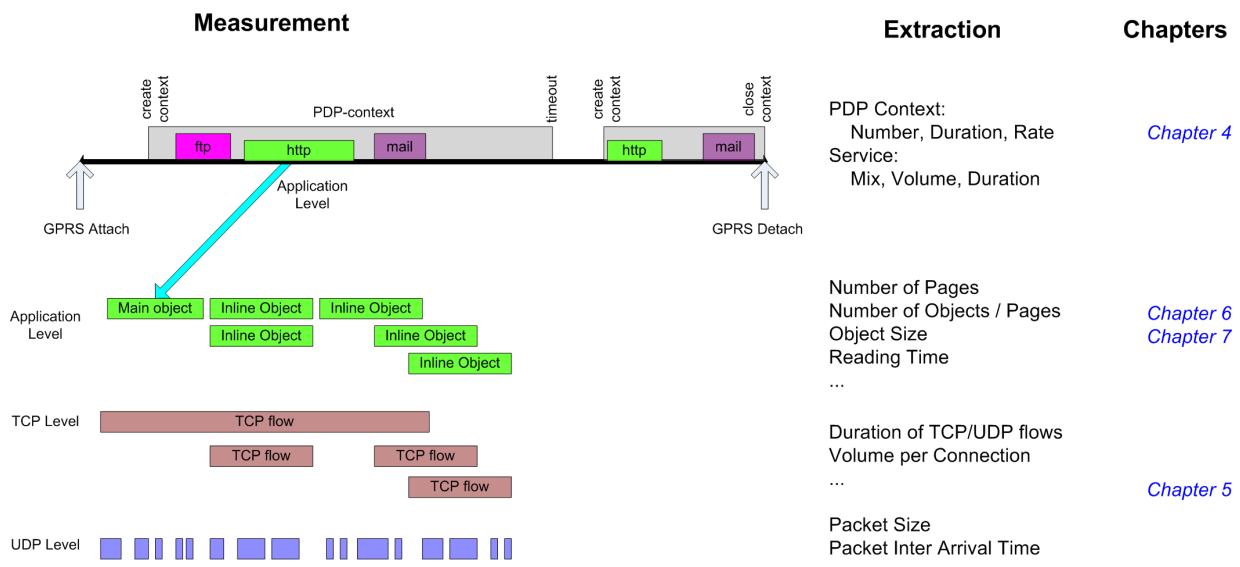


Figure 1.1: Overview of topics and the related chapters in this thesis.

In Fig. 1.1 we present the packet switched data traffic in a 3G cellular mobile core network, seen from different layers. The upper most layer presents a user data session as seen by the customer. The logical attachment between UE and SGSN is done via GPRS attach and detach-procedures indicated by vertical arrows before the beginning and after the end of the PDP-contexts. IP-connectivity then is established with the PDP-context create procedure. Within a PDP-context the customer can transmit and receive IP related data packets from and to the Internet, respectively. A PDP-context is either terminated by a timeout or by user interaction. From the 3G core networks point of view this is the

most basic data unit. We analyze and model this PDP-context in Chapter 4 of the thesis. In this chapter we also analyze the top services which we need in the following chapters.

The lower layers in Fig. 1.1 are UDP and TCP. These protocols are used in the Internet to transfer application data. Modeling these flows allows small and effective models, however missing the interaction from the application plane. We analyze the properties of the TCP and UDP flows of the top services in Chapter 5.

In order to allow application related user modeling, we analyze the network traffic from an application point of view. This is depicted in the middle layer and covered in Chapter 6, which improves existing models and Chapter 7, which introduces new models for upcoming services like online gaming and push to talk. Here we work with application related parameters, e.g., file size, page numbers, and so on.

In the following paragraphs we present a detailed overview for each chapter.

The most difficult task in this thesis was to meet the scientific interests, e.g., detailed absolute numbers, and the restrictions from the mobile operator, e.g., anonymization, and privacy laws, e.g., payload cutting, at the same time.

Chapter 2: Theoretical Background The second chapter gives a short theoretical background of the Internet paradigm, the Internet protocol suite and the basic principles in traffic engineering including heavy-tails, self similarity, and phase type renewal processes. The first section is about the Internet, explaining the main Internet protocols and the trade-offs met when measuring traffic. The next section covers traffic engineering. It gives an introduction to the topic and presents various common traffic models.

Chapter 3: Measurement Setup and Introduction to 3G Core Networks The third chapter illustrates the parts of the mobile core networks with which we work in this thesis. The structure of the UMTS network is explained first, covering protocol stack, logical channels, roaming and available bearers. Detailed descriptions on the topic can be found in the Appendices A and B.

In order to extract our data for further processing a tracing had to be set up. The second part of this chapter gives a brief overview of the measurement system to capture the traces for this thesis. The system was developed within the Measurement and Traffic Analysis in Wireless Networks (METAWIN) project in cooperation with Kapsch Carrier Com (KCC) and a leading mobile operator in Austria at the ftw. As a part of the thesis we developed and modified our own code modules, post processing scripts and verification tools **[18, 19, P. Svoboda et al.]**. Note that **green** colored citations are publications which are authored or co-authored by the author of this thesis. A more detailed description of all modules can be found in the Appendix C[2].

Finally, this section explains in detail all the interfaces we used to capture traces and defines the different traces[3] recorded for this thesis.

Our setup was hosted inside the live network. Due to the high number of different and unknown terminals in combination with the complex protocol stack, we often had to spend much more time

[2]Each metric has its limitations due to nature of a real world measurement setup.

[3]Traces were anonymized in order to guarantee privacy to the subscribers. The payload was cut away after the protocol headers and the IMSI was hashed.

verifying and rerunning the results than we actually did interpreting them. Within our work we discovered and fixed several errors in the network [20].

Chapter 4: User Related Composition of GPRS/UMTS Patterns This chapter reports the first evaluation of measured data in the thesis. We focus on the composition of the traffic from two perspectives: the user and the network. It turned out that the UMTS population, although still much smaller, dominated the share of transfered volume by a factor of two to one. Further investigation into this topic showed that there is a strong imbalance in the population between the two technologies. Data-card users are mainly found on UMTS, GPRS users tend to use handsets. The different composition of the user population is also visible in the service mix, we extracted separately for each technology, as there is nearly no WAP service — a typical service for mobile user terminals — found in UMTS. Analyzing the volume per user we realized that a small part of the population is generating the majority of the data volume **[21, P. Svoboda et al.]**. Previous to the analysis work we found that, today, even traffic traces in a mobile network have a bias due to malicious or unwanted traffic from port-scans, viruses and worms **[22, P. Svoboda et al.]**. Therefore, we introduced appropriate filtering rules based on our analysis of sample packet traces and properties of the related protocols.

From this chapter we learn basically three things: First, the main services in GPRS and UMTS are HTTP, WAP and e-mail with HTTP taking up to two-thirds of the volume, and secondly, the paradigm to use a "typical" user will fail and thirdly, we can group users to specific groups according to the services which they access in the network. The first statement leads to the fact that the network cannot be optimized by aggregate statistics which assume equal usage from every customer. Instead a group of heavy hitters has to be taken out to generate useful statistics. A so called *heavy hitter* is a member of a relatively small group of customers. This group consumes a large share of the overall traffic, e.g., 5% of the users consume 90% of the traffic.

Analyzing the recorded traces was a difficult task. The raw size of each recorded trace was more then one Terra byte. Therefore we had to code our own tools even for simple operations and analysis, e.g., generating and plotting eCDFs or fitting parameters. Standard tools, such as matlab, are not able to work with these amounts of input lines.

Another challenge was the scanning activity, which inflated the flow table metric we used in this chapter. Every probing packet from an infected PC creates a new flow entry to the metric. Therefore we had to find packet filtering rules which eliminated the probing traffic but did not change the non-malicious traffic characteristics.

Finally the eCDF curves of the PDP-context parameters showed several explicit steps we had to include in the fitting process of the parameters.

Chapter 5: Application Flow Patterns Entering the packet level domain we analyzed the properties of the UDP/TCP traffic flows. A first analysis showed that only approximately 10% of the flows are UDP, the vast majority are TCP flows **[21, P. Svoboda et al.]**. This result is consistent over the access technologies and time. Also in the following investigations the results for GPRS and UMTS show no significant differences to each other. The independence of the flow patterns from the access technology to which the subscriber was attached is one of the main findings in this chapter.

As expected, the traffic flows of the top services look quite different. The huge TCP streams for HTTP dominate the volume transfered **[19, P. Svoboda et al.]**. However, in terms of number of flows the small UDP streams generated by WAP 1.0 dominate the picture in the network. This result is very interesting as it points out that the operator has to optimize for both long and short flows.

The basic analysis of the TCP dataset revealed that most of the flows suffer from retransmission. This effect biases the flow size of TCP in mobile environments **[23, P. Svoboda et al.]**. The TCP flows and as a main contributer the HTTP flows showed signs of heavy tailed behavior. To fit these flows we split them into a body and a tail part. The tail part typically describes all events which in the CDF appear between 99% and 100%. The log-normal distribution provided the best fit for the body part **[19, P. Svoboda et al.]**.

The UDP and WAP flows instead showed no clear sign of heavy-tailed behavior. The log-normal distribution fitted best the UDP flows. The empirical distribution of the WAP flows had several discrete steps. Therefore, we fitted a phase type distribution to them.

In this chapter the most difficult task, beside the filtering and data handling already referred to in the previous chapter, was the identification of the heavy-tailed flows. To find this properties we used the scaling analysis introduced by Choi. However, due to the large size of our traces we had to improve the code of the available tool.

Chapter 6: Evolved and Extended Service Traffic Models Our main interest was to find appropriate models for mobile network services. At first we analyzed existing models for our needs and found that there are very adequate models for HTTP and FTP. The parameters for these models were quite old and did not fit to our traces [8, 24]. We therefore extracted the new parameters for these models for UMTS and GPRS **[19, 21, 25, 26, P. Svoboda et al.]**.

Further investigation for state of the art traffic models revealed that there was no e-mail traffic model which could cope with the high RTTs typically found in mobile networks. Therefore, a new model was developed which can better reproduce the TCP footprints found in the mobile core networks. We introduced TCP footprints in **[23, P. Svoboda et al.]**. To achieve this a login process was included in the model.

Although fitting the HTTP parameters seemed to be a simple task when we started, it turned out to be a very hard practical problem. The quasi standard tool for TCP payload analysis *tcptrace* was not able to handle traces of such a size in real time. We were not allowed to run any payload related analysis offline. So, we tuned the tool for more performance. However, we still had to restrict the input data. In order to extract the data for this analysis we coded a new metric called *tcptrace*. This metric converts the recorded traces to a pseudo PCAP format, that can be read by *tcptrace* and includes meta information about the radio access network.

For the extraction of the new e-mail model we extended the *Gn-duration* metric, so that this metric was able to detect login and mail receive procedures separately.

Chapter 7: Newly Developed Service Traffic Models Beside the traditional services expected in the network of a mobile operator, we also discovered the presence of new services, e.g., gaming traffic **[27, 28, P. Svoboda et al.]**. As such services are very demanding on delay and response time, a weak point of mobile Internet accesses, this was a surprise for us. However, there are only very few

traffic models in literature even for fixed Internet access technologies. Therefore, we analyzed three different types of games: FPS (First Person Shooter), RTS (Real Time Strategy) and MMOG (Massive Multiplayer Online Games). We selected them as they are very different from a traffic engineering point of view. For each type a traffic model was implemented and tested in a UMTS like scenario.

When we started our work, only very few customers played online games in our network. Therefore, we had to set up a special local configuration for each traffic model in order to measure the traffic patterns in detail. As the traces were recorded on ADSL it was then necessary to code new tools for analyzing the parameters.

The chapter closes with a traffic model for PTT (Push to Talk) based on an analytical speech model following the ITU recommendations. The model was implemented in MATLAB.

Chapter 8: Conclusions The last chapter of this thesis gives a summary of the work and presents the main conclusions found.

Chapter 2

Theoretical Background

Contents

IN this chapter we provide an overview of the basic principles in traffic engineering. Section 2.1 explains the Internet paradigm. There the Internet protocol suite, namely IP, UDP, and TCP is explained. In Section 2.2 we present the basics of teletraffic engineering. This section will introduce the important parameters and connect them. It is concluded with the explanation of heavy-tails, self similarity, and phase type renewal processes. Finally, Section 2.3 summarizes the chapter.

2.1 The Internet: A crash course

IT is crucial that we understand the service architecture of the applications which are investigated in this thesis. Therefore, this section provides an overview of the Internet applications in this work. Section 2.1.1 introduces the protocol stacks of TCP, UDP and IP. These protocols carry nearly all the user related data exchanged between applications in the Internet.

The Internet is a meshed network combining many different networks, each which has a large number of nodes. Each host can communicate with any other host in this network. The data exchange relies on a common protocol stack, namely TCP/IP, which allows flexible, fault tolerant, and connection oriented transmission of data. From its very beginning near the end of the 1960s the paradigm did not change much. Even today we are using TCP that is based on the first implementations from the 1970s.

In order to establish a worldwide data network three major steps were needed. The first step was to move the intelligence from the core devices to the boarder devices. In a phone network the end device, the telephone, is a very simple device capable only of transmitting and receiving audio data. In the Internet all functions for all services are realized in the end terminals, including reliable end to end communication. This approach simplifies the transport network and allows terminals to implement a growing number of applications. Second, the data transmission in the Internet is packet-switched. Each packet can run a different route from host-to-host, making connections simpler and more reliable. The so called Internet Protocol (IP) was designed for this task. And as a final step four layers were defined, namely application, transport, network, and sub-network layer. Precise Access Points Interfaces (API) were defined. With these functions an efficient protocol stack was implemented.

2.1.1 The Internet Protocol Suite

The Internet protocol suite consists of three major players. These are IP, TCP, and UDP. The IP covers addressing and routing issues regarding the packets, TCP supports a reliable and connection oriented data exchange, while UDP supports a connection-less data transfer on top of IP. The IP protocol can run on any subnet layer given a convergence layer. A convergence layer defines the interface to the higher layers. It contains adaption functions for both data transport and controls.

Figure 2.1 gives a mapping between the TCP/IP reference model and the services or protocols used in the Internet. The direct access to the subnet layer is not defined in this protocol suite. Moreover, the protocols are responsible for a point to point data exchange between two or more hosts. Any communication can flow only on the same layer. Thus, the e-mail client can only communicate with

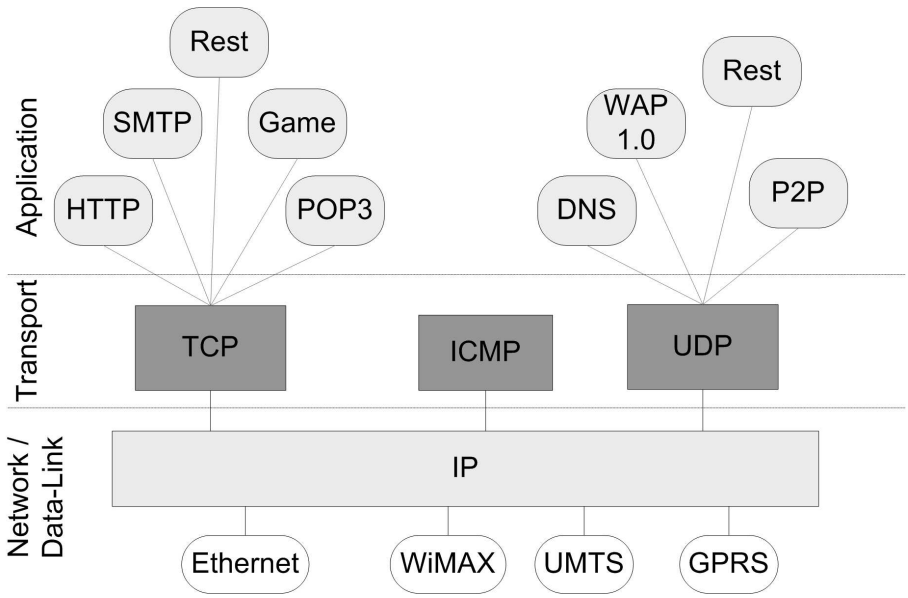


Figure 2.1: TCP/IP protocol suite.

an e-mail server in a horizontal way. To do so the application hands a Protocol Data Unit (PDU) to the next lower layer. In our case this would be the TCP agent running on the originating host. TCP will process the PDU to make it fit to the restrictions of IP, e.g., divide it into slices according to the maximum transmission unit and then hand it over to the IP layer. The IP agent will then route the packets from the first host to the destination address.

2.1.1.1 The Internet Protocol (IP)

The Internet Protocol (IP) is a network layer protocol which runs over data link layer protocols, e.g., ATM or Ethernet. It is defined in the **RFC791** [29]. Thus, the main task of this protocol is to provide a global unique addressing among the hosts of the Internet enabling routing of IP packets from the source to the destination address.

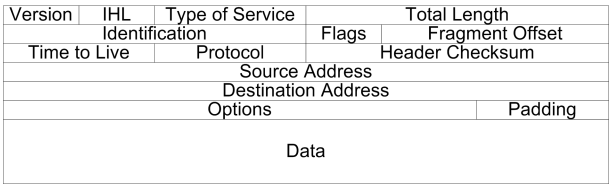


Figure 2.2: IP header (RFC 791).

Figure 2.2 depicts the header of an IP packet. The size of the header is 20 bytes in total, it consists of 13 fields with 12 of them mandatory, the *Options* field is optional. The version field indicates if this packet is coded as version four (`v4`) or version six (`v6`). The Internet Header Length (IHL) indicates the size of the header in 32 bit words. This is needed as the options can fill up a variable amount of bytes in the packet. Type of Service (ToS) is used to mark the packet according to Quality of Service (QoS) measures. It features eight bits that indicate delay, reliability and throughput demand of the

application. The *total length* can have a maximum value of up to 65,535, which is also the maximum number of bytes for an IP packet. The last two fields which we consider in this short overview are the *source* and the *destination address*. Both fields are 32 bits wide, therefore roughly two to the power of 32 hosts can be addressed by IP`v4`. Although this is quite a large number we are presently running out of IP addresses. IP`v6` is standardized to solve this problem. The addressing scheme of IP, note that we talk only about IP`v4` here, is not straightforward as there are many special addresses for special nodes, e.g., the local loopback adapter has the address 127.0.0.1, and some IP ranges are reserved for home networks. These addresses are not unique throughout the network. For more information please see [29, 30].

2.1.1.2 The Transport Control Protocol (TCP)

The Transport Control Protocol (TCP) supports reliable, in order connection oriented, end to end communication between two hosts on top of IP. TCP is defined in the RFC793 [31]. IP data transmission will not take care of packet loss, thus TCP implements features which provide an accurate delivery of data. In a TCP transmission the receiver of data sends an acknowledgment packet to the sender. The sender stores sent packets and waits with new data until the corresponding acknowledgment is received. This technique works for corrupted packets reported by the receiver. However, dropped packets will not generate any messages at all. Thus, the sender starts a timer for each packet sent. If this timer expires before the sender receives an acknowledgment he considers the packet as lost and will retransmit it. Since TCP is optimized to avoid loss of data, it sometimes introduces long delays in case of retransmissions or out of order packets. This short introduction to TCP can only be a very rough introduction to the functionality of TCP. More information can be found in the related RFCs [31, 29].

TCP has been developed for wired networks. Packet loss is thought to originate from congestion scenarios and the sender windows are reduced accordingly. The window estimation is based on the inter packet interval, smaller intervals allow for larger window sizes, packet loss increases this interval. In a wireless environment packet loss is often related to corrupted packets due to bad receiver signal strength. However, TCP will assume congestion and reduce the data rate at the sender side, leaving radio resources unused. Much research was made on this topic, and proposals exist to correct these issues, e.g., fast retransmissions. Still this problem remains unsolved and harms the performance of TCP on top of wireless access technologies.

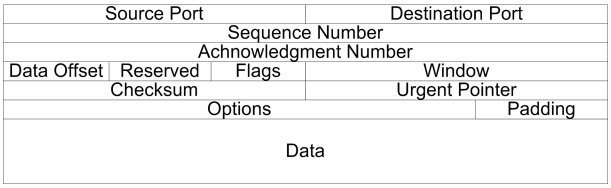


Figure 2.3: TCP header (RFC 793).

Figure 2.3 depicts a TCP packet. A TCP packet is part of a data stream between two hosts. Each application on a host may use a different port to communicate with another host, e.g., an e-mail server will listen to port 110 for requests. These port numbers are defined in the first two fields of the header,

each of them is 16 bits in size, thus allowing 65,535 different port numbers per host. Ports 1 through 1023 are so called "well known" ports, they include services such as web, e-mail, domain name service, and so on. Ports 1024 through 49,151 are registered ports as they are used by less popular applications on a fixed scheme. Finally, ports 49,152 through 65,535 are used as temporary ports used mainly by clients to communicate with servers. The Sequence Number in the header is used to define the exact position of the attached data within the transmitted stream. Note that packets transmitted over IP may not be delivered in order. In this case the sequence number allows the receiving host to reorder the packets. The Acknowledgment Number can be used to report the reception of a correct packet. It is valid only in case the ACK flag in the Flags field is set. The Flags field covers control settings for the TCP connection. There are six different flags: urgent (URG), acknowledgment (ACK), push (PSH), reset (RST), syn (SYN) and finish (FIN). They may be transmitted in various combinations. More details on the TCP mechanism can be found in [31]. The Window field indicates the number of bytes the sender of this packet is ready to receive. The Checksum is used to verify the correct data transmission, however, it is far too weak to correct errors. Again, the Option field can hold various information and is variable in size up to 32 bits. If present, the data block contains a slice of user data from the stream between the two hosts.

2.1.1.3 The User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) is the second main protocol of the Internet protocol suite. It is defined in the RFC 768 [32] and supports connection-less communication without any error recovery, flow control and/or congestion control function. Thus, data packets may arrive out of order, duplicated or even missing. The application has to provide functions similar to those provided by TCP in case it is necessary. UDP reduces the overhead in the control functions, allowing for a faster and more efficient data transmission, especially for applications that do not need the functions of TCP, e.g., video streaming as in this case retransmitted packets often arrive too late and are therefore useless.

Figure 2.4 depicts a UDP packet. Without all of the control functions featured by TCP it only consists of four fields and the data block. The first two fields are the source and the destination port similar to TCP. The third field indicates the length of the packet and the fourth field is a 16 bit checksum for the header and the data part of the packet.

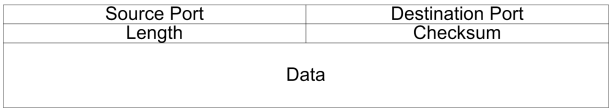


Figure 2.4: UDP header (RFC 768).

2.1.2 The Application Layer: Internet Services

The five layer TCP/IP model was developed earlier than the OSI Reference Model by the Internet Engineering Task Force (IETF). The OSI model has a seven layer structure and is often used in literature and theory. However, the IETF never made the TCP/IP model compliant with the OSI model. The OSI model therefore differs from the Internet protocol stack.

The application layer in the Internet protocol suite covers the upper three OSI layers: session, presentation, and application. Software in the Internet has to inter-operate these three layer functions. In general, applications in the Internet follow a client and server approach. At the user's host the client allows presentation of data, e.g., web-browser or e-mail client. It visualizes the data received from the server running at a central host. The user can interact with the client side and the client will interact with the server, which then processes the requested data and sends it to the client, which then displays the results of the human input, e.g., a mouse click to load a new web page.

All communication between server and client is based upon so called sockets. A socket is defined by the pair IP address and port for one host. A connection between two hosts can be defined by a quadruple of IP addresses and ports of source and destination. Note that this definition is simple in case of TCP as this protocol is connection oriented. However, in the case of UDP the definition is tricky as a monitoring node cannot detect sessions. We will later point out the definition of a UDP connection from case to case.

There are some well known services for servers, e.g., a service listening on port 80 is expected to be a web-server. The RFC 1700 provides an actual list of these ports [30]. Servers that offer the same service should listen on the same port number. The port may be used to transfer data and signaling information, e.g., web-servers, or for signaling purposes only, e.g., Session Initiation Protocol (SIP). In the second case the application choses another port for further data transfer.

In the first case the application can be monitored with a port based filter. In the second case however, the negotiated port must be tracked by the monitoring system. This is a much harder task especially if payload has to be processed online. We will see later that more than 80% of the traffic volume can be classified based on port rules and that the remaining services are rather small, e.g., below 0.4% in terms of volume. Note that in the measurement period there were no routers performing network address translation at the user side. We only monitored services using a port based filter.

If security matters these ports may be altered by the system administrators as this assignment is not mandatory. The well known services range up to 1023, ports above are not uniquely assigned to a single service. Applications such as peer to peer (P2P), online games, remote desktop sharing and so on use these ranges of ports to run servers.

2.2 Traffic Engineering

IN this section we discuss topics related to traffic engineering, also called teletraffic theory. Teletraffic engineering is an application of traffic engineering to telecommunications. A possible definition of teletraffic theory can be formulated as follows:

"to make measurable in well defined units through mathematical models and to derive the relationship between grade-of-service and system capacity in such a way that the theory becomes a tool by which investments can be planned"[1]

The main task of teletraffic theory is to find a balance between QoS, network capacity and traffic demand. In other words we are looking for a cost effective solution (network capacity) to fulfill the

[1]Source: *Handbook of Teletraffic Engineering, ITU-D, Draft 2001*

need of the customer (Grade of Service - GoS). We therefore have to forecast the demand in traffic, e.g., with measurements and forecasts.

In Paragraph 2.2.1 the connection between QoS, network capacity and traffic demand is worked out. The follow up Paragraph 2.2.2 introduces various aspects of traffic models used for Internet traffic. There properties such as self similarity and heavy-tail distributions are discussed.

2.2.1 Introduction to Traffic Engineering

The teletraffic theory was first introduced for circuit switched Public Switching Telephone Networks (PSTN). It was the Erlang formula, developed by Alan Erlang in 1975 for voice calls in PSTN, which played a very important role in the evolution of telephone networks.

While in circuit switched networks a line can only have two states: busy and available, in packet-switched (PS) networks the load on a link is much more complex. In the beginning of the Internet, therefore, teletraffic engineering played only a minor role.

The parameters of teletraffic are: QoS, network capacity, and traffic demand. These three components are closely related, given two parameters the third can be calculated. It is the task of the teletraffic theory to find the exact mathematical relationship between these three parameters. This may include some statistics and cannot be always solved in a closed form.

The definition of these three characteristics can be manifold, examples for QoS metrics are: packet-delay, -throughput, -error-rate, etc. Traffic demand can be any kind of traffic running over a network element. This traffic may be user initiated through applications or also system initiated via signaling information transmitted between nodes. Traffic models are designed to reproduce such metrics. We report more on this topic in the following Section 2.2.2.

Performance analysis is another task that can be solved using teletraffic theory. This analysis searches for the maximum achievable performance of a system, therefore, QoS is an output parameter, while demand and capacity are input parameters. Such analysis is crucial in the setup phase of a network, and later on in the optimization process. The optimization process shows, given an increasing demand, how much capacity an operator has to add to meet his target QoS.

There are three different methods to solve problems of teletraffic engineering:

- analytically
- by simulation
- measurement based.

The historical approaches in teletraffic engineering were derived analytically, e.g., Alan Erlang's formula is an analytical solution. The analytical solution is a closed form equation, or at least a numerical approximation, describing the relationship between QoS, demand, and capacity. This has several advantages for the understanding of the investigated network element. The solution can be calculated directly, is tractable, and provides a deep insight into how the three parameters are linked together. However, in case of more complex systems it is often necessary to use a high abstraction level. Analytic solutions exist for many queuing models [33, 34].

The simulation approach is based on raw computational power. The traffic demand is generated by a stochastic model. The simulation software will now simulate the properties of all network elements

between sender and receiver of the traffic, including the protocols. Ideally, the simulation provides the same results as an evaluation based on measurement. However, due to computational restrictions simulation often use simplifications, e.g., omitting the protocol stack. The output of a simulation is only a point of the function connecting QoS, demand, and capacity. Therefore, simulations have to be repeated for various input parameters, e.g., radio conditions or user load.

The big advantage of simulations is the fact that they are simpler to implement than analytical methods and more flexible than measurements. They are often used in performance evaluations and optimizations. For mobile networks there is a huge number of available simulation software: ns-2 [2], OPNET [3], GPRSSim, and QoSSim. The simulation tool ns-2 is the only freely available software. Therefore, our investigations were performed with this software.

Traffic simulation consists of two steps: first setting up the network, second finding a model for the input traffic. The first step is often quite time consuming, however, it is fixed by the given environment. Choosing the right traffic model is a much harder task. A wrong choice may render the results useless, e.g., heavy-tailed input leads to peaks in queuing delays.

Both, analytical and simulation based, approaches often have to make assumptions to simplify the problem. Measurement based evaluations are important to verify the results of these two approaches.

2.2.2 Traffic Models

Traffic models are designed to generate an input load for evaluation, either analytical or simulation based. In the case of analytical investigations it must be possible to describe the model in a closed mathematical way. Simulation based experiments are not limited to such restrictions, in fact some simulations use recorded traces as input vectors.

A traffic model can be attached to different layers of the protocol stack, e.g., packet level, flow level, or application level including high layers.

At packet level the parameters for the traffic model define the arrival process for each packet and the size distribution of all packets. This implementation is well suited to describe aggregated traffic, e.g., recorded on link trunks as found in 3G core networks due to their tree-like deployment. The models use stochastic processes to describe the arrival. At the packet layer there is no differentiating between the packets, suche as user-data and application signaling. Therefore, this approach is strongly limited in the event that QoS on the application layer is a target output for the simulation. Properties of the underlaying protocols may not be reproduced correctly, e.g., TCP retransmissions in case of a bottleneck. Such properties can be captured on the flow level.

A flow level model reproduces flows at the UDP and/or TCP layer of the network. A flow consists of at least one packet and is defined by the quadruple: host IP-address and corresponding port and client IP-address and corresponding port (see Section 2.1.1.1 for the definition of IP-address and port). The model describes the arrival process, the volume, and the duration of the flows (see [35, 36] for examples). In the case of TCP most of the models are based on the assumption that the average data rate is a good approximation for the distribution within the flow.

Both models cannot reproduce user interactions at the client side. This input can only be im-

[2] `http://www.isi.edu/nsnam/`
[3] `http://www.opnet.com/`

plemented in so called source traffic models [37]. These models reproduce the usage of applications and their objects, e.g., a source model for HTTP browser application describes the arrival process of user sessions in terms of requests to a web server and the properties of the webpages. A webpage is a resource for information in the World Wide Web (WWW). It has many important properties, e.g., size of images, size of other objects, etc., all linked to the requested page via the number of objects per page. The advantage of such an approach is the fact that the parameters are related to application properties, e.g., the question how much traffic is generated in case the user population doubles can be answered easily. In addition the source model is independent of the underlying transport network, e.g., the same e-mail model for 2G and 3G networks can be used. This is not the case for the packet and flow level models. However, in contrast to the packet modeling, where one model is able to describe all the traffic on a link, here each service has to be modeled separately. It is therefore not possible to simulate an Internet backbone based only upon source models.

A model for generating traffic often is not enough to capture special phenomena that need a precise reproduction of traffic patterns. This requires a detailed modeling of the protocol stacks below the traffic models. This effect is well known for TCP connections. Often such simulations are performed in an open loop, which means that the TCP feedback which protects against overload is not active. The setup is simpler as the TCP stack is not included in the simulation, however, the results may diverge from real world behavior (see [38, 39]).

Concluding traffic generation, the last approach to generated traffic is the playback of measurement based traces. It may be questionable to call such an approach a model. These traces are used as an input vector for network simulations. Although it looks as if it is a very precise method to reproduce traffic, as it was recorded in a real network, the playback of a trace lacks all of the interaction taking place on the network level. For example a bottleneck in the simulation will not impact the trace, or the packet arrival process, from the trace file. In a real world setup TCP congestion control would reduce the data rate of each flow as congestion arises (see [38]). In general, trace driven simulations work in a system with a small to medium load and they do not work in systems with high to overload situations.

In conclusion the traffic models which we saw showed that every method presented has advantages and shortcomings. A good traffic model is: accurate, complete, tractable, simple, and robust.

Although the need for accuracy is clear, there is often the question of how to prove accuracy for a given model and more important how to set a threshold for accuracy. These two questions can only be answered case by case.

The term *complete* differs from *accurate*. It means that the model includes all effects important to the evaluation. For example using a mean file size, derived from a trace file, in a model for an FTP application is not sufficient, although it may be accurate enough.

It is important to have a tractable model as it allows to make an in-depth analytical investigation. However, only a small group of models fulfill this point, e.g., Markovian processes.

A robust model can be used with a large variation of input parameters without becoming unstable. This is important in case of optimizations for different load situations with the same model.

Finally, a model should be simple in terms of parameters. For example an HTTP model with 30 parameters per user may be very accurate, however, in case you have to simulate `100,000` users it will

take forever to run even small simulations. Therefore, it is important that the number of parameters is kept small, in a trade-off with the accuracy accomplished.

Stochastic Process Stochastic processes are used in traffic modeling and principles of heavy-tails and self similarity will be used later in this thesis. Therefore, we introduce the basic terms. A random process $\mathbf{X}$ has a CDF defined for all x by $F(x) = P(\mathbf{X} \leq x)$. In measurements the CDF is replaced by the empirical CDF (eCDF). Given the data vector $\mathbf{x}$ with n measurement points the eCDF is

$$F_{emp}(x) = \frac{\texttt{number of elements in the sample} \leq x}{n} = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x). \tag{2.1}$$

with $I(z)$ being the indicator for an event z.

In case of heavy-tails often the complementary CDF is plotted:

$$F_{cmp}(x) = 1 - F(x). \tag{2.2}$$

Often, these plots are in double log scale.

A stochastic process is, in the simplest case, a collection of random variables $\mathbf{X}$ indexed by a set t (time). The stochastic process is discrete in time if t is countable. It is time continuous in case t is a time interval. In contrast to a deterministic process, where only one evolution is possible for the future, the future of a stochastic process is described by probability distributions. This means that for a fixed starting point different evolutions of the process are possible.

2.2.2.1 Self Similarity and Heavy-Tails

In PSTN traffic engineers worked for decades with the Poisson process, which has some favorable properties, e.g., it is memory-less. However, measurements [13, 40] showed that the Internet traffic is very bursty, or in other words not smooth. In fact a time series, generated from packets per time bin, of an observed network stays bursty if the traffic is aggregated over several links.

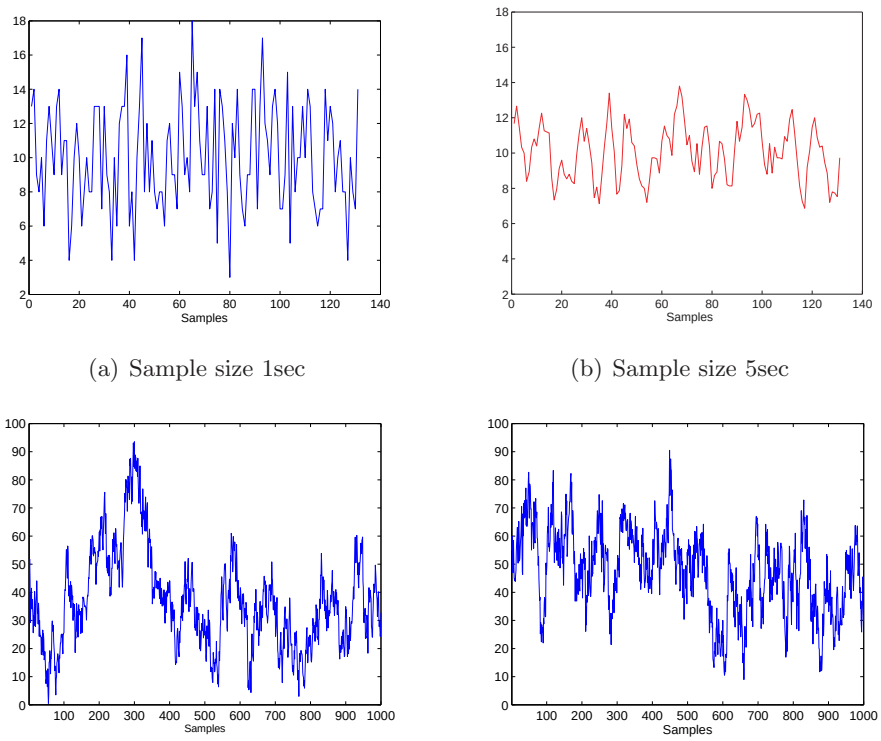
Note that in case of a Poisson process, which is a memory-less stochastic process being used to model the arrival of independent random events, characterized by a rate parameter λ. Aggregating two links, e.g., with $\mathbf{X}(\lambda_1)$ and $\mathbf{X}(\lambda_2)$, leads to the new process which is again Poisson distributed with $\lambda = \lambda_1 + \lambda_2$. This new process has a smaller variance, it smoothens out for a higher aggregation.

Burstiness is not a mathematical term, in case of the Internet traffic it has been shown by [13] that the concept of self similarity describes the burstiness well. This research showed that Local Area Network (LAN) and Wide Area Network (WAN) traffic is self similar.

A self similar object is similar to a part of itself, e.g., the whole object has a similar shape as at least one part of it. This is a typical property of fractals. In our case self similarity arises due to the fact that the statistical characteristics of a stochastic process are the same for different aggregation levels. A graph of a time series will look "similar" in different zoom levels, typically the variance of a aggregated process is decreasing, e.g., for a Poissonian process it changes with the aggregation level. As the time series is similar in different aggregations, obviously the variance, in both snapshots depicted, is the same despite the growth of the aggregation level. The process has no finite variance value. Another effect linked to self similarity is the Long Range Dependency (LRD). A LRD process

has large auto covariance values even for long gaps, in fact the decay is only hyperbolic, while the decay of a short range dependent process is exponential. The Hurst parameter, also called *index of dependence*, is an indicator for the degree of self similarity of a time series, expressing the relative tendency of the time series. It was named by B. Mandelbrot after H. E. Hurst, a British hydrologist who studied the optimal dam capacity of rivers over a long time-period [41]. A pure random process with independent events has a Hurst value of 0.5, while a value of 1.0 is generated by an exactly self similar process. A more detailed discussion of this parameter is provided in the paragraph on self similarity.

Figure 2.5 presents the effect of aggregation for a Poisson process and a self similar process. In case of the Poisson process the curve flattens out quickly, this is not true for the self similar process. The process looks "the same" for both aggregation levels.



(a) Sample size 1sec

(b) Sample size 5sec

(c) Sample size 1sec

(d) Sample size 5sec

Figure 2.5: Time series aggregation for Poissonian (top) and self similar traffic (bottom).

Heavy-Tailed Heavy-tailed distributions are closely linked to self similar processes. The "heavy-tail" refers to for a property of some statistical distributions, e.g., Power law, Pareto, etc. It describes distributions that consist of two parts: a highly populated region and a low populated one that converges asymptotically. Although the events in the tail have a very low probability they have a very large value and therefore impact the statistical moments of the distribution. In other words the tail converges slower than $1/x^2$, sometimes even slower than $1/x$. A distribution is called *heavy-tailed*, if:

$$P[\mathbf{Y} > x] \sim x^{-\alpha}, \text{ as } x \to \infty \text{ , } 0 < \alpha < 2. \tag{2.3}$$

From this definition we learn that a heavy-tailed distribution has an infinite variance and may even have an infinite expectation value. We consider two cases: first $0 < \alpha \leq 1$ and second $1 < \alpha < 2$. In the second case the variance does not exist and in the first case also the mean value is infinite.

To explain this fact we directly use the definitions of these two values. If the probability distribution of $\mathbf{Y}$ allows for a probability density function $f(x)$, then the expected value is computed as:

$$E(\mathbf{Y}) = \int_0^a xf(x)dx + \int_a^\infty xf(x)dx = \mu;\ \alpha > 0.$$

Let us consider only the tail part of this function from a to ∞. If the integral for this part is infinity, the mean is also infinite. From Eq. (2.3) we can derive $f(x) \sim x^{-1-\alpha}$ the argument within the integral is $x \cdot f(x) \sim x^{-\alpha}$. Note the lower limit for the integral is zero as we are working with positive values only. In the first case the argument in the integral stays larger than x^{-1}. The result for the tail part of the integral will therefore be infinite. Since the variance depends on the mean, it cannot exist in case the mean is already infinite.

In the second case the mean does exist. Its variance is given by:

$$Var(\mathbf{Y}) = E[(\mathbf{Y} - \mu)^2].$$

Inserted into the definition of the expectation value we obtain:

$$Var(\mathbf{Y}) = \int_0^a f(x) \cdot (x - \mu)^2 dx + \int_a^\infty f(x) \cdot (x - \mu)^2 dx,$$

Again we only focus on the tail part which is for $a > 0$ the right integral. We are in the same situation as for the mean in the first case[4].

The *Pareto distributions* (see Appendix D-5), which are defined as:

$$P[\mathbf{Y} > x] = \begin{cases} k \cdot \frac{x_m^k}{x^{k+1}} & x \geq x_m \\ 0 & else. \end{cases} \tag{2.4}$$

for all $k > 0$, are a member of this class of distributions (Note: k is the same parameter as α. In the following paragraph we will use k.). In a log-log plot the Pareto distribution is a straight line with a constant slope equal to $-k$. In fact any heavy-tailed distribution following, the definition in Eq.(2.3), plotted in a log-log plot will have a tail with a linear slope, as $\log(x^{-k}) = -k \cdot \log(x)$. The value α can be estimated from such a log-log plot, or the Hill estimator [42, 43].

Sometimes the Weibull and the log-normal distribution are also accounted as heavy-tailed [43]. Both distributions decay much faster than the Pareto does, however, they do not decay as fast as light tailed distributions. We shall use these two distributions later to describe certain flow level parameters.

A distribution as it is defined in Eq.(2.3) has no finite variance. Therefore, such a distribution has also no Laplace transformation in an analytical closed from. Note also Weibull and log-normal do not have a Laplace transform. Such distributions cannot be used in classical solutions of queuing problems. As the heavy-tail may cause problems due to the burstiness produced, it cannot be neglected

[4]Note a measured sample will always have a mean and a variance, the tail parameter can only be estimated on a part of the distribution. This fact makes it very difficult to extract heavy-tailed behavior from measurement samples

in a simulation setup. Phase type distributions are used to approximate these distributions [44]. In simulations real heavy-tail behavior can only be reproduced with a large number of cycles [45].

Self-Similar Let $\mathbf{x}(t)$ be a Wide-Sense Stationary (WSS) stochastic process. By definition the process has a constant mean $E[\mathbf{x}(t)]$ and an autocorrelation function $r(k) = E[\mathbf{x}(t)\mathbf{x}(t+k)]$.

Let $\mathbf{X}^{(m)}$ be the aggregation of $\mathbf{x}(t)$, obtained by averaging the original time series for non-overlapping blocks of size m. The Eq. (2.5) shows this aggregation process.

$$\mathbf{X}^{(m)} = \mathbf{X}_k^{(m)} = \frac{1}{m}(\mathbf{x}((k-1)\cdot m+1) + \ldots + \mathbf{x}(k\cdot m)),\ k > 0 \tag{2.5}$$

The new autocorrelation function is denoted by $r^{(m)}(k)$. If $\mathbf{X}$ is self similar, it has the same autocorrelation function as $\mathbf{X}^{(m)}$ has for all m. In particular $\mathbf{X} \approx m^{1-\mathtt{H}}\mathbf{X}^{(m)}$, where $\mathtt{H}$ is the self similarity parameter with $0 < \mathtt{H} < 1$ and $m > 0$. The Hurst parameter $\mathtt{H}$ indicates the level of self similarity, a value of 0.5 indicates a random process, a value of 1.0 indicates a self similar process. A process with $0.5 > \mathtt{H} \geq 1.0$ is called LRD. Both processes $\mathbf{X}$ and $\mathbf{X}^{(m)}$ are distributionally self similar, this means that every moment of $\mathbf{X}$ is equal to the same moment of $\mathbf{X}^{(m)}$.

In network measurements often only the first two moments are used. A zero mean covariance-stationary process $\mathbf{X}$, is second order self similar if $\mathbf{X}$ and $m^{1-H}\mathbf{X}^{(m)}$ have identical second order statistics for all $m > 0$.

LRD processes differ from Markovian birth processes. The Markovian process generates Short Range Dependencies (SRD), based on the limited number of states within the process. Therefore, the autocorrelation of a Markovian process decays faster than it does for a LRD process. For large values of m the Auto-Correlation Function (ACF) of an SRD process is similar to the ACF of white noise.

LRD behavior of traffic in a network has a big impact to network performance. In case queue lengths in nodes are heavy-tailed, the packet delay also has a heavy-tailed distribution. Therefore, buffers have to be larger than they would have been using classical traffic models.

The main conclusion of LRD traffic is the fact that congestion is unavoidable and will show up in the form of short impulses [46, 47].

Typical heavy-tailed parameters are: file size, flow size, and flow duration. The source for self similarity may also come from protocol interactions [48, 49].

2.2.2.2 Phase Type Renewal Process

The PHase type (PH) renewal process is a renewal process describing the time between two events, the resulting distributions are called phase type distributions [50]. The process can be described by an underlying Markov chain with $(n-1)$ transient states and one absorbing state, these chains are called finite and absorbing Markov chains. The number of phases in the resulting PH distribution is n, which is equal to the number of transient states in the Markov chain. The PH distribution represents a random process $\mathbf{X}(k)$ that corresponds to the time that the underlying Markov chain remains in the transient state until it reaches the absorption. The Markov chain is restarted for every k, resulting in independent, identically distributed $\mathbf{X}(k)$.

Let $[\mathbf{Y}(t), t \geq 0]$ be a finite Markov chain with a state space of n, for which $\{1, \ldots (n-1)\}$ are transient and n is the (only) absorbing state for this process. The generator matrix of the chain is

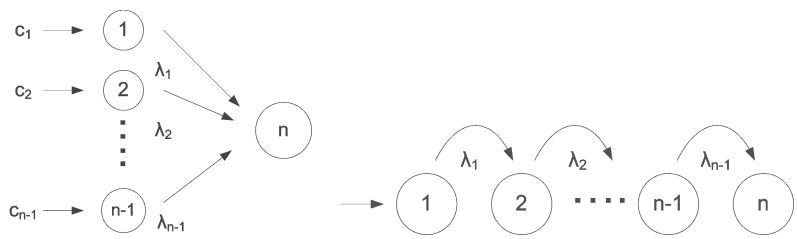
called $\mathbf{Q}$ and the probability to start in state i is q_i, the n-th element of q represents the absorbing state and is zero by definition. Now we define another matrix $\mathbf{R}$ of dimension $(n-1) \times (n-1)$, with $R_{ii} < 0$ for $1 \leq i < (n-1)$ and $R_{(n-1)i} \geq 0$ for $i \neq (n-1)$. The matrix $\mathbf{R}$ is called the phase type generator. Let $\mathbf{e}_1$ be a column vector of ones. Then the exit rate vector $\mathbf{r}$ is defined by $\mathbf{r} = -\mathbf{R} \cdot \mathbf{e}_1$. It describes the conditional intensity of absorption (r_i) starting from state one. The generator matrix $\mathbf{Q}$ is now:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ 0 & 0 \end{bmatrix}. \tag{2.6}$$

If we now define a variable $\mathbf{z}$ representing the time until the absorbing state is reached: $\mathbf{z} = \inf\{t \geq 0 : \mathbf{Y}(t) = n\}$, then $\mathbf{z}$ is PH distributed with $(n-1)$ different phases. The PH distribution is defined by $n(n-1)$ parameters, however, in practice this number is much smaller as the majority of the values in $\mathbf{R}$ is equal to zero.

