Bayerische Julius-Maximilians-Universität
Würzburg
Institut für Informatik
Lehrstuhl für verteilte Systeme
Prof. Dr. P.Tran-Gia

Modeling of Internet Traffic: Internet Access Influence, User Interference, and TCP Behavior

Norbert Vicari

Würzburger Beiträge zur
Leistungsbewertung Verteilter Systeme

Bericht 02/03

Würzburger Beiträge zur Leistungsbewertung Verteilter Systeme

Herausgeber

Prof. Dr. P. Tran-Gia
Universität Würzburg
Institut für Informatik
Am Hubland
D-97074 Würzburg
Tel.: +49-931-888-6630
Fax.: +49-931-888-6632
email: trangia@informatik.uni-wuerzburg.de

Satz

Reproduktionsfähige Vorlage vom Autor.

ISSN 1432 - 8801

Modeling of Internet Traffic: Internet Access Influence, User Interference, and TCP Behavior

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius-Maximilians-Universität Würzburg

vorgelegt von

Norbert Vicari

aus

Kaufbeuren

Würzburg 2003

Eingereicht am: 28. April 2003
bei der Fakultät für Mathematik und Informatik
1. Gutachter: Prof. Dr. P. Tran-Gia
2. Gutachter: Dr. M. Fiedler
Tag der mündlichen Prüfung: 9. Juli 2003

Danksagung

An dieser Stelle möchte ich allen danken, die mich bei der Anfertigung der vorliegenden Arbeit unterstützt haben:

In besonderer Weise gilt mein Dank meinem Doktorvater Prof. Dr.-Ing. Tran-Gia, der es mir ermöglicht hat, diese Arbeit in einer Atmosphäre gegenseitigen Vertrauens und fruchtbaren Zusammenwirkens anzufertigen. Durch sein fortwährendes Bemühen um neue Kontakte zu Industrie und Forschung ist es ihm stets gelungen, seinen Mitarbeitern den Zugang zu aktuellen und relevanten Wissenschaftsgebieten und Forschungsbereichen zu erschließen. Indem Prof. Tran-Gia die Teilnahme an internationalen Konferenzen und Projekten engagiert und großzügig fördert, lernt man, sich als junger Forscher nicht nur sich im internationalen Umfeld angemessen zu präsentieren, sondern auch die eigene Arbeit richtig einzuschätzen.

Herrn Dr. Markus Fiedler möchte ich für die spontane und bereitwillige Übernahme des Zweitgutachtens danken.

Meinen Kollegen am Lehrstuhl für verteilte Systeme – Mathias Dümmler, Dr. Notker Gerlich, Klaus Heck, Stefan Köhler, Dr. Kenji Leibnitz, Michael Menth, Jens Milbrand, Dr. Manfred Mittler, Dr. Rainer Müller, Dr. Michael Ritter, Dr. Oliver Rose, Dr. Alexander K. Schömig, Dirk Staehle, Dr. Kurt Tutschku und Patricia Wilcox – danke ich für die angenehme Zusammenarbeit. Sie haben durch fachliche Diskussionen und freundschaftliche Gespräche sowohl zum guten Betriebsklima als auch zum Gelingen meiner Arbeit wesentlich beigetragen.

Georgios Aivalis, Tobias Hossfeld und Robert Schedel danke ich für ihren Einsatz und ihre Kooperationsbereitschaft im Rahmen der Erstellung von Studien- und Diplomarbeiten.

Ein herzlicher Dank geht an unsere Sekretärin Frau Giesela Alt für die zuverlässige Unterstützung bei der Verwaltungsarbeit, insbesondere für ihre Hilfe im Rahmen des COST-257-Projekts.

Für die gute Zusammenarbeit im Rahmen des EU-Projekts COST-257 möchte ich mich bei allen Teilnehmern bedanken. Daneben danke ich ganz besonders Herrn Dr. Joachim Charzinski und Herrn Dr. Udo Krieger für die fachkundigen Diskussionen und die langjährige, produktive Kooperation.

Auch bei meinen Freunden, die mit Volleyball, Badminton und diversen anderen Ablenkungen dafür gesorgt haben, dass das Leben neben Studium und Promotion nicht zu kurz kommt, möchte ich mich herzlich bedanken. Besonders gilt das für Andrea Wutzer und Stefan Landvogt, die mich immer wieder motiviert haben, die Arbeit fortzusetzen und zum Abschluss zu bringen.

Zu guter Letzt möchte ich daran denken, meine Eltern zu erwähnen. Sie haben mir das Studium ermöglicht, haben mich auf meinem Weg immer bestärkt und sind mir stets mit Rat und Tat zur Seite gestanden. Dafür bin ich aus tiefstem Herzen dankbar.

Würzburg, im Juli 2003 Norbert Vicari

Contents

1 Introduction

The Internet has evolved in the last 30 years from a scientists network to a communication and information media for everybody. This rapid development was enabled by the open and modular standards the Internet is based on. With the exponential growth of the Internet with regard to users and carried traffic, the econonomic dimension became an important factor of commercial relevance.

Technical innovation is the main driver for the growth of the Internet. A good example for technology driven growth is the development of the hyper link technology behind the WWW in 1990 and the first WWW browser in 1993. These technologies opened the Internet for people that were not that familiar with computers – which enabled a wider range of users to participate in the Internet.

The main technological trends seen today are: steady growing bandwidth in the core and access network, user mobility, wireless access networks, convergence of telecommunication and data networks, and Quality of Service (QoS) differentiation. These trends are driven by economical interests of the network and service providers. Before implementing, performance analysis is applied for the development and dimensioning of these novel Internet protocols and services. The analysis depends often on the traffic characteristics and the user behavior. To improve the accuracy of this analysis, flexible and realistic traffic models are needed as input.

This monograph investigates the characterization of Internet traffic from a user perspective. For this goal measurements of Internet access network traffic are analyzed. The derived characterization is used to develop flexible models that can be used for the performance analysis of Internet protocols and services.

For the analysis of the measurement a profound knowledge on the observed Internet protocols and applications is required. An introduction to the relevant protocols and applications, i.e. the TCP/IP protocol stack is given in Chapter 2. The measurements, on which this monograph is based on, are addressed in the following Chapter 3. We present the setup and environment of the measurement and a basic evaluation of the measured data. The analysis of the measurement is continued in Chapter 4, where the most frequent observed applications are analyzed in detail. The protocol objects and their temporal relation is described down to the connection level. Chapter 5 discusses the influence of the access speed to the traffic characteristics. This evaluation was enabled by the measurement of the packet stream and the access log file data. The results are compared to a measurement in an ADSL environment, which provides a multiple faster access speed. To give a usable model of the most frequent observed applications, we discuss in Chapter 6 the issue of heavy-tailed distributions in the context of Internet traffic modelling. Based on the discussion we give fitting distribution and their parameters for the model components. Chapter 7 summarizes the results of the monograph and gives an outlook to the applicability of the results with regard to current development of Internet traffic.

2 Internet Traffic and Protocol Characteristics

The term “Internet traffic” describes all data traffic transported with the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. According to the hourglass model depicted in Figure 2.1, the IP protocol is the common denominator for all Internet traffic. A variety of different technologies and protocols are used for the physical transport of the TCP/IP data, such as optical and electrical

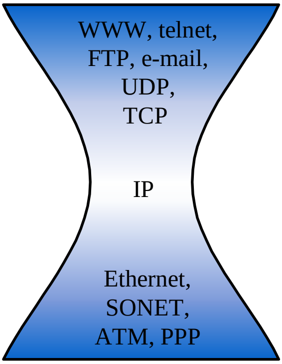

Figure 2.1: Internet hourglass model.

transmission with the protocols Ethernet (CSMA-CD), Asynchronous Transfer Mode (ATM) or Synchronous Optical NETwork (SONET). Common to all TCP/IP traffic is the utilization of the IP protocol for moving the packets around in the network. On higher protocol layer the data is transported with different protocols such as TCP or User Datagram Protocol (UDP). The data origins of a variety of applications that are invoked by the users, such as Telnet, e-mail or the World Wide Web (WWW). In this chapter, we will describe the functionality of the layers of the TCP/IP protocol suite and introduce a model for characterizing Internet traffic for teletraffic modeling purposes. The protocols below the IP layer, that is, Ethernet, SONET, ATM, Point to Point Protocol (PPP), etc., will not be reviewed in detail, since they provide the transport of the bit stream at a dedicated speed without significantly altering the traffic characteristic itself. The influence of the data transmission speed is perceived by the user and the feedback introduced thereby is discussed in Chapter 5. The protocols relevant for traffic characterization are reviewed in detail in the following sections.

2.1 Hierarchical Internet Traffic Model

The International Standards Organization (ISO) published the so called Open Systems Interconnection (OSI) Reference Model in order to allow an abstract description of inter-computer communication in an efficient and clear way. Even if the Internet protocol, that is, the TCP/IP protocol suite, was developed prior to the ISO/OSI reference model this reference model is useful for describing network protocols. Thus, we describe the functionality of the TCP/IP protocol stack by means of the ISO/OSI reference model. Besides the protocols functionality a temporal component is required for traffic characterization, which will be introduced in Section 2.1.2.

2.1.1 TCP/IP Protocol Stack

The TCP/IP protocol suite is organized, as most modern communication protocols, in a layered model. In the case of TCP/IP, the protocol suite is designed in a four layer model [82, 83, 84].