The PH distribution is very flexible in approximating other distributions. Even a power tail distribution can be approximated, however, only with an infinite number of phases. The advantage of the PH distribution is the fact that the solutions are a sum of exponential distributions. Therefore, such results are tractable and a Laplace transformation does exist. This is often used in queuing theory to replace a general distribution by an approximated PH distribution.

The Hyper-exponential and Hypo-exponential distributions are special cases of the PH distribution. A Hyper-exponential distribution can be interpreted as a result of $(n-1)$ parallel exponential distributions. A Hypo-exponential variable consists of a sum of exponential distributed variables. The following Fig. 2.6 shows the states of the Markov chain for Hypo- and Hyper-exponential distributions.



(a) Hyper-exponential PH Distribution (b) Hypo-exponential PH Distribution

Figure 2.6: Special cases of PH distributions: The hyper- and hypo-exponential distribution.

In the Hyper-exponential case only the values on the main diagonal of the matrix $\mathbf{R}$ differ from zero:

$$\mathbf{R} = \begin{bmatrix} -\lambda_1 & 0 & \dots & 0 \\ 0 & -\lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\lambda_{n-1} \end{bmatrix}, \tag{2.7}$$

and the initial state vector is $\mathbf{q}^T = (c_1, c_2, \dots, c_{(n-1)}, 0)$.

In the Hypo-exponential case the matrix $\mathbf{R}$ has the following shape:

$$\mathbf{R} = \begin{bmatrix} -\lambda_1 & \lambda_1 & \dots & 0 \\ 0 & -\lambda_2 & \lambda_2 & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\lambda_{n-1} \end{bmatrix}, \tag{2.8}$$

and the initial state vector is $\mathbf{q}^T = (1, 0, \dots, 0)$ [50].

2.3 Summary

THIS chapter provides the background for what follows in this thesis. We explain the Internet paradigm to connect heterogeneous networks with different nodes based on the Internet protocol suite. The IP protocol in combination with UDP/TCP allows a flexible and in case of TCP reliable connection between two hosts in the Internet. An IP address identifies a single host in the Internet while the port, found in the protocol header of TCP and UDP, can be used to address a specific service running on the host system. It is therefore possible for several different services to communicate in parallel from the same host or IP address.

The chapter also presents an overview of self similarity, heavy-tails, and phase type renewable processes, as these definitions are needed later in order to analyze our traces.

Chapter 3

Measurement Setup and Introduction to 3G Core Networks

Contents

THE traffic analyzed in this thesis originates from a 2G and 3G cellular core network. This chapter presents an overview of packet-switched traffic in wireless cellular core networks with a focus on GPRS and UMTS networks. Section 3.1 presents the evolution of mobile core networks, starting with GSM up to UMTS. The introduction to UMTS is found in Section 3.2. As UMTS has evolved from GPRS the core networks of these two technologies is very similar. Note, we only consider the core network components in this overview, the major changes found in the radio network components are outside the scope of this thesis and not included in the introduction. A data transmission in a 3G network does need some session management. To give a brief insight in such a session setup, a sample data session setup is presented in Section 3.3. The next Section 3.4 introduces the measurement system used within this thesis. Section 3.5 summarizes the chapter.

The appendix provides further information on the protocol stack, Appendix A for UMTS, and Appendix B for GPRS.

3.1 The Evolution of Mobile Core Networks

IN the early stages of mobile communications the goal was to offer availability everywhere using only one phone number. In this evolution phase there was just one kind of service: voice. The basic features of the mobile devices did increase from generation to generation, the coding techniques changed from analog to digital. But even the full digital GSM network, by default, had no other service available than voice. As data-traffic became more and more important a strange situation arose: although the GSM transport network was using digital signals, the end terminals could only process audio/voice data as input data. Therefore, users had to use a modem to transfer data traffic via GSM. This method of data transport is quite inefficient. To stop such shortcomings the GSM group finalized a new Release introducing GPRS (Rel. 99). To keep the changes small only minor parts were adopted. Figure 3.1 shows a top level view of a 3G core network including a GPRS and a UMTS Radio Access Network (RAN). From this figure we learn the fact that GPRS was *attached* to the existing GSM by adding the Packet Control Unit (PCU), while in UMTS the packet-switched data is processed in the same device as the voice and video calls. GPRS is the first mobile technology that was purely targeted to data-traffic. The technology was introduced to the Austrian market in fall 1999 by mobilkom austria Co AG. From there on a wide range of new services like WAP or e-mail could be directly accessed via a mobile handset. In 2002 the introduction of UMTS even further increased the number of available services to the end terminal. The higher data rate, the possibility to use advanced QoS settings and to choose between Circuit Switched (CS) and Packet Switched (PS) bearers enable new advanced services such as live video streaming, video telephony and so on.

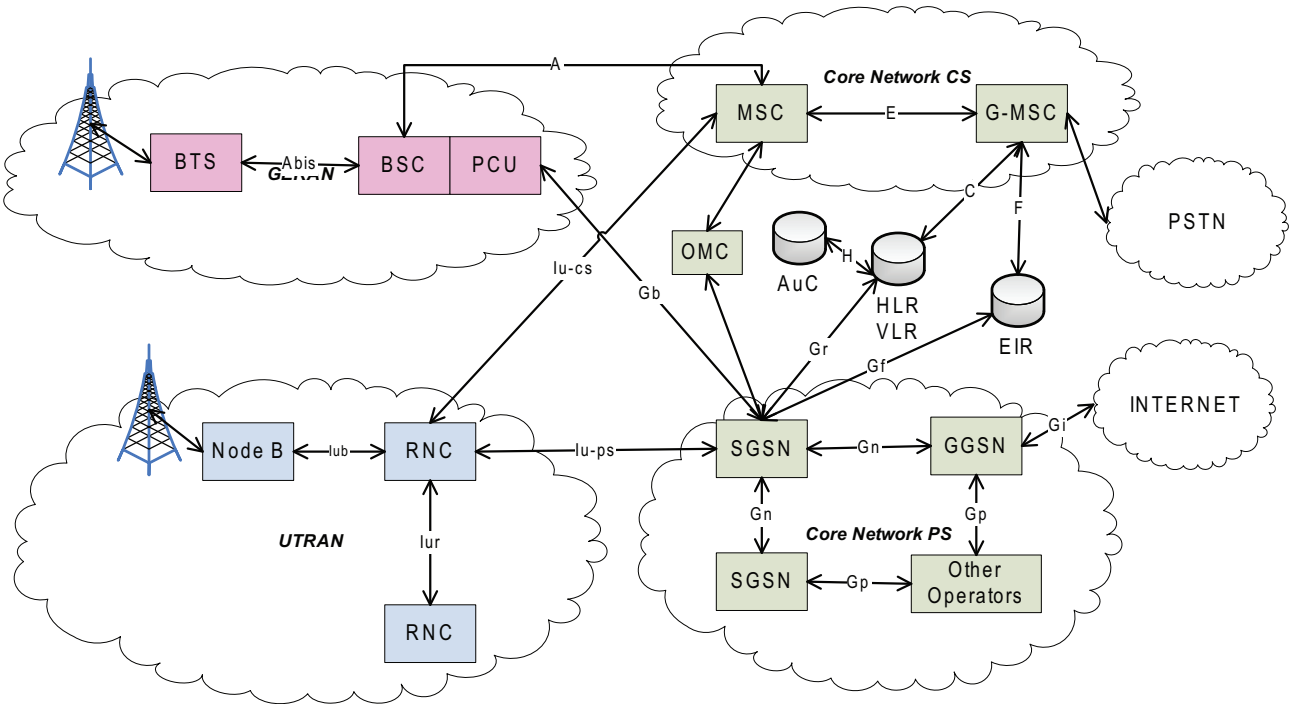


Figure 3.1: Network elements for 2.5G and 3G mobile networks.

3.2 Introduction to the 3G Packet Switched Core Network

THE initial design goal of GSM was to support voice services which are on the same level with Integrated Services Digital Network (ISDN) combined with mobility. In the late 1990s the user focus started to shift from pure voice to voice- and data-traffic. The GPRS standard was agreed upon as a basis to build for data only services. Although this was a first step to mobile packet-switched networks, it was only a placeholder for a new technology which could serve PS and CS services by default. UMTS was designed to serve both needs of the CS and the PS domain. To minimize cost of the core-network the structure and functions of the components are very similar to the GSM/GPRS units. In fact with UMTS Release 5, the UMTS units can also serve GPRS and GSM RANs. The UMTS RAN called UTRAN (UMTS Terrestrial RAN) uses Wideband Code Division Multiple Access (WCDMA) instead of Time Division Multiple Access (TDMA). WCDMA in UMTS supports two different operation modes, namely Frequency Division Duplex (FDD) and Time Division Duplex (TDD). WCDMA enables the User Equipments (UE) to be attached to several base stations. In GPRS such a feature would need two independent amplifiers. The main elements of the UMTS PS-core network are: Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN). The main elements of the UMTS CS-core network are: Mobile Switching Center (MSC) and Gateway MSC (G-MSC). Both core networks use the following system elements: Home Location Register (HLR), Visitor Location Register (VLR), and Authentication Center (AuC).

3.2.1 System Architecture

The basic system architecture of UMTS splits the components into three parts [51]. The UE represents all parts handled by the users, including the mobile and the Subscriber Identity Module (SIM) called USIM (UMTS SIM). The UTRAN contains all parts necessary for the radio connection. It includes the NodeB and the RNC. Finally, the Core Network (CN) contains SGSN, GGSN, MSC, GMSC, and so on. A first difference to GPRS is the fact that the different RNCs are inter connected (see Iub in Figure 3.1). Therefore, datagrams sent to a certain NodeB can be forwarded directly by the RNC to support user mobility (cell change). In GPRS such datagrams would have been dropped by the PCU in the Base Station Controller (BSC) and the SGSN would then have to retransmit them again, resulting in high RTT after handovers. Figure 3.2 presents a simplified view of the UMTS parts.

UE	User Equipment
UTRAN	UMTS Terrestrial Radio Access Network
CN	Core Network

Figure 3.2: Main parts of a UMTS mobile network.

Figure 3.1 shows a detailed setup of a UMTS network. It is obvious that UMTS has its roots in the GPRS network. The functions of the UE are similar to the MS. It too consists of two parts: the Mobile Equipment (ME) and the UMTS subscriber identity module (USIM). The ME provides all the hard and software that is needed to access the standard UMTS services. The USIM holds the user specific information. It is an extended SIM card to support more dynamic memory and new and better security options implemented in UMTS.

The NodeB is the UMTS version of the GPRS element Base Transceiver Station (BTS). Its main tasks are channel coding, interleaving, rate adaption, adding scrambling codes and modulation. The BTS also supported some higher functionality (setup of BCCH, . . .) which is now shifted towards the RNC components. In the new UMTS mode called High Speed Packet Access (HSPA) a part of these functions is again shifted from the RNC towards the NodeB, e.g., in HSDPA the NodeB has to fulfill a part of the scheduling and power control on its own.

Radio Network Controller (RNC) The RNC covers all radio resource management tasks. The NodeB itself has a quite simple function set, therefore, the RNC has to manage the scrambling code tree and the transmit power for each radio link active. The Radio Resource Control (RRC) protocol is established between the UE and the RNC to support the manipulation of the radio link between the NodeB and the UE. Towards the core network the RNC terminates the GPRS Tunnel Protocol (GTP, see Appendix A-1.3) tunnels. This is different from the GPRS network where the SGSN was the endpoint of the GTP tunnel. The NodeB has no caching for data packets; therefore, the RNC also has to process the flow control algorithms.

Serving GPRS Support Node (SGSN) The SGSN is the switching center for the data traffic. In downlink direction the SGSN is connected to several RNCs using Iu-Ps protocols. A certain SGSN

serves a group of RNCs and therefore covers a given geographical area. The number of necessary SGSNs is simply given by the processing power that is needed to serve the given traffic in the area. The common tasks of a SGSN are: session management including attach, detach and mobility management, ciphering, cell updates, paging, compression, and so on. The billing in mobile cellular networks is volume based, therefore the SGSN is generating billing tickets per each user and sends these tickets to a central database. The protocols used by the SGSN are the Sub Network Dependent Convergence Protocol (SNDCP), the Logical Link Control (LLC), the Base Station Subsystem GPRS Protocol (BSSGP), and GTP.

Gateway GPRS Supporting Node (GGSN) The GGSN is the boarder node between the core network of the mobile operator and the external packet data network. GPRS supports different Packet Data Protocols (PDP) like IP, Point-to-Point Protocol (PPP) and X.25. The GGSN must be able to handle all these PDPs. The type of PDP can be chosen by the mobile subscriber by creating a PDP-context. The PDP-context creation request marks the start of a data-session. It holds the information about the Access Point Name (APN) and the settings the user requests from the mobile network. The GGSN is connected to the external network via the Gi. There is a firewall in common between the GGSN and the external network, protecting the mobile core infrastructure from attacks. If the user accesses an IP network the GGSN will convert the user datagram from the mobile network to IP packets and replace the GTP identifier, which is the ID for a specific user within the mobile network, with an external IP-address. The GGSN also takes care of the QoS profiles for each PDP-context. One user can have several PDP-contexts, each with a different QoS profile. This can be used to access different services with different QoS settings (for more details see [52]).

The Home Location Register (HLR) The HLR is the heart of the GSM network [53]. It is a database holding management data for each user of the mobile operator. The HLR holds all permanent user data such as Mobile Subscriber ISDN Number (MSISDN), available services, QoS, international ID, the IMSI and further temporal data like the location area the ME was last seen, the actual VLR or the Mobile Subscriber Roaming Number (MSRN). The HLR can be accessed by the MSC via the C interface and by the VLR via the D interface. The HLR itself is closely connected to the AuC. The AuC takes care of the generation of security related data the HLR needs to authenticate users. The HLR has to hold at least one entry per subscriber and to fulfill real-time requests from the MSC units. To solve this issue a HLR unit normally consists of several discrete units managing the huge load of data and requests (consider that today mobilkom austria Co AG on its own has more than 4 million subscribers). This can also be seen in the IMSI. The IMSI is structured as shown in Figure 3.3.

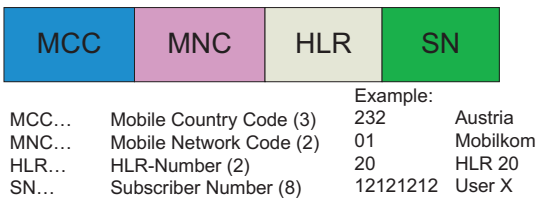


Figure 3.3: Structure of the IMSI.

The first three digits are fixed by the country of the operator. The next two digits identify the operator itself. The following HLR part identifies the HLR the user data is stored in. Finally, the last eight digits are the unique identifier at this HLR for the searched user. At the terminal side this information is stored in the SIM card.

The Visitors Location Register (VLR) The VLR is a database holding MS specific data allocated to one (or several) MSC unit. One could think of the visitors of one MSC unit, either in their home operators network or in roaming mode. As the user population will change over time this database, in contrast to the HLR, is highly dynamic. The first request of a MSC will target the VLR, this instance takes the load from the central HLR unit and can be seen as a kind of cache instance. The VLR makes more sense when thinking about an instance directly implemented in the MSC enabling local caching of user information over a period of time. More details can be found in [54, 55].

The Operation and Maintenance Center (OMC) The OMC uses the O-interface to monitor and control all the network components. The protocols are SS7 respectively X.25. The typical tasks are status reports, generation of billing tickets, user billing, security screening.

The Equipment Identity Register (EIR) The EIR holds user equipment information in the form of International Mobile Equipment Identities (IMEI) [56]. The idea of the EIR was to black list stolen or malfunctioning devices to ensure that they are not able to enter the network of a mobile subscriber. Although the intention itself was good, it suffers from the fact that the mobile providers do not update this list regularly and that the IMEI is easily re-programmable on most devices.

The Authentication Center (AuC) The AuC holds the authentication key `Ki`, which is also stored at the SIM-card. By using this shared secret a new key, called `Kc`, can be derived to secure the radio link of the mobile network. Although treated separately from the HLR in this introduction, the AuC normally is a part of the HLR because the relation between these units is very close.

3.2.2 Bearer Speed in UMTS

The bearer speed in a wireless mobile network is a term for the net data rate that is available to the UE. User data in UMTS may be transfered using two different implementations: Dedicated CHannel (DCH) or High Speed Packet Access (HSPA) [57, 58].

In case a very low amount of user data has to be transmitted, a random or common channel can also serve for data transmission. However, normal Internet applications will initiate data transfers triggering a DCH or HSPA channel assignment.

The DCH channel has different bearer speeds depending on the chosen spreading factor. For a fixed transmit power, a larger spreading factor allows more reliable transmission at the cost of a lower user data rate. Therefore, users with a higher distance to the base station will only achieve a lower data rate. In addition to this, as a part of the network optimization process, the RNC monitors the actual data rate the user needs and adjusts the data rate (spreading factor) accordingly. Table 3.1 shows the available options for the DCH from our live network.

User Data Rate	Interface Data Rate	Spreading Factor
12.2 kbit/s	30 kbit/s	128
32 kbit/s	60 kbit/s	64
64 kbit/s	120 kbit/s	32
128 kbit/s	240 kbit/s	16
384 kbit/s	480 kbit/s	8

Table 3.1: DCH data rates for different spreading codes.

HSPA extends the radio interface of the UMTS network. A data symbol on the radio interface can transmit up to four bits of data, while standard UMTS symbols transmit only two bits of data. The data rate assignment in HSDPA differs from DCH. The physical channel is set to a fixed spreading factor of 16, which equals a data rate of 14.4Mbit/s. This is a strong improvement over the 384kbit/s in the DCH. However, 14.4Mbit/s is the total rate of the entire HSDPA cell. All users have to share this resource. HSDPA uses a slot length of 2ms, within each slot 15 different code channels are transmitted. A scheduler in the NodeB assigns code channels to the specific users according to the UE capabilities and the need of data rate. A UE capable of class five can decode five code channels within one time slot, which equals a user data rate of 3.6Mbit/s.

3.2.3 Bearer Speed in GPRS and EDGE

The physical link in GPRS is a Time Division Multiple Access (TDMA) implementation. It offers eight slots in uplink and in downlink direction, respectively. A normal GSM voice call uses one time slot, GPRS UEs can allocate up to eight time slots. Each added slot, upgrades the data rate available to the user. The number of free slots is a function of the cell load and has an upper limit which is bound by the capabilities of the UE.

In addition to this, three new Code Sets (CS) were introduced with the start of GPRS. The code sets offer different strengths of data protection. Higher data protection secures the transmission and the signal is more resistant to noise and interference. However, a stronger code needs more parity bits thereby reducing the user data rate. The assignment of the code is limited due to the Signal to Noise Ratio (SNR) at the UE, the fastest code set, CS-4, can only be activated close to the base station transmitter. The maximum possible data rate for GPRS is 160 kbit/s. The following Table 3.2 shows the data rates for one time slot and different code sets in GPRS and Enhanced GPRS (EDGE).

Code	GPRS		EDGE	
Set	User	Interface	User	Interface
CS-1	8.0 kbit/s	9.0 kbit/s	22.4 kbit/s	27.0 kbit/s
CS-2	12.0 kbit/s	13.4 kbit/s	29.6 kbit/s	40.2 kbit/s
CS-3	14.4 kbit/s	15.6 kbit/s	44.8 kbit/s	46.8 kbit/s
CS-4	20,0 kbit/s	21.4 kbit/s	59.2 kbit/s	64.6 kbit/s

Table 3.2: GPRS and EDGE data rates for one time-slot and different CS.

The EDGE service uses three bits per symbol at the air interface, in contrast to the one bit per symbol of GPRS. Therefore, EDGE pushes the data rate by a factor of three to a maximum of approximately 473.6 kbit/s.

3.3 A Data Session in a 3G Network

IN a mobile cellular network several steps are necessary in order to set up a data connection for IP transmissions. The following Fig. 3.4 presents the most important steps to establish a connection.

The PDP-context must not be activated prior to a GPRS Mobility Management (GMM) activation. The mobility management transfers GPRS related subscription data of the subscriber. This subscription data is needed to clarify which PDP-contexts may be established by the user interface. After the GPRS attachment the subscriber can activate a PDP-context at any time. The session management for PDP-context activation takes place only between the UE and the SGSN. The SGSN communicates the PDP-context activation data to the GGSN via a GTP tunnel.

After a successful PDP-context activation procedure the subscriber can now transmit user data on the IP layer to an external packet data network. In the case of roaming the setup procedure has to initiate an intra SGSN handover first. The QoS profiles may be modified to the needs of the new SGSN [59].

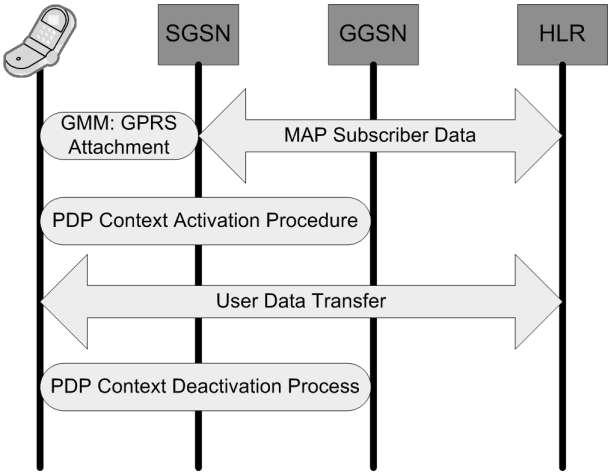


Figure 3.4: Session management procedures.

The PDP-context is similar to a dial-up session which is known from the fixed wired networks. For each context the subscriber is assigned a unique IP address. In the following chapters we sometimes use the IP address information at the Gn interface to represent a user session. A context represents a user session with volume, duration, and frequency of use.

3.4 The Measurement Setup

IN this section we provide a high-level description of the monitoring system developed jointly at ftw and used for this thesis. The METAWIN system ([19, 18, 21]) was designed to capture packet-switched data in UMTS and GPRS networks on different interfaces. Please note that this description of the measurement system reports on the first prototype. In the meanwhile the system has evolved and grown. The actual system today, 2008, is sold by Kapsch Carrier Com under the brand DataXtender.

At present the system is able to capture and correlate data from Gn, Gi, Gb, IuPS, and Gs. In this thesis we used data recorded at the Gn and the Gi interface. In the first Paragraph 3.4.1 we give a rough outline of how the system works. Paragraph 3.4.2 explains the metrics with their output and restrictions.

3.4.1 METAWIN

In this section we describe the METAWIN system. Work on the METAWIN system started in 2004 at the *Forschungszentrum Telekommunikation Wien (ftw.)* in Vienna. It was a cooperative research project together with mobilkom austria Co AG, Kapsch Carrier Com (KCC) and ftw in Vienna, Austria. The project was partly funded by the KPlus initiative of the Austrian government. The goal of the project was to build a capturing system for packet-switched data on various interfaces of the UMTS and the GPRS core networks and link this data at the user level.

The main objectives of this measurement setup, from a scientific point of view, were:

- Passive Monitoring,
- Flow/application level tracing for traffic analysis,
- Anonymization to meet privacy issues,
- Protocol decoding to allow correlation of the recorded data,
- Allow large storage to build up statistical datasets.

The next section gives a more detailed view of the measurement system. This description was part of a technical report [19].

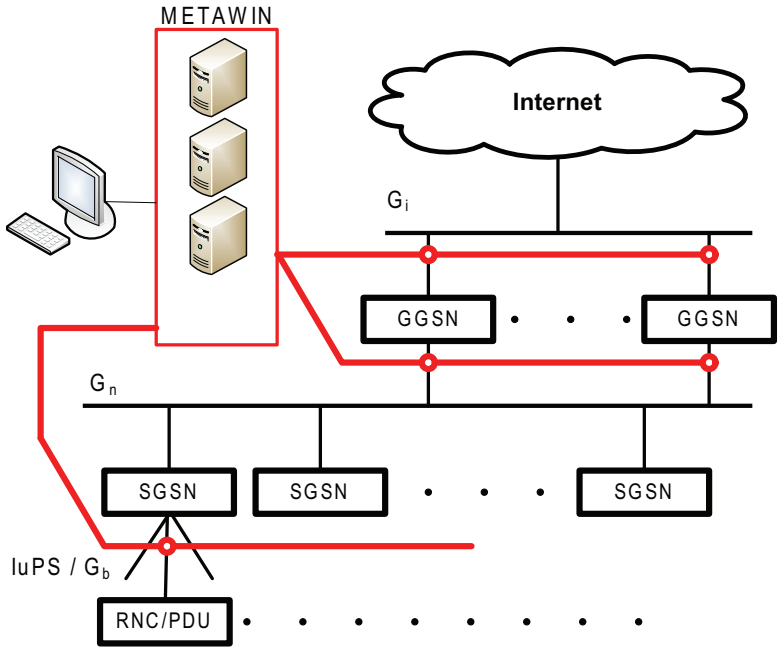


Figure 3.5: Current deployment of the METAWIN monitoring system.

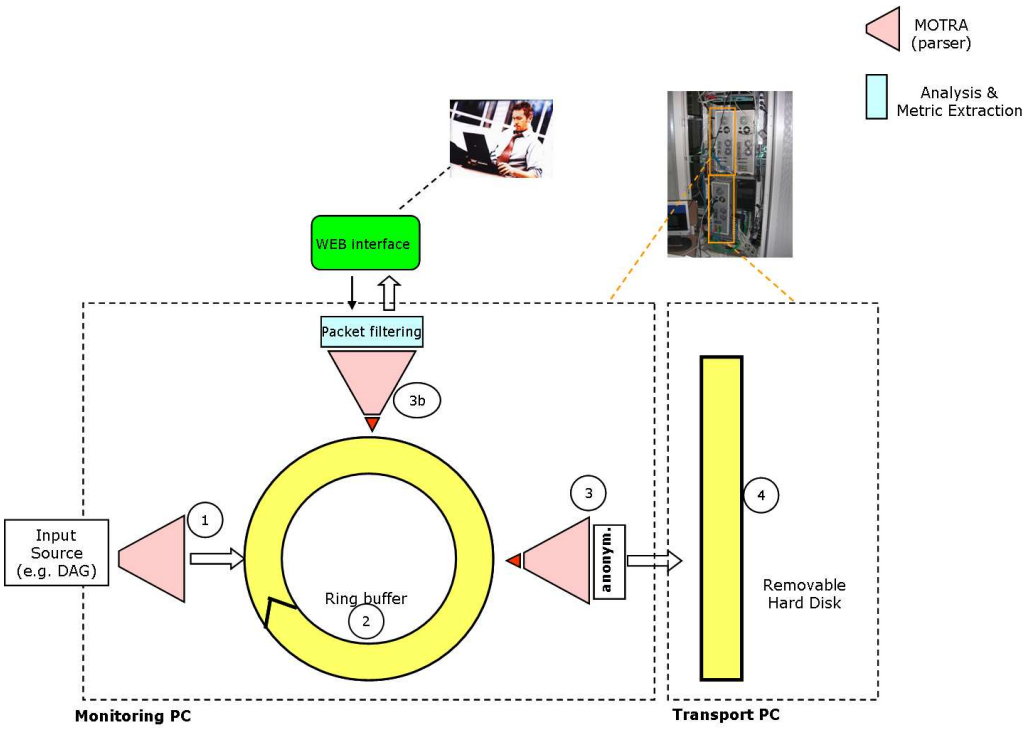


Figure 3.6: Structure of the METAWIN monitoring system.

3.4.1.1 Structure of the Monitoring System

The internal structure of the monitoring system is depicted in Fig. 3.6. Regarding the hardware, it consists of a number of high-end PC/Linux equipped with special acquisition cards (DAG), and with multiple slots for removable hard disks.

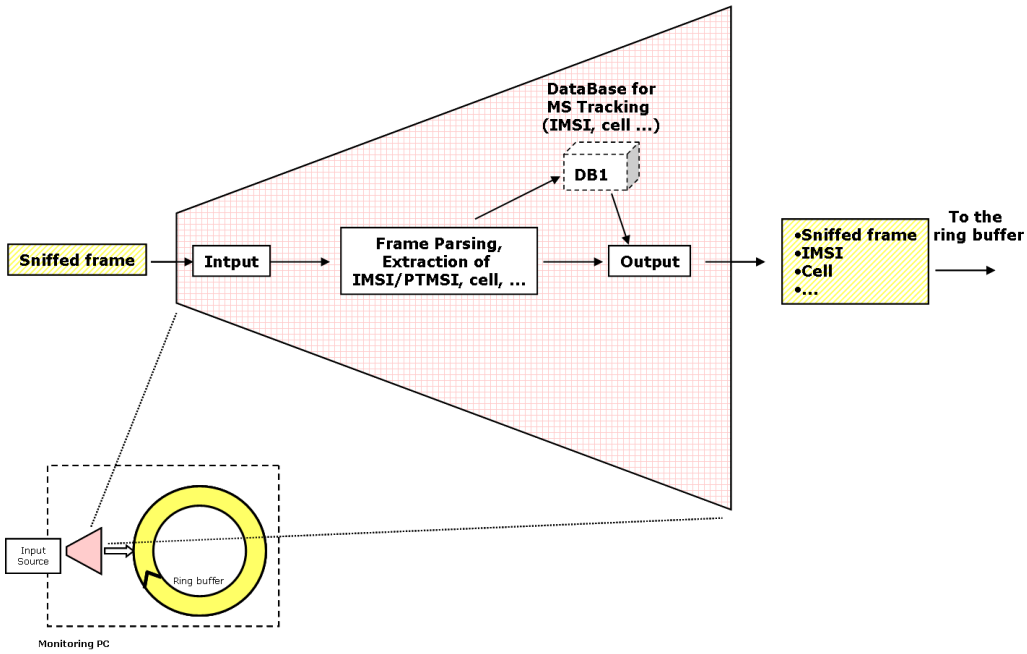


Figure 3.7: Details of the frame parser (MOTRA).

All the software components were completely developed within the project on a Linux platform. The key module is a frame parser which is able to read and interpret the full protocol stack within the frame. This module is called "MOTRA" and is depicted as a triangle in the following figures. Several instances of this module are active in the system.

A first instance of MOTRA (Label '1' in Fig. 3.6) receives the frame from the source - i.e., a DAG card for Gi/Gn/IuPS and Gb. An expanded view of this module is given in Fig. 3.7. The MOTRA module is able to extract on-the-fly the value of all fields within the frame, at any protocol layer. For instance, it can extract the IMSI or Packet Temporary Mobile Subscriber Identity (PTMSI) and the cell identifier associated to the frame. These values are then fed into a local database (DB1 in Fig. 3.7) which maintains the current value of cell and PTMSI corresponding to each IMSI. Therefore, MS identity (IMSI) and location can be tracked despite PTMSI changes.

The MOTRA outputs the frame to the short-term ring buffer ('2' in Fig. 3.6) in a special format called "CF". This format is proprietary and was designed by the METAWIN staff specifically for this purpose: it includes special additional information associated to the frame, e.g., the corresponding IMSI and current cell of the generating MS, timestamp, etc. These additional fields in the CF format allow very fast seek-and-extractions of specific sub-traces matching some specific attributes, e.g., frame in a specific time period for and/or from a specific IMSI and/or cell.

A second instance of MOTRA ('3' in Fig. 3.6) reads from the ring-buffer, anonymizes the frame and writes it on an array of removable hard-disks ('4' in Fig. 3.6) which are installed on a different PC ("Transport PC"). From there the hard-disks can be unplugged and carried to ftw. for analysis. For privacy reasons the packet trace must be anonymized. This is achieved by the MOTRA parser,

which has a built-in secret-key hash function that is applied to all MS-specific values in the frame (IMSIs, MSISDN, etc.). Therefore, while it is still possible to discriminate packets of different MSs and associate packets to the same MS, it is not possible to retrieve the subscriber identity from the anonymized traces.

Anonymisation is a very important part of the system and therefore we cite the relating paragraph from [19]. The other objectives can be found in more detail within the final report about the METAIWN project [19].

3.4.1.2 Anonymisation

Traces in a live data network contain privacy relevant data which have to be protected. In the METAWIN project one level of protection with regards to privacy is performed by means of anonymisation.

In the current stage of the project anonymisation is defined as obfuscating user identifiers which are covered in the ETSI GPRS protocol specifications. In particular the static user identifiers IMSI and the MSISDN have to be anonymised. Other identifiers like IP addresses or the PTMSIs are not privacy relevant due to their dynamic nature. The basic goal of the anonymisation process is a irreversible but unique mapping between a real user and an anonymised identifier. The currently used method for anonymisation is a MD5 hash function in combination with a secret salt value. The salt value is appended to the real IMSI and hashed with the MD5 hash function. The result is the anonymised IMSI.

The Message Digest algorithm number 5 (MD5) is a cryptographically safe hash function, which means it has the two following properties. First with the currently available computational power it is not possible to find an input value to a known output value. This means that MD5 is a strict one-way function. Second it is not possible to find two input values which cause the same output value in a reasonable amount of time. For the anonymisation requirements in the METAWIN project the first property of the MD5 is important.

After hashing a 16 bytes anonymised user identifier is available. Since this identifier is too large to fit in the IMSI and MSISDN fields it is usually truncated to 7 bytes. The 7 bytes anonymised identifier is directly written over the place in the packet where the real user identifier is located causing the real user identifier to disappear.

The anonymisation is tightly integrated into the parser modules. The protocols relevant to anonymisation are GTPv0, GTPv1, MAP, BSSGP, and GMM/SM.

With the current anonymity implementation it is statistically impossible to derive a real user identifier from an anonymised identifier.

3.4.1.3 Other Related Monitoring Tools for 3G Core Networks

Nowadays, 2008, there exist many different monitoring tools for 3G core networks. The following list shows some of the most popular systems on the market:

- acceSS7,
- RadCom,

- Tektronix,
- NetHawk M5,
- NetTest.

However, all of these systems are optimized to report on KPIs but not to store and caputre large traces. This is a totally different approach compared to the METAWIN system, as these systems only store extracted time series of selected parameters but not the full trace information.

The METAWIN system is designed to store full traces with all payload. Therefore, the operator is able to understand and investigate problems found in the network. This feature makes the system unique on the market and very useful for research. However, for the researcher it has a high entry level, as he has to understand the whole system in detail and then has to write his own extraction modules for every parameter monitored.

3.4.2 Traces from a live 3G Core Network

The traces for this work were recorded on the Gn and the Gi interface. The Gi interface connects the GGSN with the Internet service provider of the operator. It allows for high level analysis of the total traffic of the network, e.g., service mix, Internet traffic up- and downlink, etc.

The Gn interface connects the GGSNs with the different SGSNs. This interface provides the same information as the Gi interface on a much higher granularity. The data packets traveling over the Gn interface are still encapsulated in the GTP protocol. Now each packet can be assigned to a specific PDP-context and further the packet can be addressed to a specific, anonymized subscriber. This allows us to build statistics based on the subscriber level. The information on the SGSN from which the packet originated is used to allocate the traffic to either the GPRS or the UMTS subnetwork. There is no possible way to differentiate between UMTS and HSDPA, respectively GPRS and EDGE.

The Gn traces were stored in the network of the operator. We were allowed to access the PC and run analysis locally on anonymized outputs of the ring buffer. The extracted statistics were then transported to the research facility. After the extraction of data via modules of the MOTRA the data was processed with AWK, PERL and MATLAB.

To extract the data for this thesis we used the following MOTRA modules:

- Gn-Duration
- Gi-Timeseries
- PCAP
- `tcptrace`.

Please refer to Appendix C for a detailed description of these modules.

3.4.2.1 Recorded Traces

This section presents an overview of the traces upon which this thesis is based. We use traces recorded over three years starting in December 2004 and lasting to April 2007. As the traffic load increased

over the years the first traces span over several days up to one week, while the most recent traces in April 2007 are four hours around the busy hour of the network.

We recorded most of our traces on the Gn interface. At this interface we are able to distinguish an individual subscriber via the GTP tunnel ID (see Appendix A-1.3 for more details on this topic). In our network each SGSN covered a specific radio technology, e.g., SGSN1 for GPRS traffic and SGSN15 for UMTS traffic. Therefore, it is possible to split the traffic into UMTS and GPRS traffic via the information to show which SGSN the user is attached.

With the introduction of Combo-SGSN, a SGSN type capable of connecting with GPRS and UMTS RANs simultaneously, this approach is not useful any more. Future tracing will process the RAN-Type flag, which is transmitted in the GTP tunnel and was introduced in Rel. 6, in order to assign traffic according to its RAN type. There were no Combo-SGSNs active till the end of 2007, so the RAN type was evaluated using the IP addresses of the SGSNs. Therefore, this is no restriction to our trace analysis.

Trace $\mathtt{TR}_1$ is one full week in December 2004 including GPRS and UMTS traces. At this time only a limited number of subscribers had a UMTS UE and even fewer had a data-card. Therefore, in this trace the majority of the subscribers were web browsing with mobile terminals.

Traces $\mathtt{TR}_2$ and $\mathtt{TR}_3$ are from one week in September 2005 and from one week in May 2006. The data-cards were already widely deployed and EDGE technology was installed in the network. The typical UMTS subscriber is no longer a data-card-only user.

Trace $\mathtt{TR}_4$ was recorded in October 2006. We recorded three consecutive days at the end of October. These traces already include HSDPA UEs but not HSUPA.

Trace $\mathtt{TR}_5$ contains several non consecutive days from April 2007. At this time HSDPA was already introduced widely. These traces served as a basis for the traffic model for e-mail.

Trace $\mathtt{TR}_6$ contains two consecutive days in September 2006. This data was evaluated on the flow level in Chapter 5.

Note that the traces $\mathtt{TR}_1$, $\mathtt{TR}_2$, $\mathtt{TR}_3$ and $\mathtt{TR}_6$ were recorded in a UMTS `Rel 5.0` environment. In this release the IMEI and RAT information was not transmitted over the Gn interface. Therefore, the analysis based on these traces can only make assumptions about the type of mobile terminal used by the customer. Unfortunately, Chapter 4, which could be improved by these two parameters, had to be done in the beginning of this work as it builds the basement for the following analysis. Therefore, we were not able to include detailed IMEI related analysis in this work.

3.5 Summary

IN this chapter we present important aspects of the UMTS core network nodes and protocols. The functions of RNC, SGSN, and GGSN are described in detail. Most of the data we recorded for this thesis was captured on the Gn interface. There the data is encapsulated in the GTP protocol. A detailed description of this protocol can be found in Appendix A. A discussion on the different steps in the setup of a data transmission and the impact of the techniques closes this short introduction into UMTS core networks.

In the last section we introduce the measurement setup which built the basis of the data for this thesis. It can capture traces on many interfaces of the UMTS core network and correlate the results with the subscriber ID.

The subscriber privacy is protected by payload cutting and IMSI hashing[1]. The IMSI hashing delivers an anonymous but unique pseudo-IMSI that allows for run-per-subscriber analysis without violating the privacy of the users. The metrics and modules that were implemented in order to extract further information from the dataset are described in Appendix C.

[1] A hash function is any well-defined procedure or mathematical function for turning some kind of data into a relatively small integer, that may serve as an index into an array. The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes.

Chapter 4

User Related Composition of GPRS/UMTS Traffic

Contents

IN this chapter we present a high level overview of the traffic recorded. It is the initial step towards understanding which kind of applications are present and how these services are used by the subscribers in the mobile core network.

In Section 4.1 we present the overall traffic patterns at the Gn interface. There we analyze the daily profiles and the growth in traffic over the past years. In addition to this we compare how the traffic splits between UMTS and GPRS. Section 4.2 investigates the application mix found in the core network, separately for GPRS and UMTS. In this section we briefly introduce the problems of malicious traffic. This topic is covered in a separate section. Section 4.3 focuses on the properties of the PDP-context. We analyze the duration, size, and rate of PDP-contexts generated by active users. Background radiation originating from various malicious activities, e.g., port scan, worm propagation, etc., introduces a bias to measurement results in live networks. The problem of filtering malicious traffic is discussed in Section 4.4. Section 4.5 summarizes this chapter.

Please recall that, as written in the introduction part of this work, absolute volume cannot be disclosed due to our NDA with the network operator. Therefore, we will only give truncated eCDFs and relative changes of shares.

4.1 Daily Usage Profile for UMTS and GPRS

THE daily profile of the network load is an important factor in dimensioning network capacity. Especially the information concerning the busy hour is very valuable. This is the time during which the maximum load is present in the network. For network dimensioning the network engineer can focus on this time span in order to evaluate the resulting QoS parameters. In this first section of the chapter we show qualitative results for the live network separately for GPRS and UMTS.

Figure 4.1 depicts the load on a Monday in May 2006, part of $\mathtt{TR}_3$. As expected the load profiles for UMTS and GPRS follow a typical daily profile. The peak hour is around 8.00 p.m. in the evening. This is interesting as it indicates that the main load in the traffic is due to private customers and not due to business subscribers only.

Between UMTS and GPRS there is a factor of 2:1 in terms of generated data rate at the Gn interfaces. As we reveal later, the gap between UMTS and GPRS was wider in the beginning of the mobile Internet service. However, over the years and with the introduction of EDGE the gap started to close.

Figure 4.2 presents the number of active or open PDP-contexts for 24 hours of a Monday in May 2006. The first interesting point is the fact that even during the night this curve does not drop to zero, because there are subscribers with an always-on context behavior. The factor between UMTS and GPRS subscribers is 1:5 which is just the opposite of what we found in the data rate case. The smaller UMTS population generates twice the data rate in the core network. The activation rate is the first derivative of the plotted curves. The steep slope of the curves between 5.00 to 9.00 a.m., indicates a sudden increase in the activation rate which is most assumed to be linked with the wakeup of the subscribers.

Most of the active contexts are torn down during the late evening after 9.00 p.m., one hour after the peak data rate load in the network. We assume this is due to timeouts in the mobile terminals.

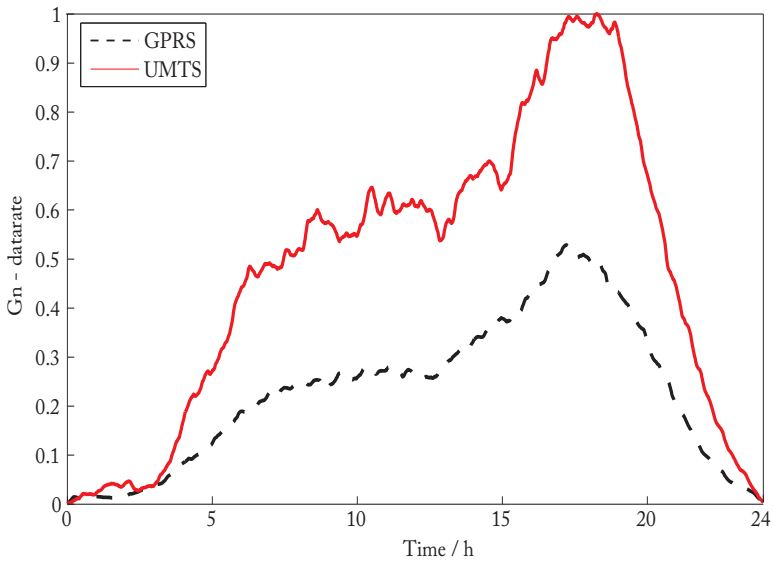


Figure 4.1: Daily profile of Gn data rate for UMTS and GPRS ($\mathtt{TR}_3$, normalized).

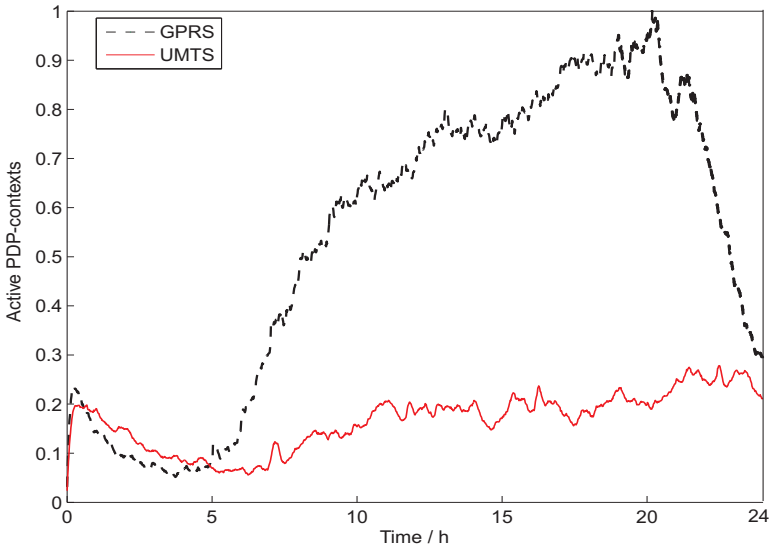


Figure 4.2: Daily profile of number of active PDP-contexts for UMTS and GPRS ($\mathtt{TR}_3$, normalized).

4.2 Volume and User Population

AT first we obtained results as to how the total traffic and user population is split between different services. We present results from two separate one-week periods in December 2004 and in September 2005, hereafter denoted respectively as $\mathtt{TR}_1$ and $\mathtt{TR}_2$ datasets. The comparison between the two provides an insight into the historical changes in the macroscopic traffic composition during almost one year.

Attached to the 3G Core Network there are two different RANs, namely GPRS and UMTS, delivering different link data rate (for further details see e.g. [4, pp. 23]). Since UMTS radio deployment is more recent, the number of UMTS base stations is smaller compared to GPRS and limited to urban and suburban areas. Therefore, GPRS traffic may be characterized by rural areas with a very small amount of traffic compared to urban city centers.

4.2.1 Volumes and User Population in GPRS and UMTS

Table 4.1 shows the relative fractions of users and volumes found in the network for GPRS and UMTS. Users are identified by their IMSI. Note that only *active* users are accounted here, i.e. with at least one PDP-context activation in the measurement window. Attached but inactive users are not accounted for. Note that in general UMTS capable terminals can also access GPRS RANs outside the UMTS coverage. We label as *GPRS users* those seen exclusively on the GPRS section during the measurement period, while those who accessed UMTS at least once are classified as *UMTS user*.

A caveat here is that a *UMTS user* can also generate GPRS traffic. While this traffic is correctly accounted as GPRS, the user count in GPRS only considers GPRS-only users. This leads to some over-estimation of the average per-user traffic for GPRS. In order to avoid such over-estimation, a more complex classification of users and traffic is required, considering the GPRS traffic produced by UMTS capable MSs in a separate class. For the sake of simplicity we omit such refinement in this work. A study of $\mathtt{TR}_1$ reveals that less than 1% of the subscribers change cell while a packet-switched connection is active. Out of this subset less than 1% had changed the RAN technology. This validates our assumption from above.

From Table 4.1 it can be seen that in $\mathtt{TR}_1$ the number of active UMTS users was approximately two orders of magnitude less than GPRS, but the two groups generated comparable traffic volumes. In fact at that time a large fraction of UMTS mobile stations were 3G data-cards mounted on laptops[1] coupled with flat-rate contracts, while UMTS hand-held devices were just starting to spread. In $\mathtt{TR}_2$ the fraction of UMTS users had increased, as the combined effect of additional UMTS subscribers plus legacy GPRS users upgraded to UMTS hand-held terminals.