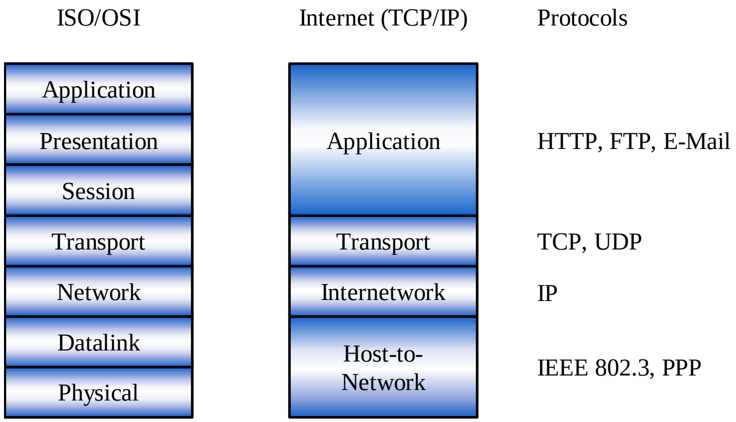

Figure 2.2: ISO/OSI network protocol stack and the equivalent TCP/IP stack.

Figure 2.2 shows the protocol layers, their relation to the ISO/OSI reference model and examples of the corresponding protocols.The layers are as follows:

Link Layer (Host-to-Network)

By means of using standard network protocols the link layer offers the functionality of the *physical* and *data link* OSI layers. This layer gathers all network functions for accessing the physical transmission media. The physical part of the link layer handles the transfer of the raw bit data to and from the medium and delivers the required interfaces. The data link part of the link layer performs the logical link control (LLC), that is, formatting the data for the medium and performing low level error and flow control. The medium access control (MAC) part of the data link layer controls the access to the medium and resolves conflicts among attached stations.

The most common protocols on the link layer are the Ethernet protocol family, that is, IEEE 802.3 [37], and the Point to Point Protocol (PPP) [65] used in modem dial-in connections.

The link layer does in general not influence or alter the traffic characteristics of applications significantly, as long as flow control mechanisms on higher layers prevent overload and the medium itself provides a low bit error rate. This is normally guaranteed for optical or electrical transmission. For wireless media the control and error correction mechanisms and their interaction with higher protocol layers are an important issue and have to be taken into account.

Network Layer

The IP protocol is used to connect all Internet nodes on the network layer. The main tasks of the network layer are to implement addressing and to enable routing. The IP protocol addressing is described in [70] and provides a means to deliver an datagram, that is, data packet, from a source to a destination. The IP transmission is called connectionless, since IP only delivers the datagram from hop to hop, without giving feedback on delivery success. The IP protocol is seconded by the Internet Control Message Protocol (ICMP) [67] to transfer state and error messages of the IP network. For delivering datagrams in local networks IP uses the Address Resolution Protocol ARP [66] to discover the hardware addresses required by the protocols on link layer.

Routing is used to direct the IP datagrams from source to destination. Therefore the vast number of end systems in the Internet is organized in an hierarchical manner, consisting of independent autonomous systems (AS), such as an university campus or the intranet of a large company. This kind of organization helps to reduce signalling overhead. For routing within ASs Interior Gateway Protocols (IGPs), such as the Routing Information Protocol (RIP) [49], Open Shortest Path

First (OSPF) [57], or the Enhanced Internet Gateway Protocol (EIGRP), a proprietary CISCO extension to RIP, are used. Between ASs the Border Gateway Protocol (BGP) [78] is used as the most common Exterior Gateway Protocol (EGP).

The influence of the routing protocol to traffic characterization is restricted to be responsible for providing connectivity at the best bandwidth possible, but does not directly alter the application or user behavior.

Transport Layer

The transport layer ensures reliable end-to-end data transport in the network. The main tasks of the transport layer are multiplexing, error checking, sequencing, and providing flow control. The transport layer acts as interface for applications to the physical and link layer. For connection oriented communications the transport layer also sets up the bidirectional communication.

Since, the addressing scheme of UDP and TCP as well as the flow and congestion control of TCP are important for traffic characterization, we will discuss TCP and UDP later in this chapter.

Application Layer

The application layer in the TCP/IP protocol stack handles not only the application details of particular applications, but also covers – if required – the functionality of the ISO/OSI session, presentation and application layer. Later in this chapter we will introduce the most common applications and its protocols, such as WWW, e-Mail, telnet, FTP.

The application layer is of high relevance for traffic modeling and characterization, since not only the applications have different characteristics for themselves, but also the user interacts directly with the applications.

Before presenting the protocols of the application and transport layer in detail, we will introduce a temporal context into the traffic characterization model.

2.1.2 IP Traffic Characterization Model

As pointed out in the section above, the upper two layers of the TCP/IP reference model impact the traffic characteristics while the lower two layers are essential for the data transfer capability of the system but are of minor influence to the traffic characteristics. The two relevant layers, that is, transport and application layer, are depicted in temporal context in Figure 2.3. Above these two layers we define a new layer, called session layer, representing the users presence and general activity in the system. This session layer is distinct from the session layer defined in the ISO/OSI reference model since a session encloses different applications. The historical reasons for the identical naming will be motivated in the next Section 2.2.

The model describes the traffic on different layers by means of object volume and temporal context, that is, the relation in time of the transmitted objects. The kind of objects and its temporal relation depends on the protocol properties and user interaction. The time period between the start of the transmission of two subsequent objects is denoted as inter-arrival time (IAT).

The time granularity increases from the session layer over the application layer to the transport layer. Within an Internet session the user starts usually at least one application, in many cases several applications. The applications could overlap, for example a user might check his mail while surfing in the WWW. Each of the applications causes data transmissions on the transport layer. These data transmissions might result in a single UDP packet stream or result in more complex structures, such as inter-depending TCP transmissions. For the characterization of some complex applications it is reasonable to group TCP connections to objects that are downloaded together.

In the following we will outline the session layer and review the applications and transport protocols.

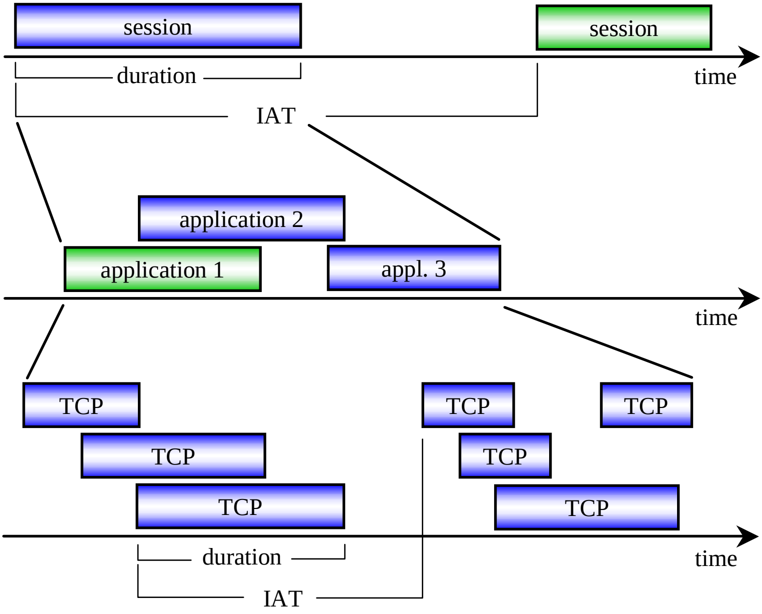

Figure 2.3: Relation of sessions, applications and TCP connections.

2.2 User Activity: Sessions

Classical teletraffic modeling is often based on (telephone-) calls and the related call holding times [1]. In the context of the OSI/ISO model the technical processing of a call would be described as session, that is, the process of capturing resources.

Similar to a call in the telephone world, we define an Internet session in our model as the process in which a user captures network resources. By this means, a session describes the user activity on topmost layer. As difference to classical telephone systems the network resources are not captured exclusively by a user for the duration of a session. The resource usage depends on the application usage and personal communication behavior.

Technically, we have to discern between two different Internet access methods, that is, dial up access and always on access.

2.2.1 Dial Up Sessions

The most common Internet access for home users is to dial up to an Internet Service Provider (ISP) with a modem connection. Usually a Point to Point (PPP) connection is established from the computer of the user to the dial-in router of the ISP. The telephone line of the user is busy for the duration of the session. The resource phone line and a port of the dial-in router are captured and busy for the Internet session, and similar to a traditional phone call the ISP records accounting data of the Internet session which could be used for modeling of the captured resources [28].

In difference to traditional teletraffic modeling, the actual network load only depends on the users activity and behavior and can not be deducted from the captured access resources. Correlating the network load with the Internet access records of the ISP enables detailed insight to the user traffic.

2.2.2 Always-On Access

The always-on access is commonly provided in companies or universities. In the case of home users with modern DSL Internet access the commonly provided flat rate makes the access to a virtual always on access even if technical a modem access is used. The user computer is connected permanently to the Internet, independent of the actual Internet usage. To distinguish the traffic characteristics of a single user is more complicated in the always on access scenario, since, no accounting is provided for single users and the authentication is aligned to the computer connected to the network.

In this case the Internet session identification is equivalent to the identification of user activity. Based on traffic measurements on packet level user activity is detected and a session is said to be a period of activity not interrupted for more than a predefined interval. This method was applied for WWW traffic for example in [89]. Reasonable intervals of inactivity reach from 10 to 20 minutes. The disadvantage of this method is that it is not applicable accurately in scenarios of high load, e.g., as in highly frequented computer laboratory where users capture workstations immediately when a working place gets idle.

In case of both access methods the network load is determined by the use of applications. We will introduce the most common used applications in the following.

2.3 Applications and Services

In the following we will describe the protocols of the most frequently observed applications. In our measurement, cf. Section 3.2.2, HTPP, FTP, e-mail, Telnet and News are the TCP based applications of relevance. For applications that use the UDP protocol we will look at game traffic, DNS, NetBios and ICQ.

Like most of the Internet application protocols the above mentioned applications use the client-server paradigm of interaction. An application consists normally of a server and a client part, which could run on different computes or on the same system. The server is an application that offers a service to the user, while the client requests this service.