The fact that for UMTS the growth factor in the volume share is substantially less than for the user share suggests that most of the new additional terminals were hand-held devices rather than 3G data-cards for laptops, under the assumption that hand-held devices generate less traffic than laptops due to a combination of differences in terminal capabilities and billing schemes. However, laptop data-cards are still an important subset in the UMTS terminal population. The comparison between

[1] See e.g. Vodafone data-cards `http://www.option.com/products/3g_edge.shtml`

the ratios of volume to user shares shows that the average per-user traffic is larger for UMTS than for GPRS. This was expected, given that UMTS delivers a higher data rate and hence a better user experience than GPRS, despite the fact that they share the same flat rates.

$\mathtt{TR}_1$ (Dec04)			$\mathtt{TR}_2$ (Sep05)		
RAN	Users	Volume	RAN	Users	Volume
GPRS	98.8%	58.3%	GPRS	90.0%	40.4%
UMTS	1.2%	41.7%	UMTS	10.0%	59.6%

Table 4.1: Fraction of users and volume in GPRS and UMTS ($\mathtt{TR}_1$, $\mathtt{TR}_2$).

As a next step we directly evaluated the weekly volume generated by individual users. The per-user volume distributions for GPRS and UMTS in $\mathtt{TR}_2$ are reported in Figure 4.3. All values are in MB/week. Figure 4.3(a) reports the CCDF. It can be seen that the median is around 8.2 MB/week for UMTS and only 102 KB/week for GPRS. The distribution spans about five orders of magnitude, denoting a large disparity in the user behavior.

This can be better seen from Figure 4.3(b). There we plot the fraction y of total volume generated by the fraction x of top users in a double log scale (loglog). We conclude that the top 1% of UMTS users generated around 16% of total traffic, while the top 1% of GPRS users generated 47% of total traffic. For the 10% of top users, the cumulated fraction of volume jumps to 59% for UMTS and 92% for GPRS. The GPRS figures reveal that the vast majority of users "seen" on GPRS are spurious users, generating sporadic or very low traffic volumes. At the other extreme of the range are a small population of heavy-users, who transfer massive volumes of traffic.

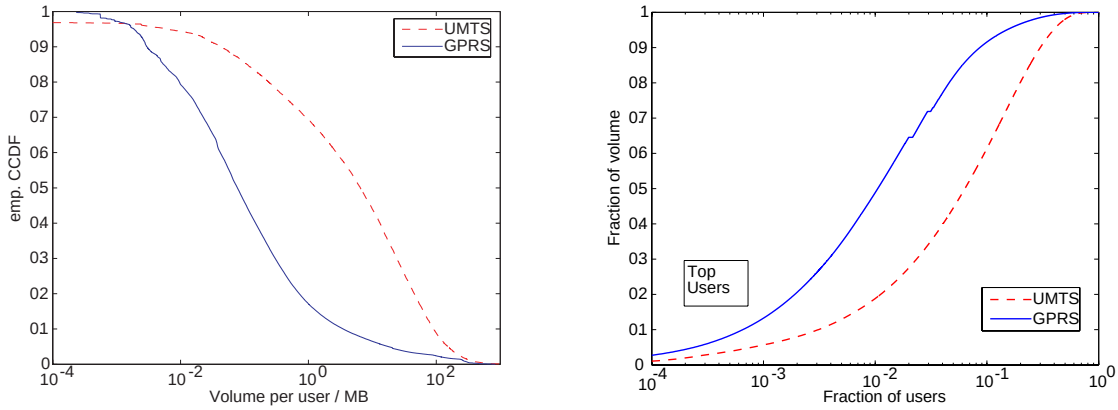
This suggests the possibility of classifying the users into three basic groups: sporadic, heavy and intermediate users. In general such classification would be service-dependent. However, as far as the total network load is concerned we can ignore service-specific metrics and consider just the total transferred volume in the measurement period. Hence we classified the users into three groups based upon arbitrarily chosen threshold values on the weekly volume v: low ($v \leq$ 1MB), medium (1 MB $< v <$ 100 MB) and high ($v \geq$ 100MB). We used the same boundary values for UMTS and GPRS. Table 4.2 shows the results for both datasets.

From Table 4.2 we see that the vast majority of users (70-85%) are in the "low" group, but they generate only a small fraction of the total traffic (1-4%). The largest change in GPRS from $\mathtt{TR}_1$ to $\mathtt{TR}_2$ is found in the "high" group: the fraction of heavy-users in GPRS increased and they account now for 70% of the total traffic.

In UMTS, we see that the "low" group was almost non-existent in $\mathtt{TR}_1$ (less than 4% of the users), while in $\mathtt{TR}_2$ the fraction of low users rose to almost 30.5%. This is probably due to the introduction of a large number of hand-held devices, while in $\mathtt{TR}_1$ the predominant terminal type was the laptop data-card.

4.2.2 Fraction of Volume per Service

Next we analyzed the volume share for each service. The following figures do not include custom services implemented for specific customers, which was pre-filtered based on the APN value on a per-



(a) Volume per user in one week (CCDF)

(b) Fraction of volume cumulated by the fraction of top users

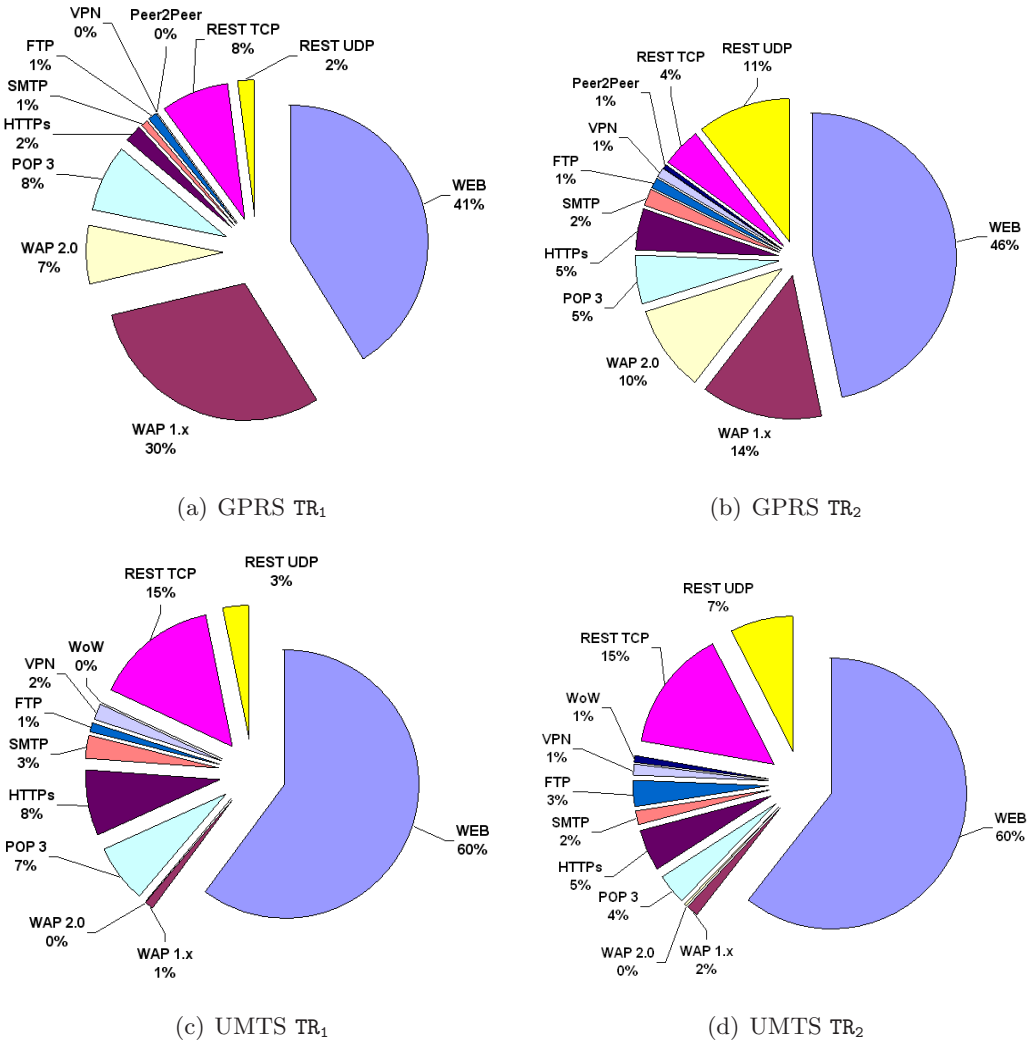
Figure 4.3: Distribution of volume per user in MB/week in September 05 ($\mathtt{TR}_2$). The x axis had to be truncated to 10^{-4} in order to not disclose the total number of users.

GPRS				**UMTS**			
$\mathtt{TR}_1$	high	medium	low	$\mathtt{TR}_1$	high	medium	low
Users	0.1%	29.3%	70.6%	Users	13.0%	83.3%	3.7%
Volume	7.0%	89.3%	3.7%	Volume	61.4%	38.6%	$< 0.1\%$
$\mathtt{TR}_2$	high	medium	low	$\mathtt{TR}_2$	high	medium	low
Users	2.9%	14.1%	83.0%	Users	7.3%	62.2%	30.5%
Volume	70.2%	28.5%	1.3%	Volume	91.2%	8.8%	$< 0.1\%$

Table 4.2: Grouping per transferred volume ($\mathtt{TR}_1$ (Dec04) and $\mathtt{TR}_2$ (Sep05), Values in fraction of total).

PDP-context basis. Also, we filtered traffic on ports tcp:135 and tcp:445, which are used by several scanning worms (see discussion further ahead in Section 4.2.5).
The results are presented in Figure 4.4 for UMTS and GPRS separately. UMTS shows only minor changes in the per-service volume distribution. In GPRS the main changes occur for the Wireless Application Protocol version 1.0 (WAP 1.0) share, which halved from 30% to 14%. This was only partially compensated for by the increase of WAP 2.0.

In both GPRS and UMTS the largest volume share is on WEB. In Figure 4.4, we compare relative shares. Table 4.3 shows the growth factors of each individual service during the last year (recall that absolute volume cannot be disclosed). From Table 4.3 we learn that in GPRS the WEB traffic has grown considerably more than WAP. Curiously, it appears that the opposite applies to UMTS, with a WAP growth rate of 800%, double than that of WEB. This is due to the fact that WAP was only marginally present in UMTS in $\mathtt{TR}_1$, and as noted above, UMTS capable handsets started to spread later.



(a) GPRS $\mathtt{TR}_1$ (b) GPRS $\mathtt{TR}_2$

(c) UMTS $\mathtt{TR}_1$ (d) UMTS $\mathtt{TR}_2$

Figure 4.4: Service shares for UMTS and GPRS ($\mathtt{TR}_1$, $\mathtt{TR}_2$).

4.2.3 Service Mix Diurnal Profile

Next we analyzed the stability of service shares on a day-by-day basis, limited to the top services. Table 4.4 shows the fraction of volume per service per RAN accumulated on a day-by-day basis. We found that they have nearly constant day-by-day shares, with only the exception of e-mail traffic which displays a significantly lower share on weekends. We assume that this is due to the fact that business people tend not to use their mobile access on the weekends. An interesting detail is, that GPRS users tend to download mainly e-mails (POP3/SMTP = 8/1), while the same ratio for UMTS

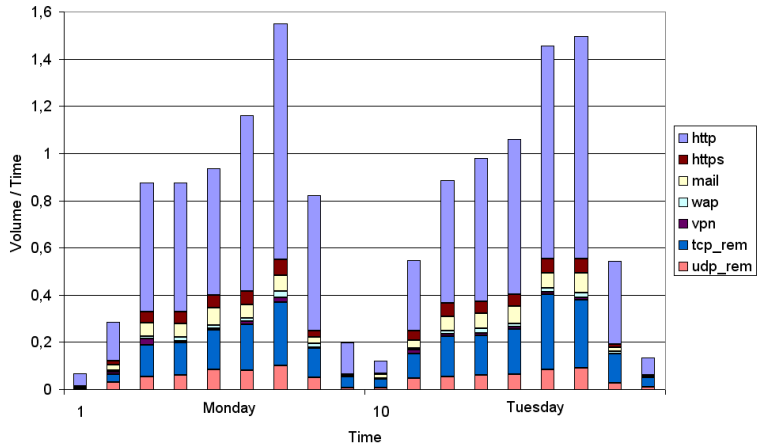
Service	WAP 1.x&2.0	HTTP	POP3, SMTP, IMAP	HTTPs	VPN (port1000, etc)
GPRS (%)	+132	+437	+122	+850	+113
UMTS (%)	+852	+401	+178	+241	+397

Table 4.3: Growth factor of per-service volumes normalized to one year ($\mathtt{TR}_1$ - $\mathtt{TR}_2$).

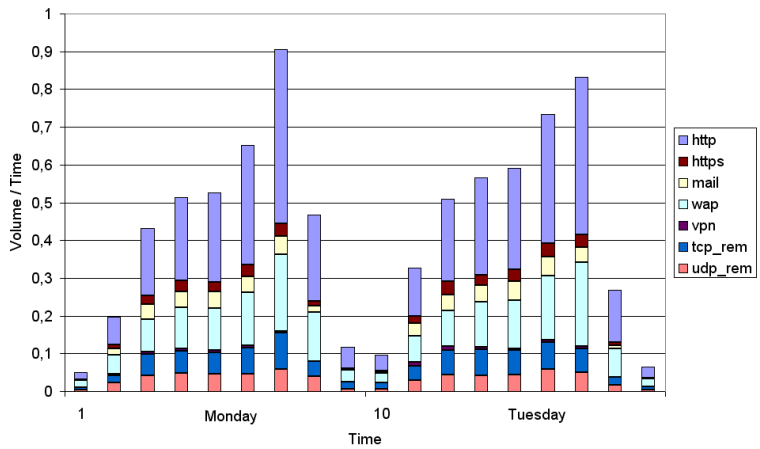
users is around two. We can state that the weekly service share is rather representative of single days.

Fraction of Volume														
	GPRS							UMTS						
Service	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
HTTP	39.4	40.1	39.1	38.8	40.1	40.3	41.8	60.7	60.6	62.8	61.0	60.5	60.3	61.0
WAP	35.8	35.4	36.0	34.7	35.4	39.2	32.1	2.1	1.9	1.8	2.1	1.7	1.9	2.0
POP3	8.0	8.4	8.2	8.5	8.9	7.0	7.3	3.9	4.3	4.4	4.5	3.8	2.0	1.9
SMTP	1.0	1.2	1.1	1.0	0.8	1.0	1.0	2.0	2.1	2.1	1.9	1.9	1.5	1.7
HTTPs	1.8	2.2	2.0	2.0	2.4	2.2	2.1	5.3	5.3	5.5	5.5	5.2	5.8	5.7
Rest TCP	10.3	9.7	10.9	11.1	8.9	7.8	10.7	15.7	14.3	14.6	14.9	14.7	16.3	16.2
Rest UDP	3.7	3.0	2.7	3.9	3.5	2.5	5.0	10.3	11.5	8.8	10.1	12.2	12.2	11.5

Table 4.4: Daily service shares - GPRS & UMTS ($\mathtt{TR}_1$).



(a) UMTS $\mathtt{TR}_2$



(b) GPRS $\mathtt{TR}_2$

Figure 4.5: Time of day effect in $\mathtt{TR}_2$, volume normalized.

Service	HTTP	HTTPS	POP3 & SMTP	WAP	VPN	TCP-Rest	UDP-Rest
GPRS	45.1 %	3.7 %	5.3 %	22.4 %	0.6 %	16.4 %	6.5 %
UTMS	64.2 %	5.5 %	8.9 %	1.7 %	2.8 %	10.8 %	6.1 %

Table 4.5: Average fraction of volume per service during the busy hour ($\mathtt{TR}_1$).

Network-planning departments often refer to the busy hour to dimension their networks. To answer the question if the weekly traffic share can also be used to adjust traffic shares in a busy hour simulation, we split the traffic further into bins of 10,000 sec. The result is presented in Figure 4.5 separated for GPRS and UMTS. The graphs have been rescaled by a factor in order not to disclose absolute volumes. From the shape of the curves we see that the busy hour is found between 7 p.m and 9 p.m. The actual patterns are stable on a day-by-day basis. It can be seen that WEB is the major player all over the day. The UMTS and GPRS patterns are similar to each other if we neglect WAP traffic. The service volume share in the busy hour is shown in Table 4.5. The values are averaged over three busy-hours in $\mathtt{TR}_2$.

4.2.4 Grouping Subscribers per Service Access

We grouped the users according to their services. We defined three groups, similar to [60]:

1. WAP only
2. WAP and E-Mail
3. Internet Services (no WAP).

Table 4.6 shows the relative numbers of users in each of the three groups for GPRS and UMTS in December 2004 ($\mathtt{TR}_1$) and September 2005 ($\mathtt{TR}_2$). The first group represents the vast majority of users who use only a mobile phone to browse WAP pages or download ring tones, send MMS via WAP and so on. In $\mathtt{TR}_1$ about 89.3% of all GPRS users are part of this group, but less than 0.1% of the UMTS users. As noted above in $\mathtt{TR}_1$ most users active on UMTS were data-card users. In $\mathtt{TR}_2$ the share of WAP UMTS has already grown to 11.4%. The second group is able to browse WAP, transfer e-mails but does not have WEB traffic. This group of users is small in UMTS, 1.4%, but populated with $\geq$5% in GPRS. In the last group there is no WAP traffic and mainly WEB volumes. This includes the mobile offices, users who access the internet outside their company by their notebooks or smart-phones. In the GPRS part of $\mathtt{TR}_2$ there is less than 17% of the users in this group, which consumes all of the WEB volume share, which is $\approx$50% of the total traffic. In the older traces on GPRS less than 2% generated all of the WEB traffic. This is consistent with a result presented in [60] by Vodafone and Ericsson, reporting a study carried out by a mobile provider in Hungary .

GPRS				UMTS			
Usage:	WAP only	WAP and E-Mail	Internet	Usage:	WAP only	WAP and E-Mail	Internet
$\mathtt{TR}_1$	89.3%	9.3%	1.4%	$\mathtt{TR}_1$	0.1%	1.4%	98.5%
$\mathtt{TR}_2$	78.1%	5.1%	16.8%	$\mathtt{TR}_2$	11.4%	0.9%	87.7%

Table 4.6: Grouping customers per service ($\mathtt{TR}_1$ (Dec04) and $\mathtt{TR}_2$ (Sep05), values in fraction of total).

4.2.5 Filtering in the Port Analysis

During the exploitative analysis we detected the presence of a large number of packets directed to ports tcp:135 and tcp:445, mainly TCP SYN in the uplink direction. This is due to several self-propagating worms attached to some infected 3G terminals. The presence of such undesired traffic should be expected since laptops with 3G data-cards - often equipped with popular operating systems - coexist nowadays with handsets and smart-phones in the 3G network. For years it has been well-known that undesired traffic is a steady component of the traffic in the wired networks (see for instance [61]). It is important to understand that such traffic is not the expression of user preference. Therefore we filtered it out of our analysis. The filtering rules were set to:

TCP: $\sum(\texttt{Uplink-Packets}) \geq 2 \,\&\, \sum(\texttt{Downlink-Packets}) \geq 2$

and

UDP $\sum(\texttt{Uplink-Packets}) \geq 1 \,\&\, \sum(\texttt{Downlink-Packets}) \geq 1$.

Please note that the filtering was mainly applied to avoid an inflation of the flow table in the analysis, e.g., each probing packet of a virus would have been counted as a new flow and in the measurement system each flow holds a set of state information which have to be maintained. This would put additional load to the system, making some of the analysis impossible. The goal of this filtering rule is not to exclude all malicious traffic but to reduce the flows created by the probing of infected terminals. Although we exclude the so called background radiation from the analysis, a network operator should always understand the presence of these activities. Systems like a stateful packet inspection tool could get overloaded due to the tree like structure of the mobile core network, in which all the traffic has to be routed over one single interface. A failure of this system will logically disconnect the whole core network from the Internet.

On TCP this rule is set according to the TCP handshake procedure, which generates three packets (pkts), plus a single data packet. In UDP there is no connection setup, therefore we set the threshold to two packets: one sent and one received packet. These rules filtered 3.4% of the TCP and 2.1% of the UDP traffic in $\mathtt{TR}_2$.
Finally, note that EDGE was introduced in August 2005, therefore the dataset taken in September 2005 includes data traffic from EDGE terminals, which we account simply as “GPRS”.

4.3 Analysis of the PDP-Context Activity

IN this section we investigate the activity of each user at the level of PDP-contexts. The PDP-context in a GPRS/UMTS network is comparable to a dial-up process in wired networks: a user has to establish a PDP-context to transfer data via a mobile network (see [52]). The dataset sample for this analysis was $\mathtt{TR}_2$ (September 2005), divided into GPRS and UMTS PDP-contexts. To reduce the processing time we took the first three days of $\mathtt{TR}_2$.

4.3.1 Per-User Activity

For each PDP-context j generated by MS i we extracted the total volume v_{ji} and the duration d_{ji}. For each MS i, we considered the following attributes: total number n_i of PDP-contexts, total transferred volume $s_i = \sum_{j=1}^{n_i} v_{ji}$ and total on-time $t_i = \sum_{j=1}^{n_i} d_{ji}$. Figures 4.6 and 4.7 show scatter plots of these three attributes with logarithmic binning, split to UMTS and GPRS. Each point represents the number of users i within the loglog binning[2]. In all plots there are boundaries due to admissible regions.

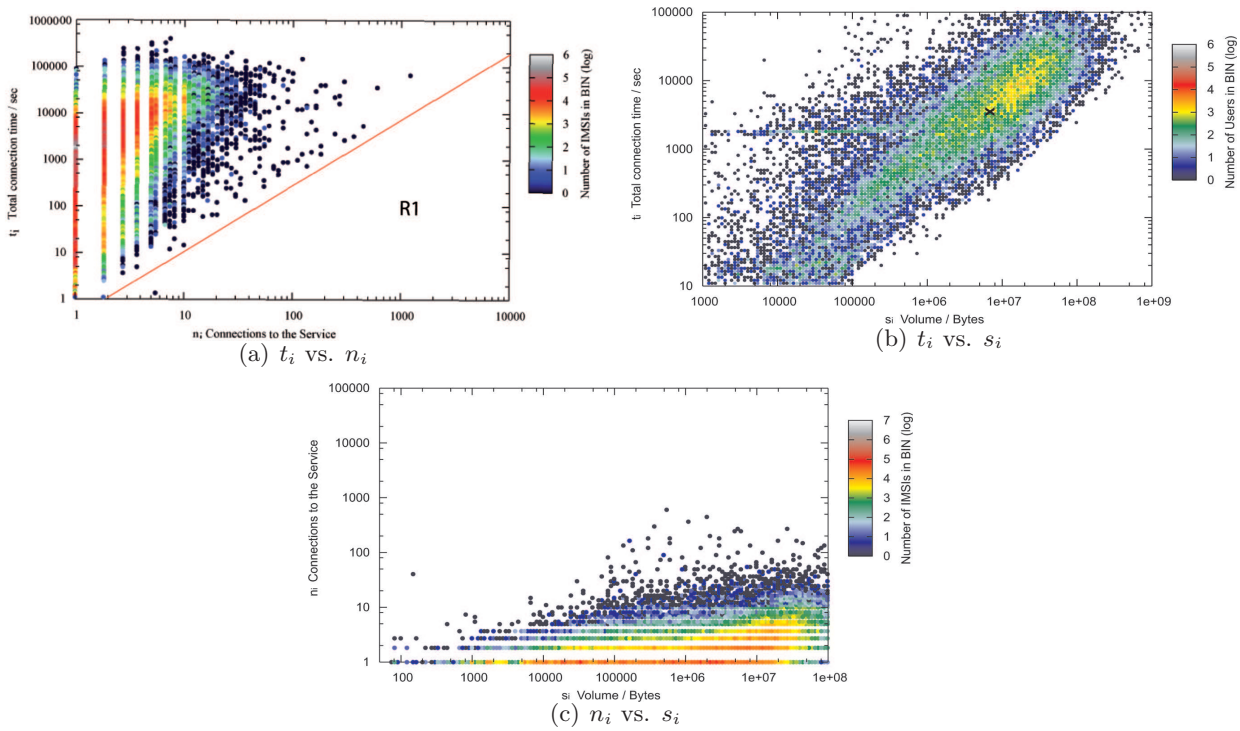
The empty region $\mathtt{R}_1$ in the Figures 4.6(a) and 4.7(a) relates to the presence of a minimum value for the duration of a PDP-context, say d_{min}, which forces a user with n_i PDP-context to stay on-line for at least $s_i \geq n_i \times d_{min}$. We obtain that in this case GPRS and UMTS have a similar footprint. In GPRS (Fig. 4.7(a)), for $n_i > 100$ there is a region $\mathtt{R}_2$ with correlations. The average PDP-context duration is 12.1 sec for this region. We observed a d_{min} of 32ms.

In the Figures 4.6(b) and 4.7(b) the lower limit relates to the maximum available data rate (cumulated uplink and downlink) UMTS: 320.3kbit/s; GPRS 81.2kbit/s. The maximum achievable data rate is 384kbit/s in UMTS and 85.6kbit/s in GPRS. This result is only a validation of the measurement, as there will always be idle times within a PDP-context. We observe that Fig. 4.6(b) is shifted to the right compared to 4.7(b), the mean values are indicated by the black **x** in the scatter plots. This indicates a higher net data rate in UMTS. The spreading is also smaller in UMTS. The upper limit is due to the presence of a time-out for long inactive PDP-contexts.

In Fig. 4.7(c) we can spot two different regions, not present in Fig. 4.6(c). For $n_i \leq 100$ there is no significant link between the number of connections and the volume transfered. The cluster of highly active users, with many PDP-contexts (>100) shows a correlation between these two values. This indicates service messages of constant size. The slope of the cluster cloud yields an average volume per PDP-context of around 1.8kB (see also Fig. 4.10 further ahead). More detailed analysis of the size of the PDP-context (in up- and downlink) can be found in the following section on the PDP-context size.

In Figure 4.8(a) there are four curves, two per RAN. The "IMSI - [RAN]" curves are the CDF of the variable n_i: the point at $x = X$ shows the *fraction of users* with $n_i \leq X$. The "PDP - [RAN]" curves show instead the total *fraction of total PDP-contexts* generated by all the users with $n_i \leq X$.

[2]The scale has again been multiplied by an arbitrary value to not reveal any absolute number.



(a) t_i vs. n_i

(b) t_i vs. s_i

(c) n_i vs. s_i

Figure 4.6: Scatter plots of *Volume*, *Number of Contexts* and *Total Duration* per IMSI [$\mathtt{TR}_2$, UMTS] (the figures have been truncated in order not to show the detailed numbers).

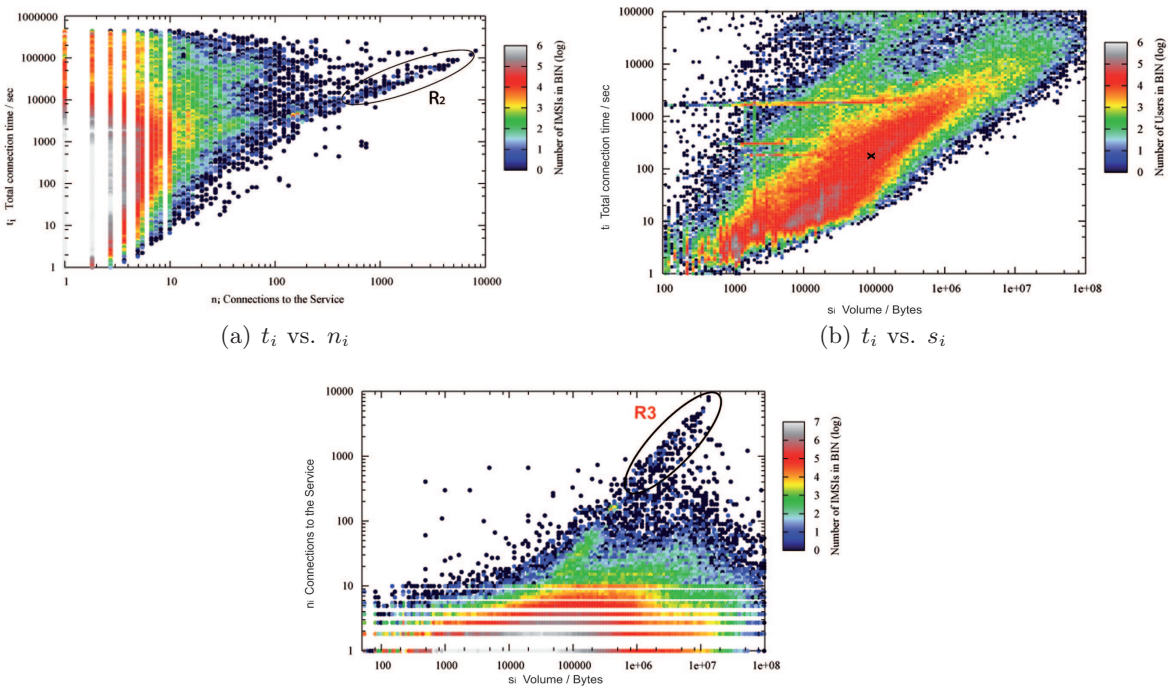
It is interesting to note that 48.3% of the UMTS-users and 31% of the GPRS-users had only one PDP-context within the observation period. Considering 100 PDP-contexts it is realized that >99.9% of the users are covered by this limit, however we obtain only ≤57.4% for UMTS, respectivly ≤88.4% for GPRS of all PDP-contexts for these users. From this we learn that the last 0.01 % of the population is generating a huge amount of the PDP-contexts (difference to 100 %). Further analysis will reveal that these contexts characterize the ePDFs of the population.

4.3.2 Distribution of PDP-Context Duration

In this section we investigate the distribution of PDP-context duration. Based on the high dispersion of n_i revealed by Fig. 4.8(a), a small fraction of users (0.01 %) generated values of n_i one to two orders of magnitude larger than the rest of the population. We know that a simple PDF of the duration d_{ji} would be mainly characterized by those few users with a very high number of PDP-contexts. These are assumed to originate from periodic polling of custom services with an intrinsically different behavior from other users. An offline inspection of the payload would confirm this assumption. However, as mentioned in the introduction this is not allowed due to data privacy issues.

We therefore considered the distributions of the variable d_{ji} *conditioned* to a particular value of n_i. This is represented in a compact way by the plots in Figures 4.9(a) and 4.9(b), where each discrete value of x represents the empirical distribution of the variables d_{ji} conditioned to the value of n_i given in abscissa. Note that the values of the empirical histogram (in log binning) across each column have been normalized by the counts per bin multiplied by the size of the corresponding bin.

The plot allows a comparison of the distribution of PDP-context duration for users with different



(a) t_i vs. n_i (b) t_i vs. s_i

(c) n_i vs. s_i

Figure 4.7: Scatter plots of *Volume*, *Number of Contexts* and *Total Duration* per IMSI [$\mathtt{TR}_2$, GPRS] (the figures have been truncated in order not to show the detailed numbers).

activity.

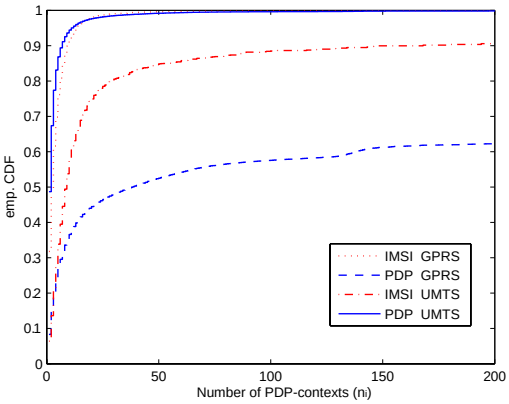
The plot reveals that low activity users (say ≤100 PDP/week) generate PDP-contexts with a different distribution of the duration from that of high activity users.

The low activity group exposes a distribution with a plateau ranging from one second up to 100 seconds.

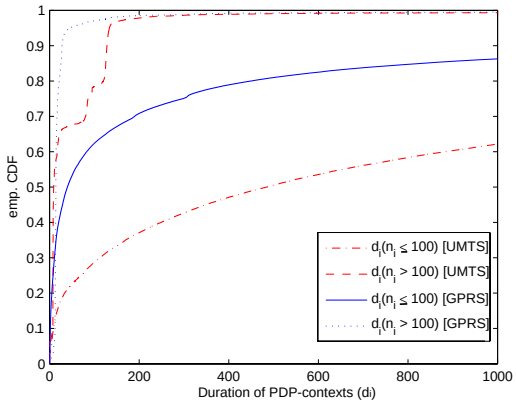
For the high activity users the PDP-context duration tends to be concentrated on smaller values. In UMTS there are two peaks around 8 and 80 seconds. In GPRS the high activity group has a peak at 11 seconds and a high plateau below five seconds duration. For more details see the results in Table 4.7.

Based on such figures, we decided to split the users into two main groups based on the value of n_i and an arbitrary set boundary of 100 PDP-contexts/week. The first group, with $n_i \leq 100$, gathers 99.6% of the users (see Figure 4.8(a)) and ≤57.4% (UMTS), respectivly ≤88.4% (GPRS) of the PDP-contexts, while in the second group 0.4% of users generate 40% of all contexts. Assuming a constant arrival rate of the second group, the resulting fraction is a function of the observation period, e.g., 6 days would double the number of PDP-contexts in the second group. It is likely that the second group is dominated by automatically accessed services (e.g. polling, remote-control). The eCDF of the PDP-context duration for each group and RAN are shown in Figure 4.8(b). The curves for IMSIs with ≤100 PDP-contexts fit a log-normal function, with parameters provided in Table 4.7.

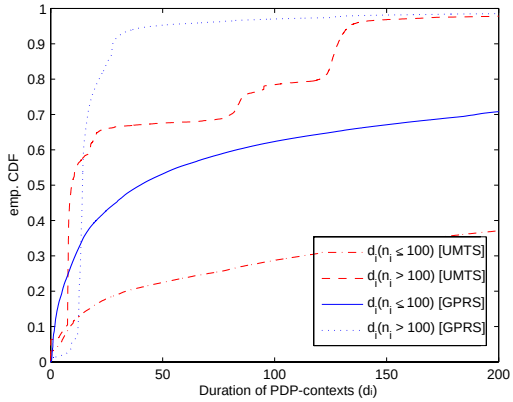
We observe that the duration of a PDP-context is larger for UMTS users. For IMSIs with a larger number of PDP-contexts, the distribution has discrete steps. In GPRS we see one major step at 11.3 sec for users with $n_i > 100$, in UMTS the first step is already at 8.2 sec. In UMTS this group has two



(a) Number of PDP-contexts



(b) Duration of PDP-contexts



(c) Duration of PDP-contexts - Zoom

Figure 4.8: Number of PDP-contexts and duration analysis low and high PDP-context count ($\mathtt{TR}_2$).

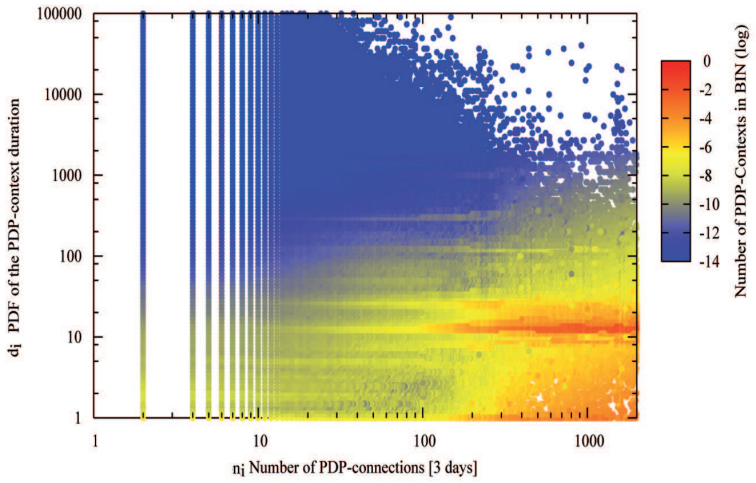
additional steps at 88.2 sec and at 131.2 sec. This indicates that there are additional services active on UMTS.

We split the fit for this CDF into two different components. The first component is a log-normal distribution as is used for the first group. The second component is a step function, which likely

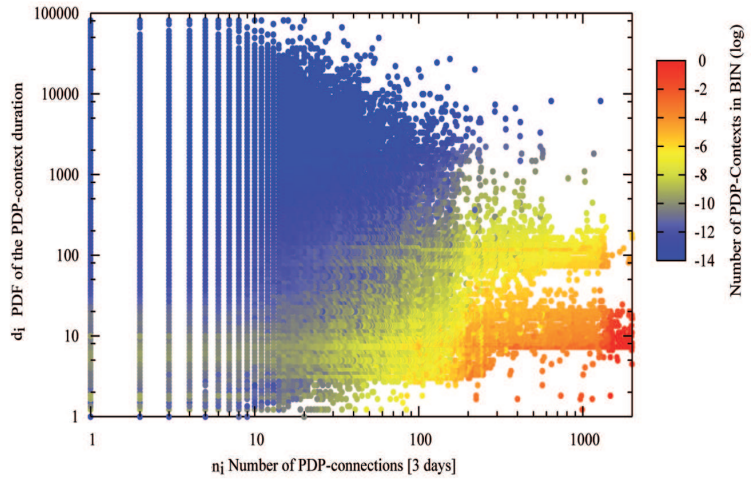
Technology	GPRS				UMTS			
Group	Weight	Distribution			Weight	Distribution		
$n_i \leq 100$	1	Weibull	$\lambda = 187.82$	k = 0.368	1	Weibull	$\lambda = 935.88$	k = 0.48
$n_i > 100$	+ 0.179	log-normal	$\mu = 5.13$	$\sigma = 2.44$	+ 0.36	log-normal	$\mu = 4.91$	$\sigma = 2.13$
	+ 0.821	step	11.3 sec	-	+ 0.421	step	8.2 sec	-
					+ 0.053	step	88.2 sec	-
					+ 0.167	step	131.2 sec	-

Table 4.7: Model parameter for PDP-context ON-time ($\mathtt{TR}_2$).

originated from the automatic services. Table 4.7 shows the parameters for the two groups per RAN. The second group is split into several lines, which have to be added to gain the total CDF.



(a) d_i vs. n_i (GPRS)



(b) d_i vs. n_i (UMTS)

Figure 4.9: PDF of PDP-context duration for users with x PDP-connections within three days (weighted, see text) ($\mathtt{TR}_2$).

4.3.3 The Volume of a PDP-Context

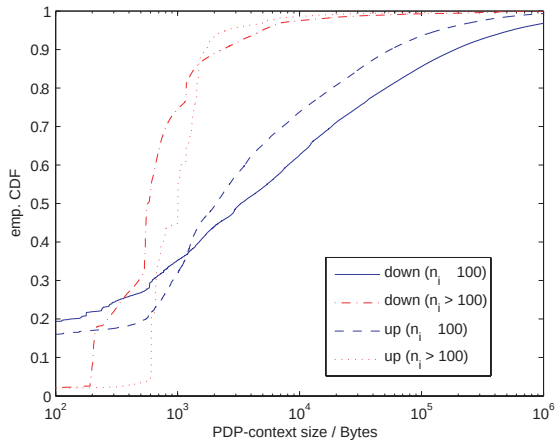
In this section we investigate the size of the PDP-contexts in terms of total volume. Recall that the volume is defined as the sum of the IP layer bytes (GTP payload) of user traffic, without considering the volume of signaling messages. In this section we distinguish between uplink and downlink volumes. We used the same grouping as above (n_i $\leq$100 and n_i >100). In Figure 4.10 we plot the CDF of downlink and uplink volume per PDP-context split into the two user groups. Figure 4.10(a) shows the results for GPRS and 4.10(b) for UMTS.

There is obviously a lower limit for the size of a PDP-context. This is due to the fact that we pre-filtered all PDP-contexts with no data packets, i.e. spurious PDP-contexts and erroneous procedures (they accounted for 2% of all PDP-contexts in our dataset). Note that in GPRS ≈20% and in UMTS ≈8% of the PDP-contexts in each group have less than 100 bytes. Table 4.8 shows the

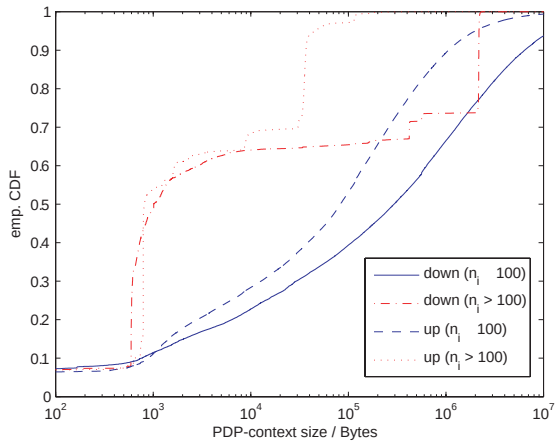
fitting for the up- and downlink bytes per PDP-context.

In GPRS more than 70% of all contexts are accumulated in the step near 1 kByte. In UMTS only ≈50% of the contexts are covered by this step. However, there are two additional steps. Together these steps add up to more than 85% of the PDP-contexts in UMTS.

This indicates that the services with many PDP-contexts not only have a step in the duration of their PDP-context, but also in the size of the transferred volume.



(a) GPRS



(b) UMTS

Figure 4.10: Size of PDP-contexts divided into two user groups ($\mathtt{TR}_2$).

$n_i \leq$ **100**	UMTS			GPRS		
Direction	Distribution			Distribution		
Uplink / Bytes	log-normal	$\mu = 8.22$	$\sigma = 2.23$	Weibull	$\lambda = 169.27$k	k = 0.39
Downlink / Bytes	Weibull	$\lambda = 10.52$k	k = 0.30	Weibull	$\lambda = 633.19$k	k = 0.34

Table 4.8: Number of bytes per PDP-context ($\mathtt{TR}_2$).

4.3.4 Total Volume and Number of PDP-contexts per Group

In conclusion we summed up the number of contexts and volume generated per group and compared the results. Using the same grouping as above we accumulated the volume and the number of PDP-contexts for $n_i \leq 100$, $n_i > 100$. Table 4.9 shows how the total volume maps to the different groups and how many users are inside. It is interesting to see that for GPRS less than 0.1% of the users generate ≈42% of all PDP-context activity in the network but only 2.1% of the total volume. In UMTS the effect is smaller, however more than 10% of the traffic is generated by less then 0.1%. The conclusion we have drawn from the numbers is that the group with $n_i \leq 100$ generates the main amount of the payload volume.

Technology	UMTS		GPRS	
Group	$n_i \leq 100$	$n_i > 100$	$n_i \leq 100$	$n_i > 100$
Number of Users	$\geq 99.9\%$	$< 0.1\%$	99.9%	0.1%
Generated PDP-contexts	88.6%	11.4%	57.6%	42.4%
Total Volume	97.6%	2.4%	97.9%	2.1%

Table 4.9: Fraction of *Users*, *Number of PDP-contexts* and *Volume* divided into two groups: $n_i \leq 100$, $n_i > 100$ ($\mathtt{TR}_2$).

4.4 Detecting and Filtering of Malicious Traffic

THE dataset $\mathtt{TR}_2$ for this analysis was recorded in September 2005. Starting with the raw dataset (Fig. 4.11 (a)) we observed several problems. The first problem was that we detected a high number of TCP connections with only one SYN packet. This is an indication for port scans and/or virus activities.

A (computer) virus is a self propagating program which reproduces itself by infecting other programs. Like a biological worm the program harms the resources of its host system. In contrast to a (computer) worm, a virus will not spread to other computers on its own. The so called (computer) worm is a computer program, which is capable of spreading via networks from one infected host to many other computers, e.g., by sending out spam e-mails.

A worm program will try to use exploitations on other system in the network in order to gain access to the system. Therefore, an infected computer will intensively scan its network to find other computers that can be infected. A worm will target typical services hosted on most computers (like Server Message Block (SMB) protocol on Windows machines), therefore the number of connection setup attempts on such ports infer worm activity.

The fact that the TCP port for SMB is seen in the top ten in Figure 4.11(a), supports the hypothesis of a self spreading worm performing port scans. More information on this topic can be found in [62]. SMB is used by Windows driven operating systems for sharing files, printers and communication interfaces. Some SMB implementations of Microsoft suffer from buffer-overflows, which can be exploited by various worms and viruses. Taking a look at Fig. 4.11(b) we find that SMB takes

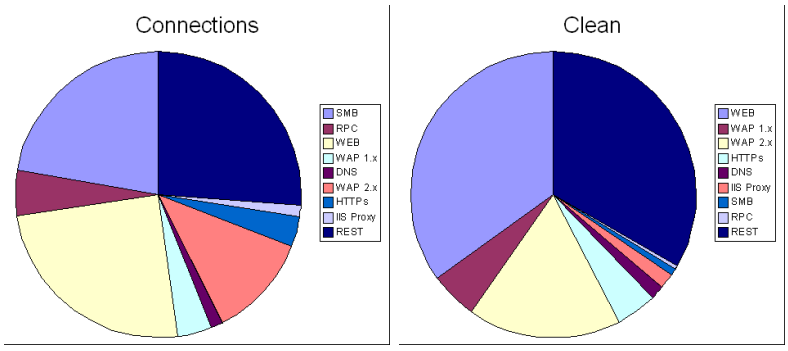
the major share of the connections. We therefore conclude that this is due to a virus spread and we set filter rules to eliminate this spurious traffic. At this point we should also mention that filtering undesired traffic can have strong impact on the reliability of traffic prediction. For example it can happen that the huge number of ignored TCP connections takes down some WEB-caching device located in the core network. As soon as we have filtered all TCP-connections with fewer than three packets (this is equal to the number of packets for one TCP-handshake [31] nearly 99.53% of the SMB traffic disappeared, depicted in Fig. 4.11(b).

The second port used by worms is the Remote Procedure Call (RPC) protocol. This port became infamous by the outbreak of the BLASTER-worm (see [62]). The filtering rules also had considerable impact on this port. Table 4.10 shows the impact of the filtering rules on the different services. We filtered for TCP $\geq 4pkts$, and for UDP $\geq 2pkts$.

We learned that valid DNS connections are too affected by this rule. This indicates problems with the DNS resolution for some customers. A valid connection for an address resolution has to have at least two packets [63].

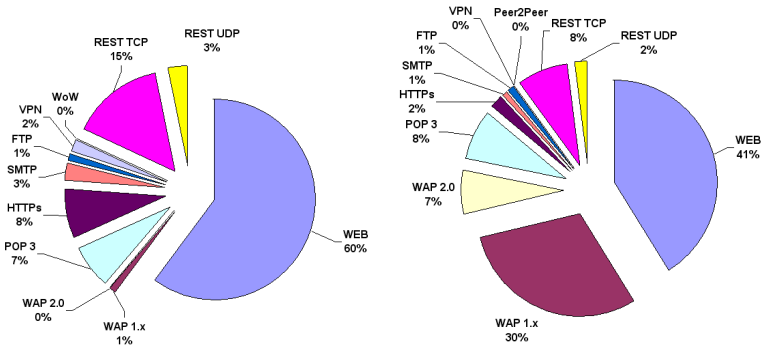
Protocol	Port	description	total number	clean number	clean/total %
TCP	445	MS-SMB	105,966,578	497,273	0.47 %
TCP	135	MS-RPC	25,068,717	256,533	1.02 %
TCP	80	HTTP	7,226,108	6,499,828	89.9 %
UDP	53	DNS	1,624,423	587,311	36.15 %
TCP	8001	WAP 2.0	1,663,524	1,593,378	95.78 %
TCP	443	HTTPs	799,929	707,273	88.41 %
TCP	4662	edonkey	292,737	204,920	70.0 %
ALL			154,234,535	14,650,613	9.50 %

Table 4.10: Number of connections in UMTS sorted in terms of SYN packets ($\mathtt{TR}_2$).



(a) Unfiltered (b) Filtered result

Figure 4.11: Number of connections in UMTS per service ($\mathtt{TR}_2$).



(a) UMTS (b) GPRS

Figure 4.12: Volume service share ($\mathtt{TR}_2$, filtered).

4.5 Summary

IN this chapter we present a high level analysis of the user traffic in a live 3G network. The first section studies the daily subscriber patterns. The busy hour of the network is found to be around 8:00 p.m. in the evening. The busy hour is roughly the same for active PDP-contexts and data rate on the Gn interface of the network. There are fewer UMTS subscribers (20 %), but they generate twice the traffic of the GPRS subscribers.

We discovered that the main part of the traffic is generated by a small fraction of users and that two-thirds of this traffic is due to browsing activity over HTTP. The share of wireless applications for web browsing, which was mainly found in the GPRS service shares, decreased over the measurement period in relative terms. However, in raw numbers the volume of all applications did increase with HTTP simply growing faster than all other services.

The analysis of the PDP-context behavior showed several effects. First, a small group of subscribers did generate the majority of the PDP-contexts. This group has a constant ON/OFF pattern indicating some application related periodic polling of service information. Excluding this data allowed us to fit the remaining datasets to a Weibull distribution.

Related Work: There are only a few papers available reporting usage patterns based on real measurements in live mobile networks, namely [64, 16]. The second paper reports only on GPRS.

The master thesis of Kivi [64] presents measurements done in Finland within a 3G network. The service shares were characterized by VPN connections, which account for more than 45% of the volume in the network.

In [16] Vargas et al. show the service shares of different operators in Hungary. These shares contained more than 40%, in some cases even up to 70%, of WAP traffic. There are no absolute numbers given in the paper, but we assume that these traces were collected in networks where no flat rate contracts were available.

The service shares presented in these papers are different from our analysis.

To our knowledge there was up to now no work done on the modeling of PDP-Contexts. However, there is a single eCDF plotted in [60]. This paper is about GPRS traffic only and the values are one

order of magnitude smaller than the results from our measurements.

Self Criticism: The split into different RANs relies only on the RAN of the cell in which the PDP-context was started. The traces for this thesis did not include the Radio Access Type (RAT) of the mobile. With this information a reliable classification would be possible. Therefore, our traffic shares are not 100% correct, although we showed in an example that less than 0.01% of the users is changing its RAN while using an active PDP-context.

The classification for data-card users is not precise. As the IMEI is not transmitted in this network setup, we had to use a heuristic method which we could not prove. Only further research will show if our assumptions were correct.

Chapter 5

Application Flow Patterns

Contents

IN this chapter we analyze the network traffic of the TCP and UDP flow level at the Gn interface of the mobile core network. We analyze for each flow duration, size in up- and in downlink direction. A TCP or UDP flow can be seen as the next smaller entity when compared with the PDP-context introduced in Chapter 4. A flow is, in the perspective of a network engineer, a set of packets transmitted by an application between two nodes.

Internet measurements carried out in Wide Area Networks (WAN) environments showed heavy-tail behavior in the size of flows, especially of TCP flows [13, 47, 48, 12]. Heavy-tailed distributions show a relatively high probability in regions far from the mean or median. We are interested in discovering if this heavy-tail behavior also holds true for Internet traffic in mobile cellular networks.

Section 5.1 starts with a motivation and introduces several basics such as the definition of a TCP or a UDP flow in the network. In case of a UDP stream the definition of a flow is not unique, therefore it is important to clarify the usage of these terms [65].

In this chapter we will fit a number of parameters according to empirical data. Section 5.2 introduces a set of methods to fit such parameters. We apply these functions in the follow up sections.

In Section 5.3 we first depict the length of the flows. We then analyze the tails, far from the mean or median, and the body, close to the mean or median, of the empirical distributions. We repeat this analysis for several datasets recorded over three years.

In our studies we revealed that TCP footprints are useful to analyze bottlenecks in a core network. Section 5.3.3 thus introduces the term footprint and presents some practical examples from our live network. Although this is not the direct scope of this thesis, it gives an important insight into which way flows may be affected by boundary conditions such as network congestion.

Section 5.4 presents a summary of the results and concludes this chapter about application flow patterns.

5.1 Introduction to Flow Analysis

IN the introduction to teletraffic engineering we stressed the fact that heavy-tail properties of a flow length are very important. However, for the performance of TCP the body of the distributions is also an important input parameter for simulation setups. We therefore focus on the tails and analyze separately the body of the various distributions found in the network.

Short TCP flows have an impact on the performance of TCP as many retransmission strategies of this protocol need a certain amount of outstanding packets to work properly [66, 67]. In applications with flows of fewer than ten packets (e.g. WAP 2.0), retransmission events especially in a high RTT environment decrease the QoS.

In wired networks short flows make up for a very small percentage of the traffic [68]. This is due to the fact that optimizing an application for wireless channels is equal to transmitting the data efficiently, e.g., by compressing the sent data. In a wireline network applications are not optimized in this way, therefore the flows are much larger. Heavy-tails are important for network performance of network elements, but short tails may be important for user perceived performance in several applications. Next we have to define the term *flow*.

5.1.1 The Flow

In this paragraph we introduce the definition of a flow as we implemented it in our tracing modules. We analyzed the output of the *Gn-Duration* metric (see 3.4.2) for the flow statistics.

A flow in a network is the set of all packets exchanged between two nodes in the Internet which belong to one application. It does not have to be bi-directional, e.g., a push service will generate a uni-directional flow. The flow is described by three numbers: {size in bytes, number of packets, duration}. For each flow, the size and the number of packets is split into up- and downlink direction.

In a flow the uplink direction is defined from the client to the server. This is not always equal to the network definition of uplink which is defined from the UE to the GGSN node. Therefore, in case the server is at the UE side the flow direction differs from the network direction.

5.1.1.1 The TCP Flow

The TCP protocol is connection oriented. Therefore, we can directly implement the definition of the TCP standard to define a connection in the tracing system. A TCP connection is defined via the quadruple: IP address and port for each source and destination.

The flow starts with a three-way handshake. The source sends a packet with the *TCP SYN* flag set. The destination replies with the *TCP SYN ACK* flag set. The handshake is finalized with a *TCP ACK* flag granting the receive of the last packet. Now the connection is established and the two nodes may transfer data. The flow ends with a similar procedure where the *TCP FIN* flag is set. In case of an out of order tear down, one node signals a *TCP RST* flag, closing the connection immediately. The HRO metric Gn-duration, described in Appendix C-1, has a small state machine following these procedures.

The TCP protocol allows bi-directional data transfers. However, often the data is nearly uni-directional, only ACK packets flow in the downlink direction. To identify the flow direction we analyze the up- and downlink flow size and order it according to the larger value.

5.1.1.2 The UDP Flow

For UDP traffic we define a connection as the union of all packets featuring the same quadruple (source/destination addresses and corresponding ports) within the same PDP-context and with a maximum inter-packet spacing of less than T (we followed [42] which proposes T=10 min).

This choice might be critical when comparing the two implementations of the same service adopting different transport protocols, e.g.WAP 1.x (over UDP) and WAP 2.0 (over TCP). We verified that the average number of WAP connections per-user is approximately the same for both implementations: 1.3 and 1.4 for WAP 1.x and WAP 2.0, respectively. This comforts our definition of connection for UDP.

Note that in literature there exist several other definitions for the UDP timeout [69, 68].

5.1.1.3 The Application Flow

The definition of an application flow is important for the following analysis. Based on the knowledge of the destination port we can use the well known ports, e.g., ports below 1024, to identify the involved

service. For example a TCP flow with a destination port 80 or 8080 is accounted as an HTTP flow.

This rule set works well in case there is no software tunneling its data over well known ports to bypass firewall systems in place, as for example *Skype* does (see [70, 71, 72, 73] for more details on the methods implemented in *Skype*).

An analysis of the flows, for the top ten services, showed that less than 0.1‰ of the flows was affected by software tunneling through well known ports in our recorded traces. However this number may change with an increasing number of mobile Internet subscribers.

HTTP Flows from any port to the destination ports 80 and 8080 are accounted as HTTP traffic.

In a first attempt we tried to run `tcptrace` over our traces in order to obtain a more detailed picture if these flows really carry HTTP traffic. However, `tcptrace` was not able to cope with the huge amount of traffic numbers we were dealing with. The actual load numbers in September 2006 were in the order of hundreds of gigabytes of traffic on the UMTS link for one day, this is far above the limit `tcptrace` can process. Therefore, we stayed with the assignment for flows stated above.

FTP The rule set for FTP is the same as for HTTP: all TCP flows with a destination port 20 are accounted as FTP traffic. The signaling port of FTP was ignored for our analysis. This is not a strong limitation as only a small amount of traffic is generated by the signaling part of this application.

WAP 1.X The assignment for WAP 1.x flows was different. WAP 1.x runs on UDP traffic and therefore we have to use the flow definition from 5.1.1.2 to estimate a UDP flow. In a WAP session the terminal connects with the WAP Gateway, which then connects to the page requested by the browser of the terminal. Monitoring on the Gn link allows us to track the destination IP address of a request. If we find the address of the WAP 1.x gateway in the destination address field we account the traffic as WAP 1.x traffic. Since the WAP traffic can only work with this gateway we were able to identify 100 % of the WAP 1.x traffic.

WAP 2.0 The WAP 2.0 case is similar to the WAP 1.x solution. WAP 2.0 runs on top of a TCP connection. Again the terminal connects with a WAP gateway, which then fetches and processes the data for the terminal. In our network there are different gateways for WAP 2.0 and WAP 1.x. Therefore, we are able to track these types of traffic via the destination IP addresses at the Gn link trace. The assignment is precise again.

5.1.2 Protocol Shares

In this paragraph we analyze the number of connections sorted out by the specific protocols. We split the graphs into UMTS and GPRS to gain a better overview on the traffic.

In Section 4.2 we have already learned that the service mix, derived from the volume of each connection, on GPRS and UMTS is evolving towards a service share measured in wireline networks. This analysis focuses on the number of connections. Again, we filtered out all connections with fewer than three packets and also those with no packet in either up- or downlink direction.

Table 5.1 depicts the TCP/UDP share for UMTS and GPRS for different datasets. The share of TCP connections is around 90%. The value is stable over the last three years. Still there is a difference compared to traces from wireline networks [74].

The TCP/UDP connection share is nearly identical for UMTS and GPRS. This fact again indicates that subscribers use mobile Internet services.

UMTS			
Date	Nov 05	Sep 06	Sep 07
Avg. over	7 days	2 days	2 days
TCP	86.2 %	92.4 %	88.0 %
UDP	13.8 %	7.6 %	12.0 %

GPRS			
Date	Nov 05	Sep 06	Sep 07
Avg. over	7 days	2 days	2 days
TCP	87.4 %	91.0 %	87.2 %
UDP	12.6 %	9.0 %	12.8 %

Table 5.1: TCP/IP flows from 2005 to 2007.

5.2 Fitting of Distributions to Empirical Data

IN this chapter we require several techniques to fit our empirical data to analytical distributions. This section introduces the necessary steps.

To fit an empirical dataset to an analytical distribution four basic steps are necessary. First the data has to be independent of a sample-by-sample basis and stationary without a trend. Secondly, we have to chose a class of analytical distributions which we think fits our data best. Thirdly, the parameters for the distribution are estimated. And finally, testing the goodness of the fit tells us if our choice at step two was correct. In case step four fails we have to restart at step two.

For the following paragraphs the empirical sample is given by $X = \{x_1, x_2, \ldots, x_n\}$.

5.2.1 Pre-Evaluation of the Dataset

5.2.1.1 Stationarity and Ergodicity

A stationary process has the same probability density function for all times or positions. We introduced the terms strict and weak stationarity already in Chapter 2.

A stochastic process is called ergodic if its ensemble averages equal appropriate time averages. In our case we investigated for weak stationarity, considering only the first and second order moment of

our samples. The mean was calculated in a sliding window of size k:

$$X'(k) = \frac{1}{k} \sum_{i=k'}^{k'+(k-1)} x_i. \tag{5.1}$$

In order that the sample averages $X'(k)$ converges too a constant mean μ', corresponding the statistical mean of the process, for $k \to \infty$, Equation (5.1) requires the process to be ergodic. We assume ergodicity from the nature of the underlying network (many subscribers, random radio and network conditions). Now we can define a variance σ' within the sliding window:

$$\sigma'^2 = \frac{1}{k-1} \sum_{i=k'}^{k'+(k-1)} (x_i - \mu')^2. \tag{5.2}$$

The mean was then visually inspected for trend components. None of our traces showed a trend component. We expected such a result, as our largest traces spans only one week. This is more than enough data to analyze the flow parameters, however under normal network operation there should be no detectable trend in such a short period of time. In this context *enough* is related to the number of events we need to record in order to display heavy-tailed behavior, e.g., 10^7 events.

5.2.1.2 Serial Correlation

In a time series, where observations have a natural order in time, there may be a correlation between two successive values. The autocorrelation function (ACF) calculates the correlation for all possible time lags of the time series. It is the correlation between two values of the same sequence x_i and x_{i+k}. The normalized autocorrelation of a WSS sequence is defined as:

$$r(k) = \frac{E\left\{(x_i - \mu')(x_{i+k} - \mu')\right\}}{\sigma'^2}. \tag{5.3}$$

High values of the ACF indicate that values with this specific time lag are correlated strongly whereas for time lags with less correlation the ACF is also low. For white noise it can be shown that 95% of the sample autocorrelation values should lie between $\pm\sqrt{n}$ since the sample autocorrelation values are asymptotically normally distributed with zero mean and variance $1/n$. If the autocorrelations are below this boundary the sequence is likely to be random (i.e. white noise). For further reading see [75, 76].

There are different notations to indicate independence, one is:

$$|r(k)| \leq \frac{\Phi^{-1}(1 - \delta/2)}{\sqrt{n}}. \tag{5.4}$$

Where Φ is the percent point function, inverse of the cumulative distribution function, of the standard normal distribution and δ is the significance level to assume independence (see Appendix D-1).

5.2.2 Parameter Estimation

The parameter estimation is to extract an estimator. In the optimal case this estimator extracts all information from the input and is implementable with reasonable effort. The estimator takes measured data as an input and produces an estimate of the parameters.

In the general case we used the Maximum Likelihood Estimator (MLE) to estimate the parameters of the distributions. If heavy-tails were present in the input data we split the distributions into a body and a tail part. The body part was fit to several analytic distributions. The parameters for the tail part, which we define to be dominated by a heavy-tail effect, were extracted with the scaling method [42].

A method we already introduced in Section 2.2.2.2 is the phase type approximation. We also describe the estimation method based on the Expectation Maximization (EM) algorithm below.

5.2.2.1 Maximum Likelihood Estimation

The MLE is a method of calculating the best way of fitting a mathematical model to some measured data. The method was originally introduced by Sir R. A Fisher in 1912 [77].

Loosely speaking, given a set of data and a chosen underlying probability model, the MLE selects the values for the model parameters which make the data more likely than other values would do. For the normal distribution the MLE has a unique solution. However in more complicated scenarios this may not be the case.

A MLE estimator is defined in the following way:

Given a family of probability distributions D_Θ with the parameter Θ and a Probability Density Function (PDF) f_Θ, we compute the probability density associated with our observed data, $f_\Theta(X(x_1, x_2, \ldots, x_n|\Theta))$. We want to find a fit for the parameter Θ.

The likelihood function of Θ with given $x_1, x_2, \ldots, x_n$ is:

$$L(\Theta) = f_\Theta(x_1, x_2, \ldots, x_n|\Theta). \tag{5.5}$$

The value of Θ is now estimated by maximizing the function $L(\Theta)$. This is the MLE of Θ, defined as:

$$\hat{\Theta}_{MLE} = \arg\max_{\Theta} L(\Theta). \tag{5.6}$$

The problem simplifies considerably if we now assume that our dataset drawn from a particular distribution is independent and identically distributed. In this case the likelihood is a product of n univariate probability densities:

$$L(\Theta) = \prod_{i=1}^{N} f_\Theta(x_i|\Theta). \tag{5.7}$$

A monotone transformation does not affect the position of an extreme value. Therefore we only have to search the maximum of the logarithmic sum:

$$\log L(\Theta) = \sum_{i=1}^{N} \log f_\Theta(x_i|\Theta). \tag{5.8}$$

The maximum of this expression can be found by solving the partial differential equations:

$$\frac{\partial \log L(\Theta)}{\partial \Theta} = \mathbf{0}. \tag{5.9}$$

5.2.2.2 Phase Type Approximation of an Empirical Distribution

The phase type distribution was introduced in Chapter 2. This method delivers an analytical tractable approximation of a general distribution. The paragraph will focus on hyper-exponential phase type distribution in order to approximate an empirical distribution derived from measurements.

There are several different methods to estimate the parameters [78]. Here we introduce the EM-algorithm from [44]. It is an iterative method which needs only the empirical data and has a complexity that grows in a linear way with the number of phases present in the model.

The following paragraph describes the EM-algorithm for phase type distributions [44]. The measured dataset is again $X = \{x_1, x_2, \ldots, x_n\}$ and let the number of phases of a hyper-exponential distribution be I. There are three parameters in the algorithm: $c_{1,\ldots,I}$, $\lambda_{1,\ldots,I}$ and the precision ϵ defining the terminating condition of the iterative algorithm.

For every phase let:

$$p(x_j|\lambda_i) = \lambda_i \exp^{-\lambda_i x_j}, \tag{5.10}$$

$$p(x_j|(c_i, \lambda_i)) = \sum_{i=1}^{I} c_i p(x_j|\lambda_i). \tag{5.11}$$

Each iteration starts with $c_i = c'_i$, $\lambda_i = \lambda'_i$ and then computes:

$$p(i|x_j, \lambda_i) = \frac{c_i p(x_j|\lambda_i)}{p(x_j|(c_i, \lambda_i))}, \tag{5.12}$$

$$c'_i = 1/N \sum_{j=1}^{N} p(i|x_j, \lambda_i), \tag{5.13}$$

$$\lambda'_i = \frac{\sum_{j=1}^{N} p(i|x_j, \lambda_i)}{\sum_{j=1}^{N} x_j p(i|x_j, \lambda_i)}, \tag{5.14}$$

until $|c_i - c'_i| < \epsilon$ and $|\lambda_i - \lambda'_i| < \epsilon$. The resulting values for c_i and λ_i fully describe the phase type distribution.

5.2.2.3 The Scaling Method

An empirical distribution derived from Internet measurements typically consists of two parts: a body and a (heavy) tail. The Hill estimator [79] offers a simple way to estimate the tail parameters. However, this method needs a starting point well away from the body of the distribution.

Another method of finding the slope (α) of the tail parameters was introduced by Corvella and Taqqu in [42]. It utilizes a special behavior of variables with a heavy-tail distribution. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a dataset and x_i^m the aggregated sums of non-overlapping blocks of observations of size m. If $\mathbf{X}$ has a heavy-tail property, the distributional properties of x_i^m will also show heavy-tail behavior. This method aggregates the dataset of several different steps and then estimates for each step the slope parameter α. The heavy-tail region is chosen on a comparison of the different α estimates at the different aggregation levels.

In our analysis we applied the method of Corvella et al. from [42] which was coded into a tool called AEST. This tool outputs the α estimate for a preset number of aggregation levels. The parameter α corresponds to the parameter k of the Pareto distribution. It marks heavy-tailed regions in the output of each aggregation level. In an ideal case the entire distribution is marked and all aggregations are parallel in a log-log plot[1].

5.2.3 Goodness of Fit

The last step is the verification of the estimated parameters. A goodness of fit describes how well a statistical model fits the observed data. A typical measure of goodness of fit summarizes the discrepancy between observed and expected values from the chosen model. We looked at two methods: the chi-square and the Kolmogorov-Smirnov (KS) test.

The chi-square test is more general in the application. It compares a sum of differences between observed and expected outcome frequencies. The goodness of fit is found by comparing the resulting value to the chi-square distribution. However, this test needs an input parameter set in advance, which has an impact on the result. If the data follows arbitrary distributions, setting this parameter is not trivial. Therefore we use the KS test in this thesis.

The KS test determines whether two underlying one-dimensional probability distributions differ. We assume F_n to be an empirical distribution function of n independent and identically-distributed observations of x_i. Next we choose $F(x)$ to be the underlying distribution the dataset follows. Then the Kolmogorov-Smirnov statistic is:

$$D = \sup_x |F_n(x) - F(x)| \,, \tag{5.15}$$

A critical value for D can be defined. Note sup Z is the supremum of the set Z. Loosely speaking if Z is a subset of T then sup Z is the smallest element of T which is greater than or equal to all elements of Z. Based on the value of D the assumption that $F_n(x)$ is a member of $F(x)$ will then either be accepted or rejected. The value depends on the distribution and the level of significance chosen. For empirical measurements D is often above the critical value as the data does not originate from a single analytical distribution. Therefore the value of D is used to find the distribution that fits best the measurement data.

[1] Our measurement sets were very large with more than 10^8 connections per day. Therefore, we applied some improvements to the original tool in order to make it run faster and fixed some bugs which only showed up for more than 10^6 input lines. However, these changes did not modify the core functionality of the code.

5.3 Flow Statistics

IN this section we investigate the flow length statistics. In Section 5.3.1 we compare the eCDFs of different datasets recorded between fall of 2005 and fall of 2007. Paragraph 5.3.2 validates the dataset with respect to independence and stationarity. After its validation, in 5.3.4 we check for heavy-tail behavior in the dataset. After the preprocessing we fit the datasets to analytic distributions.

5.3.1 Evolution of the TCP/UDP and Application Flow Lengths from 2005 to 2007

In this section we provide a first overview to our dataset. Figure 5.1 depicts the eCDF for all flows separately split into UMTS (left) and GPRS (right). Figures 5.1(c) and 5.1(d) show how the TCP flow size grew over three years, again for UMTS and GPRS, separately. Although the traffic amount did increase more than one order of magnitude between 2005 and 2007, we learned from the figures that the shape of the distribution did not change at all. In the GPRS case the distribution was stable between 2005 and 2006 and grew only slightly in 2007. Therefore we decided to analyze only one dataset in more detail. We chose the dataset from September 2006 ($\mathtt{TR}_6$), as it is right in the middle of our observation time period.

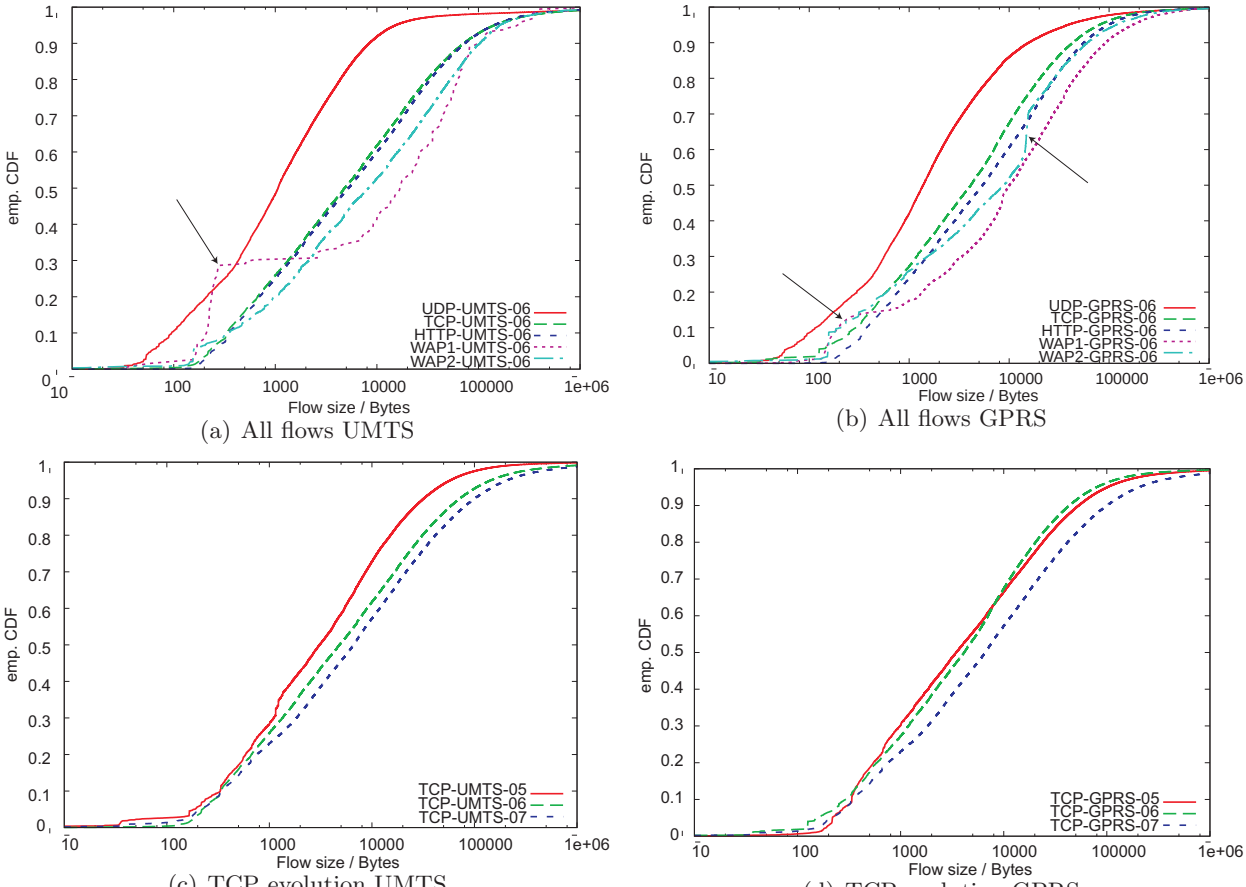
As already discussed Figures 5.1(a) and 5.1(b) are based on a dataset from September 2006, $\mathtt{TR}_6$. The UMTS flows (Fig. 5.1(a)) for WAP have a dedicated step at around 200 bytes. We assume these steps originate from WAP chat applications which are very popular for younger subscribers. Often such steps in an empirical measurement cannot be modeled with a simple analytic distribution alone. We thus use the phase type approach instead. The UDP flows in UMTS are one order of magnitude smaller than the TCP flows. This result fits our numbers for the volume transported via UDP and TCP. It is interesting to note that the HTTP curve is nearly identical to the TCP curve. We assume that most of the TCP flows are in fact generated from HTTP or similar protocols such as HTTP secure (HTTPS).

The results for the GPRS flows (Fig. 5.1(b)) are similar to UMTS. Again there is a step of $\approx 13\,\%$ in the WAP distributions. The step in GPRS is smaller ($\approx 13\,\%$) than in the UMTS case ($\approx 30\,\%$). Note that the overall number of transfered bytes over WAP is one order of magnitude larger in GPRS, therefore the smaller step includes more traffic than in the UMTS case.

In GPRS there is a second step in the WAP distributions at around 20 kByte. Based on manual inspection of a part of these samples, we assume the step is caused by MMS services. The HTTP flows are again very similar to the TCP flows, with a bit larger gap this time. From this first look we assume that fitting TCP, UDP, HTTP is straight forward.

5.3.2 Example Validation of the Datasets

In this section we tested our dataset for stationarity and independence. We inspected all flows from the September 2006 ($\mathtt{TR}_6$) dataset. We plotted the results for the TCP flow dataset (Fig. 5.2(a)) for the length of the flows. The lack of correlation was tested with the ACF and a FFT analysis.



(a) All flows UMTS

(b) All flows GPRS

(c) TCP evolution UMTS

(d) TCP evolution GPRS

Figure 5.1: Flow length of all services and growth over the years.

For all flows the ACF stayed below a 1% significance level and the FFT plot did not show any dominating frequency. Figure 5.2(b) shows the ACF for the TCP flows.

To test for a trend we analyzed moving average plots with different levels of aggregation. The different moving average plots did not show any trend. Figure 5.2(c) displays the cumulated average for the TCP flow dataset.

Note that for the FFT we reduced the datasets to the peak hours of 7 p.m. to 11 p.m., a full dataset of several days has a daily periodicity due to the lack of traffic in the night. None of the FFT analysis had a dominant frequency. We depict an example of this in Fig. 5.2(d).

In order to test the long time stationarity we split the trace into bins of 1000sec and calculated the eCDF for each bin. A Kullback-Leibler Distance was used to evaluate the distance between the subsamples. The KL distance was introduced by Kullback and Leibler in 1952 [80]. It is a non-commutative measure to compare the difference of two probability distributions, e.g., $Q(x)$ and $P(x)$. The value of the metric is zero, if and only if $P(x)$ equals $Q(x)$ and is larger than zero in any other case. The results are all below 0.1 indicating a very small difference between the curves. Therefore we conclude that the flow distributions are valid all over the day. Note that in some traces only very low traffic occurs during the night hours so we excluded these bins from the investigation as they did not have enough events to produce a meaningful PDF.

Figure 5.3 depicts a sample of this analysis in the form of a scatter plot. The x-axis represents the time of day, the y-axis is the size of the flow. Both axis are binned and each bin is colored accordingly to the number of events that fell into it. The color bar is normalized for every 1000 sec subsample in order to produce an empirical PDF. In other words for each $x = \text{const}$ the color information along the y-axis represents the PDF of the dataset.

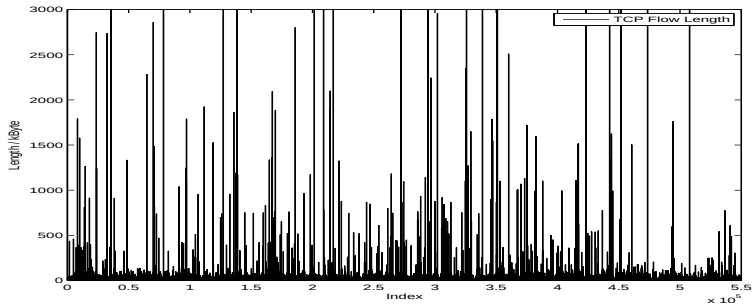
5.3.3 TCP Retransmission Footprints

In wireless cellular networks the bit errors on the radio link are recovered by Forward Error Correction (FEC) and Automatic Repeat reQuest (ARQ) methods. However in practice there is a hard limit on the maximum number of packet retransmissions, e.g., three, leaving room for a certain level of residual packet loss. This effect, together with the handover procedures due to terminal mobility, cause a certain "physiological" level of packet loss [81].

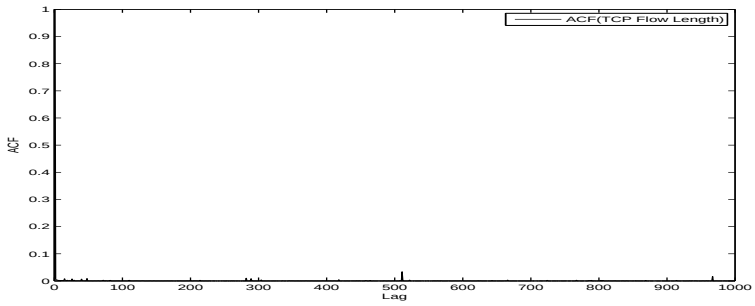
In this section we investigate how the number of packets in a TCP stream is linked to the number of retransmissions. We denote by n_i and N_i respectively, the number of retransmissions and total packets observed for the generic ith user in a given time period (e.g. 30 min). The dataset consisted of four hours during the peak hours in October 2007.

We used scatterplots to visualize the process $[N_i, n_i]$ and to create *TCP footprints*. As both variables span several orders of magnitude we introduced a logarithmic binning with 150 bins on each axis. In a first step we filtered the datasets using a median filter to remove outliers. In a second run we applied a mean filter in order to smooth the footprint.

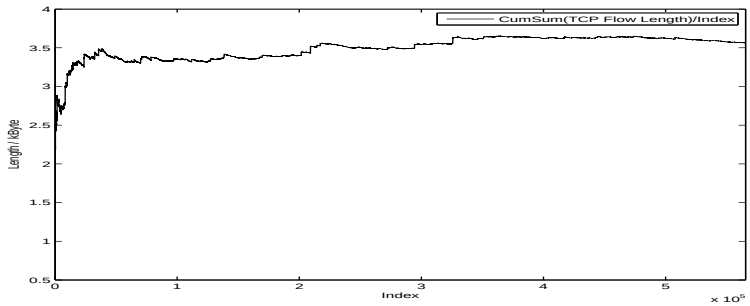
Finally we normalized the value of the bins by the number of events in the trace. The datasets are now equal to an empirical binned bidimensional histogram function. This step compensated for the varying amount of user traffic in the different measurement periods. The number of events found within one bin is given by the color of the pixel. The color bar on the right is in linear scale.



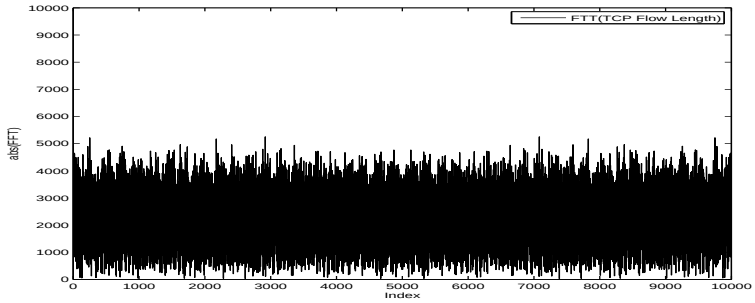
(a) TCP flow length (unchanged)



(b) ACF



(c) Cumulated average



(d) FFT

Figure 5.2: Data validation for TCP flow length.

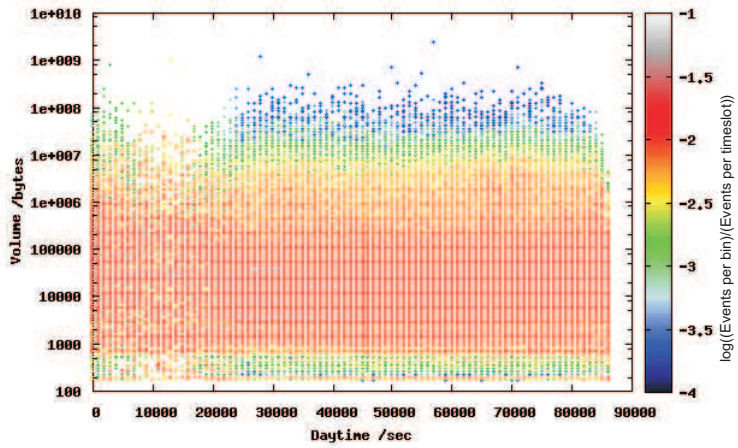


Figure 5.3: TCP flow size in 1000 sec time-bins plotted for one day (UMTS, September 2006 ($\mathtt{TR}_6$)).

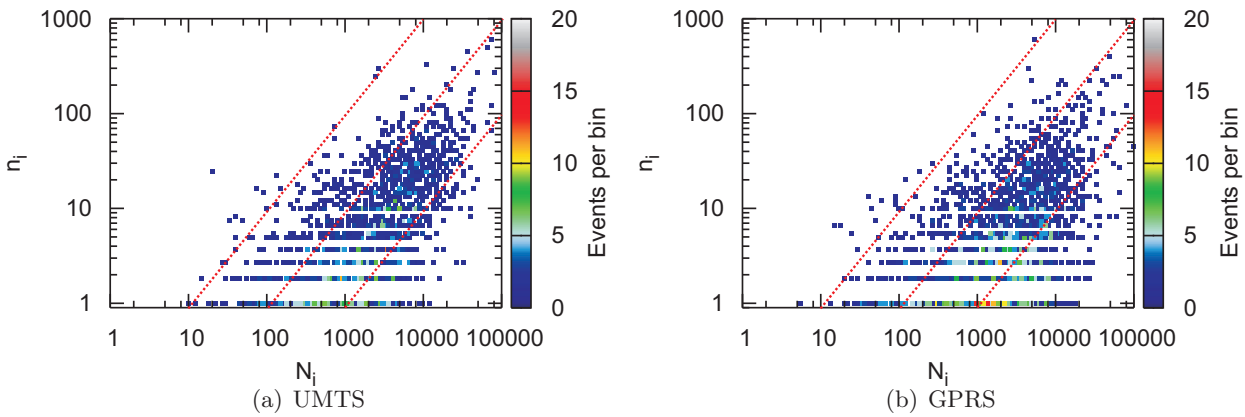


Figure 5.4: Scatterplot of N_i over n_i in the peak hours (log-binning, lin. scale).

Figure 5.4(a) displays the retransmission footprint for UMTS. We observe a correlation between the N_i and n_i, since larger flows tend to have more retransmissions. This corresponds to a device in the communication chain introducing packet loss. Given constant packet loss δ results in a direct relationship of N_i and n_i: $n_i = \delta N_i$. This relation represents a line parallel to the red dotted lines added in Figure 5.4(a).

As in the previous section the results for UMTS and GPRS are very similar (see Fig. 5.4(b)). This is an interesting observation since the two radio technologies have different capacity and usage patterns.

We can conclude that a TCP flow recorded on a mobile network is larger in terms of packet, length and size compared to a TCP flow transferring the same application data in a wired network, as they are biased by retransmissions caused by the nature of a radio link. Therefore results from wired and mobile networks cannot be compared directly.

5.3.4 Scaling Analysis of the Heavy-Tail Parameter

In this paragraph we investigate the flows for heavy-tail behavior. We analyzed TCP, UDP, HTTP, WAP 1.x, and WAP 2.0 flows from the dataset recorded in September 2006 ($\mathtt{TR}_6$) with the modified AEST tool.

In the following figure we depict on the right side the GPRS traces and on the left side the UMTS traces. The first group of Figures 5.5 depict the observations for TCP and UDP flows. The blue crosses on top of the curves are a first sign of a heavy-tail region. A heavy-tail distribution would be entirely marked. If only a part of the distribution is marked this indicates the fact that the underlying distribution has two parts: a body and a (heavy) tail. A very small marked region should not be considered as a presence of heavy-tail behavior [42]. For TCP the estimator delivers a estimated slope, α, close to one, regardless of the technology (Note: the values for α are printed on top of each figure. Here α corresponds to the k parameter of the Pareto distribution defined in Appendix D-5). Comparing the Figures 5.5(a) and 5.5(b) we learn that the tail of the TCP flows in UMTS (left) is shifted half an order of magnitude towards larger values.

In the UDP Figures 5.5(c) and 5.5(d) a much smaller region (see arrow) is marked by the algorithm. We consider these flows not to be heavy-tailed.

The next group of Figures 5.6 depicts the scaling analysis for HTTP flow length. A large part of the distribution, especially the tail, follows the heavy-tail assumption. The parameter α is estimated with $\alpha_{UMTS} = 1.04$, respectively $\alpha_{GPRS} = 1.13$. This is consistent with measurements obtained in the wired network. It is interesting to see that the results for GPRS and UMTS are similar, although the access technology of UMTS provides a higher data rate.

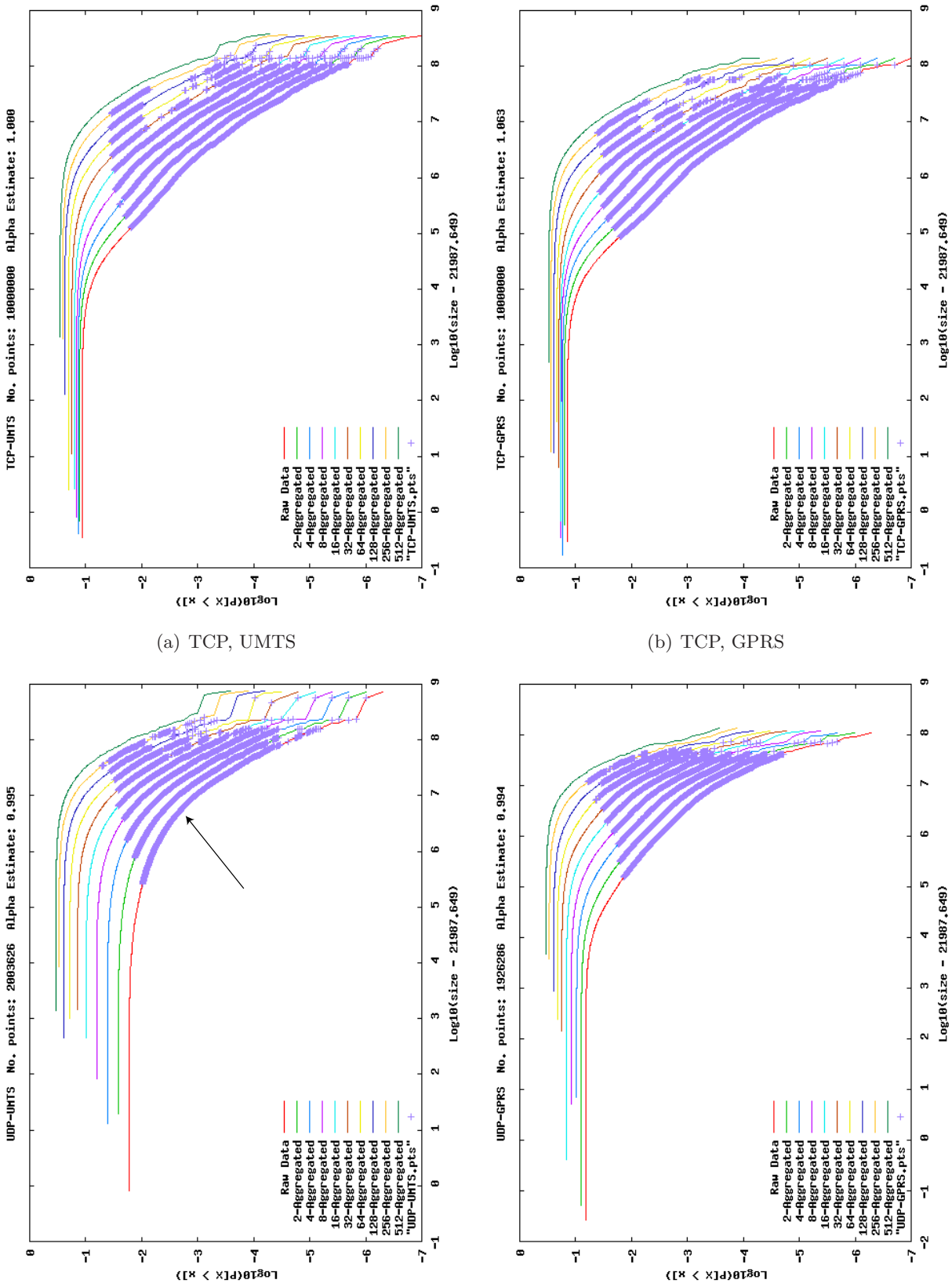
The last service tested was WAP. These results are presented in Fig. 5.7. The regions are scattered along the curve. In such cases Crovella et al. [42] suggested that the distribution may follow a log-normal or Weibull but is not heavy-tailed. The scaling method serves only as an indication for heavy-tails but not as a proof. Visual inspection is used to decide from case to case.

The estimated values for α, representing the Hill estimator, can be found in the title of the graphs. As in the UDP case we consider these distributions free of heavy-tails. From Figure 5.7 we learn that in 2006 WAP access originated mainly from GPRS terminals. Note that due to the small number of WAP terminals [42] the UMTS dataset contains fewer than 4000 events.

5.3.5 Fitting Flow Size and Duration

In this paragraph we search for an analytical distribution which best fits our measured data. First we applied fits for single analytical distributions to keep the modeling simple. In the previous section we identified heavy-tails in the TCP and the HTTP flows. Therefore we split the data for these flows into a body and a tail part. In the literature of wired Internet measurements [82] a split at 10^4 is chosen. However, in our trace the heavy-tail region starts at values that are one order of magnitude larger (10^5). We assume that this comes from the fact that nearly 10 years passed between the measurements in the paper and our traces.

We used the MLE to fit the parameters of the analytical distributions. The MLE solution can often be calculated in one run, this was an important decision criterion as our sample sizes were very large, for example one day in September 2006 had more than $50 \cdot 10^7$ flows. In case one distribution



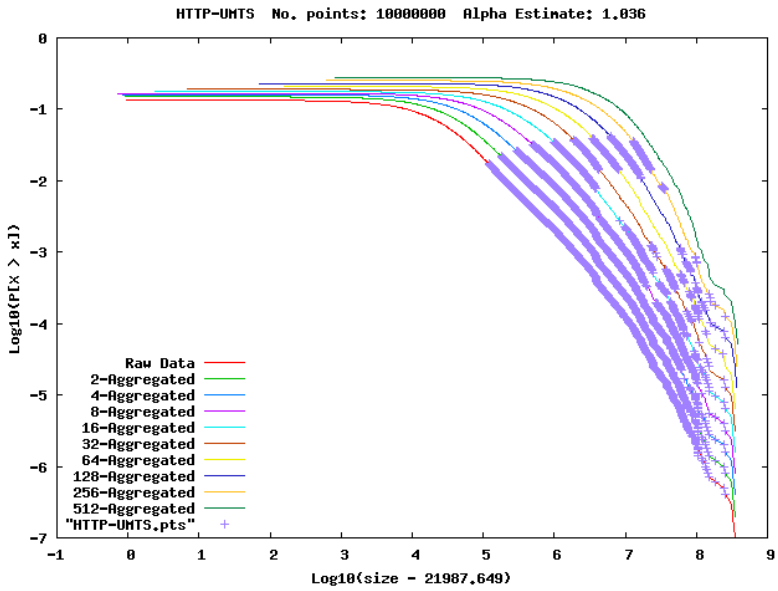
(a) TCP, UMTS

(b) TCP, GPRS

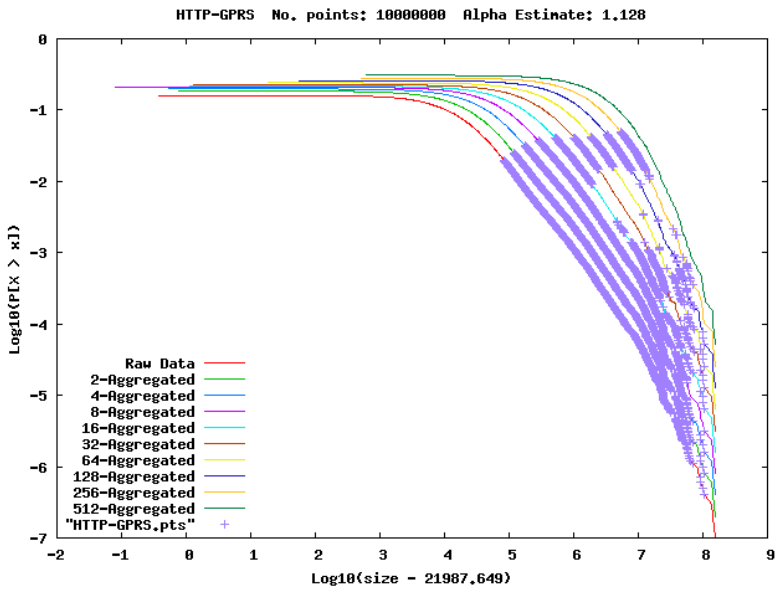
(c) UDP, UMTS

(d) UDP, GPRS

Figure 5.5: Scaling analysis for heavy-tail regions in TCP/UDP flows (September 2006, $\mathtt{TR}_6$).