Users execute the client and request some service from the server. A server receives the request and replies to the client. Servers usually are able to process several requests of different clients at the same time.

Addressing between client and server is done in the TCP/IP protocol suite by the socket concept. For TCP a socket consists of the transport protocol identification (TCP, UDP,...), the IP address and a port number. Servers are usually addressed by the clients at the so called *well-known* port numbers [79], while on the client side arbitrary port numbers are used.

2.3.1 Web Surfing with the Hypertext Transfer Protocol

Today, most of the data transferred in the Internet is transferred with the Hypertext Transfer Protocol (HTTP). It was designed in 1990 to enable the transfer of information available in the Internet in an easy way and to bypass the complexity of state-keeping protocols, such as the File Transfer Protocol (FTP). In the initial form the protocol was intended to transfer only text or hypertext documents by requesting them from a server with a GET command [4].

At the time of the investigation two versions of HTTP are in use, that is HTTP/1.0 [6] and HTTP/1.1 [30]. These versions of HTTP have, besides the core functionality to retrieve documents, the following properties. HTTP allows clients to signal their capabilities, e.g. the presentation capability, such as the character set supported, to allow the server to adapt its response. Caching is supported, that is, a previous downloaded file could be used by only retrieving the file properties. HTTP/1.1 is optimized in performance and allows to specify byte ranges, enabling the continuation of previously interrupted downloads. The file type transferred is indicated by MIME encoding and file extensions.

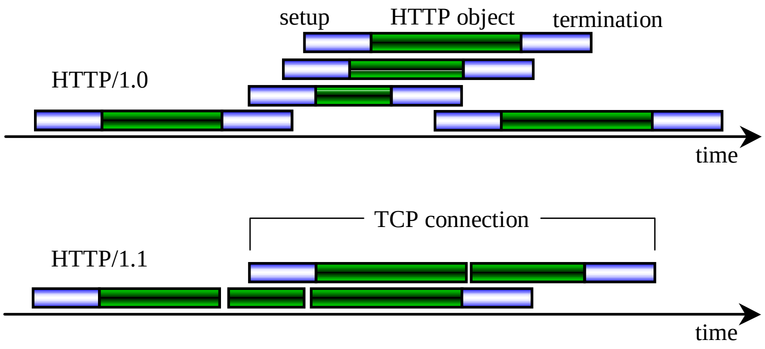

Figure 2.4: Differences of web page download with HTTP/1.0 and HTTP/1.1.

The default applications utilizing the HTTP protocol are WWW browsers. In this case the well-known port number 80 is used to access documents structured with the Hypertext Markup Language (HTML). These documents, also called web-pages, are structured pages with tags and frames. A page can contain text, images, and other inline objects. HTML was originally defined in the IETF [5, 60, 74] and is now under development at the World Wide Web Consortium, w3c.org, cf. [19]. A web-page is normally addressed by the use of an Uniform Resource Locator (URL) [7], that describes the main access method of a document, its Internet location and name.

When a web page is interpreted by a WWW browser, the included inline objects are retrieved by subsequent requests. Depending on the protocol version of HTTP a different number of TCP connections is used to download these inline objects, what also affects the overall download performance. Figure 2.4 depicts the differences related to TCP connection usage in HTTP/1.0 and HTTP/1.1 for an example web-page with four inline objects.

In HTTP/1.0 a single connection is used for every object contained in the web-page. By default, most browsers open 4 TCP connections simultaneously for every browser window. The overhead of connection establishment and the properties of the TCP congestion control, described in Section 2.4.2, reduce the efficiency of HTTP/1.0 since most inline objects are too small to allow a efficient utilization of available bandwidth.

To cope with these performance problems a “keep-alive”-option was introduced to HTTP/1.0 clients [29] and included as persistent connections with optimized proxy handling and caching in HTTP/1.1. With persistent connections multiple objects can be transmitted in a single TCP connection. In doing this, the connection setup and termination overhead is reduced and the connections are enabled to utilize higher bandwidth. A discussion of the efficiency of HTTP/1.0 and HTTP/1.1 under various conditions is given in [86, 62].

2.3.2 File Transfer Protocol

The File Transfer Protocol (FTP) [72] is the standard Internet protocol to transfer files. It utilizes the well-known port numbers 20 and 21 on the server side to give access to files on remote systems with the TCP protocol.

FTP is designed to operate between different hosts with different file and operating systems. This broad applicability was achieved by detailed control options with regard to file type, format control, structure, and transmission mode of the data transfer. Nowadays most implementations restrict the control options to the choice of binary or ASCII transfer. The remaining options are either antiquated or replaced by mechanisms on higher protocol level.

As indicated in Figure 2.5 FTP uses two parallel connections to transfer files. First a control connection is opened actively from the client to the server on port 21. This control connection remains open during the whole time of the clients and servers communication. The control connection is used the transfer the users commands to the server and the servers replies to the user. The type of this connection is similar to a telnet connection. Besides the transfer of control information the control connection is used to authenticate the user and to keep track of the state of the transfer process.