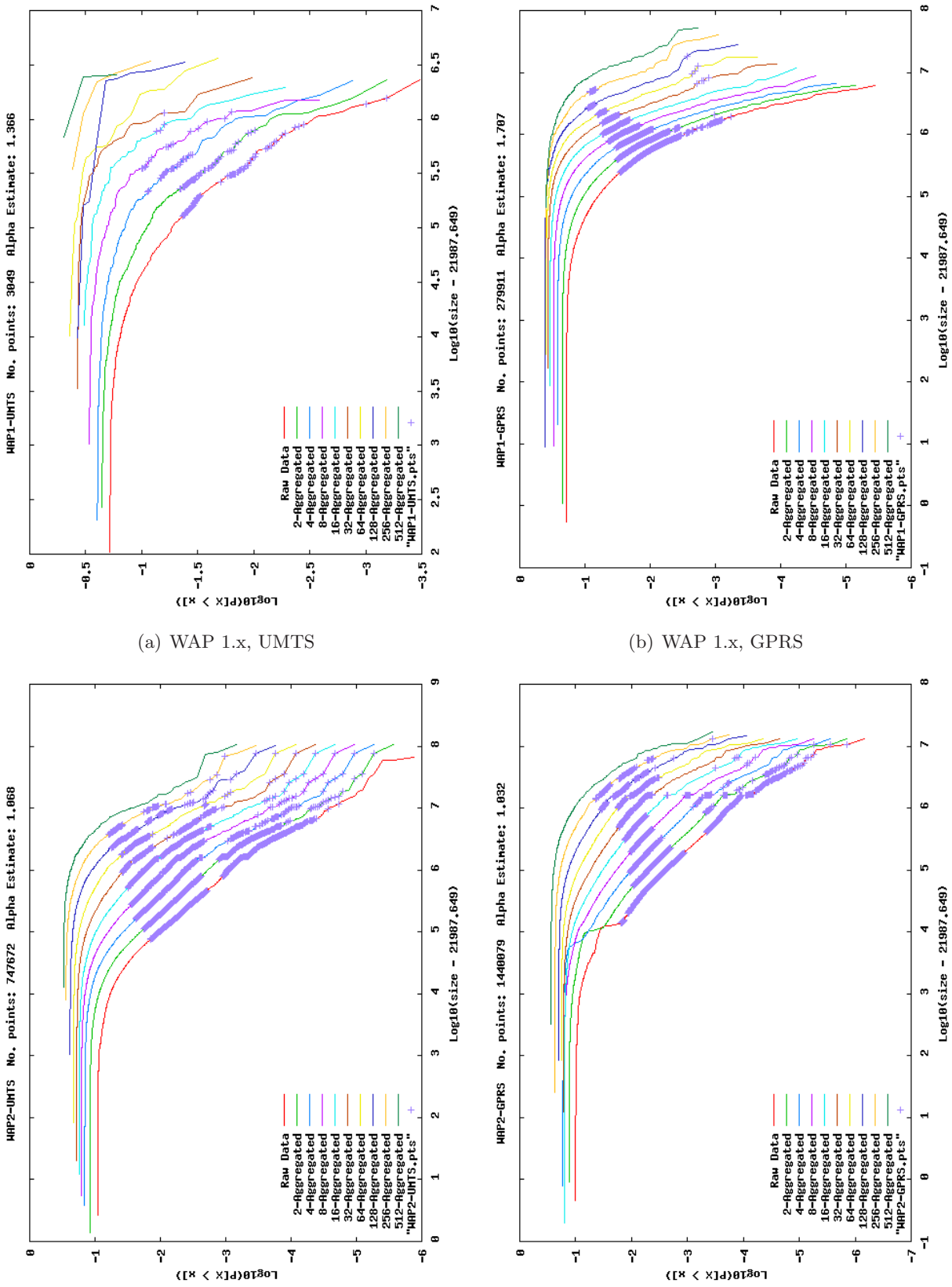


(a) HTTP, UMTS



(b) HTTP, GPRS

Figure 5.6: Scaling analysis for heavy-tail regions in HTTP flows (September 2006, $\mathtt{TR}_6$).



(a) WAP 1.x, UMTS

(b) WAP 1.x, GPRS

(c) WAP 2.0, UMTS

(d) WAP 2.0, GPRS

Figure 5.7: Scaling analysis for heavy-tail regions in WAP 1.0 and 2.0 flows (September 2006, $\mathtt{TR}_6$).

did not fit the eCDF we fitted a phase type distribution. We implemented the EM algorithm in `C++` to achieve a better performance figure.

Table 5.2 lists the best fits for the flows. First of all we have to mention that all analytical fits failed the KS test. None of the MLE solutions could fulfil the critical value of the KS statistics. Hence, we only present the analytical distribution with the best KS score.

Note that we found that the MLE solution could be easily improved by manual adjustments to the parameters. The reason that the method performed so poorly is that the large range which the input values span forces the algorithm to fit the tail rather than the body.

The results in Table 5.2 display the log-normal distribution which fitted best the non heavy-tail datasets, in respect to the body parts. This again is consistent with previous results from Internet measurements.

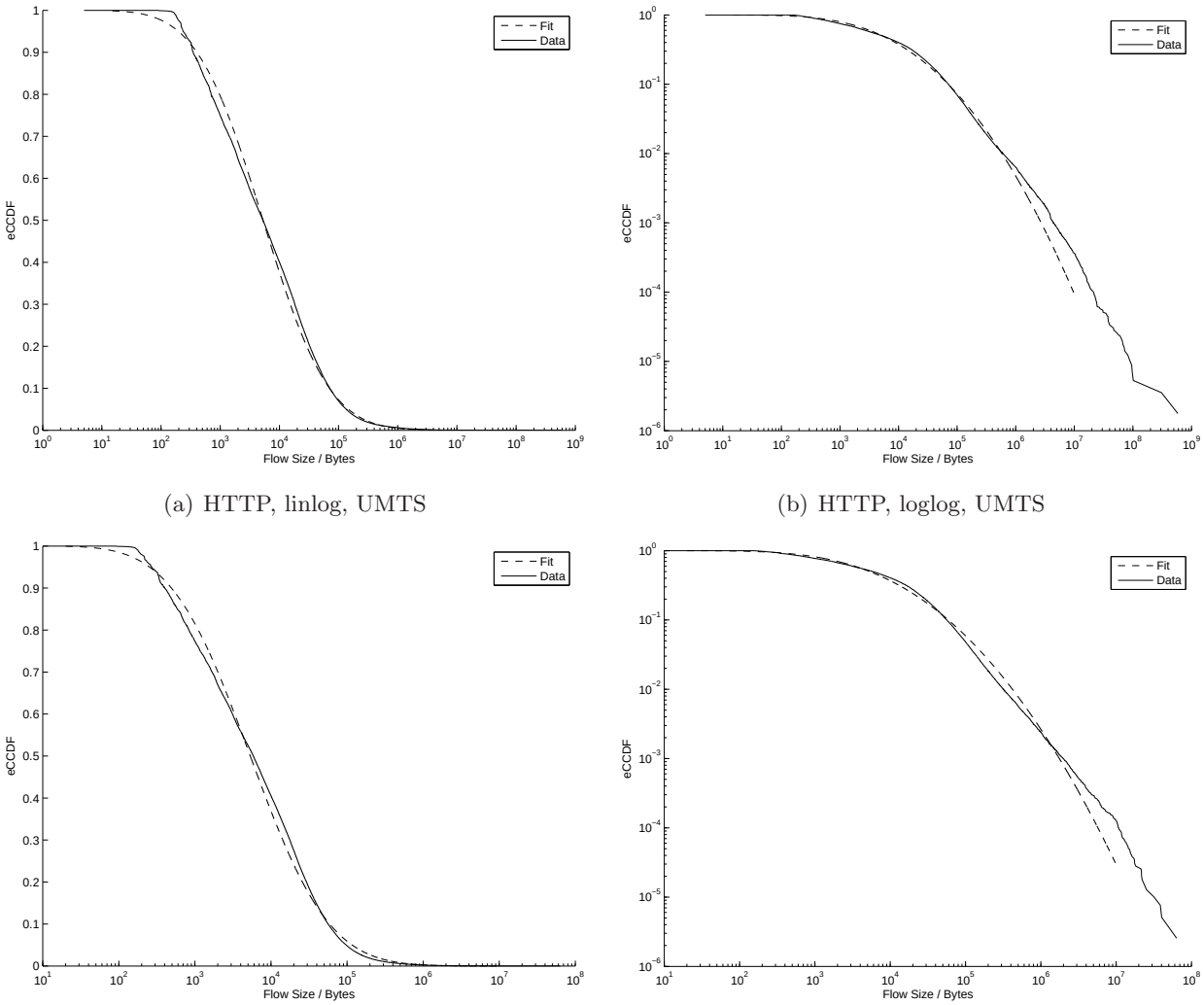
UMTS				
Flow	Distribution	Parameters		KS-stat
TCP tail	Pareto	x_m=100,000	k=1.1070	0.0366
TCP body	log-normal	μ=8.5085	σ=2.0565	0.0913
UDP	log-normal	μ=6.8780	σ=1.8541	0.0832
HTTP tail	Pareto	x_m=100,000	k=1.0890	0.0331
HTTP body	log-normal	μ=8.5739	σ=2.0200	0.0933
WAP 1.x	PH-2	$\lambda_0 = 3.12\ 10^{-4}$	$c_0 = 6.12\ 10^{-1}$	–
		$\lambda_1 = 6.89\ 10^{-5}$	$c_1 = 3.77\ 10^{-1}$	–
WAP 2.x	PH-2	$\lambda_0 = 4.33\ 10^{-4}$	$c_0 = 7.12\ 10^{-1}$	–
		$\lambda_1 = 9.72\ 10^{-5}$	$c_1 = 4.01\ 10^{-1}$	–
GPRS				
Flow	Distribution	Parameters		KS-stat
TCP tail	Pareto	x_m=100,000	k=1.2070	0.0955
TCP body	log-normal	μ=8.2979	σ=1.9436	0.1132
UDP	log-normal	μ=7.1892	σ=1.9011	0.0832
HTTP tail	Pareto	x_m=100,000	k=1.2130	0.0310
HTTP body	log-normal	μ=8.5830	σ=1.8763	0.1132
WAP 1.x	PH-2	$\lambda_0 = 1.33\ 10^{-4}$	$c_0 = 5.34\ 10^{-1}$	–
		$\lambda_1 = 7.68\ 10^{-5}$	$c_1 = 2.12\ 10^{-1}$	–
WAP 2.x	PH-2	$\lambda_0 = 3.44\ 10^{-4}$	$c_0 = 6.89\ 10^{-1}$	–
		$\lambda_1 = 7.41\ 10^{-5}$	$c_1 = 2.99\ 10^{-1}$	–

Table 5.2: Best fits of the empirical distributions in downlink direction (September 2006, $\mathtt{TR}_6$).

HTTP makes up for more than 60% of the TCP traffic, the graphs for HTTP and TCP are nearly identical. Therefore, we investigate solely the HTTP flows. The following block of figures presents the empirical complementary CDF (eCCDF) of the HTTP flows with the log-normal fit added to the graph.

The four plots in Fig. 5.8 display the eCCDFs for HTTP flows. For each technology we plotted

two graphs, one in loglinear which shows the body and one in loglog, which focus on the tail. The loglinear plots (Fig. 5.8(a) and Fig. 5.8(c)) show that the log-normal distribution follows the body part well, if we neglect to take into consideration the lower end of the empirical data. From the loglog plots we can clearly identify that the tail cannot be fitted with a log-normal, which starts to decay much faster than the empirical data at around 10^5. This result is consistent with the estimation of the heavy-tail region we derived with the scaling method of the AEST tool. An interesting finding is the fact that the body part of UMTS and GPRS, Fig. 5.8(a) and Fig. 5.8(c) respectively, are very similar. We conclude that the body part of the distribution is linked to the protocol type rather than to the access speed of the user terminal.



(a) HTTP, linlog, UMTS

(b) HTTP, loglog, UMTS

(c) HTTP, linlog, GPRS

(d) HTTP, loglog, GPRS

Figure 5.8: Fitting the body of the HTTP flows.

5.3.6 Mice and Elephants in Traffic Flows

In Section 5.3.4 we showed that the TCP flows in our trace are heavy-tailed. A side effect of heavy-tailed distribution is the presence of so called mice (small flows) and elephants (very large flows).

Many papers discuss this effect, see [83] for further references.

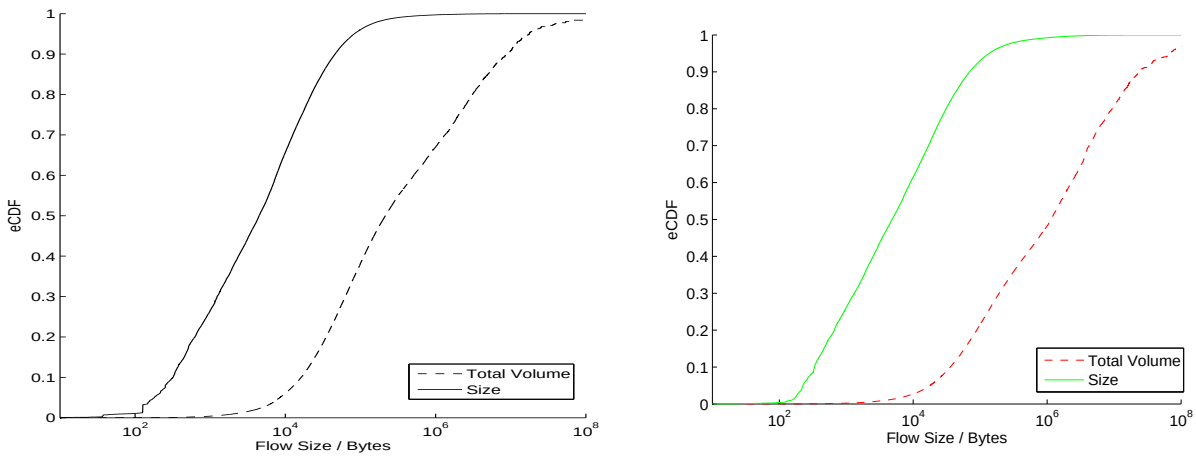
The mice generate the large part of the connections. However, the elephants generate the traffic in the network. The presence of elephant flows allows a strong simplification in simulation setups. First only a small number of TCP connections have to be simulated and secondly these connections are long and therefore boundary effects (e.g. slow start) can be omitted.

Figure 5.9 displays the effect for the UMTS and the GPRS networks. Both plots show two CDFs. The *Size* curve is the eCDF of the flow sizes of the TCP connections. The *Total Volume* curve depicts the volume generated by all flows less or equal to x.

The UMTS CDF curve of the flows reveals that 99% of them is below 94.62 kByte. However, the remaining 1% generates 63.2% of the total volume. In GPRS the 99% limit is 76.11 kByte. Here the remaining part accounts for 48% of the traffic. UMTS and GPRS have about the same upper bound for mice flows. Due to the higher data rate the elephants are larger in UMTS.

In the WAP 1.x and 2.0 flows the elephants have a smaller impact of 22% and 19% respectively. In this application the limited capability of the terminals do not allow for very large flows.

As the distribution of the flow sizes of HTTP is very similar to the ones of TCP the results are also valid for HTTP flows. In a high RTT environment we cannot neglect the mice flows. A simulator will have to consider effects like the slow start.



(a) UMTS

(b) GPRS

Figure 5.9: TCP flow size and originating volume.

5.4 Summary

IN this chapter we investigated the TCP/UDP flows found in our traces. A first analysis showed that the vast majority are TCP flows and only about 10% of the flows are UDP. This result is consistent with the access technologies and in time.

In the following investigations the results for GPRS and UMTS show no significant difference with each other. The independence of the flow patterns from the access technology to which the subscriber was attached is one of the main findings in this chapter.

The basic analysis of the dataset revealed that most of the TCP flows suffer from retransmission. This effect biases the flow size of TCP in mobile environments.

The TCP flows and as a main contributer the HTTP flows had signs of heavy-tailed behavior. To fit these flows we split them into a body and a tail part. The log-normal distribution provided the best fit for the body part which is also consistent with the literature.

The UDP and WAP flows instead showed no clear sign of heavy-tailed behavior. The log-normal distribution fitted best the UDP flows. The empirical distribution of the WAP flows had several discrete steps. Therefore we fitted a phase type distribution to them.

Concluding this chapter we can say that many flow properties of the mobile Internet accesses are identical to results found in wired measurements. However, there are some important differences, e.g., the presence of a non negligible level of TCP retransmissions.

Related Work: Beside publications from our research group, we found only one further paper reporting on flow length of HTTP and WAP session. In [84] Dahmouni et al. report on measurements from a GPRS live network. The figures published in this paper show flows which are one order of magnitude smaller and do not show signs of heavy-tailed behavior. However, the sample size was maybe too small.

Self Criticism: In the paragraph in which we fitted size and duration of the flows we stated that we had problems fitting the curves automatically. Often manual adaptation provided better fits. The main improvement came from visual inspection, manual separation of the tail and body parts and the usage of computer added search for the best parameters. We think that this part could have been improved by a better theoretical analysis.

Chapter 6

Evolved and Extended Service Traffic Models

Contents

IN this chapter we present application specific traffic models. Network traffic is obviously driven by users interacting with applications. Therefore these models emulate the user behavior at the application layer of the network model.

The first Section 6.1 gives a motivation why we try to model user behavior on the application level.

Section 6.2 presents new parameters for HTTP models. We use a model from Choi et al. and fit the parameters according to our measurements. The same approach is followed in Section 6.3 for the FTP model.

Section 6.4 presents two different models for e-mail traffic. For POP3 we propose a new model that extends older models with a login process as an important add-on for high delay networks. Secondly, we present a SMTP model similar to FTP.

The chapter is concluded with Section 6.5 presenting a summary of the traffic models.

6.1 Motivation

IN teletraffic engineering we can chose different layers of the protocol stack in which we want to model our traffic. Moving to lower layers reduces the parameters we have to model, e.g., the Medium Access Control (MAC) layer can be modeled by the packet size and rate. However, such models cannot answer questions that are related to the application layer of the network, e.g., how does the traffic change if a bottleneck is present?

Due to the small number of parameters such models are favored in simulation scenarios with many nodes. We presented such an approach in Chapter 5 where we investigated the distribution of the TCP/UDP flows. Simulating the traffic on top of TCP/UDP flows allows simulation scenarios that investigate interaction at layer three of the OSI model, e.g., TCP retransmissions or TCP RTT change in the presence of a bottleneck [20].

However, interaction at the user level can only be partly mapped onto TCP/UDP properties. For example it is not possible to simulate changes in the application such as an increase of the mean webpage size. These events can only be evaluated with a model which acts at the application level. Figure 6.1 displays three different layers to simulate the traffic.

In this chapter we present models for the top services observed in our network. For applications where appropriate models exist (e.g. HTTP and FTP) we fitted the parameters according to our mobile network. In case the models did not meet our needs we improved them (e.g. POP3) or for new applications such as online gaming introduced new traffic models.

6.1.1 Advantage of Traces from a Cellular Mobile Network

Constructing a traffic model on the application level is often difficult. The main drawback to such models is the fact that some parameters of the models have to be extracted on a per user basis. For example we know from the service mix that there are 10 MByte of FTP traffic per hour and that there are 10,000 subscribers in the network. If we had no knowledge of the number of subscribers accessing FTP services, the resulting average FTP traffic per user would be 1 kByte per hour. Although this example exaggerates the problem, a good application model needs additional information how the

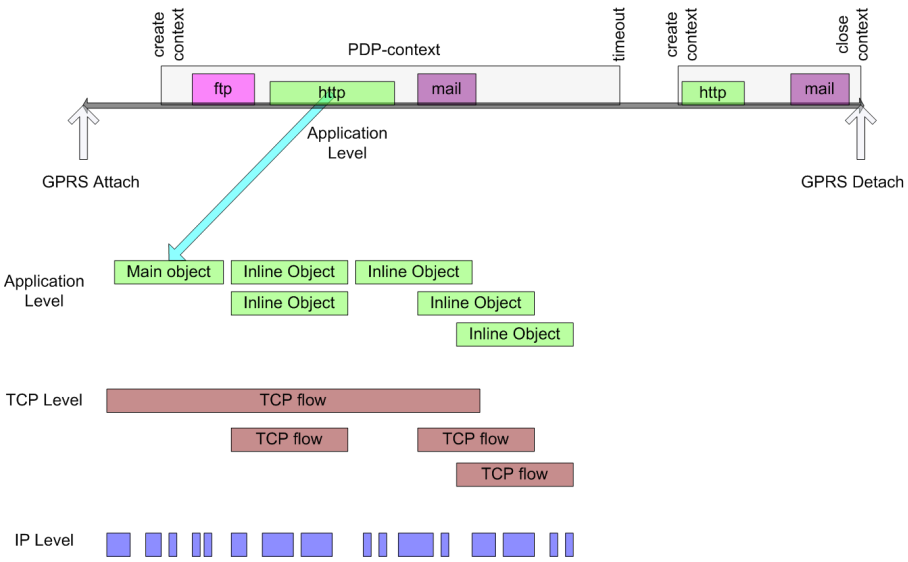
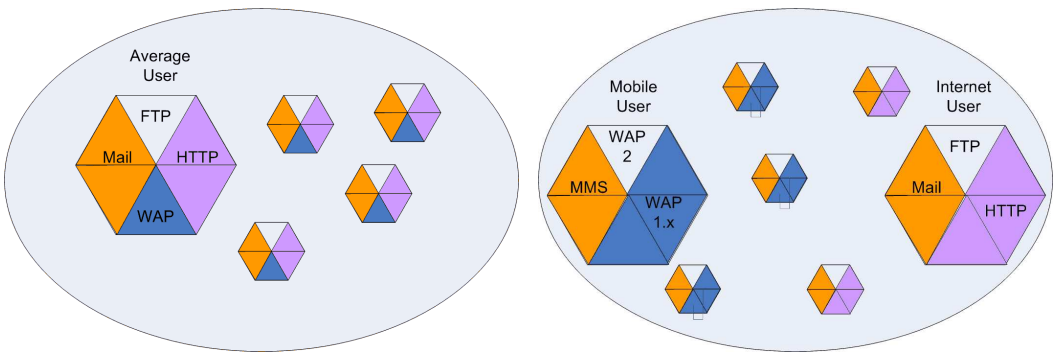


Figure 6.1: Traffic modeling on different layers of the network.

service usage is distributed among the user population. Figure 6.2(a) illustrates the idea of an average user. We have shown in Section 4.2 that the service usage per user can be divided into three large groups: WAP only, WAP & E-Mail, and Internet Services (no WAP). Figure 6.2(b) visualizes this approach.

There are two ways to collect these parameters: a modified application software logging the data or an Internet access where each user has to authenticate himself. A measurement based approach on a modified application software is limited due to the limited number of participants. However, in our case every IP packet, in fact every GTP packet, can be assigned to the subscriber. Therefore we are able to track and extract the parameters on a user base. We therefore also investigate the distribution of the service call rate for the different traffic sources.



(a) Identical service mix for all users

(b) Individual service mix for each users

Figure 6.2: HTTP-model measurements.

6.2 Modeling HTTP Browsing Sessions for the Mobile Internet Access

THE Hypertext Transfer Protocol (HTTP) is a protocol to transmit data at the application layer between hosts. The protocol was designed in 1989 by Tim Berners-Lee at CERN in combination with the Uniform Resource Locator (URL) and the Hypertext Markup Language (HTML). It is a communication scheme to transmit data units which are parts of websites in the WWW and defined in the RFC 2616 [85]. HTTP is a stateless protocol allowing asynchronous connections between client and server. It needs a reliable connection to transmit data. Mainly TCP is used for this purpose although it can run on other reliable protocols too.

The data on the server side is encoded using a markup text format called HTML or eXtended Markup Language (XML). The URL interconnects the objects embedded within one webpage.

The basic operation between two hosts follows a request/response scheme (see Figure 6.3). The client establishes a connection with the server and sends a request to the server (Get) including additional information about protocol version, client capabilities, and modifiers. The server responds with a message (Reply) about his status including server capabilities, possible error codes, page content and the main object, which is the main webpage including all URLs necessary to fetch all embedded objects on this page. If there were embedded URLs in the main object, the client starts another Get message subsequently for all objects on the page. This process stops in case the last embedded object has been downloaded by the client and the page transfer is now complete.

Up to now there exist two different versions of HTTP: 1.0 and 1.1, the later is a down-compatible extension of the previous version. In HTTP 1.0 the client/server relation can only set up separate connections for every request. In this case a client creates a new TCP connection for each object request separately, e.g., images on the page, closing it right after the response was received. The TCP protocol is not optimized for this kind of data transfer. In fact the slow start mechanism will harm the performance of such protocols. As one of the main advantages HTTP 1.1 offers so called persistent connections. In such a connection the client can request multiple objects at once. Hence, there is only one open TCP connection for the whole webpage. The drawback to this method is the fact that there are more open connections that must be handled by the servers.

The second performance improvement in HTTP 1.1. is pipelining. This feature enables the client to request multiple objects without waiting for the response from the server. In combination with the persistent connection this feature fills the available resources much more efficiently.

In general the HTTP header may hold optional information not standardized, which allows special applications to implement modified data communications.

6.2.1 HTTP Traffic Model

We expected the HTTP model parameters to be different for UMTS and GPRS. Therefore, we collected the values separately. To extract the application level data we used the `tcptrace` tool [86] with the add-on HTTP-parser. Although the huge size of our traces sometimes led to program crashes, we were able to extract a useful number of user data parsed. A shell script parsed the HTTP data and

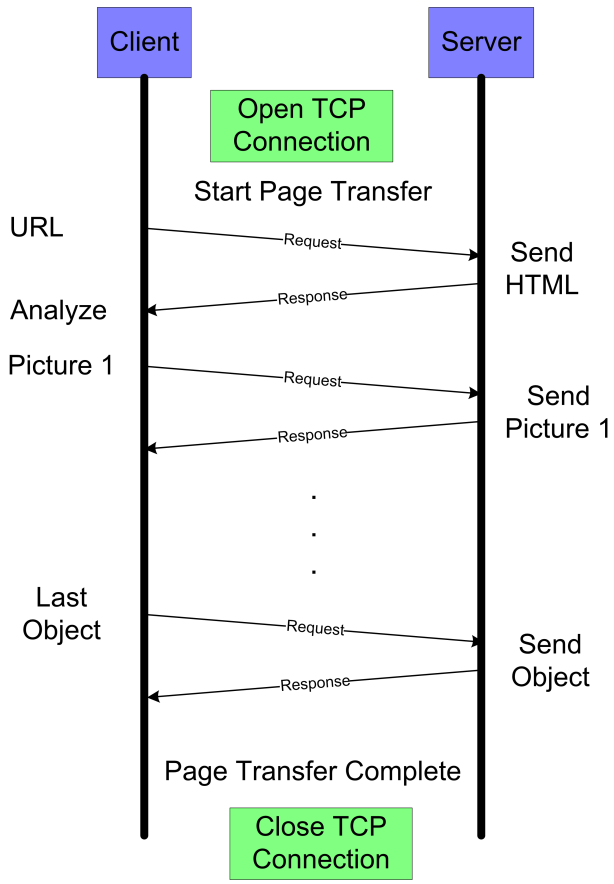


Figure 6.3: HTTP transfer of a webpage (persistent connection, HTTP 1.1).

dumped the session parameters: reading time, objects per page, pages per session, object size, page size, download time and so on. The results were then fitted and plotted. In the following sections we show the results for UMTS and GPRS separately to distinguish the RAN settings.

Related works like in the thesis and the publications of P. Stuckmann [87, 88] on this area reuse the typical traffic models, with the standard parameters, from [8, 9, 89, 90]. Therefore, we directly compare our measurements with these models.

In this section we applied a filter rule at the user level. Users that generated less than 100 kByte over the whole trace and the top 1% were removed from the dataset. The following figures (Fig. 6.4(a) and 6.4(b)) show the eCDFs for the page and the object parameters extracted from the traces. They show a solid curve for GPRS and a dashed line for UMTS and the dotted lines are the fits obtained for the parameters.

Comparing these figures we can reveal two facts concerning the different access technologies. First, UMTS users tend to visit larger HTTP pages (mean in UMTS $\approx$ 6.2 `kByte` vs. mean in GPRS $\approx$ 2.1 `kByte`) - see Fig. 6.4 (a.2), than GPRS users. This can also be seen in the eCDF of the object sizes in one page (Fig. 6.4 (b.2)). Secondly, the download time of the UMTS HTTP pages is the same as for the GPRS pages. This point is important because it could indicate a hard limit on the user side, e.g., a maximum tolerance time a user will wait for a page. The RAN in UMTS is approximately three times faster than in GPRS, this manifests itself in a measured difference of the mean HTTP page size.

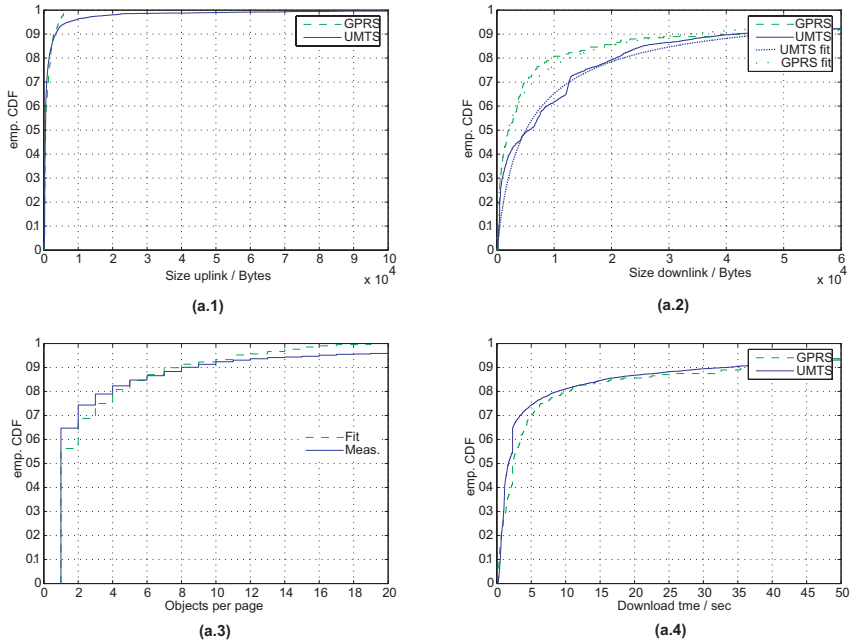
Figure 6.4.a shows the number of objects per page as an eCDF. It is interesting to see that there are over 50% of requests resulting in a page with only one object. The literature [91] proposes a gamma distribution with mean 2.5 for the object per page, which is higher than we have measured. To clarify this misfit we looked into the raw traces. We found out that many automatic program updates. For example Anti-virus or Windows updates were using the HTTP protocol. These programs accessed only one big object - the update file. In addition to this many HTTP requests originated from cache updates in the network. We can only guess on this topic as, in order to protect the privacy of the subscribers, we are not allowed to investigate the data at full length.

Figure 6.5 displays the parameter reading- or thinking-time. We define the reading-time as the timespan between the arrival of the last packet of the previous page and the first packet of the new page. Comparing UMTS and GPRS we found out that this parameter does not depend on the radio access network. The values in Figure 6.5, which were fitted with an MLE and some manual fine tuning, depict similar results as found in [8]. The curve has two regions a steep start indicating fast hopping from page to page and a long tail for the largest 20%. Only every 5th reading-time is above 18sec, therefore we conclude that users read less than 20% of the pages to the full extent.

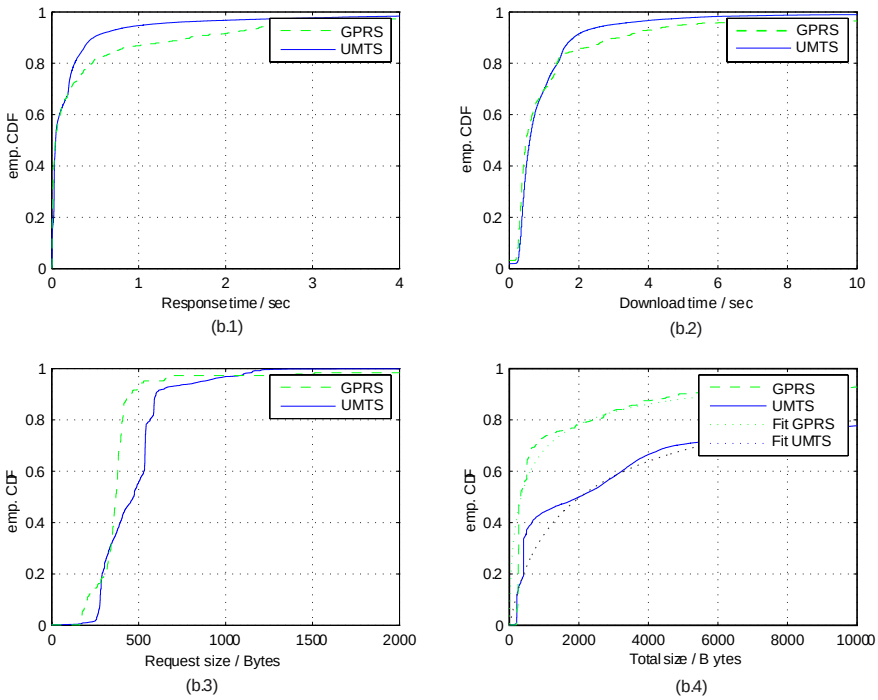
6.2.1.1 Fitting the Model Parameters

In this paragraph we present the fittings for our dataset and compare it to the distributions found in the literature [8],[24]. Figure 6.6 depicts the model of an HTTP application. The page is constructed from one main object and several inline objects (e.g images). After the transmission the user will read the page indicated by the reading time between two pages.

For this evaluation we analyzed HTTP traces of about 1,000 users, part of the $\mathtt{TR}_1$. Due to the



(a) Page level parameters



(b) Object level parameters

Figure 6.4: HTTP-model measurements and their best fits.

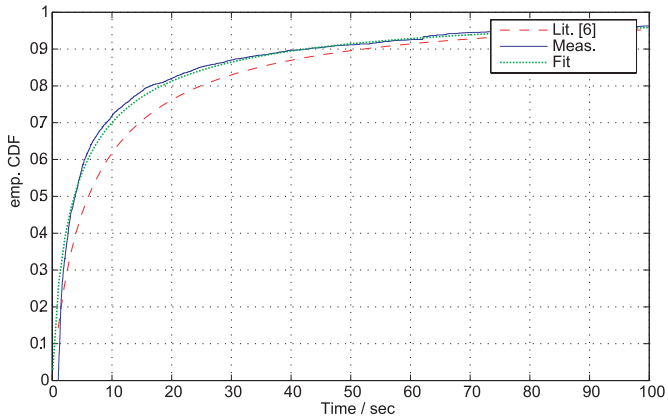


Figure 6.5: HTTP reading time.

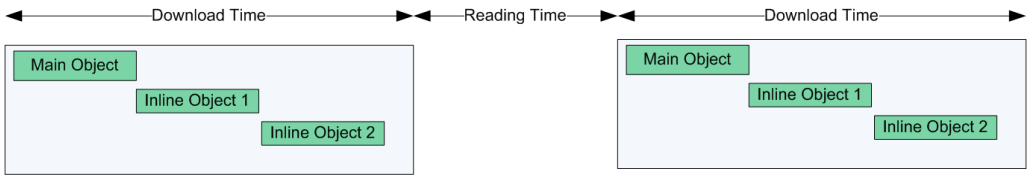


Figure 6.6: HTTP browsing session.

small size of our dataset there would for sure be no heavy-tailed behavior visible. Note that the observation of a heavy-tailed behavior needs a large dataset, e.g., if we want to detect events with a probability of below 10^{-6} we would need at least 10^6 samples. Therefore we fitted the parameters to different distributions and chose the best fit based on the KS-statistic as we did in Chapter 5.

Table 6.1 displays the best fits for the parameters of the HTTP model for UMTS and GPRS. Here, in contrast to the results in Section 5.3.5 we see a clear difference between the fitted parameters for UMTS respectively GPRS. As this special analysis is based on data from Fall 2004, a time when UMTS was deployed only for a few months, we conjecture, based on the results from Section 5.3.5 that for actual results the parameters for UMTS and GPRS are identical.

The fit for the reading-time parameter delivers smaller results than found in [89] in both technologies. We assume that multi tabbed browsing, one browser transfers several pages in parallel, affects this parameter toward lower values. Although tabbed browsing was not widely in use at that time, we have to note the fact that this analysis was done for the top users in the network, not for an average or sporadic web browsing customer. To verify the numbers we extended the metric `Gn-duration` so that it reports page requests starting in parallel to active page downloads. Visual inspection showed a similar result from this data source. For both technologies the best fit was achieved with a log-normal distribution.

The size of the main object and the object size also followed a log-normal distribution. The GPRS and UMTS fits for the object size have a large difference. We assume that UMTS subscribers tend to download larger objects within the HTTP pages visited, e.g., embedded streams or images.

In the following section we focus on the difference of our measured distributions compared to ones from the literature.

UMTS			
Model Parameter	Distribution	Parameters	
		μ	σ
Reading Time	log-normal	1.620	1.501
Size Main Object	log-normal	8.530	1.743
Object Size	log-normal	5.830	2.270
Number of Objects	gamma	5.614	213.4
GPRS			
Model Parameter	Distribution	Parameters	
		μ	σ
Reading Time	log-normal	1.631	1.544
Size Main Object	log-normal	7.600	2.194
Object Size	log-normal	7.620	1.503
Number of Objects	gamma	5.321	246.2

Table 6.1: HTTP parameter fittings.

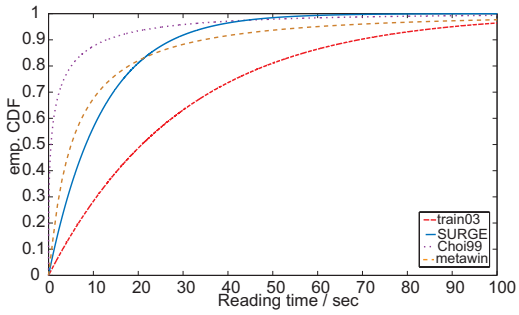
6.2.1.2 Comparing the different parameters

In this paragraph we investigate how the parameters of the various models differ in detail. In addition we added some information taken from the top 1,000 sites in the Netbooster log-file extracted with the IE_Control tools [92, 26]. The Netbooter is an HTTP caching device located at the Gi interface. Modifying the IE_Control tool delivered the following information: number of inline objects and size of main object. Figure 6.7 shows the reading time (a), the size of the main object (b) and the number of inline objects (c) for the different models.

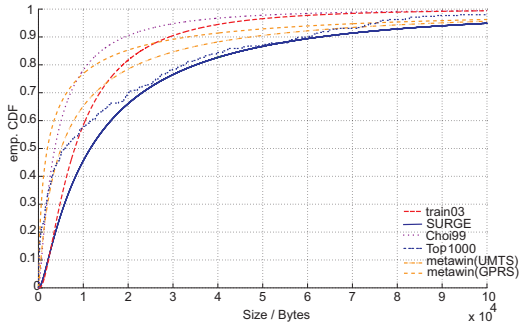
From Figure 6.7(a) we can conclude that the reading time in the UMTS network behaves like the SURGE model. The HTTP model from Choi and the model in the 802.20 standard do not fit the dataset. If only the size of the main object, depicted in Fig. 6.7(b), is considered, the measurements are following the model of Choi. The best fit for the parameter of the SURGE model still showed a big error when compared to the empirical measured data. We assume that the distribution does not fit our measured data set.

6.3 Modeling FTP Sessions in a Mobile Network

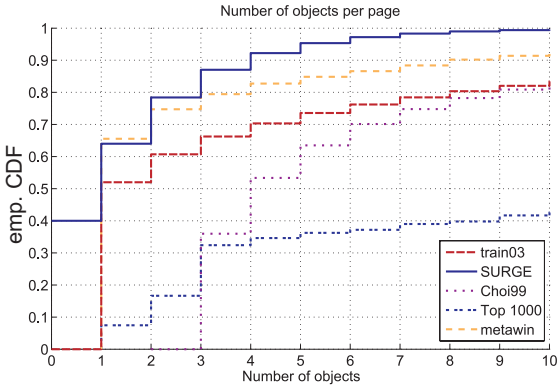
THE File Transfer Protocol (FTP) allows bulk transfers of large files through the Internet. FTP is a network protocol defined in the RFC 959 [93]. The RFC states the following objectives for the use of FTP: sharing of files, indirect or implicit use of remote computers, hide variations in file storage systems from the user, transfering data reliably and efficiently. It supports exchange of files over any TCP/IP based network as well as manipulations of files at the server, e.g., deleting or listing files. It uses exclusively TCP. An FTP server will by default listen at port 21 for incoming connection



(a) Reading time



(b) Size of the main object



(c) Number of inline objects

Figure 6.7: Comparison of HTTP-model parameters.

attempts of clients. The client is granted access to the file system of the remote host in an operating system style manner. It is possible to down- respectively upload several files within one session.

FTP implements out-of-band control, thus data and signaling is transmitted over different connections (see Figure 6.8). An FTP client will access the server at port 21 to browse the file system. The download itself will take place in a second TCP connection. In active mode the FTP client opens a port, randomly chosen from above 1023, and sends the FTP server this number waiting for the server to initiate the data transfer. The server will open port 20 and connect to the client port starting the data transfer.

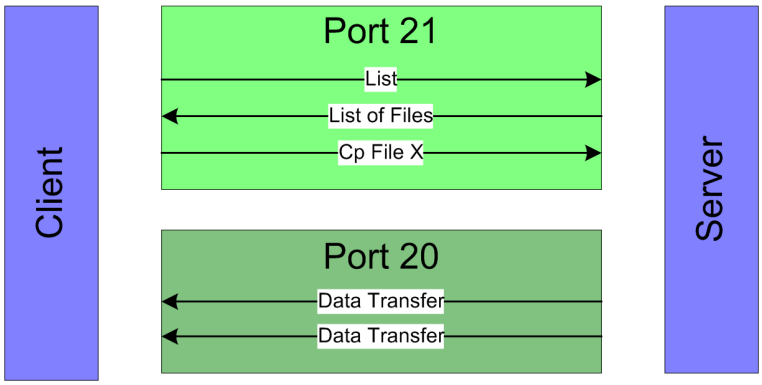


Figure 6.8: FTP file transaction in active mode.

FTP allows for two different data formats: American Standard Code for Information Interchange (ASCII) and Binary. In the binary mode a transaction takes place bit by bit. This mode is used for program binaries or data that is not plain text. For plain text files the FTP application supports ASCII transfers. In this mode the ASCII code of a symbol is transmitted between server and client. This enables the client to translate text to the local ASCII settings, e.g., replacing an end of line code from a Unix server with the symbol for Windows computers.

The original FTP definition has some drawbacks such as clear text passwords, high latency and no integrity checking. Today, in 2007, FTP transfers are largely replaced by HTTP, e.g., web-mail applications contain parts of HTTP and FTP services.

6.3.1 Modeling FTP Sessions

The data transfer of an FTP session can be interpreted as a single large TCP flow. Therefore, we propose a very simple model for FTP transfers which is depicted in Figure 6.9. The application has two states corresponding to *on* and *off*. In the *off* state the application is inactive and remains until a timeout expires. After the timeout the application enters the *on* state. In this state a file transfer via TCP is initiated, the size of the file is derived from a distribution fitted to the empirical data we extracted from our traces. Such a model can be found in many papers [37, 15].

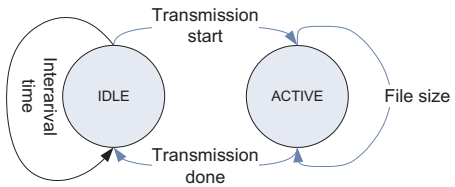


Figure 6.9: State diagram for an FTP traffic model.

6.3.2 Fitting the Parameters

The FTP model we propose has the following parameters: file size, service calls per user and number of users. We used the traces from September 2006 to evaluate the parameters. The best MLE fit for the file size was log-normal ($\mu = 8.5534$, $\sigma = 2.112$). This is a similar result as we found for the TCP connection in Section 5.3.5.

A scaling analysis revealed no clear signs of a heavy-tailed behavior. This result is in contrast to the literature [87]. However, the FTP share in the network is very small and therefore the sample size is not significant enough. Note that the resolution of a empirical generated CDF cannot exceed the reciprocal of the events in the sample, e.g., given 10,000 measured events the smallest step, equal to a unique event x_{unique} at a specific value, is $P(x_{unique} = 10^{-4})$.

The share of FTP users among the subscribers was 2.43%. The service call pattern per user is depicted in Fig. 6.10. Again the service call distribution is independent from the access technology. The best MLE fit for both curves is an exponential distribution with $\mu = 6.542$.

Summarizing the FTP model we have found out that only a small part of the subscribers access this service and that the file sizes we obtained did not show signs of heavy-tails.

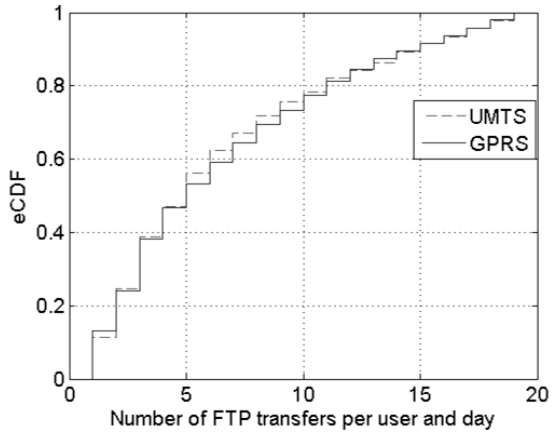


Figure 6.10: FTP service calls per user and day.

6.4 E-Mail Traffic Model: An Extension to High Delay Networks

IN this section we introduce a traffic model for the e-mail service POP3 in a mobile environment. As we have shown in Chapter 4, this service is in the top three of the service share. Although it generates smaller traffic loads than web browsing does, most business customers rely upon a proper function of this service.

In contrast to wire line connections 3G networks impose a high delay in their links. Therefore, a proper modeling of this service needs a different approach than in the low delay case. In a low delay scenario a login process can be omitted and the e-mail transfer is reduced to some kind of push service only.

The trace $\mathtt{TR}_5$ used in this section consists of four different one-day periods in April of 2007 which were recorded on the Gn Interface.

Measurements in a live network are likely to encounter an impact generated from some heavy hitters. So called heavy hitters are users generating the main amount of traffic. Normally, this user group holds only a small part of the population. However, these users still characterize many measured PDFs. Also in this case we found out that more than 20% of all e-mail requests originated from less than 0.25% of the users. We filtered out this group in a preprocessing step.

The detection was implemented by marking all IMSIs as “bad” which had more than 500 requests in six hours in any of the traces. This limit is according to public rules for DNS blacklists. More than 99% of the users had less then 80 requests in the full week.

This rule excluded only less than 0.32% of the users from the traces, but astonishing more than 25% of the e-mails.

6.4.1 E-Mail Protocols of the Internet

In the Internet the following three protocols are used to transmit most of the e-mail traffic: SMTP (Simple Mail Transfer Protocol) [94], POP3 (Post Office Protocol v.3) [95] and IMAP (Internet Message Access Protocol) [96]. SMTP is a protocol to send e-mails between local systems. The protocols

POP3 and IMAP were designed for clients who only sporadically connect to the e-mail server. E-mails are stored on the server and transfered to the client on demand. All protocols rely on the TCP protocol for an error free end-to-end delivery.

6.4.1.1 Simple Mail Transfer Protocol (SMTP)

SMTP is a text based, simple protocol which can send a message and other objects to one or more recipients. The user's client will look up his SMTP server in the local configuration and deliver the message to this server. An SMTP server, by default, listens on port 25 for incoming connections. TCP is used underneath as a transport protocol. The first SMTP server will then look up the e-*M*ail e*X*change (MS) DNS (Domain Name Service) entry of each recipient's domain name and will then forward the message to the SMTP server of the recipient. SMTP only features push deliveries of e-mail. To pull messages from a server the recipient has to use protocols like POP3 or IMAP. Figure 6.11 depicts this process.

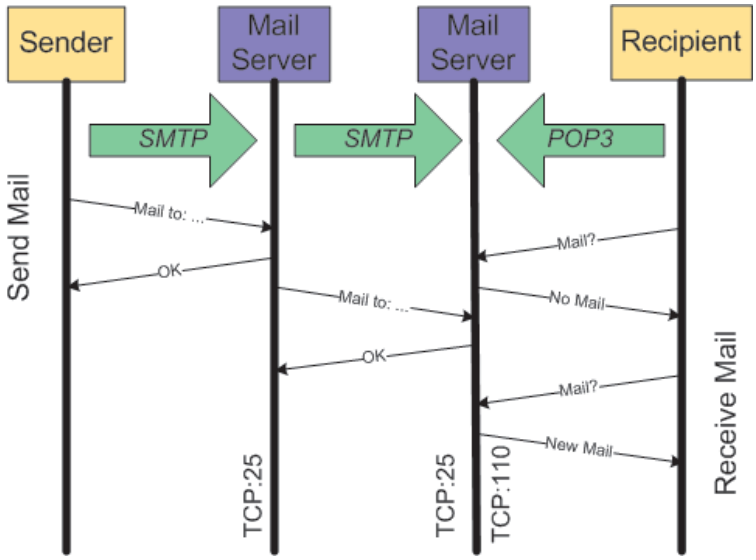


Figure 6.11: Mail transport between two users.

The original SMTP has no authentication functions for the sender as every user could use any SMTP server to send his messages. Therefore, spam e-mails are a massive problem for this server. There are some extensions to the original SMTP standard, such as STMP-AUTH. However, they are impractical and therefore not implemented. Some operators only allow their users to access the SMTP server to minimize the problem of spam attacks.

Although there are strong shortcomings in this protocol there is no real competitor for delivering e-mail through the Internet.

6.4.1.2 Post Office Protocol v.3 (POP3)

The first version of the POP was defined in the RFC 918[97] in 1984. Today, in 2007, clients implement the third version of POP, called POP3, which is defined in the RFC 1939 [95]. The protocol allows for end users with dial-up connections to retrieve e-mails from servers (see Fig. 6.11). POP3 definition

intends that the user downloads all available messages at once and deletes them from the server hereafter. There is an option to leave messages stored on the server. However, most POP3 commands identify the messages by their number only. That causes problems if someone deletes an e-mail manually at the server. To avoid this the Unique Identification Listing (UID) command is used. This command adds a unique ID to the message at the server side. Now the client can distinguish new from old e-mails.

In contrast to SMTP, the POP3 application needs authentication. A standard server listens on TCP port 110 for incoming connections. After a session the TCP connection is closed, several e-mails may be downloaded with the same connection.

6.4.1.3 Internet Message Access Protocol (IMAP)

The basic operation of the IMAP is similar to POP3. It is an application layer protocol. The server operates on TCP port 143. The current version of IMAP is IMAPv4r1, it is defined in the RFC 3591 [96, 98].

It offers more sophisticated functions to manipulated e-mails. While POP3 was designed to download e-mails from the server and read them offline, IMAP also supports an online mode, which leaves messages on the server until the user deletes them. IMAP supports connected and disconnected operations, offering more flexibility. In POP3 only one client at a time can be connected to a mailbox, while IMAP supports simultaneous interaction of multiple clients to the same mailbox. In addition to this, IMAP provides message state information. Within POP3 this information is locally generated by client interaction. Some e-mail manipulation can be executed directly on the server side (e.g. searching for keywords).

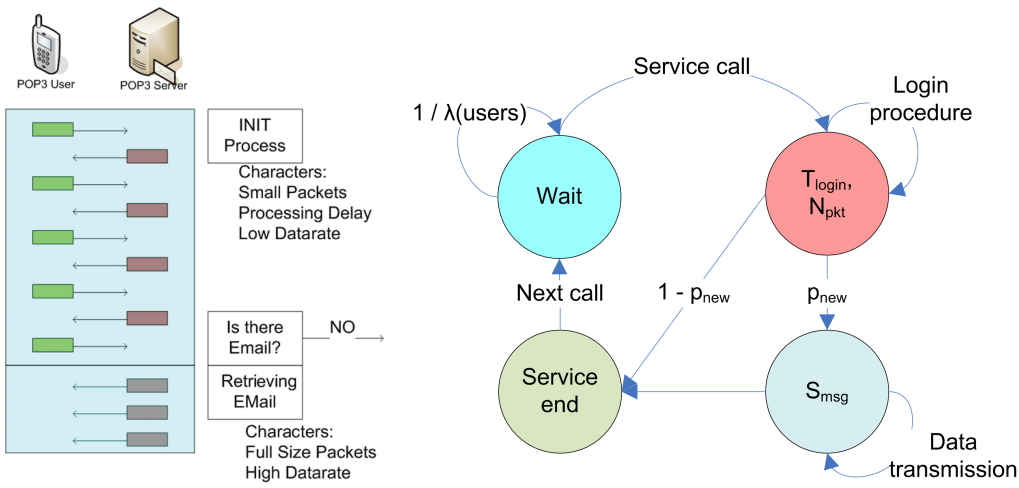
In spite of the advantages of IMAP there are also several drawbacks. IMAP is much more complex than POP3 and users may consume more resources on the server side (e.g. via a message search).

6.4.2 A POP3 E-Mail Model for high RTT Networks

In the literature e-mail models cover only the e-mail message size [99, 24]. However, in order to receive his e-mail a user has to authenticate his account with his password. This login process transfers only a small amount of data and is neglected in common models. We decided to use our measurement data to build up a model for e-mail traffic better suited for mobile internet access technologies. Fig. 6.12(a) gives the principle idea behind these thoughts. The goal is to separate the login procedure from the e-mail download itself. This has two advantages: first, we can model the different payloads on the IP-level separately (login = small payload, download = large payload) and secondly, we can make use of the model to estimate a Mean Opinion Score (MOS) value for this service. To estimate the MOS it is important that the login process is a well defined procedure that can be mapped to a number of small packets traveling in up- and down direction. A high RTT, like in GPRS, will then result in a lower service experience.

6.4.2.1 Extracted Parameters

For the first step of the extraction process we evaluated the number of users connecting to the e-mail services. Due to the NDA restrictions we cannot disclose the absolute numbers. However, we are



(a) Service call of a single user

(b) Model states for traffic generation

Figure 6.12: Simulation model for POP3 service.

allowed to present normalized values. In the measurement period 22.1% of the customers used e-mail. This is consistent with the results we obtained in Section 4.2.

Note in this context we calculated the fraction of active subscribers to e-mail users. An active subscriber has to have at least one PDP-context, meaning that only data active users were counted. Figure 6.12(b) presents the state model for the e-mail model using the introduced parameters.

Service Request Rate Next we evaluated the number of service calls generated per subscriber per day. Fig. 6.13 shows the eCDF of service calls per user. As in the previous sections we see that both technologies, GPRS and UMTS, represented by the solid and the dashed line respectively, are very similar to each other. This indicates that the e-mail service usage is decoupled from the technology. Further investigation in this topic showed that other parameters such as the e-mail size does not differ between the technologies. Therefore, we did not extract the following parameters for GPRS and UMTS separately.

Considering a request-generation-process combined for all users in the network we extracted the inter-arrival time between two consecutive service calls. The parameter for 1,000 active users in the busy hour follows an exponential distribution with the rate $\lambda_r^{-1} = 0.01102$.

We implemented the service generation using an exponential distributed rate.

Probability of Login-only Session According to our e-mail model we had to discriminate between a simple login with no e-mail download from a login with an e-mail download. Without a parser at the application layer we had to find a compromise in detecting a service call with only a login process. An analysis of the POP3 protocol (see [100]) revealed that a normal login process should have less than 13 packets, including the TCP handshake. We set an arbitrary limit to 15 packets in up- and downlink direction for connections we consider to be login only. Further, we also set a lower limit to five packets in up- and downlink direction, as the lower limit for a valid connection. This rule was set to filter out all port scans and attacks against any e-mail server. The final numbers were 22.4% of

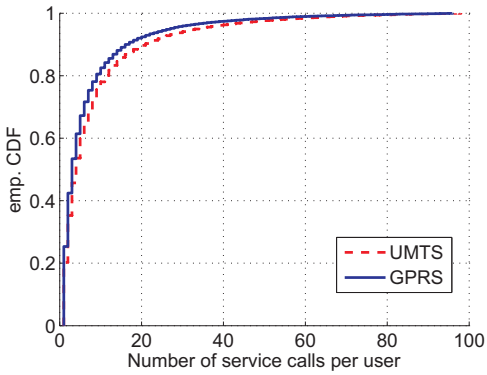


Figure 6.13: Number of POP3 service calls per user and day for UMTS and GPRS.

the logins with an e-mail transfer and 77.6% with no e-mail transfer. In other words the probability that a user has a new e-mail in his mailbox, called p_{new}, equals 22.4%. Therefore, simulating the login process separately is an important part of the model. The volume for all detected login connections is below 300 byte.

Size of an E-Mail Message The e-mail size, S_{msg} is the next parameter evaluated. In our case the name e-mail size is a bit misleading, as the POP3 service can transfer many e-mails in one connection. The parameter is the volume a user will transfer if he is connected to the service once there is a new e-mail in his mailbox.

A scaling analysis of the e-mail sizes did not show any heavy-tailed behavior. As in Chapter 5 we applied an MLE fit to a set of distributions and chose the best fit based KS statistic.

Figure 6.14 presents the fitted values for the login and the e-mail size part. We also included values from the FUNET (Finish University NETwork) [99] and the values suggested by Paxson [24, 87]. The FUNET model was derived from traces collected from the Finish university network and its research network. It uses a Cauchy distribution (see Appendix D) with a cut-off, $S_{\texttt{max}}$, for the maximum message size of 10 kByte.

$$f_{\texttt{Cauchy}}(x) = \begin{cases} \frac{1}{\pi\, F_{\texttt{Cauchy}}(S_{\texttt{max}})} \frac{1}{1+(x-0.8)^2} & 0 \leq x \leq S_{\texttt{max}},\ x \in \mathbb{N} \\ 0 & else. \end{cases} \tag{6.1}$$

Paxson [24] characterizes the mail size using two log2 distributions (similar to a log-normal but with a base of two, see Appendix D for more details), one for small and one for large e-mails. A fixed quota of 300 byte is added to both distributions in order to model the application overhead, S_{head}, within the login process of every e-mail download. The paper already mentions that it is problematic to model the overhead as it is a function of the number of servers the e-mail has to pass, similar to a hop count, in order to reach its destination. Every server adds some lines to the header of the e-mail, thereby increasing the size of the header.

It is assumed that 20 % of all e-mails follow the large distribution while the remaining 80 % originate from the small distribution. The crossover between the two distributions was set to 2 kByte. The e-mail size is limited to 100 kByte.

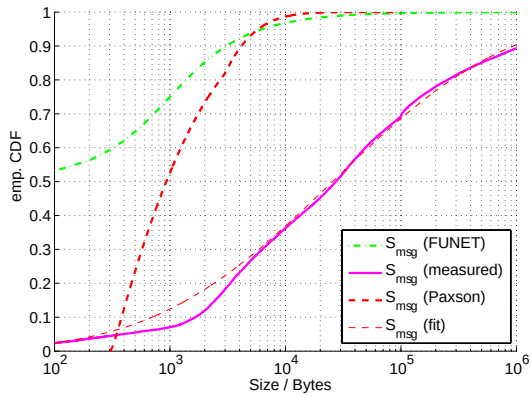


Figure 6.14: Volume transfered per service call.

The values for both models already include the login process. Therefore, the CDF starts at higher values than our e-mail size. The median value from Paxson for the e-mail size is more than one order of magnitude smaller than the actual values measured in the 3G mobile network. The FUNET estimation for the e-mail size is even smaller than in the Paxson model. We assume that this was due to the age of the models. We fitted the parameter mail size with a log-normal distribution. The login process consists of a number of packets, N_{pkt}, with a fixed packet size of 30 bytes payload each. A uniform distribution was used to model the varying number of packets per service call. The processing time at the server side, T_{login}, is also implemented as a uniform distributed variable.

In this paragraph we summarize the model. Table 6.2 compares the fittings of the parameters for the different models. Note that the Paxson values are based on a log-normal distribution with base two. The parameter λ_r^{-1} given in Table 6.2 is based on 100 active users within the busy hour, which lasts from 7.45 to 8.45 p.m., averaged for four days. The parameters of the model are depicted in Fig. 6.12(b). If a service request is generated the model enters the login phase. This phase is executed for every user. At the end of the login process the service call will start downloading an e-mail with the probability p_{new} and terminates the session after the successful transfer of the e-mail. If the model decides not to download an e-mail message the service call will terminate right after the login phase. The number of users in the network can be changed by increasing the request rate.

6.4.3 Simulation Setup

In this section we describe the simulation setup that we used to verify our new model. We used ns-2 to verify our e-mail model. A basic setup with two nodes was used to simulate the underlying network conditions. The link parameters were adopted to parameters similar to the RAN we have been monitoring. The data rate was set to discrete values: 40, 64, 128, and 384 kbit/s. These values reflect the setting in the network for UMTS-DCH and GPRS. We simulated the RTT as a uniform distribution from 200 to 500 ms, according to values we have measured for the TCP-RTT in [20].

Into this artificial scenario the new e-mail agent was implemented. The agent supports e-mail transfers with and without login. It is therefore possible to extract the impact of the login session separated from the e-mail download process. The application uses the e-mail size according to our measurements in the network. In addition to this we implemented the e-mail sizes as suggested by

Paxson			
Model Parameter	Distribution	Parameter	
S_{msg} small / byte	$\log_2$-normal	μ =10.0	σ =2.75
S_{msg} large / byte	$\log_2$-normal	μ =9.5	σ =12.8
S_{head} / byte	Constant	x =300	—
S_{max} / byte	Constant	x =100 k	—
FUNET			
Model Parameter	Distribution	Parameter	
S_{msg} / byte	Cauchy	a=0.8	b=1.0
S_{max} / byte	Constant	10 k	—
New model			
Model Parameter	Distribution	Parameter	
S_{msg} / byte	log-normal	μ =10.151	σ =2.8096
N_{pkt} / byte	Uniform	min =10	max =12
T_{login} / ms	Uniform	min =100	max =200
λ_r^{-1} / s	Exponential	μ =0.0110	—
p_{new}	Constant	μ =22.4 %	—

Table 6.2: Model parameter for the email model for both literature and our model. (Note that the $\log_2$-normal distribution is similar to the log-normal distribution, with a different base parameter of *2*.)

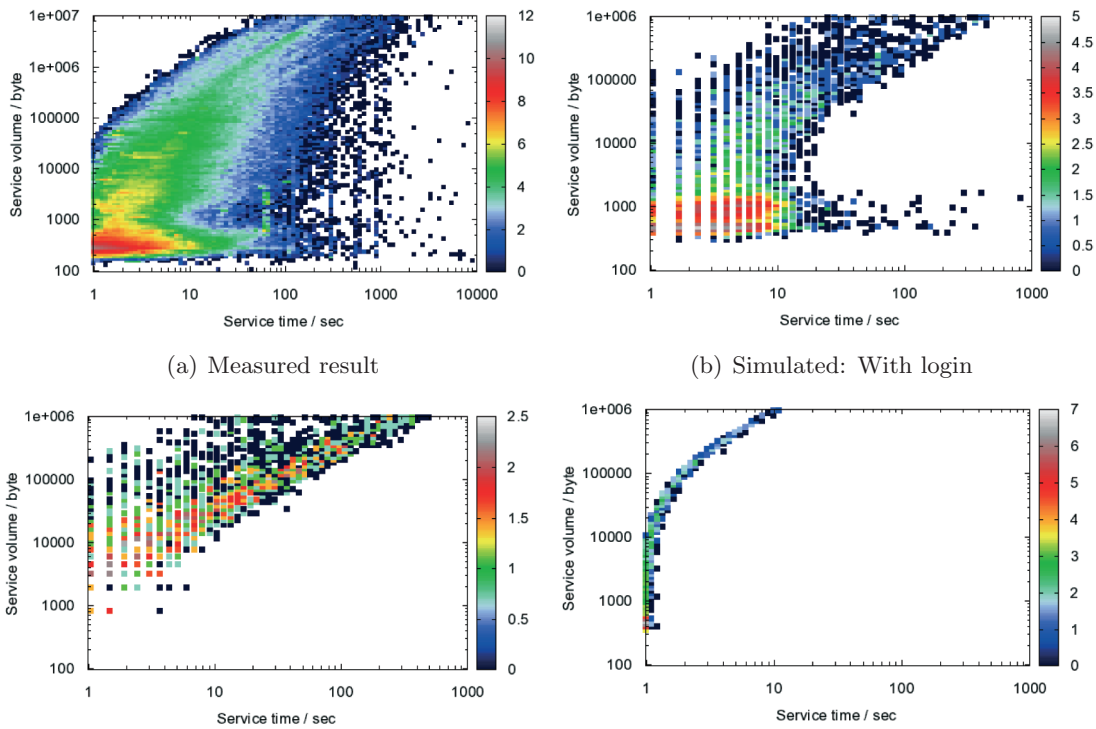
Stuckmann [87].

6.4.3.1 Measured Service Footprint

For the first result we extracted a scatter plot of the service time over the data volume transfered. Each point in the scatter plot is equal to one POP3 service call of a client in the network. Hence a single user can be represented with several points in these scatter plots. In this scatter plot both axis were binned into logarithmic bins. The color of each dot indicates the number of occurrences at this specific [x,y] position in the plot. The Fig. 6.15(a) gives the service footprint recorded in the mobile core network at the Gn interface of the provider.

There are two main clusters visible in Fig. 6.15(a). The lower cluster is due to users who only access the system and do not find any new e-mail in their account. They complete the login procedure but do not transmit any e-mails. This is visible from the fact that they only have a very limited traffic volume per service call. The wide spread in time reveals the large variation in the round trip time for the different terminal connections. This is consistent with our finding in [20].

The second cluster above 1 kByte is generated by terminals which have new e-mails waiting at the server for delivery. This cluster is modeled by the e-mail size parameter we extracted in the previous section. The data rate of these terminals is limited to the parameters of the RAN. Therefore, there is an upper boundary at a net data rate of approximately 1 Mbit/s.



(a) Measured result

(b) Simulated: With login

(c) Simulated: No login

(d) Simulated: With login, ADSL

Figure 6.15: Scatter plot of the e-mail service: service time vs. service volume (log binning, linear coloring).

6.4.4 Simulation Results

We now setup a simulation according to Section 6.4.3 in ns-2. In this section we present the results of this simulation. To compare the output of our model with the measured results presented in Fig. 6.15(a) we ran three different setups: with login and without a login process, which is similar to the Paxson model, where in fact only the size of the e-mails is distributed differently. The first two scenarios should analyze the impact of a separate login process emulation. The third setup is used to show possible gains over common models.

The outcome of the first scenario is depicted in Fig. 6.15(b). The figure is very similar to the measured result in Fig. 6.15(a). The 2D correlation coefficient of this result with the measured data is 0.91, which indicates a high similarity. However, the variation in the direction of the x-axis is smaller than for the recorded results. This is due to the fact that we use only a simple emulation of the underlying RAN system, introducing only small variation.

The result of the second scenario in Fig. 6.15(c) differs strongly from the recorded trace in Fig. 6.15(a). Here the correlation is only 0.3, which indicates the large difference between the results. As the login emulation is not in place there is no cluster and no additional delay from the network. The data is pushed directly to the mobile terminal.

Finally, we simulated the first scenario with new link parameters set accordingly to an ADSL line: 1 Mbit/s of data rate and 10 ms of RTT. In Fig. 6.15(d) the results for this setup are presented. The link delay is one order of magnitude smaller than in the mobile RAN case. Therefore the result is

similar to a scenario where the login process was neglected. We conclude that this is also the reason why common e-mail models, designed for fast LAN connections, skipped the login process.

6.4.5 Concluding the Model

In concluding the POP3 model we learned that only 20% of the subscribers access this application. As in most of the previous sections the parameter distributions are similar for UMTS and GPRS. The e-mail size did not show heavy-tailed properties.

A comparison based on TCP footprints showed that a simulation with an explicit implementation of a login process delivers better results.

6.5 Summary

IN this chapter we presented various traffic models for the application layer. These models try to emulate the user and application behavior instead of the flow parameters following the lines of Chapter 5.

The models in this chapter were not designed from scratch. In fact for HTTP and FTP we only extracted parameters for existing models according to our traces. The HTTP parameters were the only ones where the results for UMTS and GPRS did differ in a significant way. We assume that this is due to a small sample size of 1,000 users.

We extended the common e-mail models for POP3 with a login process. This change led to simulation results close to the footprints we measured in the 3G network. Comparing our e-mail sizes with older models we found out that the value grew one order of magnitude every 10 years.

Self Criticism: The HTTP parameter extraction is based on rather old data from 2004. At this point most of the users still worked with GPRS data-cards. Today most of the customers use HSPA and EDGE instead. Therefore, we think that the parameters will be different. There were two reasons why we did not update the parameters. First, we did not have the time to improve `tcptrace` so that a higher number of parallel connections could be processed. In the actual version `tcptrace` can handle only the first trace TR_1. Secondly, the privacy policy changed and we were no longer allowed to cache the customer data with payload. This is necessary in order to run `tcptrace` on the traces. Therefore, we would have to implement all the `tcptrace` functions into the MOTRA tracing system. We were not able to spend this effort within this work.

The traffic model for e-mail covers POP3 traffic quite well. However, we did not analyze IMAP traffic. Recent traces taken in 2008 show that this kind of service together with Blackberry is starting to replace POP3 to some extent. We did not expect such a change, which was a clear misjudgement from our side.

Chapter 7

Newly Developed Service Traffic Models

Contents

IN this chapter we again present service traffic models. In contrast to the previous chapter these models were build from scratch and are related to the recently emerging and fast growing applications in the sector of online gaming and push to talk (PTT). A motivation for model behavior related simulation can be found at the beginning of the previous chapter (see Section 6.1).

Section 7.2 introduces three new models for online games. Each model covers one genre of online games: first person shooter, real time strategy, and massive multiplayer online games.

A purely theoretical model is presented for PTT in Section 7.3. The model was implemented in MATLAB according to Nokia PTT specifications and ITU 59.9 artificial conversations. The chapter is concluded with Section 7.4 presenting a summary of the traffic models.

7.1 Motivation

THE service share at Internet backbone links of today (2007) show that new services take over the lead from HTTP. In terms of volume, peer-to-peer networks generate more traffic than all web browsing customers. However, in a mobile cellular network volume is a costly resource for the customer. Therefore we did not focus on such applications but rather investigated new applications for which the customer may be willing to pay extra in order to stay connected everywhere.

We chose online gaming and PTT as new emerging markets. The reason for online gaming was the fact that today (2007) most of the games have a kind of social platform. The users want to stay online even on the weekends at the country side. The fees for the games are charged by the month. Therefore, the customers want to be online as often as possible. Figure 7.1 depicts the growth of the number of subscribers for a popular massive multiplayer game, namely World of Warcraft (WoW). Already in 2006 we found more than 1,000 gamers of WoW in the mobile network in Austria (population eight million). In total more then 7% out of the *high* group, as defined in Table 4.2, mobile customers played online games. Therefore online games are already an important service for the customers.

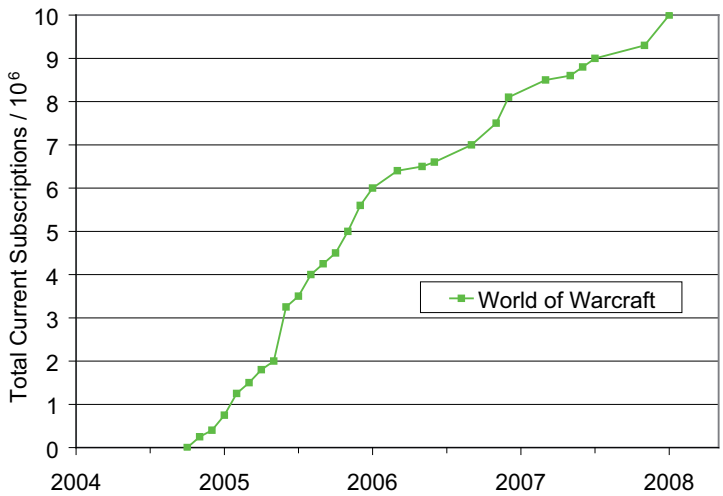


Figure 7.1: Growth of subscriptions to the online game WoW (Source `http://www.mmogchart.com/`).

PTT is a similar system to VoIP, which we consider as an important service that may replace classical phone communication. Therefore we focused on this service for building a traffic model.

In the beginning of our work (2004) only by few analyses for online games existed [101, 102] and PTT was not lunched until then. Therefore, we decided to develop new models for this type of service.

7.2 Traffic Models for Online Gaming

ONLINE Gaming is an application that evolved from applications called multiplayer games. In multiplayer games two or more players enter the same game either in competition or in cooperation mode. The early adaptations were realized at the same machine; later the players took part on separate computers connected via a network. The first known implementation was coded in 1959 by William A. Hinginbotham [103]. He used an oscilloscope to show a virtual tennis game at the Brookhaven National Laboratory.

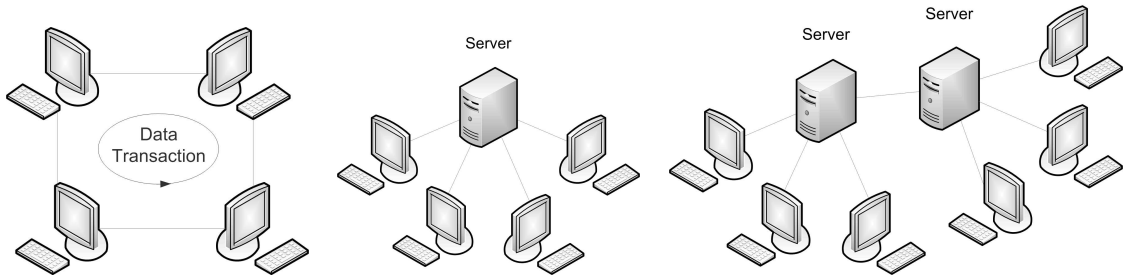
The commercial success started with Galaxy Wars in the 1970s [103]. However, at this point in time a computer was far too expensive for a normal household, therefore games did not spread quickly. The introduction of cheap home computers such as the Commodore C64 changed the scene. In the early 1990s `id`Software released a game called Doom [103]. It was a first person shooter, although not the first of its kind, it had an important new function implemented: network gaming. This feature yielded a big success and was from then on a *must have* for follow up games. Today some games can be played online only.

Network Architecture There are many different network architectures for online games to establish communication between nodes. The simplest realization is depicted in Fig. 7.2(a), the events of all players are exchanged between the nodes and then executed synchronously. There is no need for a dedicated server node, however in case one computer crashes the communication is broken. In real time strategy games a setup like Fig. 7.2(a) is often used. Here, this is not a short coming as such games need all players to stay connected till the end. The architecture does not scale well as every added client increases the information to be transfered between the clients.

The server-client architecture, as shown in Fig. 7.2(b), scales better in terms of participating users. All clients report the input updates from their players to a central server node which can physically run on one of the clients. The server processes the inputs and broadcasts the results to the client computers. This setup scales better for a large number of players, as the client information is only reported once to one node. The server then sends only the updates relevant to the specific client, e.g., only information within the view of the player has to be updated.

In case of a very large user population, as in Massively Multiplayer Online Games (MMOG) where more than 10,000 users join the same online game, the client load may be split to several servers that run as a loose cluster, see Fig. 7.2(c).

Modeling Online Games In the recent past the popularity of online games has grown quickly [104]. Even in our mobile cellular network the traffic share of online gaming traffic has reached a level of around 2% in 2006 [28]. The relatively high load combined with the fact that this type of traffic



(a) Peer to peer (b) Client - server (c) Client - server cluster

Figure 7.2: Online gaming network architectures.

shows significant different characteristics compared to common and well known traffic sources such as WWW, e-mail or FTP makes it interesting to analyze. In Asia MMOGs are much more popular than in Europe. Therefore, there are already some publications based on games such as ShenZhou-Online[1] [105, 106, 102].

In this thesis we provide new traffic models for the following three different types of online games: First Person Shooter (FPS), Real Time Strategy (RTS) and Massively Multiplayer Online Game (MMOG).

7.2.1 Traffic Model for a Fast Action Game: Unreal Tournament

FPS games are the ultimate benchmark for mobile systems. These games tolerate only a small amount of delay and jitter [27], which are system inherent for a wireless connection. Today's 3G networks are not able to provide low delay connections suitable for FPS games. However, with the deployment of HSDPA and HSUPA delays in mobile network may be low enough for FPS games. Therefore, we investigated the traffic patterns of an FPS game.

Unreal Tournament 99 (UT 99) is a game quite popular in the world of FPS. It has a dedicated server infrastructure as depicted in Fig. 7.2(b). In an FPS the games are viewed in a first person perspective and the goal is: shooting opponents. An important factor in this type of games is timing and accurate movement, making it challenging to set up on wireless systems.

UT 99 sends data via the UDP transport protocol, as most real-time applications. In the previous section we modeled services running on top of TCP. In contrast to UDP, which offers only error correction codes for the packets, TCP offers reliable transmission of data including retransmission of lost segments. Therefore, data transmission in TCP has to be modeled as data units from an application. In UDP however, there is no protocol interaction. We can directly model the traffic patterns of an application via the packet size and the *interarrival-times* of the packets.

The following model for UT assumes that the communication between server and client is independent from the packet loss at the interface level. While in wired networks packet loss often can be neglected this is not possible in a mobile cellular network. However, there are several publications which show that the applications we simulate, e.g., UT and SC, can operate with up to 30% of packet loss [103, 107, 108]. This is achieved by the complex prediction of user movement and actions. The target packet loss in the network under test is set to 1%. We showed in [20] that even in case of a

[1] `http://www.ewsoft.com.tw/`

bottleneck in the core network the packet loss did not exceed 5%. Therefore, we did not consider correlation between packet loss and source rate.

7.2.1.1 Packet Level Analysis

As mobile access technologies of today are too slow to serve as a transport network for FPS applications we could not collect our traces from the METAWIN system. Therefore, we recorded eight hours of packet-switched traffic on a PC connected to the Internet via an ADSL modem, game version `1.32`.

Figure 7.3 displays the packet level analysis of a UT 99 trace. As discussed above we focus on packet size and packet interarrival-times for server and client separately. Figures 7.3(a) and 7.3(c) depict the histograms for the server node. We can observe that the server is transmitting packets of varying size at a nearly constant rate. We assume that the server sends regular packets to keep all clients updated. The size of the packets change depending upon the amount of events the server has to report.

In contrast to this, the client, as we recognize from Fig. 7.3(b) and Fig. 7.3(d), transmits packets at a varying rate with nearly constant size. The client sends packets depending on the user interaction. There are some events visible at 200 ms interarrival time, notice the arrow at the right side in Fig. 7.3(d). A visual inspection of the traffic time series in combination with the ingame videos revealed that there are periods of low uplink traffic for the time span a user is dead and has to wait until he can rejoin the game. Based on these findings we designed a state space to model the different levels of traffic.

7.2.1.2 Traffic Model for UT 99

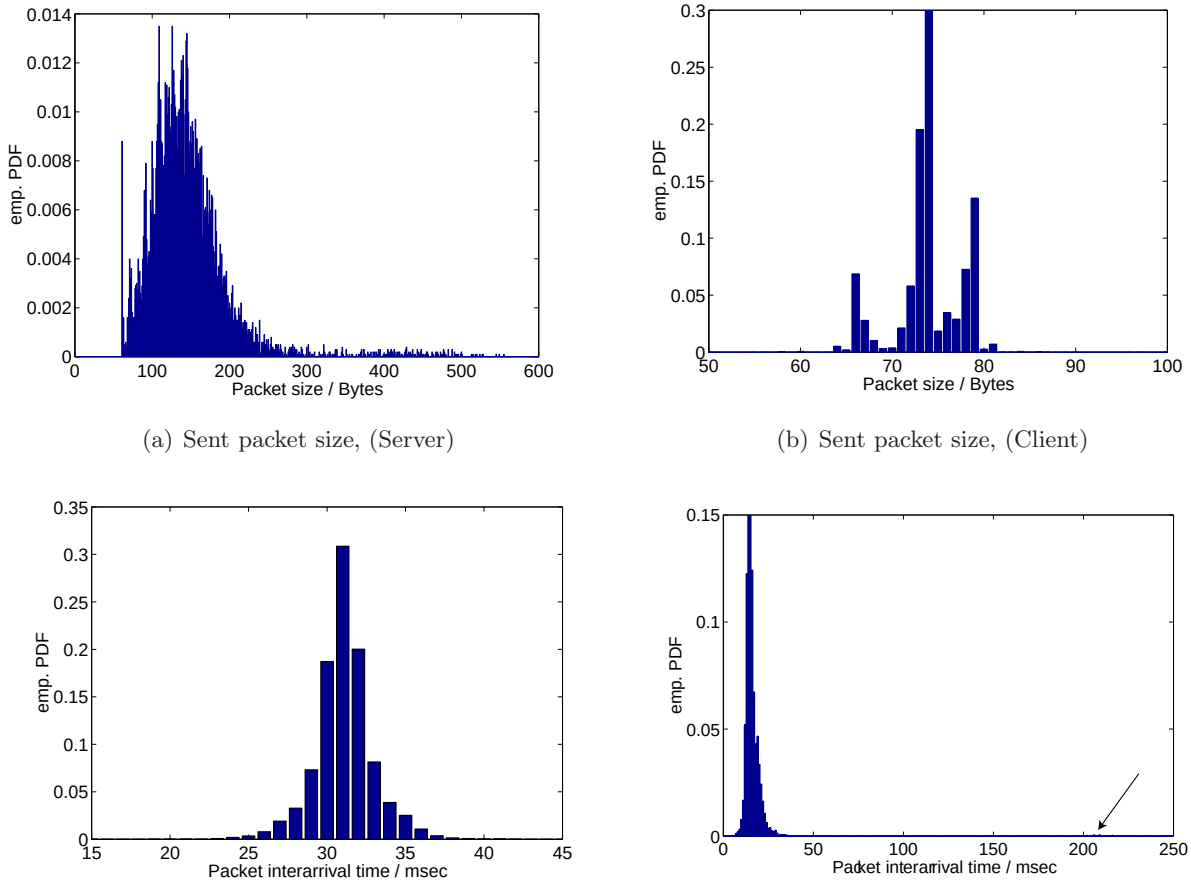
The traffic model for UT 99 consists of three states. A player starts the session in the "Change Level" state. After a timeout the state is automatically changed to "Active". The game starts and server and client begin to transmit packets. The client transmits packets with an interarrival time on the left side of the histogram in Fig. 7.3(d). If the player is killed, the interarrival time of the client is around 200 ms corresponding to the second cluster in Fig. 7.3(d). The player will stay for some timeout in the killed state until he can rejoin the game again. After the session time the state automatically changes to "Change Level" and the cycle can restart again. In the "Change Level" state we assume as a simplification that both server and client do not transmit any packets

The parameters which we needed to implement our model were: packet size and interarrival time for client and server, session time, respawn timeout and the time it takes to change a level. In the following paragraph we extracted these parameters and implemented the model into ns-2.

7.2.1.3 Parameter Fit and Simulation Results

In this paragraph we present the fit for the parameters which we defined for the traffic model. A UDP packet has an upper limit defined by the Maximum Transfer Unit (MTU), therefore we are not required to search for heavy-tailed behavior.

Table 7.1 shows the best MLE fits for the packet size and interarrival times. The respawn timeout and the time between two consecutive levels were set to discrete values according to the observed mean value from the trace.



(a) Sent packet size, (Server)

(b) Sent packet size, (Client)

(c) Interarrival times, (Server)

(d) Interarrival times, (Client)

Figure 7.3: Packet level analysis for Unreal Tournament 99.

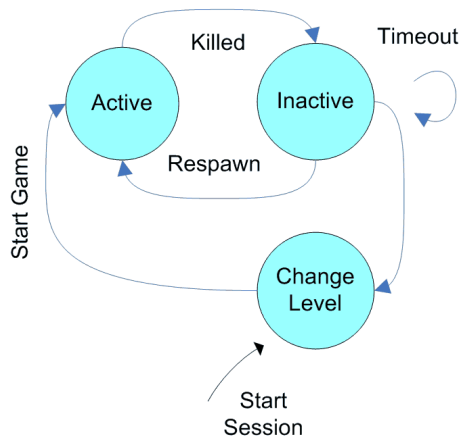


Figure 7.4: State diagram for an UT99 traffic model.

Model Parameter	Distribution	Parameter			
Packet Size (Server)	**log**-normal	$\mu =$	4.21	$\sigma =$	0.657
Packet Size (Client)	normal	$\mu =$	73.53	$\sigma =$	2.331
Interarrival Time (Server)	normal	$\mu =$	31.23	$\sigma =$	1.221
Interarrival Time$_{low}$ (Client)	normal	$\mu =$	32.70	$\sigma =$	8.739
Interarrival Time$_{high}$ (Client)	normal	$\mu =$	218.50	$\sigma =$	15.130
Respwan Timeout	constant	4.329 sec			
Session Time	constant	15.42 min			
Change Level Timeout	constant	23.61 sec			

Table 7.1: Model parameters for UT 99. (Note the first variable is log-normal!)

Figure 7.5 displays two eCDFs for the uplink data rate, the measured data in blue and the ns-2 simulation result in red. For this graph we simulated one gaming session of 15 min. We realize that the model produces an eCDF similar to the empirical data we measured.

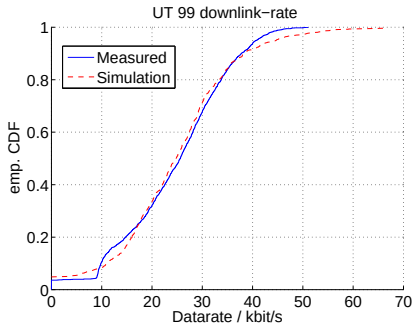


Figure 7.5: Simulation result for ns-2 implementation of the UT99 traffic model.

Concluding the UT 99 traffic model we learned that this FPS has different states and each of them has different traffic patterns. As the application uses UDP as a transport protocol the traffic can be modeled by simply fitting size and interarrival time of the packets to an analytical distribution. The output from our ns-2 implementation yielded results close to the empirical data. In our model we did not look into a correlation between the up- and downlink traffic as we did not have access to the source code in order to understand the origin of the correlation.

7.2.2 Traffic Model for a Real Time Strategy Game: Star Craft

Star-Craft (SC) is one of the most popular RTS games. There are three tasks to master in these kind of games: First, collect resources (often two types), secondly, use them wisely to build units and thirdly, finally go into combat. Timing in these games is not that crucial, but higher delays also result in problems for micro management of the built units.

The average delay of a 3G network is approximately 200 ms. SC could be played well when we ran the game via a UMTS DCH connection [27].

Like UT 99, SC uses UDP packets in the local network. We again recorded packet traces and extracted size and interarrival time of such packets. SC has a network structure as it is displayed in

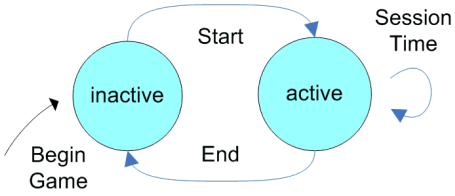


Figure 7.6: State diagram for a SC traffic model.

Fig. 7.2(a). All nodes participating in the local network run synchronously. The failure of a single node results in blocking the others.

Again we assume that the parameters of the model are independent to the packet loss at the interface. This decision is based on the same arguments we presented for the UT traffic model in 7.2.1.

7.2.2.1 Packet Level Analysis

We recorded a packet trace of a one-hour-game session. From this dataset we extracted the parameters packet size and interarrival time.

Both parameters had a nearly constant value. The interarrival time was uniformly distributed between $31 - 33$ ms. The packet size sent from each client was between $75 - 78$ byte.

7.2.2.2 Traffic Model

In the last paragraph we have shown that the traffic patterns of SC are very simple. A two-state model, as it is depicted in Fig. 7.6, can generate appropriate traffic.

The only missing parameter is the session time. We extracted this parameter from 400 games, which were recorded over a timespan of three years. The empirical data for the session time had two peaks. One at around eight minutes and a smaller one around 19 min. We modeled the session time as $t_s = 0.62 \cdot N(8.2, 2.3) + 0.38 \cdot N(18.7, 3.7)$ (values obtained with an MLE).

7.2.3 Traffic Model for a Massive Multiplayer Online Game: World of Warcraft

Within the last few years, games such as World of Warcraft (WoW) made MMOGs very popular. An MMOG intends to create a virtual environment which is populated simultaneously by thousands of players. The big attraction arises from the fact that only a few characters are simulated by the server, called NPCs (Non Playing Character), while the rest are humans.

The session times of MMOGs are typically higher than those of round based action games, such as FPSs. FPS games join short matches in the order of 15 min. Users stay for several matches in one session. Due to these tournaments the sessions have a periodical behavior. It is therefore common to model only one tournament and repeat the result [105]. MMOG gamers will play one continuous session with varying parameters, this impacts the traffic patterns. MMOGs motivate to go online regularly to interact with the virtual environment. Therefore, the timespan between consecutive sessions may show a periodic profile.

From a network traffic engineer's perspective this type of application pushes a new class of service: the real-time interactive service over TCP. The real-time restriction is caused by the interactive nature of the application. The gamer is interacting with others via the server node. Therefore, responsiveness of the underlaying transport network is crucial for a good subjective gaming experience. The maximum tolerable delays are much larger than those known from other online games. Common real-time systems use UDP because retransmitted packets will have too high a delay and is discarded anyway. In WoW and many other MMOGs, TCP serves as a transport protocol. TCP is connection oriented and offers a reliable connection. This attribute is well suited for MMOGs, preventing error propagation during long sessions. The RTT can be reduced by using small packets. In WoW we are now dealing with an application generating large numbers of very small TCP packets in one long stream. Services like web browsing generate many small connections.

WoW is the first game we could monitor on our 3G live network. In fact we were quite surprised to find WoW in the top ten TCP services in one of our traces. However, this was only true for a single day in May 2006, part of $\mathtt{TR}_3$.

7.2.3.1 Measurement Setup

In order to obtain the different parameters we used two setups for our measurements, an active and a passive approach.

The first setting is based on active measurements. This setup is able to monitor user session parameters, IP packet size and interarrival time. The IP parameters were recorded using a monitoring PC running the WoW client, which was connected to the Internet via a 1Mbit ADSL link. In addition to this an ethereal[2] session running at the PC recorded the packet traces. The datasets consist of several gaming sessions with a sum of about 20 hours net active time, recorded in May 2006.

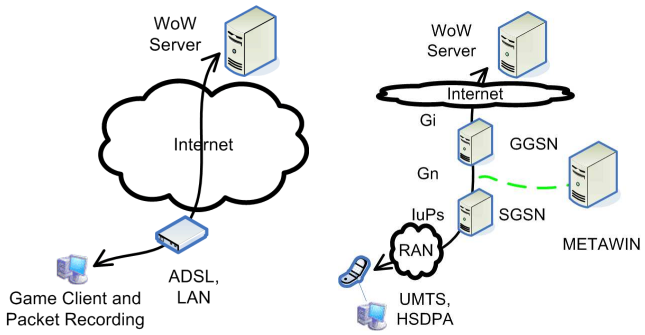
The user behavior was extracted using a shell script recording the session time of two groups with five players each. The script ran as a background job, dumping start and stop time of the *wow.exe* file. These outputs contain the session duration for each player. To verify these values from the small sample we used a portion of the $\mathtt{TR}_6$ data set.

The $\mathtt{TR}_6$ dataset contains two days of traffic from September of 2006. Analyzing this trace we found approximately 1,000 different users playing WoW. We were surprised by the fact that these users added up to nearly 1% of the total TCP traffic in the core network. We showed in Chapter 4 that for non-HTTP services a share of 1% is a relatively large number in mobile networks. In fact other common services such as FTP had a smaller volume share.

Considering the two sources we were able to verify the session time of the small sample with the one collected from the large population. Note that the users monitored in the mobile core network most likely used a flat rate contract, therefore a similar behavior is expected.

We used two setups to record three different datasets. From the wired Internet access we obtained two datasets. The first dataset is a packet trace dump from one WoW client. This dataset was used to extract parameters at the IP level. The second dataset contains the session times of 10 users using wired Internet accesses. A shell script recorded the session duration for each user. From the second setup we obtained the third dataset, $\mathtt{TR}_6$, containing the session times of a large population.

[2] `http://www.ethereal.org/`



(a) Wired networks (b) Mobile networks

Figure 7.7: Measurement setups.

7.2.3.2 Traffic Characterization

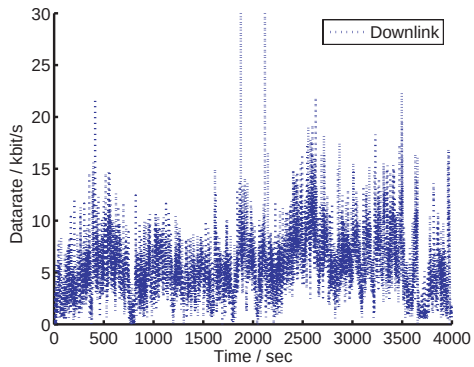
MMOGs focus on accurate execution of client inputs. This has an impact on the transport protocol. WoW sends its packet through TCP connections, while common FPSs transmit their packets via UDP. The TCP ACK mechanism will protect the delivery of the data packets by the underlying physical network.

Basic Analysis of the Traffic Analyzing the TCP stream from WoW we observed that the two flags ACK and PSH are set for most of the packets. In near real-time usage both flags are used, although uncommon in normal TCP transmission. A TCP agent has several performance features [31], one of them is delayed acknowledgment. Instead of sending an ACK packet for each datagram, the receiver will wait for some time and then acknowledge a bunch of packets all together. On low-delay links this will reduce the number of sent ACK packets. However, in near real-time applications which do not tolerate delays, this is a bad choice as it will introduce extra delay in case of a retransmission.

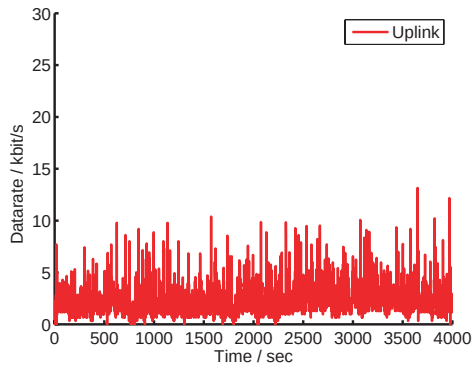
Another throughput optimization is achieved by the gathering of small chunks of data. The Nagle algorithm in the TCP agent uses a lower threshold in order to send blocks of data [109]. As TCP is connection oriented, the way programs transmit data has fundamental differences compared to UDP. In UDP the program has to take care of packetizing the data, whereas in TCP the program hands over its payload to the agent without preprocessing, e.g. there is no problem for the program to hand over more than MTU bytes because the TCP agent will split them later. Due to this behavior the TCP agent has no detailed information on a time stamp to send a datagram. To avoid the transmission of packets with very small payload that leads to a large overhead, a threshold was established. In an application with relative low data rates this will introduce a high delay. Known applications such as telnet suffer from this fact. Non varying data rate will add jitter. The PSH flag forces the TCP agent to send the output buffer immediately, bypassing the threshold.