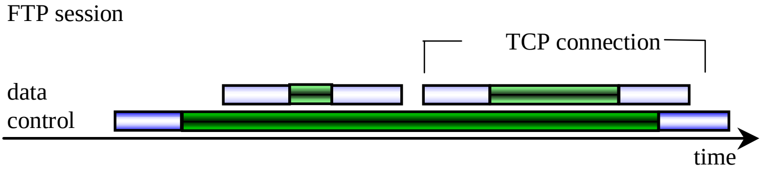

Figure 2.5: Control and data connection in FTP sessions.

The data connections are identified by the well-known port 20 on the server side and are used for sending files from the client to the server, fetching files from the server to the client and to transfer listings of files or directories from the server to the client. The data connections are opened passively by the client, indicating which port number to use, and actively opened on demand by the server. The end of a file is indicated by closing the data connection.

Compared to HTTP the FTP protocol is more complicated and involves more overhead. But it provided means to access files interactive in times before hypertext was used for navigation. Further it provides an simple method to authenticate the user, which is not given in HTTP.

2.3.3 Electronic-Mail

Electronic-mail (e-mail) is one of the most popular applications in the Internet. It allows users to personally exchange letters in ASCII format, and, by the use of extensions, such as MIME [11], also the exchange of non-text documents.

Three protocols commonly are used for e-mail transport. These protocols are described in the following.

Simple Mail Transfer Protocol

To send e-mail to mail servers and to exchange e-mail between mail servers the Simple Mail Transfer Protocol (SMTP) [68] is used. In an access network such as in the measurements evaluated in this work, only client traffic is recorded. Thus, no traffic between mail servers is measured and the SMTP traffic can be identified as sending mail.

The SMTP server waits on the well-known port 25 for e-mail transfers. A transfer is accomplished by simple ASCII commands, identifying the communicating systems, sender, receiver and body of the mail. It is possible to transfer multiple messages in a single TCP connection, allowing effective batch processing of e-mails. The connection is closed when the transfer is finished.

Since SMTP requires the server to be up to receive e-mails, most organizations use relay servers that store the e-mail for the user into a mailbox until the mail is fetched. The following two protocols are used for this purpose.

Post Office Protocol

The Post Office Protocol version 3 (POP3) [59] is used to fetch messages from a mail server. Similar to SMTP POP3 accesses e-mails with simple ASCII commands. It also provides commands to authenticate the user, methods to request the number of messages, and is able to list the size of the waiting messages.

The server is connected at the well-known port 110. Normally all waiting messages are transferred in a single TCP connection, that ends when the transfer is finished. Client implementations of POP3 might also allow only to download selected messages and limit the message length.

Internet Message Access Protocol

The Internet Message Access Protocol IMAP4 [20] has basically the same functionality as the POP3 protocol, that is, to access e-mail on a remote server. Additionally it provides methods to administrate remote mailboxes and to filter and search the stored e-mails.

An IMAP server is accessed on the well-known TCP port 143. Multiple messages are transferred in a single TCP connection. In many implementations this connection stays open awaiting for new commands utilizing the TCP keep-alive option.

2.3.4 Telnet

The Telnet Protocol [73] enables remote access to any host system by defining the network virtual terminal (NVT). This NVT emulates the remote hosts terminal. The telnet protocol connects the server on the well-known port 23 and handles data and signaling transfer in a single TCP connection. Telnet transports user keyboard input and server responses in small packets, either a character at a time or a line at a time.

The telnet protocol capabilities are used not only for terminal emulation but also for applications that require keyboard input, such as FTP.

2.3.5 Network News

Network News is a system to read and exchange articles – similar to e-mails – on a worldwide system of servers. The related Network News Transfer Protocol (NNTP) [43] is used to access the articles on a local server as well as to update the servers. Nowadays the Network News functionality is often moved into a web based interface, which enables the access to news groups by the HTTP protocol.

A news server is contacted on the well-known TCP port 119. The TCP connection defines also the session context. It allows to navigate within the news group hierarchy, to request and post articles.

2.3.6 Network Games

Network Games are observed in several variants. Typically we discern first person shooters (Doom, Quake, Unreal), real time strategy (Star Craft, Age of Empires), role playing (Diablo, Ultima Online, EverQuest) and simulation/adventure games. First person shooters have also the most stringent network delay requirements but also real time strategy and role playing require short response times. The bandwidth required is normally designed for modem usage. Different communication models are supported by network games, from peer-to-peer messaging to pure server client communication [9, 10].

Similar to other applications these games could be identified by the port numbers they use. Due to the real-time requirement UDP is used instead of TCP as transport protocol. In our measurement, cf. Section 3.2.2, we detected battlenet games, i.e. StarCraft or Diablo on the UDP port 6112 and first person shooters, such as Quake II on UDP ports 27910, 27911, 27912 and 27920.

2.3.7 Domain Name System

TCP applications use the Domain Name System (DNS) [54, 55] to map between host names and IP addresses. As such, the DNS represents a distributed database, which is queried before being able to open a TCP connection to an host with unknown IP address. To reduce traffic, DNS resolvers cache IP addresses, avoiding repeated queries for the same address.

DNS requests and replies are transported with UDP and the server is connected usually on port 53. Normally the request is a single UDP packet, as well as the response is a single UDP packet.

2.3.8 NetBIOS

The NetBIOS service defines a name system for computers in a LAN. NetBIOS is a defined API and not a protocol [58, 61]. A NetBIOS group is discovered and joined by a host sending broadcast packets on UDP port 137. The name system of such a group is independent from the hierarchical DNS. Since the scope of NetBIOS is the LAN, NetBIOS packets are normally not routed.

2.3.9 ICQ

ICQ is an Internet tool that allows to contact other users in form of an online chat. ICQ normally runs on UDP port 4000, enabling peer-to-peer communication in written form. Newer implementations also offer the option to integrate file transfers and e-mail, while the initial version was a pure tool for Internet chat.

2.4 Transport Level

On transport level mainly two protocols are utilized: TCP for providing a connection oriented, reliable byte-stream service and UDP as simple datagram-oriented transport protocol. TCP is used for the majority of all data transfers in the Internet, while UDP is applied for real time applications and services as DNS lookup, that require low overhead. In the following we describe the features of these protocols, as far as they are relevant for traffic measurement and characterization.

2.4.1 TCP and UDP Addressing

Both transport protocols, TCP and UDP, utilize the same addressing scheme. Packets are identified by the TCP/UDP and IP header. For a complete description of the header the user is referred to [82]. We will describe the addressing scheme that allows to identify the hosts and applications involved in an Internet communication. The addressing scheme is the basis to evaluate the measurements.

The IP header contains the IP address of the source and destination end systems. An IP address is given by an 32-bit number, which is used to route the packets in the Internet. For higher routing efficiency these IP addresses could be discerned in variable length parts, describing the network and subnetwork a host belongs to, and the identity of the host. The protocol field in the IP header identifies the protocol, which is carried over IP, normally one of TCP or UDP.

A complete TCP/UDP address is given by an IP address and a port number, which is a 16 bit number, identifying the application or task the data is delivered to or origins from, respectively. According to the client-server concept of most Internet applications, the server side application is addressed with the so called *well-known* port number of the range 1–1023. On the client side usually arbitrary port numbers in the range 1024 – 5000 are used. These ephemal port numbers are utilized only for the time a connection exists and then released for further usage. The port numbers in the range up to 65535 are used to identify user defined servers [82], for example online games. The well-known port numbers are defined in [79] and are now maintained by the Internet Assigned Numbers Authority (IANA). The combination of IP address, port number and protocol type is often named socket, and describes one end of a TCP or UDP connection.

2.4.2 Transmission Control Protocol (TCP)

The Transmission Control Protocol (TCP) [71] is a connection-oriented and reliable protocol for the transmission of a byte-stream. The protocol is called connection-oriented, since an end-to-end connection is established before transmitting data over the connectionless IP service. The TCP protocol ensures the reliable transport of packets by keeping the state of the connections. With help of the state TCP ensures the correct packet ordering, and protects against packet loss or duplication. The reliable transport is achieved by the usage of positive acknowledgements (ACKs). These ACKs are also interpreted for flow and congestion control issues. Normally flow and congestion control is a task of the network layer, but in the TCP/IP protocol stack this functionality is assigned to the transport layer, that is, TCP is responsible for flow and congestion control.

TCP Signaling

TCP forms a full duplex data connection, and piggybacks the signaling information used for flow and congestion control in the header of the data packets. If required, TCP sends empty data packets, that contain only signaling information. Table 2.1 lists the flags used for signalling purposes and their meaning.

flag	meaning
SYN	synchronize to sequence number
ACK	acknowledge data
FIN	finished sending data
RST	reset connection
PSH	pass data to application as soon as possible
URG	urgent pointer

Table 2.1: TCP flags.

The SYN and ACK flag refer to the 32-bit sequence and acknowledgement number fields in the TCP header, respectively. The sequence number is used to unambiguously identify the first byte sent in a data packet and the acknowledgement number is used to inform the sender of the next byte expected by the receiver. With these numbers the receiver is able to reconstruct the byte stream and to inform the sender which data was received. To simplify the explanation of flow and congestion control, we will speak of acknowledged packets, that is, the receiver informs the sender of a successful received packet by signaling the byte it expects to receive next. The calculations for flow and congestion control are always based on multiples of the maximum segment size (MSS), which denotes the maximum number of bytes that could be transmitted in a single packet.