Packet Traces In this paragraph we analyze packet traces recorded during several sessions. Our datasets contain $\approx$ 20 hours of gaming.

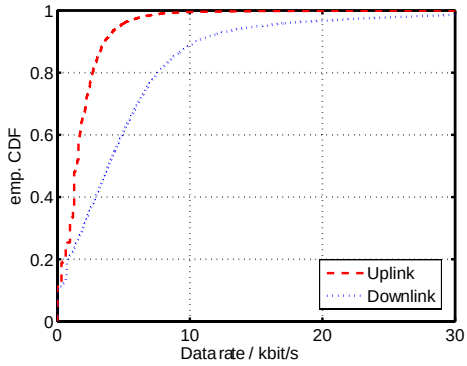
Figures 7.8(a) and 7.8(b) give a snapshot of the down- and uplink data rate time series. To plot the data rate, we used a bin size of 1sec and no running average. The median downlink data rate is 6.9kbit/s. In uplink we observe less traffic with a median of 2.1kbit/s. From the figure we also



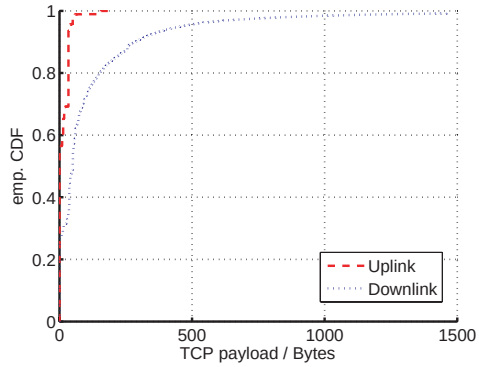
(a) Snapshot of a downlink data rate time series



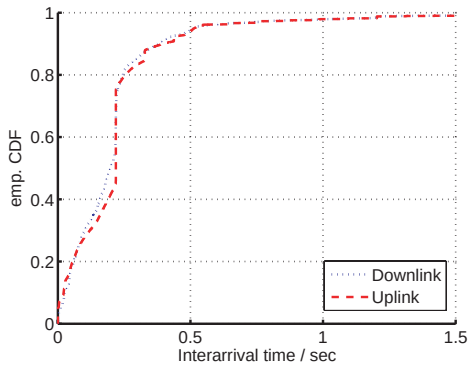
(b) Snapshot of an uplink data rate time series



(c) Data rate



(d) Packet sizes



(e) Packet interarrival time

Figure 7.8: Traffic snapshot and summary of the packet-level analysis.

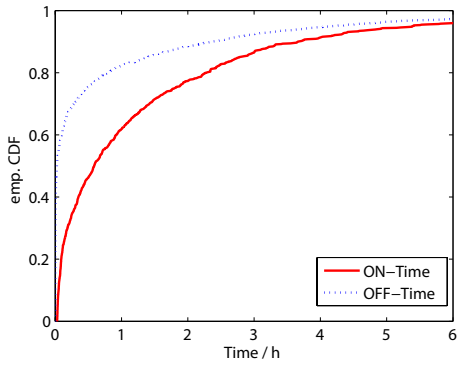


Figure 7.9: Session times from wired Internet access.

learn that there are some high peaks ($\geq$ 64kbit/s) in downlink direction. Comparing the traces with a recorded video we can correlate the peaks to scenes with high environment interaction, e.g., many players nearby. We did not use this information in our model.

Figure 7.8(c) presents the CDF of the up- and downlink data rate. Compared to the snapshot in Fig. 7.8(a) we observed some not-negligible part of high downlink bandwidth. A detailed analysis of the packet trace time series showed that these peaks occur mainly in the beginning of the sessions.

Figure 7.8(d) illustrates the eCDF of the packet sizes in the up- and downlink direction. The figure displays that in the up- and downlink, 28% and 57% of the packets, respectivly, have no payload. This is due to the fact that these packets are only acknowledgments (ACK).

Figure 7.8(e) shows the eCDF of the interarrival time of packets in up- and downlink direction. From this we learn that the client has a target update interval of approximately 220 ms indicated by a step in the CDF. The maximum values range up to 1.5sec. Up- and downlink curves are similar.

User Behavior With the METAWIN system we recorded the session times for a large population ($\approx$1,000 gamers). Next we focused on the question of whether or not using a mobile network had a negative impact on the active time (ontime). Therefore, we also extracted session times from two groups of users, each consisting of five gamers connected via fixed line access. Ten persons were monitored by a shell script for one month. We split the users into two groups according to their Internet access speed. Group one used a standard ADSL link with 1M/256kbit/s and group two had a link speed of 1.5M/512kbit/s.

Figure 7.9 visualizes the eCDF for the session times obtained from $\mathtt{TR}_6$. The ontime exceeds the value of 2.75 hours in more than 20% of the samples. It is interesting to see such high values of active service times in a mobile core network. This indicates that the mobile Internet access is used to replace wired Internet access links. The tail of the off-time curve indicates that the players are online on a regular time frame.

The stars and crosses in Fig. 7.10 show the values obtained by the second dataset extracted from wired Internet accesses. We concluded that the results of all three datasets were similar. Therefore, we fit our model parameters according to the $\mathtt{TR}_6$ dataset.

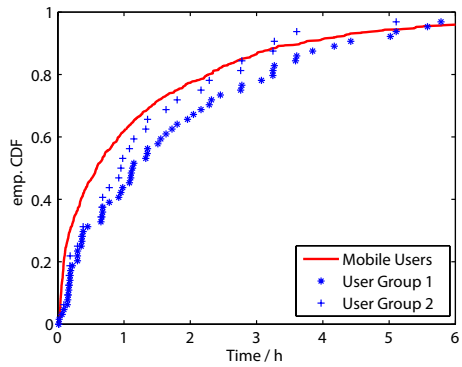


Figure 7.10: Session times: mobile vs. wired Internet access.

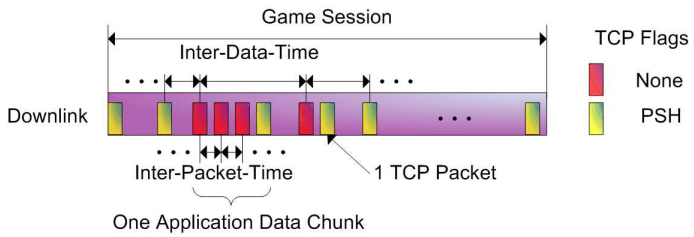


Figure 7.11: Definition of application-data-size and inter-data-time.

Simulation Results Based upon our findings in Section 7.2.3.2 we developed an ns-2 script to model a WoW-like client to sever connections. The following paragraph discusses the introduction of two new parameters that adapt the measured values to inputs for the simulation. In contrast to UDP, using a TCP connection between server and client introduces interaction at the protocol level. Therefore it was necessary to transform the input parameters obtained in the packet traces into suitable parameters to drive a simulation. In other words we could not make direct use of the packet sizes measured, but had to derive a data unit from the view of the application. We then fitted these new parameters and implemented them into the ns-2 simulation environment.

Filtered Packet Trace Analysis In the previous paragraph we have shown that there is a high fraction of ACK packets in the trace. The first straight forward approach was to filter all packets transmitting no payload. Please note that removing all ACK packets would also remove payload packets, as the client can mark a normal payload packet to confirm previous traffic.

Analyzing the resulting packet trace we realized that not all of the packets carry a PSH flag. Extracting these packet sizes we obtained mainly full MTU size. We assume that in this case the application sends datagram units larger than the MTU. The TCP service then splits the units down to MTU in order to transmit them. The last packet of this block is assigned with a PSH flag. Based on this idea we created two artificial values. The inter-data-time replaces the interarrival time for packets and the application-data-size substitutes the packet size. Figure 7.11 illustrates a detailed picture of the parameter mapping.

The following figures visualize the eCDFs for the two parameters. Figure 7.12 displays the eCDFs of the application-data-size in up- and downlink direction. The uplink curve shows several high discrete

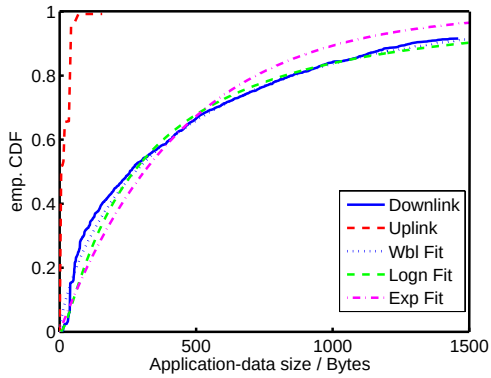


Figure 7.12: Model parameter: application-data-size.

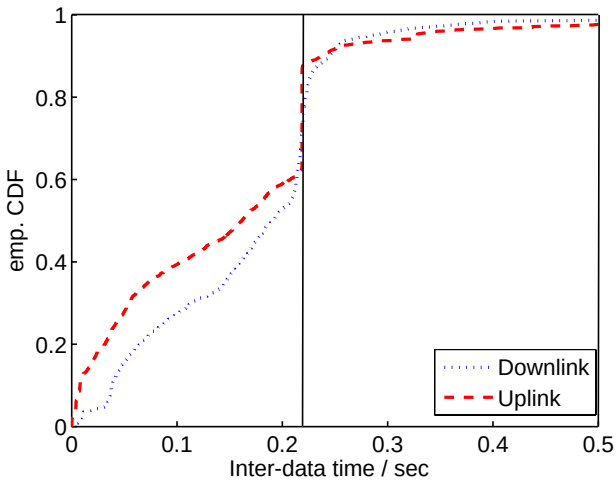


Figure 7.13: Model parameter: inter-data-time.

steps. We conjecture that these values represent simple commands that are often used, such as "walk", "attack" or "open". More complex commands, represented by a larger data size in the CDF, have a low probability. As we will show later the update rate is around 220 ms. Using scripts which execute multiple commands may change the distribution.

The downlink curve ranges up to 3,000 bytes. In the downlink the application-data size is larger than in the uplink. We applied a continuous distribution for the smooth downlink curve. A Weibull distribution fitted best. A comparison between Fig. 7.8(d) and 7.12 points out that only the downlink stream is affected by the chosen add-up rule.

Figure 7.13 shows the eCDF of the inter-data-time. The filter process reduces the step size at approximately 220 ms. The step occurs in up- and downlink. We assume that this is some threshold coded to the software. Both curves have a relative linear part followed by a step at around 220 ms, indicating a uniform distribution. The tail of the curves shows that the threshold is sometimes exceeded by network delays.

Simulation Model In ns-2 standard TCP agents only support unidirectional communication. Although client and server were modeled separately from each other, a full featured ACK process needs a bidirectional TCP implementation on the ns-2 level. A bidirectional agent permits a realistic communication between two nodes, e.g. a payload packet that carries an ACK information. We used the `FullTCP` [110] agent implemented in ns-2 to model the two-way connection, the agent generates ACK packets according to the TCP algorithms. Therefore, we directly implemented the parameters extracted in the last paragraph to drive our simulation. The following paragraphs discuss the parameter fit. The parameters were fit by an MLE fit and by selecting the distribution with the best KS statistic.

Server → Client We fit a Weibull distribution to the dataset as shown in Eq.(7.1). The resulting parameters for the distribution were $\lambda = 426$ and $k = 0.8196$. The real distribution has an upper limit $l = 3010\,\text{byte}$, so we implemented a cutoff to model this limit

$$n = f_{wbl}(l; k, \lambda)$$

$$f_{wbl}(x; k, \lambda) = \begin{cases} \frac{1}{n} \cdot \frac{k}{\lambda} \cdot \left(\frac{x}{\lambda}\right)^{k-1} \cdot e^{-\left(\frac{x}{\lambda}\right)^k} & x \geq 0 \\ 0 & else. \end{cases} \tag{7.1}$$

The inter-data time was harder to synthesize. We decided to use a joint distribution of three random variables. All three processes are modeled uniformly distributed. The resulting description of the PDF is given in Eq.(7.2). All parameters were fitted by regression to the steps in the dataset ($a = 218.3$, $b = 251.2$, $c = 1500\,\text{ms}$). Hence the curves for inter-data time in up- and downlink direction are similar, we applied Eq.(7.2) to both

$$f(x) = \begin{cases} 0.620 \cdot \frac{1}{a-0}, & x = [0 \ldots a) \\ 0.257 \cdot \frac{1}{b-a}, & x = [a \ldots b) \\ 0.123 \cdot \frac{1}{c-b}, & x = [b \ldots c] \\ 0, & else. \end{cases} \tag{7.2}$$

Client → Server The application-data size in the uplink had discrete steps (Fig. 7.12). We implemented a source generating three different sizes of packets (Eq.(7.3) with $a = 6$, $b = 19$, $c = 43\,\text{byte}$). The parameters were estimated using the average over the different datasets. The original probability of those steps adds up to 98% in the CDF

$$f(x; a, b, c) = 0.52 \cdot \delta(x - a) + 0.14 \cdot \delta(x - b) + 0.34 \cdot \delta(x - c)\,. \tag{7.3}$$

To model the session times we compared several fittings using Weibull, log-normal and neg-exponential distributions. The best fit for the on-times was obtained using a Weibull distribution with the parameters $\lambda = 4321$ and $k = 0.7813$ in Eq.(7.1). The off-time fitted best a log-normal distribution Eq.(7.4). The resulting parameters were $\mu = 5.512$ and $\sigma = 2.434$

$$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} \cdot e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}\,. \tag{7.4}$$

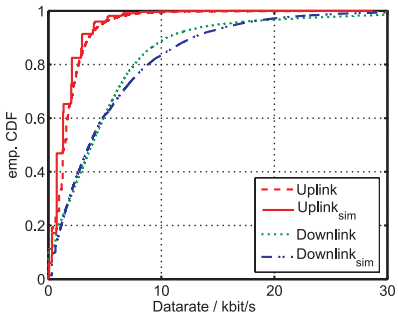


Figure 7.14: Simulated data rate.

7.2.3.3 Simulation Results

These distributions were implemented in a ns-2 script. The simulation setup consists of a server client pair using two way TCP agents. The data rate recorded in 10 hours of simulation time is plotted in Fig. 7.14. For a direct comparison we presented the CDF and added the original values given in Fig. 7.8(c). We noticed that the simulation produces a similar distributed data rate. However, a closer look to the time series reveals the weakness of the simple modeling as there are no activity bursts like in Fig. 7.8(a).

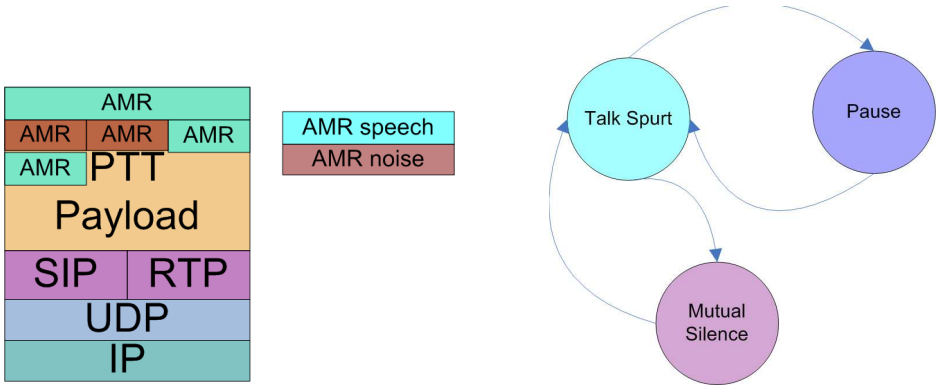
Concluding the traffic model for WoW we learned that this MMOG uses reliable TCP connections to transmit data between server and client. It is the first game we could monitor within our live 3G network and in September of 2006, $\mathtt{TR}_6$ was, in terms of volume, within the top ten TCP services. In TCP, due to interaction of the protocol with the network we cannot directly model packet size and interarrival time. We had to extract a pseudo element called the data element by mending together consecutive TCP packets. The ns-2 simulation code based on our model generated a traffic pattern with a CDF nearly identical to the measured traffic.

7.3 A Traffic Model for Push to Talk (Nokia)

THE push-to-talk service in mobile networks is a new service and therefore most of the information presented here will be of theoretical nature. The task Push-to-talk over Cellular (PoC) should provide voice services on cheap packet-switched links. To fulfill this requirement, the code rate of the Adaptive Multi-Rate (AMR) codec is set to a minimum and relatively high delays will be accepted. The push to talk service is an one-way communication using the PoC standard. The voice data is encoded using the AMR codec with a code rate of 5.17 kbit/s. The transport protocol is Real Time Protocol (RTP) on top of UDP. Signaling is transmitted via Session Initiating Protocol (SIP) to a media gateway managing the connections.

7.3.1 AMR: Facts from the Data Sheets

The voice data from one user is coded using AMR. The AMR codec splits the data stream to frames with a timespan of 20 ms [111]. The output signal consists of three different classes of bit streams (A,B,C), which are coded differently (A holds the important information, and for this reason is protected by the best code). The resulting frame length is 103 Bit. The AMR encoder supports an activity detection. With this feature it is possible to stop sending data while the user does not speak but listens to the conversation. Absolute silence is not very comfortable for the conversational partner, therefore the so called "comfort noise" was introduced in the AMR codec. In the silent period smaller packets containing random noise are transmitted. These packets have 39 bits of payload. An AMR frame is equal to the total payload that has to be transfered for a 20 ms time slot over the PTT bearer. Additionally, we have to consider the overhead of the surrounding protocols. Let us start from the bottom. In the PoC standard the transport protocol is IPv6 but recently there has been an extension so that also IPv4 will be possible. IPv6 has a 40 byte over-head. The next protocol on top of IP is UDP. UDP consumes 12 bytes of overhead information [32]. UDP supports only connectionless data exchange. The session management has to be processed in the higher layers, see RTP or application level. Finally, the last protocol is RTP, introducing additional 8 bytes [112]. In total we have now 60 bytes of overhead and only 15 byte of information. This problem is solved by adding up 4 to 20 AMR frames in one RTP packet. This increases the payload up to about 300 bytes per packet and reduces the overhead to 20%. Figure 7.15 shows the described protocol stack for PTT services.



(a) Protocol stack for PoC

(b) PoC state diagram

Figure 7.15: Push to talk over cellular: protocol stack and state diagram (ITU 59.9).

A further part of the traffic are the heartbeats. PTT services do not have a constant connection. The server has to determine if the user is still there waiting for data. This problem is solved by sending "empty" packets every 300sec of inactivity, the so called heartbeats.

7.3.2 Parameters for an Artificial Conversational Speech

There are several studies about the talk and silent periods in a human conversation [113]. The ITU P.59 [114] standard recommends four different parameters (see Table 7.2) to model artificial conversational

speech. The distribution of the talk duration, and talk spurt (the activity period of one user is called spurt in ITU P.59) duration is approximated with an exponential function, the pause parameter by an exponential function plus a constant. The activity factor for speech is assumed with $0.3 - 0.4$. The activity factor states the ratio between talk time and listening time. The likelihood of around 0.5 indicates silent periods of both speakers at the same time.

Parameter	Duration (sec)	Rate (%)
Talk spurt	1.004	36.53
Pause	1.587	61.47
Double talk	0.228	6.59
Mutual silence	0.508	22.48

Table 7.2: Parameters for artificial conversational speech.

7.3.3 PTT Model

We used the parameters found in the ITU standard and implemented this to a MATLAB simulation file. There were not enough active users in the network to find an empirical model, and so we decided to build a theoretical model for this type of traffic. The MATLAB script uses the ITU recommendation for the length of a connection and the information of the PoC standard for generating traffic in the activity phase of a user. The overhead of the RTP is also included in the simulation. In a recent publication [115] the delay limit for VoIP is given with 200 ms. Setting the tolerated delay to a fixed value is a strong simplification of the problem. In reality this parameter is linked to the packet loss as can be seen in the quality metrics for VoIP, e.g., E-model (ITU G.107) or Perceptual Evaluation of Speech Quality (PESQ, ITU P.862). With no real measurements to evaluate the packet loss we neglected the packet loss in the simulation setup.

This corresponds to $\approx$ 10 AMR-packets per RTP frame. The MATLAB simulations use these settings to evaluate data rate consumption of various numbers of users per cell. The simulation evaluates x-users per run and is rerun until the resulting total average data rate changes $\leq 0.1\%$ as compared to the last run.

The results are shown in Figure 7.16. From this figure we can conclude that the overhead depends on the working point of the system. This is due to the fact that small bursts suffer from the large header introduced by RTP. For 100 users, the 99% mark is found at 259.7 kbit/s. The simulation was performed for 100 users over 60 hours, using the parameters described above. Figure 7.17 depicts the number of parallel active users ($\equiv$conversations) and the number of parallel talking users ($\equiv$acting as data source) in the scenario. This is equal to the probability of parallel used lines (GPRS slots or UMTS bearers) for 100 users. In case of 100 users in a cell, a blocking probability of less than 1% needs at least six channels. Although GPRS is a packet-switched system, there are some hard limits to the assignment of slots for uplink and downlink . This could result in a higher blocking probability than shown in these results.

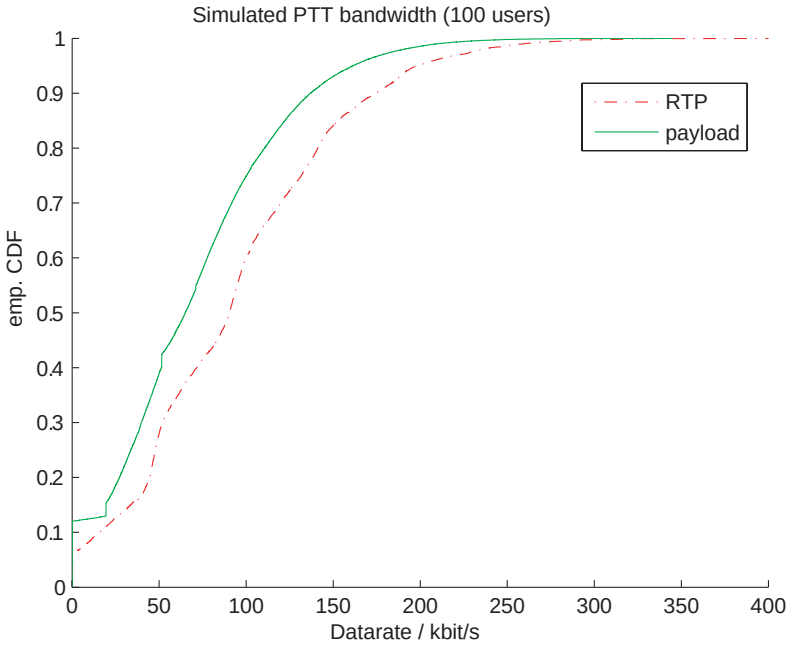
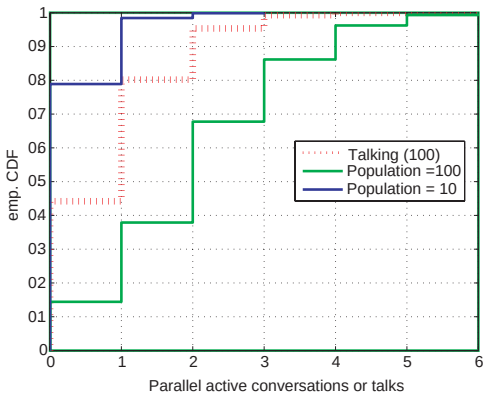
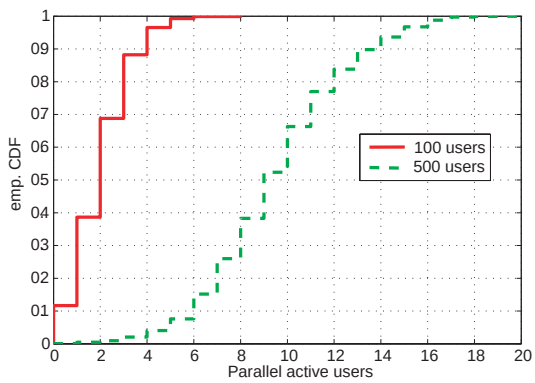


Figure 7.16: Simulated data rate for 100 PTT users.



(a) 100 users



(b) 500 users

Figure 7.17: Simulated number of parallel active users (Simulation time: 10^5 sec).

7.3.4 Results

In this chapter we provided basic information on PTT. As there were not yet enough active users to derive a model from the measurements, we used the ITU recommendations [114] to build a one way communication model for PoC services. Using this model we derived the minimum number of parallel channels for different user groups and plotted the result in Fig.7.17. With further monitoring and a growing popularity of this service we can adapt our MATLAB script to the datasets measured.

- 100 users with 1% blocking → 6 slots
- 500 users with 1% blocking → 16 slots

As a final remark we want to point out, that we did not consider the signaling traffic from the SIP protocol. Therefore, the results are only rough estimations for the minimum number of time slots consumed by this service.

A standard GSM cell offers 8 slots per frame. Each slot can carry up to two different voice calls. Therefore, the maximum load for a single cell, using the Erlang-B model, is equal to 8.9 Erlang. Based on the model parameters given in Table 7.2 and a blocking probability of 1% a GSM cell can serve up to 53 customers. The PTT simulation can serve 162 customers within one cell, improving the performance of the cell by a factor larger then three. Note in the PTT case we only use seven slots for the simulation as the first slot has to boardcast the signaling information in the GPRS cell.

7.4 Summary

IN this Chapter we presented new developed traffic models for the application layer of online games and PTT.

In a first part of the chapter we introduced new models for online gaming. We designed three different models for the different types of online games, RTS, FPS and MMOG and learned that the models for RTS and FPS can be realized straight forward while modeling the MMOG takes some considerable effort.

The PTT model we introduced is purely theoretical. It is based on the Nokia implementation of PTT and ITU recommendation of artificial conversations. A MATLAB simulation showed that in GPRS 100 users can be served with only six time slots if we consider a call blocking rate of 1%.

Self Criticism: We verified our traffic models with network parameters such as data rate or flow size. In the real world such models, especially if we consider gaming, should allow operators to improve the mean opinion score of their customers. The models here lack this kind of verification.

The PTT model was based on a theoretical approach only. Due to the lack of interest from the subscribers we were not able to record traces in the network. The model should therefore be accepted with reservations.

Another interesting point is the impact of the traffic aggregation of different online gaming applications onto the core network. However, the traces only showed small gaming activity, in terms of the

data rate. The only possibility would have been to use a simulation based approach, which is very complex as radio and core-network component must be simulated accurately in order to understand the detailed effects. We did not find the time and the appropriate tool to do this analysis.

Chapter 8

Conclusions and Outlook

Contents

IN this thesis we focus on the measurement, analysis and modeling of packet-switched user related traffic in a 3G network. The area of traffic modeling has a strong impact on network operations, planning and optimization. To design accurate models we evaluated live traffic from a mobile operator in Austria. This is an important improvement, since most traffic models for cellular mobile networks are based on extrapolations of models generated from wired measurements. This chapter summarizes the contributions and achievements of this thesis and specifies open points for further works.

8.1 Conclusions and Results from this Thesis

THIS work is based on measurements from a 2G and 3G live network of an operator in Austria over a time span of more than three years (2004 - 2007). We analyzed the traffic in three phases: first we extracted the general usage of UMTS and GPRS mobile Internet accesses, secondly we modeled traffic on the flow level for the HTTP, WAP, TCP, and UDP separately for UMTS and GPRS, and thirdly we introduced source level models for the main applications of the mobile networks.

The collection of traces on a per subscriber basis and with the feature to correlate several different traces was only possible due to a special measurement system used. This system, called METAWIN, is capable to trace all the packet-switched traffic of a 3G network on different interfaces (Gn, Gi, Iu, and Gb) and to correlate them afterward. Our work is mainly based on traces at the Gn Interface, where each packet can be assigned to a specific user which allowed a deeper understanding of the user population found in the network.

With the trace system in place we started to dissect the global traffic into specific applications. We found that UMTS and GPRS subscribers behave differently, e.g., there are more WAP users on GPRS but the UMTS users generate much more traffic. The service mix for the top application evolved throughout the monitoring period. The share of WAP decreased while HTTP increased. At the end of our monitoring phase the service shares for UMTS and GPRS were very similar. For a better understanding of the user-population in the network we split the users up according to their service usage. It turned out that the majority of the subscribers use packet-switched transmission for WAP services and other mobile applications. However, the majority of the traffic is generated by another group: the Internet enabled terminals. These subscribers represent only around 10% of the population but nearly 80% of the traffic seen in the network. In this first analysis we already found strong virus activity, which we had to take into account in the analysis. We conclude this part of the work with a detailed analysis of the PDP-context. From this step we learn that there are effects, although originating from a few users, which can bias the entire measurements.

With the top services defined we analyzed the flow length of TCP/UDP, HTTP and WAP. The TCP and HTTP flows did show heavy-tail behavior; this is expected as such behavior is also present in wired measurements. However, the mobile applications WAP 1.x and WAP 2.0 did not show heavy-tail behavior and as most of the UDP traffic originates due to WAP also UDP did not have a heavy-tail distribution.

We then modeled the flow lengths of the different applications of analytic distributions that provide a closed form. It turned out that the log-normal distribution best fitted all applications without heavy-tails. We split the distribution of HTTP into a body part, and a heavy-tailed tail, which we modeled

as Pareto Distribution. The distributions for GPRS and UMTS are nearly identical for one service. The main finding here was the fact that the flow properties depend on the service which generated it rather than on the technology with which it was transmitted.

Next we moved from the transport layer (e.g. TCP flows) towards the application layer in the form of source level traffic models. We first evolved existing models for applications such as HTTP, FTP, and e-mail. The model for HTTP is based on the model from Choi [89] and updated to parameters from our measurements. Again there was nearly no difference between the parameters extracted from GPRS or UMTS traces.

The new traffic model for e-mail extends existing models with an authentication part and data on the e-mail sizes. This interactive part has a strong impact on the flow properties in networks which feature high RTTs, as we have shown with an ns-2 simulation. We showed that the e-mail size grew by one order of magnitude in the last ten years.

In order to provide models for emerging services we developed new models for services such as Online Gaming and PTT. We designed three different models for the different genres of online gaming: FPS, RTS, and MMOG. The models for FPS and RTS run on top of UDP and were therefore quite simple in design. The model for the MMOG *World of Warcraft* runs on top of TCP, which needed much more effort to implement. All three models were implemented in ns-2 in order to test their the accuracy.

The last source level traffic model we designed addresses PTT traffic. We implemented this model based on the specifications from Nokia and the ITU recommendations for artificial conversations.

This summarizes the highlights of this thesis. We now have to address some notes to the interested reader. The traffic in all time periods showed strong virus activity and other malicious traffic. To the best of our knowledge, we filtered these patterns from the traffic.

8.1.1 Results seen from an Operator's Point of View

In contrast to the first part of the conclusions where we discussed the results from a scientific point of view, in this part we want to focus on important results for the mobile operator himself.

The traffic mix is dominated by less than 10 services. HTTP is the largest one generating more than 60% of the volume share. This result underlines the need for the operator to optimize his network for web traffic, e.g., implementing a caching node at the Gi interface. In contrast to a normal caching node this system should support a dynamic change of TCP connection settings towards the UE in order to improve even further the download times.

It turned out that there were strong probing activities ongoing in the network. From a scientific point of view we excluded these patterns as we only were interested in non-malicious user activities. However, stateful devices, e.g., GGSN, Firewall or the web proxy, which track connections could easily be overloaded if only dimensioned to handle the amount of non-malicious traffic. Even worse is the problem due to the fact that data packets traveling between two UEs of the same operator do not leave the GGSN and therefore do not pass the security systems on Gi. The probing stays undetected for the operator. It is important also to loop the internal traffic of the network through a firewall located at the GGSN.

The flow length for TCP traffic did show heavy-tailed behavior. It is important to consider this

fact when running network simulations for performance optimizations of the core network components.

The extension for the traffic model of POP3 did show that wireline traffic models can fail in mobile environments with their high RTT values. The network planing department of the operator has to verify that the traffic models of the simulation tools in use fit the RAN under test.

8.2 Open Points and Outlook for Future Work

IN this thesis we addressed the modeling of packet-switched traffic in cellular mobile networks. During this work we discovered several points and open issues for further research.

A large topic is the filtering of data, especially the question of how to filter malicious traffic while keeping the good traffic. There is recent work on this topic for wired connections and it has an even stronger impact to mobile networks in which mobility issues interact with the traffic patterns. Certainly this topic cannot be solved easily and may be worth another thesis.

Regarding the flow analysis we relied on methods from wired networks. However, there is no research upon the topic how flows interact with the UMTS core network topology and the radio access network. A high error rate could lead to strong problems for short TCP connections where the congestion avoidance methods will react unpredictable to packet loss. As the 3G network has a tree-like structure the number of parallel open TCP connections could be limited through some devices. These are only two aspects that may change the focus in analysis of mobile core network traffic away from heavy-tails.

In this part of the work we encountered the problem of fitting heavy-tail distributions. As we are interested in both the tail, for core network performance simulation and the body for login and RTT analysis, we searched for a way to fit both parts at once. However we did not manage a setup of an automatic fitting algorithm that was able to fit body and tail to one function, e.g., a linear tail (in loglog scale) and a log-normal body (in linear scale). The main problem originates from the fact that an error metric based on the distance between the two curves is biased by the high values of the tail. In addition to this a heavy-tail effect can exist only on a part of the curve as the measurement sample is limited in size. Even a small error in this value will render the fit for the body useless as the tail mainly characterizes the function. All effort could not match separation values based on visual inspection. We think that this is an major open point for further research, as a stable way to estimate such distribution allows for a more automatic analysis of network traces, e.g., fitting flows separated by the destination port or the service, respectively.

A third large area is the way in which users experience services or applications. We think that future modeling of traffic should not only focus on network aspects, like a comparison of simulated and measured data rate in order to prove accuracy, but also on parameters that are important for the user experience. Such models would allow the operator to optimize the network according to the need of the user in order to gain higher service quality with fewer resources.

All the topics above are large areas of research and beyond this work. We think that this thesis provides a good basis for further research conducted on these topics.

Appendix A

Universal Mobile Telecommunications System (UMTS) Protocol Stack

In UMTS the lower layers rely on the Asynchronous Transfer Mode (ATM). This allows for a simple integration of high speed optical fiber systems as physical layer. The ATM protocol connects RNC, SGSN, GGSN, and GMSC. For user data the ATM tunnels even reach up to the NodeB. The CS domain uses the AAL2 version of ATM which is connection oriented. AAL2 supports the transmission and multiplexing of many real-time data streams, offering a low delay, small jitter and less bandwidth fluctuation. The PS domain uses the ATM Adaptation Layer v5 (AAL5). It is connection-less and implements more or less a best-effort approach, suitable for non- real-time services such as Internet traffic. The AAL5 layer does not support connection management. Therefore, the PS domain in UMTS needs an additional layer using the GTP. The GTP protocol builds up a user specific data tunnel between the GGSN and the RNC. At the RNC the GTP protocol is converted over to the Packet Data Convergence Protocol (PDCP). The PDCP supports a more efficient coding of the headers, which is suitable for the radio link where resources are expensive.

A-1 The Protocols

Figure A.1 presents the UMTS protocol stack for all three domains. From this figure we learn that there is a second splitting into “Access Stratum” and “Non Access Stratum”. The idea is the separation of the services in the upper layers. The break down allows the UMTS network to migrate the different parts on its own. For example the change to an all IP network in the “Access Stratum” will have no effect on the “Non Access Stratum”. As long as the interface stays the same both systems can still interact.

A-1.1 Radio Access Network Part (RANAP)

The Radio Access Network Part (RANAP) handles the signaling between the UTRAN and the core network (Iu). It is responsible for tasks such as booking ATM lines, changing radio setup and so on (see [116]). All control procedures needed by the UTRAN can be executed by using instances from the three elementary classes:

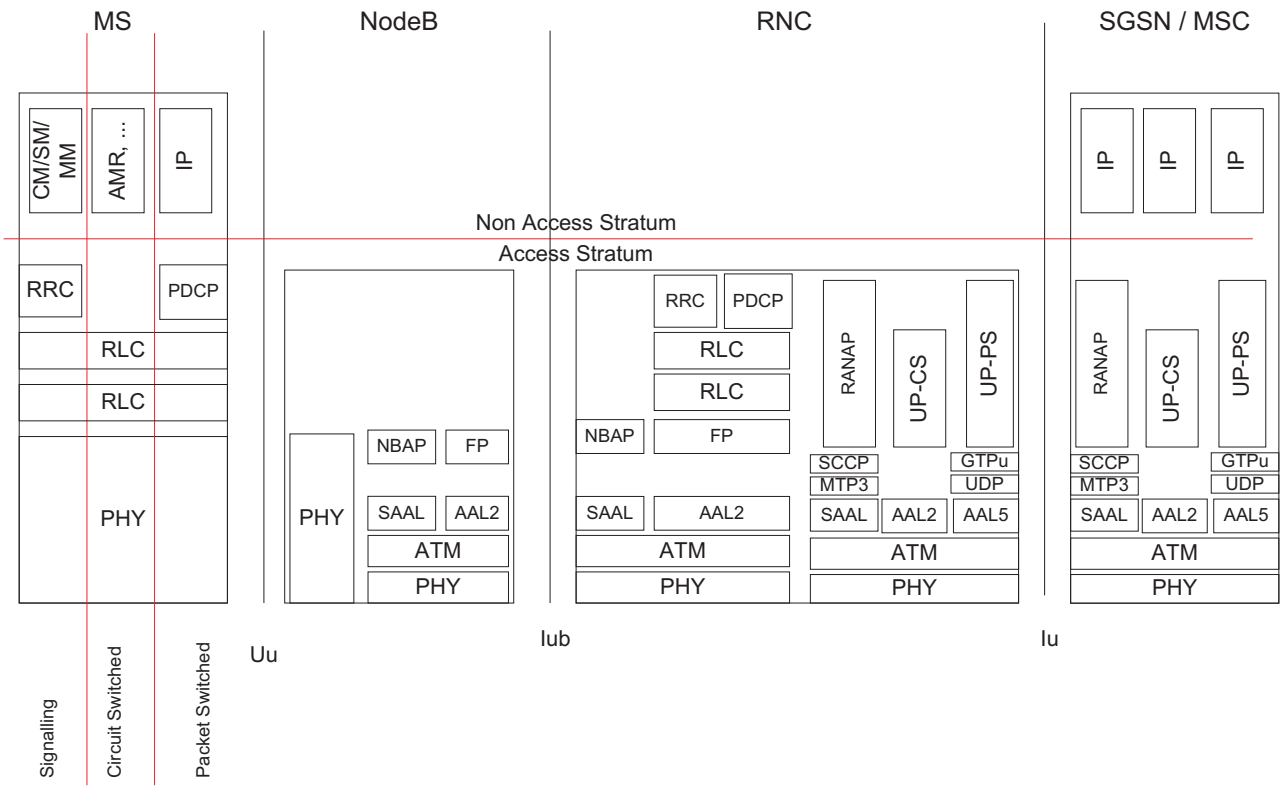


Figure A.1: UMTS protocol stack.

- general control service,
- notification service,
- dedicated control service.

All necessary functions can be constructed by using these three elementary classes. Examples for these procedures are

- Iu release,
- Overload control,
- RAB assignment.

A-1.2 Signaling Connection Control Part (SCCP)

The Signaling Connection Control Part (SCCP) delivers an abstraction between UMTS related layers and the used transport layers [117]. It allows to use different transport systems (ATM, IP). The main functions are:

- connection-less and connection-oriented extension to MTP,
- Address translation,
- Full OSI layer 3 compatibility,

- Below SS7 Protocol.

A-1.3 GPRS Tunneling Protocol (GTP v0)

The GPRS Tunneling Protocol (GTP v0) is the main protocol in the core network. It allows the end users in a GPRS or UMTS network to move between different cells while having continuing connects to the Internet. This is achived by transmitting the subscriber's data from the current subnetwork to the GGSN. It is used for connections between RNC, SGSN and GGSN. The data payload is attached to the GTP headers (8Bytes). It can handle signaling and data traffic [118, 119]. The header of the GTP v0 protocol is shown in Figure A.2.

The GTP-C(ontrol) is used to transport control information. It transmits GPRS mobility management messages between GSNs. Logically GTP-C is attached to the GTP-U(ser) tunnel - physically it is separated. The main functions are:

- Create/Update/Change PDP Context,
- Echo Request/Response,
- RAN Information.

The GTP-U(ser) is used to transport user data. It basically hides terminal mobility from IP layer of the user supporting the reordering of Transport-PDUs. The used TEID is always unique. The main functions are:

- Data Transmission,
- Tunnel Setup/Release/Error,
- Echo Request/Response.

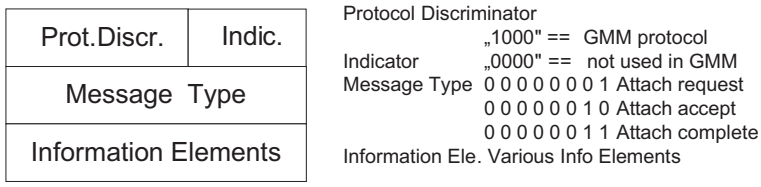


Figure A.2: GTP header and description.

A-1.4 GPRS Mobility Management (GMM)

This protocol is defined in [52]. It offers in UMTS the same functions as in GPRS: managing the mobility of the terminals. This protocol was designed to reduce the number of terminals in active state consuming radio resources. Therefore, three states were defined:

- Idle,
- Ready,

- Standby.

The transition between these states is initialized by well-defined events. A normal mobile sending data will be in the ready state. After a time period (set timeout) of not sending data the mobile will drop to standby. The state indicates that the mobile is expected to get active again. If there is no data transmission up to a second timeout the mobile will finally drop to the IDLE state. The algorithm is known by the RNC and the mobile terminal. Therefore, we do not need any signaling to initialize the state transitions. The following Figure A.3 presents the state transmission diagram.

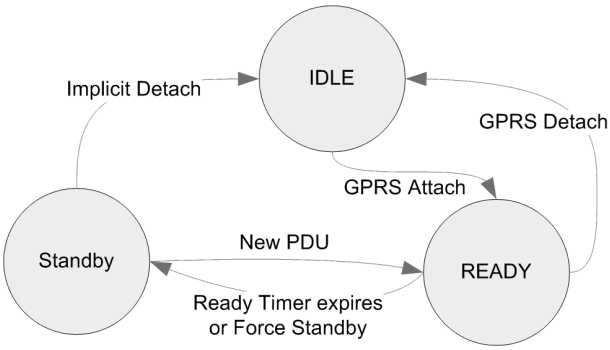


Figure A.3: States of the GMM protocol.

However if a mobile terminal tries to change its state it has to send a signal to the higher instances in the core network. GMM offers this functionality. It can handle basic procedures in the attachment process, such as “attach”, “accept”, “request”, and “complete”.

A-2 Logical channels

There are four different control channels in UMTS: The Broadcast Control Channel (BCCH) is a down link channel to spread the cell information (i.e. free scrambling codes, code for the random access channel, and so on). The PCCH (paging control channel) is a down link channel to accomplish paging of mobiles in a cell. The DCCH (dedicated control channel) is a bidirectional channel to establish point-to-point connections between a UE and the UTRAN. It is used for Radio Resource Configuration (RRC) procedures. The Common Control Channel (CCCH) is also a bidirectional transport channel, which is used to exchange control information between the UE and the UTRAN. UMTS supports two types of data channels: The Dedicated Traffic CHannel (DTCH) is a bidirectional point-to-point channel assigned to a unique UE for user data transmission. The Common Traffic CHannel (CTCH) is similar to the DTCH for multicast (point-to-multipoint) transmissions.

A-3 Transport channels

There are two types of transport channels: dedicated and common channels. A dedicated channel is assigned to one specific user while the common channels can be used by all users in one cell [120]. The only dedicated channel is the DCH. It is used to transport the user data and signaling from higher network layers. UMTS knows six types of common channels. The function of the common channel

differs from GSM, as these channels also can be used to transport PS data. The Broadcast CHannel (BCH) is used to transmit cell information broadcasts from the RNC. The channel mapping shows that this channel is used by the BCCH. The BCH is downlink only. The Forward Access CHannel (FACH) carries the user control information. The FACH is needed by the mobile stations to establish a radio link. Therefore it uses a physical channel with the largest scrambling code (low data rate, good SNR) and no power control. As there is no power control for this channel the inter cell interference from this channel is quite strong. The FACH channel only supports downlink communication. The Paging CHannel (PCH) is used to accomplish the paging procedures (call setup initiated by the RNC). The channel only supports downlink communication The Random Access CHannel (RACH) is an uplink channel that, together with a FACH, can be used to set up a radio link. Some operators also use a RACH/FACH combination to transmit a small amount of data. The common packet channel is an extended version of the RACH enabling PS services that need to transmit several consecutive frames. It is used again together with the FACH. The channel supports only uplink communication. The Downlink Shared CHannel (DSCH) can (it is an optional channel) be used to transmit user and control data which should be readable by several mobile stations. The channel is downlink only. The functions are similar to the FACH, but the channel also supports a power control. The channel has to be paired with a DCH. The channel mapping is shown in Figure A.4.

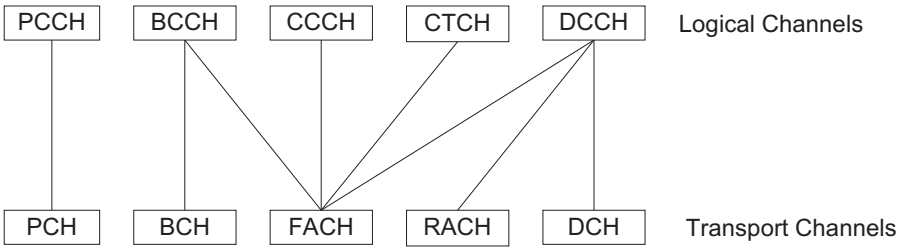


Figure A.4: UMTS channel mapping

A-4 Roaming

Roaming is one of the major features in a mobile network. Up to now we only discussed mobility in terms of intra network mobility (e.g. the user moves from one cell to another). However, there is also the case in which a user moves from one operator to another. In their home country, users are normally forced to stay with their operator. In foreign countries the user has to choose a new network that is not directly connected to the home network. The mobile network has to solve several tasks in order to establish this connection. First of all the user has to authenticate himself to a network which does not have an HLR entry for him. The problem can be solved by accessing the home network through a Virtual Private Network (VPN). In this scenario the foreign RNC and SGSN will be used to establish the connection. The used GGSN is in the home network. In this case the GTP protocol is vital. It can hide the fact that the underlying traffic flow has to go abroad. This approach also has the benefit of unified billing. A user can be directly billed correctly, as the traffic is counted by the GGSN and the billing tickets are generated in the correct network. Another problem that often arises is the usage of network intern service and the billing. As some special services can only be used

from inside the network. However, there is an option to use the foreign GGSN. This will be replaced in IMS, which offers the possibility to route traffic and signaling separately. The signaling can be routed as it is today while the real data traffic can go directly to the Internet. This will result in a performance gain.

Appendix B

General Packet Radio Service (GPRS) Packet Switched Core Network.

GPRS was initially standardized in GSM phase 2+. Today, in 2007, it is hosted by the 3GPP. The integration of GPRS into GSM was introduced in a very smooth way, the physical channels stayed unchanged and most of the infrastructure was reused. The only two new nodes introduced were the GGSN and the SGSN.

GPRS offers packet-switched IP based services to users in GSM environments. The IP routing is available through the entire network beginning at the UE and ending at the GGSN. In contrast to GSM, where each active user occupies exactly one time slot, GPRS users can use up to eight time slots in order to boost their bandwidth.

B-1 System Architecture

The network elements of the GPRS core network are very similar to the elements found in UMTS. Figure B.1 displays all interfaces and core nodes of a GPRS network. In order to make the integration of GPRS smooth the Base Station Subsystem (BSS) extended the Base Station Controller (BSC) with the Packet Control Unit (PCU) which handles the new packet-switched signaling procedures. It converts data received via the Gb interface from the SGSN in order that it can be processed by the BSC. The second component of the BSS is the Base Station Transmitter (BST), which is only a relay station transmitting the information via the air interface. This element was also present in GSM.

The GGSN and SSGN nodes are already introduced in the UMTS Section 3.2. A more detailed description of the nodes can be found in the standard [52].

The GGSN, SGSN, HLR and VLR nodes have the same functions as in UMTS. Some of the interfaces connecting the nodes have names that differ from the UMTS scheme, as the protocol stack is not identical for these interfaces, e.g., Gb replaces Iu.

B-2 Protocol Stack

The protocol stack of GPRS is split into transmission and signaling planes. Figure B.2 depicts the transmission plane for GPRS [118]. As in UMTS the GTP protocol routes user-packets within a tunnel

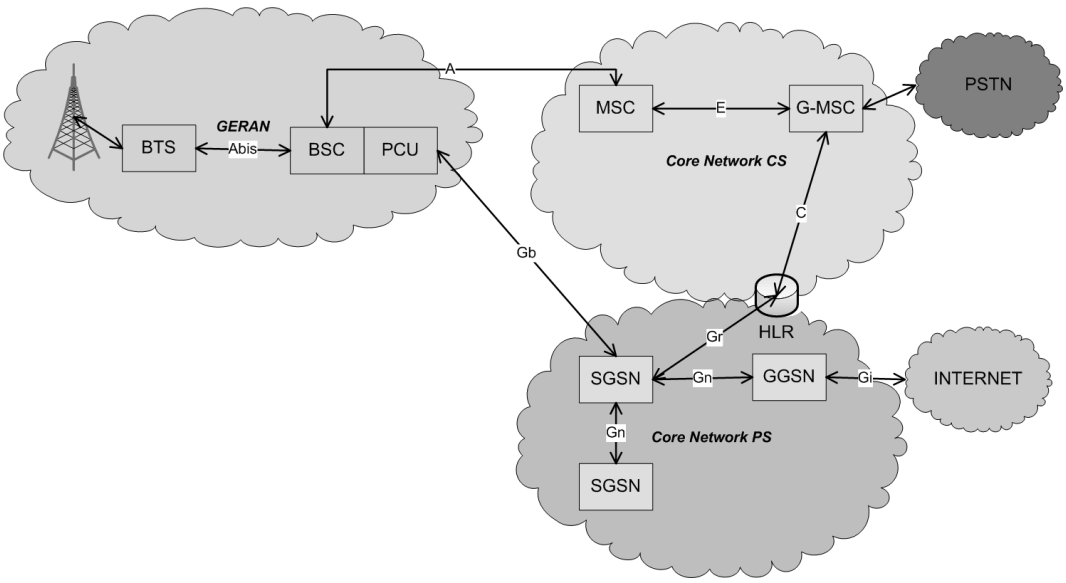


Figure B.1: GPRS core nodes and interfaces - 3GPP 23.060.

from the GGSN to the actual position of the UE.

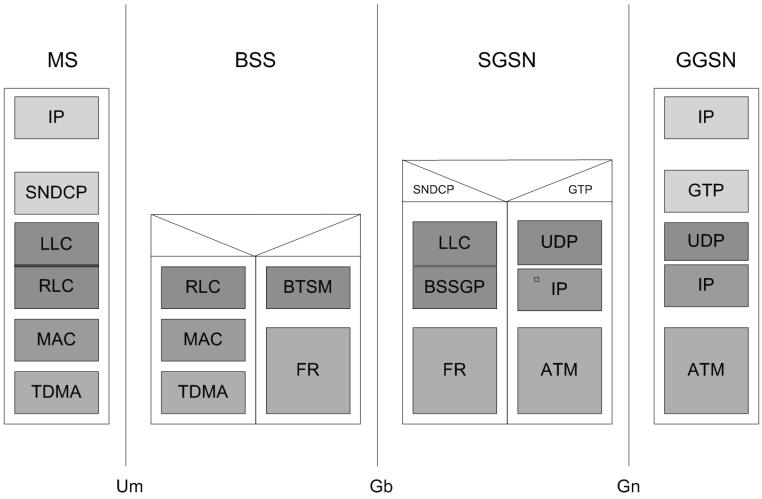


Figure B.2: GPRS stack: transmission plane.

The Sub-Network Dependent Convergence Protocol (SNDCP) is a transparent network layer protocol for IP data [121]. The protocol offers two important features: header and data compression. Therefore, it can improve performance as it reduces the amount of data transfered.

The Radio Link Control (RLC) protocol layer transfers PDUs from the Logical Link Layer protocol. The LLC offers a logical link from SGSN to the UE over Gb and Um interfaces. It covers flow control and ciphering for the logical link [122, 123]. Finally, the MAC protocol takes care of the physical properties of the radio channel.

Appendix C

Processing Modules

This appendix describes the most important MOTRA modules used in this thesis including their restrictions. These modules are called metrics or HRO (human readable output). The core program of the MOTRA handles the packet deciphering, recording and basic manipulation, e.g. adding the IMSI to each stored GTP packet. The modules are needed to interpret the stored data.

C-1 Metric: `Gn-Duration`

This metric extracts the following information, each in one file, from a Gn trace: PDP-context, TCP and UDP connection table, and routing area updates. The PDP-context file reports duration, volume up and down with and without signaling, and hashed subscriber id. The connection table reports type of protocol, e.g., UDP or TCP, duration, volume up and down with and without signaling, and hashed subscriber id. The routing area file holds the information with which SGSN the subscriber was connected.

The files are linked via an index that is basically a numbering of the PDP-contexts. Merging PDP-context with the connection table allows subscriber analysis at the IP level. This information splits the traffic into GPRS and UMTS parts of the network.

The service mix is extracted from the connection table. A service is defined by the server port used by this application, e.g., TCP:80 traffic is assumed to be HTTP traffic. Special servers inside the network of the operator are identified via their IP address, e.g., the video streaming servers, WAP 1.0 gateway, WAP 2.0 gateway, etc.

For bandwidth figure we assume an average distribution of the volume within the PDP-context or IP connection. The resulting bandwidth is calculated by adding up all instantaneous connections. A comparison with *Gi-Timeseries* verified this method to have an error below one percent for a time unit of 5 min.

The metric has some system inherent restrictions. Only if the start of the PDP-context lies within the analyzed trace, it is recognised. Therefore, in the beginning there is some warming-up phase in process which takes normally around 30 min. Open contexts at the end of a trace will be cut by the measurement system.

C-2 Metric: `Gi-Timeseries`

The metric extracts a time series from Gi traces. In defined time bins the number of bytes transfered in up and downlink and split for the top ten services is recorded to a database. The metric is used for service mix analysis of the whole network.

There are no system inherent restrictions to this metric. However, a set time bin is equal to an averaging process, which reduces the peak values. Therefore, granularity should be set to 1s.

C-3 Metric: `PCAP-Output`

The metric dumps traces to a hard disk in PCAP format. The cut payload is filled up using pseudo data. The metric serves as input feeder for `tcptrace` analysis regarding HTTP parameter extraction.

There are no restrictions to this metric.

C-4 Metric: `tcptrace`

This metric adds meta data to the PCAP format readable by `tcptrace`. The information added allows our modified `tcptrace` module to analyze the TCP information with respect to the terminal radio type, the cell information and the attached SGSN. The metric serves as extended input feeder for the analysis in Chapter 4.

There are no restrictions to this metric.

Appendix D

Analytical Distributions

D-1 Normal Distribution

The normal distribution, also called Gaussian distribution, is defined by two parameters: the mean (μ) and the standard deviation (σ), location and scale, respectively. Due to the central limit theorem the normal distribution is quite important and likely to occur in measured data samples.

A real valued random variable $\mathbf{X}$ following a normal distribution can be written as:

$$\mathbf{X} \sim N(\mu, \sigma^2). \tag{D.1}$$

The general form of the normal probability function (PDF) is:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\,\sigma^2}} . \tag{D.2}$$

Figure D.1 gives several PDFs for different settings.

The corresponding cumulative density function (CDF) is:

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp^{-\frac{(x-\mu)^2}{2\,\sigma^2}} \, dt \, . \tag{D.3}$$

The integral is the so called error function, which is solved numerically. The CDF can then be written as:

$$F(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right) . \tag{D.4}$$

In case of the normal distribution often the so called point percent point function (PPF) is used (often called Φ). The PPF is the inverse of the CDF. For the CDF we take a value x and calculate the probability that the variable we investigate is less or equal. The PPF computes for a given probability the corresponding value of x. The PPF does not exist in a simple closed form for the normal distribution. It is calculated numerically.

The general characteristics of this distribution are:

$$\begin{aligned} &\text{Mean:} && \mu, \\ &\text{Variance:} && \sigma, \\ &\text{Median:} && \mu. \end{aligned}$$

The maximum likelihood estimators for the parameters of this distribution are:

$$\widehat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i \,, \tag{D.5}$$

and

$$\widehat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \widehat{\mu})^2 \,. \tag{D.6}$$

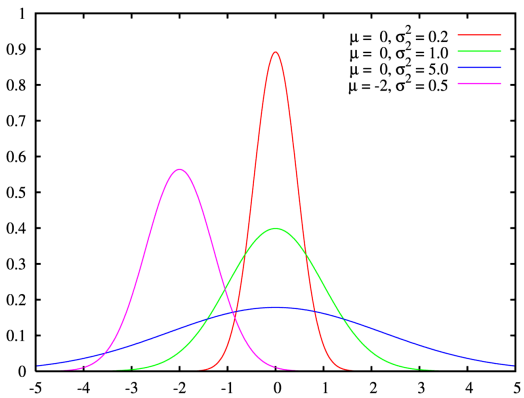


Figure D.1: PDF of normal distributions (source: Wikipedia).

D-2 Log-Normal Distribution

The log-normal distribution originates from any random variable whose logarithm is normally distributed. Given a normally distributed random variable Y, the variable $X = exp(Y)$ is log-normal. Log-normal distribution occurs in the case of many independent factors combined via multiplicative products. In the area of computer science often a so called $\log_2$ distribution is applied. It follows the same formulas as the log-normal distribution but is based on a base of 2.

The probability density distribution is not symmetric any more. The general formula for the PDF is:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp^{-\left[\frac{(\ln(x)-\mu)^2}{2\,\sigma^2}\right]} \,. \tag{D.7}$$

Figure D.2 shows several PDFs for different settings.

The cumulative density function (CDF) can be written using the error function:

$$F(x) = \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\ln(x)-\mu}{\sqrt{2}\,\sigma}\right) \,. \tag{D.8}$$

The general characteristics of this distribution are:

$$\begin{aligned}
&\text{Mean:} && \exp^{\frac{(2\mu+\sigma^2)}{2}}, \\
&\text{Variance:} && \left(\exp^{\sigma^2-1}\right)\exp^{(2\mu+\sigma^2)}, \\
&\text{Median:} && \exp^{\mu}.
\end{aligned}$$

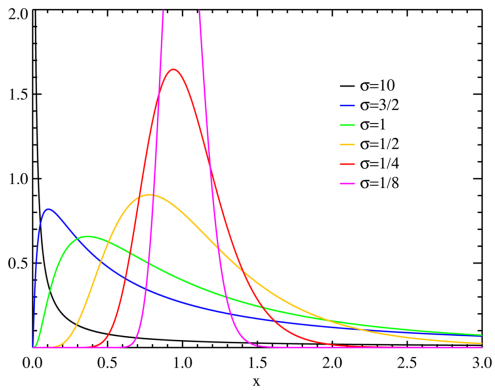


Figure D.2: PDF of log-normal distributions (source: Wikipedia).

The MLE function for this distribution is the same as for the normal distribution if instead of x the $\log(x)$ is used.

D-3 Exponential Distribution

The Exponential distribution is often used to model the inter arrival time between independent events originating from a source with a constant average rate. It has a continuous probability function.

The general formula for the PDF is:

$$f(x) = \begin{cases} \lambda \exp^{-\lambda x}; & c \geq 0 \\ 0 & else. \end{cases} \tag{D.9}$$

Figure D.3 shows several PDFs for different settings.

The cumulative density function (CDF) can be written as:

$$F(x) = \begin{cases} 1 - \exp^{-\lambda x}; & c \geq 0 \\ 0 & else. \end{cases} \tag{D.10}$$

The general characteristics of this distribution are:

$$\begin{aligned}
&\text{Mean:} && \lambda^{-1}, \\
&\text{Variance:} && \lambda^{-2}, \\
&\text{Median:} && \frac{\ln(2)}{\lambda}.
\end{aligned}$$

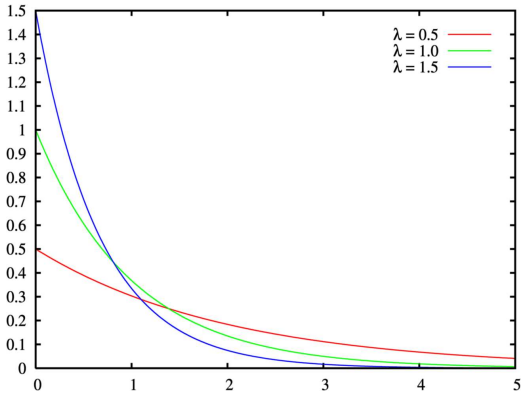


Figure D.3: PDF of exponential distributions (source: Wikipedia).

The MLE for the parameter of the exponential distribution is:

$$\widehat{\lambda} = \frac{N}{\sum_{i=1}^{N} x_i} \,. \tag{D.11}$$

D-4 Weibull Distribution

The Weibull distribution is often used in data analysis. Although it only has two parameters: shape (k) and scale (λ). It is able to reproduce the behavior of other distributions such as the exponential ($k = 1$) and the normal ($k = 2$). It is therefore able to model increasing or decreasing failure rates within the same set of distributions.

The general formula for the PDF is:

$$f(x) = \begin{cases} \frac{k}{\lambda}\frac{x}{\lambda}^{k-1} \exp^{-(\frac{x}{\lambda})^k} & x \geq 0;\ k = 0, 1, \ldots \\ 0 & else. \end{cases} \tag{D.12}$$

Figure D.4 shows several PDFs for different settings.

The cumulative density function (CDF) can be written as:

$$F(x) = \begin{cases} 1 - \exp^{-(\frac{x}{\lambda})^k} & x \geq 0;\ k = 0, 1, \ldots \\ 0 & else. \end{cases} \tag{D.13}$$

The general characteristics of this distribution are:

$$\begin{aligned} &\text{Mean:} && \lambda\Gamma\left(1 + \tfrac{1}{k}\right), \\ &\text{Variance:} && \lambda^2\Gamma\left(1 + \tfrac{2}{k}\right) - \mu^2, \\ &\text{Median:} && \lambda \ln 2^{\frac{1}{k}}. \end{aligned}$$

To find the MLE for the parameters of the Weibull distribution we have to solve the following equation for $\widehat{k}$:

$$0 = \frac{\sum_{i=1}^{N} x_i^{\widehat{k}} \ln x_i}{\sum_{i=1}^{N} x_i^{\widehat{k}}} - \frac{1}{n}\sum_{i=1}^{N} x_i^{\widehat{k}} - \frac{1}{\widehat{k}} \,. \tag{D.14}$$

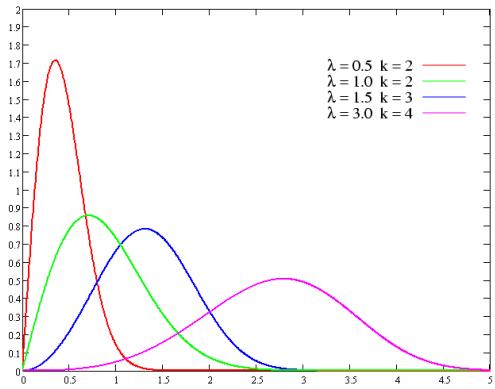


Figure D.4: PDF of Weilbull distributions (source: Wikipedia).

If this equation was solved iteratively we can calculate $\widehat{\lambda}$ as:

$$\widehat{\lambda} = \left(\frac{1}{N} \sum_{i=1}^{N} x_i^{\widehat{k}} \right)^{-\frac{1}{\widehat{k}}} . \tag{D.15}$$

D-5 Pareto Distribution

The Pareto distribution was originally used to describe the distribution of wealth among a population. It showed that a larger portion of the wealth of any country is owned by a small percentage of people. It also applies to file size distribution of Internet traffic using the TCP protocol, values of oil reserves and frequencies of words to name only a few of the major applications. It belongs to the group of power law probability distributions. The parameters k and x_m (also called α) have to be larger than zero and the support of x exists only for values larger than x_m In case $k < 2$ the distribution has no finite variance and if $k < 1$ even no finite mean. In this case the Pareto distribution is heavy tailed. Such distributions have a uncommonly high probability for very large events. In this case one can not neglect the tail of the distribution. This is why such a distribution is called heavy tailed.

The general formula for the PDF is:

$$f(x;\, k,\, x_m) = \begin{cases} \frac{k x_m^k}{x^{k+1}} \text{ for } x \geq x_m & x \geq 0 \\ 0 & else. \end{cases} \tag{D.16}$$

Figure D.5 shows several PDFs for different settings.

The cumulative density function (CDF) can be written as:

$$F(x) = \begin{cases} 1 - \left(\frac{x_m}{x}\right)^k & x \geq 0 \\ 0 & else. \end{cases} \tag{D.17}$$

The general characteristics of this distribution are:

$$\begin{array}{ll} \text{Mean:} & \frac{k\,x_m}{k-1} \text{ for } k>1, \\ \text{Variance:} & \frac{x_m^2 k}{(k-1)^2(k-2)} \text{ for } k>2, \\ \text{Median:} & x_m \sqrt[k]{2}. \end{array}$$

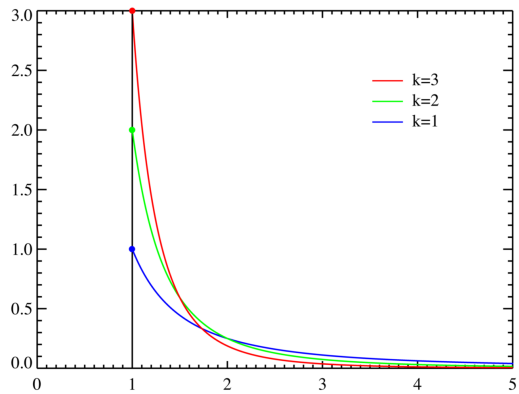


Figure D.5: PDF of Pareto distributions (source: Wikipedia).

The MLE for the parameters of the Pareto distribution are:

$$\widehat{x}_m = \min_i x_i \,, \tag{D.18}$$

and

$$\widehat{k} = \frac{N}{\sum_{i=1}^{N} \ln x_i - \ln \widehat{x}_m} \,. \tag{D.19}$$

D-6 Generalized Extreme-Value Distribution

The Generalized-Extreme-Value distribution was designed to combine the Gumbel, Frechet and Weibull families, which are also called extreme value distributions of Type I, II and III. It can model the limit distribution of independent and identically distributed random variables and therefore approximated the maxima of long sequences of random variables. The distribution has two parameters: μ, σ and ξ. Except σ, which is only defined starting at zero the parameters are defined for all real valued numbers. However there are some restrictions in the relations between the three parameters.

The general formula for the PDF is:

$$f(x,\mu,\sigma,\xi) = \begin{cases} \frac{1}{\sigma}\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-\frac{1}{\xi}-1} \exp^{-\left[1+\xi\frac{x-\mu}{\sigma}\right]^{-\frac{1}{\xi}}} & x>0 \\ 0 & else. \end{cases} \tag{D.20}$$

The cumulative density function (CDF) can be written as:

$$F(x) = \exp^{-\left[1+\xi\frac{x-\mu}{\sigma}\right]^{-\frac{1}{\xi}}} \,. \tag{D.21}$$

The general characteristics of this distribution are:

$$\begin{aligned}
&\text{Mean:} && \mu - \tfrac{\sigma}{\xi} + \tfrac{\sigma}{\xi}\Gamma(1-\xi),\\
&\text{Variance:} && \tfrac{\sigma^2}{\xi^2}(\Gamma(1-2\xi) - \Gamma(1-\xi)^2),\\
&\text{Median:} && \mu + \sigma\tfrac{\ln^{-\xi}(2)-1}{\xi}.
\end{aligned}$$

D-7 Cauchy Distribution

The Cauchy distribution is a continuous probability distribution. It is named after the French mathematician Augustin Cauchy The distribution has two parameters: a and b. The parameters are defined for all real valued numbers.

Often a truncated version of this distribution is used to model parts of a curve.

The general formula for the PDF is:

$$f(x; a, b) = \frac{1}{\pi}\frac{1}{a^2 + (x-b)^2}. \tag{D.22}$$

The cumulative density function (CDF) can be written as:

$$F(x) = \frac{1}{\pi}\arctan\left(\frac{x-b}{a}\right) + \frac{1}{2}. \tag{D.23}$$

The general characteristics of this distribution are undefined as the PDF is of form $1/x^2$. That leads to the integral $\int_{-\infty}^{\infty} 1/x dx$ for the mean. This integral has a non finite value, the mean therefore is not defined.

$$\begin{aligned}
&\text{Mean:} && -,\\
&\text{Variance:} && -,\\
&\text{Median:} && a.
\end{aligned}$$

D-8 Evaluating Parameter Fits

In the thesis we fitted different distributions to our dataset. The process of fitting generates a set of parameters for each distribution. In the follow up step it is now important to find the distributions which model the underlying process best.

There exists various different metrics to analyze the quality of a fit. Starting from simple distance metrics, e.g. the sum of the absolute differences between the modeled PDF and estimated PDF, up to more sophisticated methods like the KS-test. In this thesis we used two approaches: the Kulbeck-Leibler (KL) distance and the Kolmogorow-Smirnow (KS) test.

The KL distance was used to find the distribution that fitted the dataset best. A larger value of the KL-distance indicates a larger error between model and data.

The KL distance is a measure between a given probability distribution P and an arbitrary distribution Q. Often P represents some reference data obtained by measurements and Q is generated by a model approximating P. Eq. (D.24) shows the KL distance metric

$$\Delta_{KL}(P\|Q) = \sum_{i,j} P(i,j) \cdot \ln\frac{P(i,j)}{Q(i,j)}. \tag{D.24}$$

Note that the KL distance is not symmetric. We used a symmetric version of the metric proposed in [124]. It is shown in Eq. (D.25)

$$\Delta_{sKL}(P,Q) = \frac{1}{2} \cdot (\Delta_{KL}(P\|Q) + \Delta_{KL}(Q\|P)). \tag{D.25}$$

The output of Eq. (D.25) is $\Delta_{sKL}(P,Q) \geq 0$, with equality if P equals Q. As the equation uses a logarithmic fraction, bins of value zero in either P or Q will lead to an undefined result.

However the KL distance does not, if not equal to zero, tell if model and data were likely drawn from the same distribution. For this we used the KS-test in a second step.

The KS test is a non parametric and very stable method to compare for equality of one-dimensional PDFs and a data samples. If we again consider a given probability distribution P and an arbitrary distribution Q, the first step of the KS test searches for the largest difference between Q and P:

$$\max_x d_i = |Q(x) - F(x)| \,. \tag{D.26}$$

In case d_i is below a critical value d_α the dataset P follows the same distribution as Q. The values for d_α are tabulated, for a large number of samples, e.g. more than 40, there exist simple approximations based on $\sqrt{n}$.

Appendix E

Abbreviations and Symbols

AAL2	ATM Adaptation Layer type 2
ACF	Auto-Correlation Function
ADSL	Asymmetric Digital Subscriber Line
ALCAP	Access Link Control Application Part
AMR	Adaptive Multi-Rate
APN	Access Point Name
ARQ	Automatic Repeat reQuest
AS	Access Stratum
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
AuC	Authentication Center
BCCH	Broadcast Control Channel
BSC	Base Station Controller
BSSGP	Base Station System GPRS Protocol
BTS	Base Transceiver Station
CC	Call Control
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
CN	Core Network
CRNC	Controlling RNC
CS	Circuit Switched
DCH	Dedicated Channel
DL	Downlink
DNS	Domain Name Service
DRNC	Drift RNC
DRNS	Drift RNS
DSCH	Downlink Shared Channel
EDGE	Enhanced Data Rates for GSM Evolution
EIR	Equipment Identity Register

EP	Elementary Procedure
ETSI	European Telecommunications Standards Institute
FDD	Frequency Division Duplexing
FEC	Forward Error Correction
FPS	First Person Shooter
FTP	File Transport Protocol
GERAN	GSM/EDGE Radio Access Network
GGSN	Gateway GPRS Supporting Node
G-MSC	Gateway Mobile Switching Center
GMM	GPRS Mobility Management
GPRS	General Packet Radio System
GSM	Global System for Mobile Communications
GTP	GPRS Tunneling Protocol
GWCN	GateWay Core Network
HLR	Home Location Register
HSPA	High Speed Packet Access
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPs	Hypertext Transfer Protocol Secure
IE	Information Element
IMAP	Internet Message Access Protocol
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IPv4	Internet Protocol (version 4)
IPv6	Internet Protocol (version 6)
ISDN	Integrated Services Digital Network
LLC	Logical Link Control
MAC	Media Access Control
MBMS	Multimedia Broadcast Multicast Service
ME	Mobile Equipment
MLE	Maximum Likelihood Estimation
MM	Mobility Management
MMOG	Massive Multiplayer Online Game
MOCN	Multi Operator Core Network
MS	Mobile Station
MSC	Mobile Services Switching Centre
MSISDN	Mobile Subscriber ISDN Number
MTP	Message Transfer Part
MTU	Maximum Transport Unit

NACC	Network Assisted Cell Change
NAS	Non-Access Stratum
NPC	Non-Playing Character
NRT	Non-Real Time
N-PDU	Network - Protocol Data Unit
OMC	Operation and Maintenance Center
OSI	Open Systems Interconnection
P-TMSI	Packet TMSI
PCM	Puls Code Modulation
PCU	Packet Control Unit
PDCP	Packet Data Convergence Protocol
PDF	Probability Density Function
PDP	Packet Data Protocol
PDU	Protocol Data Unit
PH	Phase Type
PLMN	Public Land Mobile Network
PoC	Push to Talk over Cellular
PPP	Point-to-Point Protocol
PS	Packet Switched
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAB	Radio Access Bearer
RAN	Radio Access Network
RANAP	Radio Access Network Application Part
RAT	Radio Access Type
RFC	Request for Comments
RLC	Radio Link Control
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RPC	Remote Control Protocol
RRC	Radio Resource Control
RTP	Real Time Protocol
RTS	Real Time Strategy
SAP	Service Access Point
SCCP	Signaling Connection Control Part
SCTP	Simple Control Transmission Protocol
SDH	Synchrony Digital Hierarchy
SDU	Service Data Unit
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SMB	Server Message Block

STM-1	Synchrony Transport Mode 1 (= 150MB/s Payload)
SGSN	Serving GPRS Support Node
SM	Session Management
SMTP	Simple Mail Transfer Protocol
SNAC	Shared Network Area Code
SNDCP	Subnetwork Dependent Convergence Protocol
SRNC	Serving RNC
TCP	Transport Control Protocol
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
TEID	Tunnel Endpoint Identifier
TMSI	Temporary Mobile Subscriber Identity
ToS	Type of Service
TRAU	Transcoder Rate Adapter Unit
UE	User Equipment
UEA	UMTS Encryption Algorithm
UESBI-Iu	UE Specific Behavior Information - Iu
UDP	User Datagram Protocol
UID	User Identification
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitors Location Register
VPN	Virtual Private Networks
WAN	Wide Area Networks
WAP	Wireless Application Protocol
WCDMA	Wideband Code Division Multiple Access
WSS	Wide Sense Stationary
XML	eXtensible Markup Language

List of Tables

List of Figures

Bibliography

[1] E. Feyerabend, H. Heidecker, F. Breisig, and A. Kruckow. Zugfunk. *Handwörterbuch des elektrischen Fernmeldewesens, Springer Berlin*, 2:871 – 874, 1927.

[2] Generaldirektion für die Post und Telegraphenverwaltung Bundesministerium für Handel und Verkehr. Versuchsbewilligung Nr. 58329, Dez 1930.

[3] G. Heine. *GPRS - Signaling & Protocol Analysis*. Inacon, 2002.

[4] H. Holma and A. Toskala. *WCDMA for UMTS, Radio Access For Third Generation Mobile Communications, Third Edition*. Wiley, 2004.

[5] TS 25942. *Radio Frequency (RF) System Scenarios*. ETSI/3GPP.

[6] M. Taferner and E. Bonek. *Wireless Internet Access over GSM and UMTS*. Springer-Verlag, 2002.

[7] TS 25.211. *IP transport in UTRAN*. ETSI/3GPP.

[8] B. Mah. An Empirical Model of HTTP Network Traffic. *Proceedings of IEEE INFOCOM 97*, 2:592–600, 1997.

[9] P. Barford and M. Crovella. Generating representative Web workloads for network and server performance evaluation. volume 26, pages 151–160, 1998.

[10] T. Halonen, J. Romero, and J. Melero. *GSM, GPRS and EDGE Performance: Evolution Towards 3G/UMTS*. WILEY, 2003.

[11] A. Klemm, C. Lindemann, and M. Lohmann. Traffic Modeling and Characterization for UMTS Networks. *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, 3:1741–1746, 2001.

[12] G. Giambene and C. Tommasi. 3G Wireless Core Network Dimensioning in the Presence of Self-Similar Traffics. *2nd International Symposium on Wireless Communication Systems, 2005.*, pages 223–227, Sept. 2005.

[13] W.E. Leland, M.S. Taqq, W. Willinger, and D.V. Wilson. On the Self-Similar Nature of Ethernet Traffic. *SIGCOMM Comput. Commun. Rev.*, 25(1):202–213, 1995.

[14] A. Clemente and J. Velez. Parameters for Tele-traffic Characterization in Enhanced UMTS. *COST 273; Barcelona, Spain*, pages 22–34, 2003.

[15] H. Graja, P. Perry, and J. Murphy. Development of a Data Source Model for a GPRS Network Simulator. *First Joint IEI/IEE Symposium on Telecommunications Systems Research, Dublin*, pages 1–10, 2001.

[16] T. Varga, H. Bert, and B. Sanders. Analysis and Modelling of WAP Traffic in GPRS Networks. *Proceedings of ITCSS Antwerpen*, 1:122–129, 2004.

[17] T. Berners-Lee, R. Fielding, and H.F. Nielsen. *RFC 1945 : Hypertext Transfer Protocol – HTTP/1.0*. IETF, 1996.

[18] F. Ricciato, P. Mely, J. Motz, and W. Fleischer. Traffic Monitoring and Analysis in 3G networks: Lessons learned from the METAWIN Project. *e&i Elektrotechnik und Informationstechnik*, 123(7-8):22–28, July 2006.

[19] METAWIN. Report on the Monitoring System (development and deployment). Project deliverable D1.1. Technical report, ftw., July 2004.

[20] F. Ricciato, F. Vacirca, and P. Svoboda. Diagnosis of Capacity Bottlenecks via Passive Monitoring in 3G Networks: an Empirical Analysis. *Computer Networks*, 57:1205–1231, March 2007.

[21] P. Svoboda and F. Ricciato. Composition of GPRS and UMTS traffic: Snapshots from a Live Network. *IPS MoMe 2006, Salzburg*, 4:42–54, Feb 2006.

[22] F. Ricciato, P. Svoboda, E. Hasenleithner, and W. Fleischer. On the Impact of Unwanted Traffic onto a 3G Network. In *SECPERU '06: Proceedings of the Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pages 49–56, 2006.

[23] P. Svoboda, F. Ricciato, and M. Rupp. Bottleneck Footprints in TCP over Mobile Internet Accesses. *Communications Letters, IEEE*, 11(11):839 – 841, November 2007.

[24] V. Paxson. Empirically-Derived Analytic Models of Wired-Area TCP Connections. *IEEE/ACM Transactions on Networking*, 3:226–244, June 1995.

[25] P. Svoboda, W. Karner, and M. Rupp. Modeling E-Mail Traffic for 3G Mobile Networks. In *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, page 6, Athens, Greece, September 2007. IEEE.

[26] P. Svoboda, F. Ricciato, W. Keim, and M. Rupp. Measured WEB Performance in GPRS, EDGE, UMTS and HSDPA with and without Caching. In *Proceedings of the The 8th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, page 6, Helsinki, Finnland, June 2007.

[27] P. Svoboda and M. Rupp. Online Gaming Models for Wireless Networks. In *Proceedings of the 18th Internet and Multimedia Systems and Applications*, pages 417–422, Grindelwald, Swiss, 2005.

[28] P. Svoboda, W. Karner, and M. Rupp. Traffic Analysis and Modeling for World of Warcraft a MMOG. In *Proceedings of the ICC 2007*, Glasgow, Schotland, June 2007.

[29] J. Postel. *RFC 791: Internet Protocol.* IETF, 1981.

[30] J. Reynolds and J. Postel. *RFC 1700: Assigned Numbers.* IETF, 1994.

[31] J. Postel. *RFC 793: Transmission Control Protocol.* IETF, 1981.

[32] J. Postel. *RFC 768: User Datagram Protocol.* IETF, 1980.

[33] L. Kleinrock. *Queuing Systems, Vol.1.* Wiley and Sons, 1975.

[34] L. Kleinrock. *Queuing Systems, Vol.2.* Wiley and Sons, 1975.

[35] M. Roughan, A. Erramilli, and D. Veitch. Network Performance for TCP Networks Part I: Persistent Sources, 2001.

[36] I. Khalifa and L. Trajkovic. An Overview and Comparison of Analytical TCP Models. *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04.*, 5:V–469–V–472 Vol.5, May 2004.

[37] D. Staehle, K. Leibnitz, and P. Tran-Gia. Source Traffic Modelling of Wireless Applications. Technical Report 261, Universität Würzburg, June 2000.

[38] Y. Joo, V. Ribeiro, A. Feldmann, A.C. Gilbert, and W. Willinger. TCP/IP Traffic Dynamics and Network Performance: A Lesson in Workload Modeling, Flow Control, and Trace-Driven Simulations. *ACM SIGCOMM Computer Communication Review*, 31(2):25–37, 2001.

[39] A. Erramilli, O. Narayan, A. Neidhardt, and I. Saniee. Performance Impacts of Multi-Scaling in Wide Area TCP/IP Traffic . *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 1:352–359 vol.1, 2000.

[40] V. Paxson and S. Floyd. Wide area traffic: the Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[41] H. E. Hurst. Long Term Storage Capacity of Reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770 – 799, 1951.

[42] M.E. Crovella and M.S. Taqqu. Estimating the Heavy Tail Index from Scaling Properties. In *Methodology and Computing in Applied Probability*, pages 55–79, 1999.

[43] Allen B. Downey. Evidence for Long-Tailed Distributions in the Internet. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pages 229–241. ACM Press, 2001.

[44] R. el A. Khayari, R. Sadre, and B.R. Haverkort. Fitting World-Wide Web Request Traces with the EM-algorithm. *Performance Evaluation, Special issue: Internet performance and control of network systems*, 52(2-3):175–191, 2003.

[45] D. Gross, J.F. Shortle, M.J. Fischer, and D.M.B. Masi. Difficulties in Simulating Queues with Pareto Service. *Proceedings of the Winter Simulation Conference, 2002.*, 1:407–415, Dec. 2002.

[46] A. Popescu. Traffic Self-Similarity. In *Proceedings of the ICT2001*, Bucharest, Romania, June 2001.

[47] P.R. Morin. The Impact of Self-Similarity on Network Performance Analysis. Technical report, Carleton University, December 1995.

[48] K. Park, G. Kim, and M. Crovella. On the Relationship between File Sizes, Transport Protocols, and Self-Similar Network Traffic. In *ICNP '96: Proceedings of the 1996 International Conference on Network Protocols*, page 171, Washington, DC, USA, 1996. IEEE Computer Society.

[49] A. Feldmann, A.C. Gilbert, P. Huang, and W. Willinger. Dynamics of IP traffic: A Study of the Role of Variability and the Impact of Control. *SIGCOMM Computer Communication Review*, 29(4):301–313, 1999.

[50] J. H. Dshalalow. *Advances in Queueing: Theory, Methods, and Open Problems.* CRC Press, 1995.

[51] TS 23.002. *Network Architecture.* ETSI/3GPP.

[52] TS 23.060. *General Packet Radio Service (GPRS), Service description.* ETSI/3GPP.

[53] TS 11.31. *Home Location Register Specification.* ETSI/3GPP.

[54] TS 29.016. *Serving GPRS Support Node SGSN - Visitors Location Register (VLR); Gs Interface Network Service Specification.* ETSI/3GPP.

[55] TS 11.032. *Visitor Location Register Specification.* ETSI/3GPP.

[56] TS 22.016. *International Mobile Equipment Identities (IMEI).* ETSI/3GPP.

[57] TS 25.213. *Spreading and Modulation.* ETSI/3GPP.

[58] TS 25.308. *High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2.* ETSI/3GPP.

[59] TS 04.08. *Mobile Radio Interface Layer 3 Specification*. ETSI/3GPP.

[60] R. Kalden, T. Varga, A. Woutersc, and S. Sandersd. Wireless Service Usage and Traffic Characteristics in GPRS Networks. Technical report, Technical Report Vodafone, 2004.

[61] R. Pang et al. Characteristics of Internet Background Radiation. In *Proceedings of the IMC'04, International Measurements Conference, Taormina, Sicily, Italy.*, October 2004.

[62] S. Staniford, V. Paxson, and N. Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium*, 2002.

[63] E. Brunner-Williams and B. Manning. *RFC 2929: Domain Name System*. IETF, 2000.

[64] A. Kivi. *Mobile Internet Usage Measurements - Case Finland*. PhD thesis, Helsinki University of Technology, 2006.

[65] B. Trammell and E. Boschi. *RFC 5103: Bidirectional Flow Export Using IP Flow Information Export (IPFIX)*. Internet Engineering Task Force (IETF), 2008.

[66] M. Allman, E. Blanton, K. Fall, and L. Wang. *RFC 3517: A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP* . Internet Engineering Task Force (IETF), 2003.

[67] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *RFC 2018: TCP Selective Acknowledgment Options*. Internet Engineering Task Force (IETF), 1996.

[68] K.C. Claffy, H.W. Braun, and G.C. Polyzos. A Parameterizable Methodology for Internet Traffic Flow Profiling. *IEEE Journal of Selected Areas in Communications*, 13(8):1481–1494, 1995.

[69] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the Characteristics and Origins of Internet Flow Rates. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 309–322, 2002.

[70] C. Ronald and Jr. Dodge. Skype fingerprint. In *HICSS '08: Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 484, 2008.

[71] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli. Revealing Skype Traffic: when Randomness Plays with You. *SIGCOMM Computer Communications Review*, 37(4):37–48, 2007.

[72] D. Rossi, S. Valenti, P. Veglia, D. Bonfiglio, M. Mellia, and M. Meo. Pictures from the Skype. In *Proceedings of ACM SIGMETRICS Demo Competition*, page 7, 2008.

[73] K. Chen, C. Huang, P. Huang, and C. Lei. Quantifying Skype User Satisfaction. *SIGCOMM Comput. Commun. Rev.*, 36(4):399–410, 2006.

[74] K. Thompson, G. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *Network, IEEE*, 11(6):10–23, November 1997.

[75] R. Schlittgen. *Zeitreihenanalyse*. Oldenbourg, 1995.

[76] R. Davis and P. Brockwell. *Introduction to Time Series and Forecasting*. Springer-Verlag New York, 2002.

[77] J. Alderich. R. A. Fisher and the Making of Maximum Likelihood 1912-1922. *Statistical Science*, 12(3):162–176, 1997.

[78] A. Feldmann and W. Whitt. Fitting Mixtures of Exponentials to Long-Tail Distributions to analyze Network Performance Models. *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 3:1096–1104, Apr 1997.

[79] B.M. Hill. A Simple General Approach to Inference About the Tail of a Distribution. *The Annals of Statistics*, 3:1163–1174, Sep. 1975.

[80] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[81] F. Vacirca, T. Ziegler, and E. Hasenleithner. An Algorithm to detect TCP Spurious Timeouts and its Application to Operational UMTS/GPRS Networks. *IEEE Computer Networks*, 50(16):2981–3001, 2006.

[82] M.E. Crovella, M.S. Taqqu, and A. Bestavros. Heavy-tailed Probability Distributions in the World Wide Web. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, pages 3–25, 1998.

[83] N. Brownlee. Understanding Internet Traffic Streams: Dragonflies and Tortoises. *IEEE Communications Magazine*, 40:110–117, 2002.

[84] H. Dahmouni, D. Rosse, B. Morin, and S. Vaton. Impact of traffic composition on GPRS performance. *ITC 19th*, pages 1405–1409, 2005.

[85] R. Fielding, J. Gettys, and J.C. Mogul. *RFC 2616: Hypertext Transfer Protocol – HTTP/1.1*. IETF, 1997.

[86] S. Ostermann. `http://www.tcptrace.org`, 2005.

[87] P. Stuckmann. *Traffic Engineering Concepts for Cellular Packet Radio Networks with Quality of Service Support*. PhD thesis, Technischen Hochschule Aachen, 2003.

[88] P. Stuckmann and C. Hoymann. Performance evaluation of wap-based applications over gprs. *Communications, 2002. ICC 2002. IEEE International Conference*, 5:3356–3360, 2002.

[89] H.K. Choi and J.O. Limb. A Behavioral Model of Web Traffic. In *ICNP '99: Proceedings of the Seventh Annual International Conference on Network Protocols*, pages 327 – 338, Washington, DC, USA, 1999. IEEE Computer Society.

[90] B.V. .Iversen, J. Arne, and J. Rasmussen. Internet Dial-Up Traffic Modelling. pages 11–23, August 2000.

[91] A. Reyes-Lecuona, E. Gonzalez-Parada, and E. Casilari. A page-oriented WWW traffic model for wireless system simulations. In *Proceedings of 16th ITC, Edinburgh*, pages 1271–1280, 1999.

[92] P. Svoboda and M. Hogl. *IE-Control Tool for HTTP Transfer Times and Page Analysis in Mobile Networks,* `http://fil.nerdcamp.net/iecontrol.htm`, 2005.

[93] J. Postel and J. Reynolds. *RFC 959: File Transfer Protocol (FTP)*. IETF, 1985.

[94] Ed. J. Klensin. *RFC 2821: Simple Mail Transfer Protocol*. Internet Engineering Task Force (IETF), 2001.

[95] J. Myers and M. Rose. *RFC 1939: Post Office Protocol Version 3*. Internet Engineering Task Force (IETF), 1996.

[96] M. Crispin. *RFC 2060: Internet Message Access Protocol Version 4rev1* . Internet Engineering Task Force (IETF), 1996.

[97] J. K. Reynolds. *RFC 918: Post Office Protocol*. Internet Engineering Task Force (IETF), 1984.

[98] M. Stewart, A. Huynh, and H. Lam. *RFC 3591: Internet Message Access Protocol Version 4* . Internet Engineering Task Force (IETF), 1999.

[99] G. Brasche and B. Walke. Concepts, Services and Protocols of the New GSM Phase 2+ General Racket Radio Service. *IEEE Communications Magazine*, 35:94–104, 1997.

[100] B. Lloyd and W. Simpson. *RFC 1334: PPP Authentication Protocols*. IETF, 1992.

[101] J. Färber. Traffic Modelling for Fast Action Network Games. *Multimedia Tools Application*, 23(1):31–46, 2004.

[102] J. Färber. Network Game Traffic Modelling. In *NETGAMES '02: Proceedings of the 1st workshop on Network and system support for games*, pages 53–57, New York, NY, USA, 2002. ACM Press.

[103] G. Armitage, M. Claypool, and P. Branch. *Networking and Online Games - Understanding and Engineering Multiplayer Internet Games*. John Wiley and Sons, 2006.

[104] An analysis of MMOG Subscription Growth. `http://www.mmogchart.com/`.

[105] T. Lang, P. Branch, and G. Armitage. A synthetic traffic model for Quake. *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 233–238, 2004.

[106] S. Zander and G. Armitage. A Traffic Model for the Xbox Game Halo 2. In *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pages 13–18, New York, NY, USA, 2005. ACM Press.

[107] G. Armitage and L. Stewart. Some thoughts on emulating jitter for user experience trials. In *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 157–160, New York, NY, USA, 2004. ACM Press.

[108] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, and M. Claypool. The Effects of Loss and Latency on User Performance in Unreal Tournament 2003. In *NetGames '04: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, pages 144–151, New York, NY, USA, 2004. ACM Press.

[109] J. Nagle. *RFC 896: Congestion Control in IP/TCP Internetworks*. IETF, 1984.

[110] T. Issariyakul and E. Hossain. *Introduction to Network Simulator `ns-2`*. Springer-Verlag, 2008.

[111] TS 26.071. *AMR speech Codec; General Description*. ETSI/3GPP.

[112] S. Casner, R. Frederick, and V. Jacobson. *RFC 1889 - RTP: A Transport Protocol for Real-Time Applications*. IETF, 1996.

[113] S. Deng. Traffic Characteristics of Packet Voice. *IEEE International Conference on communications*, June 1995.

[114] ITU. Telephone transmission quality objective measuring apparatus: Artificial conversational speech. *Telecommunication Standardization Sector of ITU*, March 1993.

[115] J.H. James, C. Bing, and L. Garrison. Implementing voip: a voice transmission performance progress report. *Communications Magazine, IEEE*, 42(7):36–41, July 2004.

[116] TS 29.108. *Application of the Radio Access Network Application Part (RANAP) on the E-interface*. ETSI/3GPP.

[117] TS 29.800. *Signalling System No. 7 (SS7) Security Gateway; Architecture, functional description and protocol details*. ETSI/3GPP.

[118] TS 29.060. *GPRS Tunnelling Protocol across the Gn and Gp interface*. ETSI/3GPP.

[119] GSM 09.060. *GPRS Tunnelling Protocol across the Gn and Gp Interface*. ETSI/3GPP.

[120] TS 25.211. *Physical channels and mapping of transport channels onto physical channels (FDD).* ETSI/3GPP.

[121] TS 44.065. *Mobile Station - Serving GPRS Support Node; Subnetwork Dependent Convergence Protocol (SNDCP).* ETSI/3GPP.

[122] TS 44.064. *Mobile Station - Serving GPRS Support Node; Logical Link Control (LLC) Layer Specification.* ETSI/3GPP.

[123] TS 43.064. *Overall Description of the GPRS Radio Interface; Stage 2.* ETSI/3GPP.

[124] D. Johnson and S. Sinanovic. Symmetrizing the Kullback-Leibler distance. *Technical Report*, 2001.