Connection Setup and Termination

In normal client-server operation the server performs an passive open, that is, the server waits to accept connections on a well-known port. The client performs an active open, by initiating a 3 way handshake. As depicted in Figure 2.6, the client sends a SYN packet to the server, requesting a connection. The server replies acknowledging the clients SYN packet by issuing a SYN/ACK packet, with an own sequence number for the byte stream from the server to the client. The client completes the 3-way handshake with an ACK. Within the two first packets, the sender and receiver could specify the MSS they are willing to accept. This option reduces the probability that the end-systems cause fragmentation. After the 3 way handshake the connection is established and data can be exchanged.

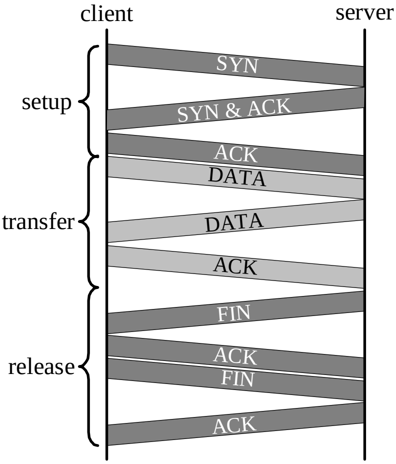

Figure 2.6: TCP connection setup and termination.

The regular termination of a TCP connection is also shown in Figure 2.6. Since TCP operates in full duplex mode, each of the directions has to close the connection. By performing a half close, one side of a connection ends the transmission of data while being continuing to receive data. Such a half close is done by sending a FIN packet, which in turn is confirmed with an ACK packet. Closing the complete connection requires 3 or 4 packets in total.

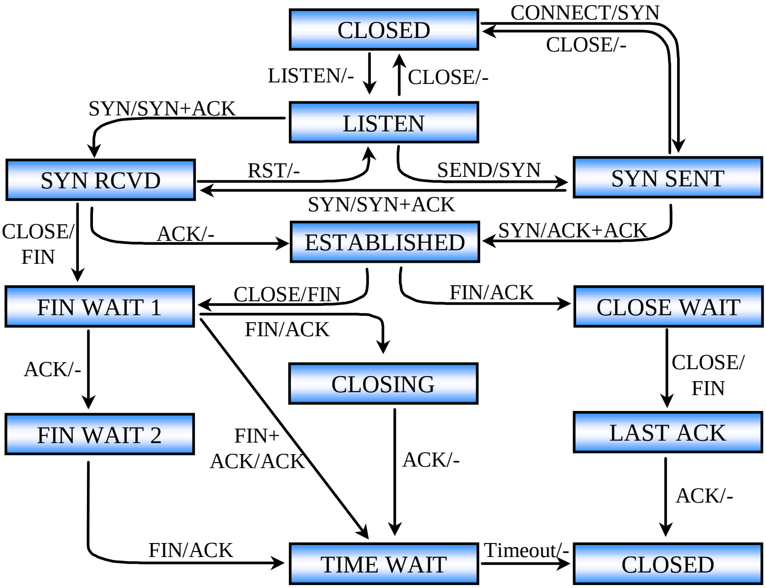

Figure 2.7: TCP connection state machine.

Connections could also be terminated by a reset, which is initiated by sending a RST packet. This is done when either a connection is refused by the server or a data transfer is aborted, e.g. by user interruption. A RST packet is not acknowledged and all queued data should be discarded and the data transfer should be stopped immediately. All options for the setup and termination of a TCP connection can be derived from the TCP state transition diagram depicted in Figure 2.7.

Reliable Transmission

TCP ensures the reliable transmission of data by positively acknowledging each data packet. The sender starts a retransmission timer for a sent data packet and resends the packet if the packet is not acknowledged before the retransmission timer expires. The retransmission timer interval is computed in dependence of the measured round trip time and variance. A wrong estimation of the parameters for the derivation of the retransmission timer would lead to either unnecessary retransmissions or to additional delay before retransmission. The retransmission timer for repeated retransmissions is extended exponentially to ensure system stability by avoiding frequent retransmissions.

Flow and Congestion Control

The initial version of TCP [71] implemented flow control by a receiver controlled window mechanism that enables the receiver to restrict the amount of data the sender is allowed to transmit. With every ACK the receiver signals an offset to the acknowledged sequence number, indicating the byte range the receiver is able to buffer. While this mechanism provides an efficient means to give feedback of the receivers state to the sender, the network state is not taken into account. In the case of network congestion, lost packets are repeatedly retransmitted without reducing the senders data rate. This behavior was discovered in October 1986, when overload induced a breakdown of Internet connections [41].

To improve this basic TCP behavior, the receiver based window control was extended by a sender window control. The senders window is determined in dependence of the successfully delivered data. The stability of this window control is guaranteed by admitting new packets to the network at the same speed packets leave the network. The window on the sender side, controlling the number of unacknowledged packets in the network is denoted as congestion window (CWND). The actual number of packets in the network is determined by the minimum of the CWND and the receiver window. We will describe the most important mechanisms of different TCP implementations for determining the CWND in the following.

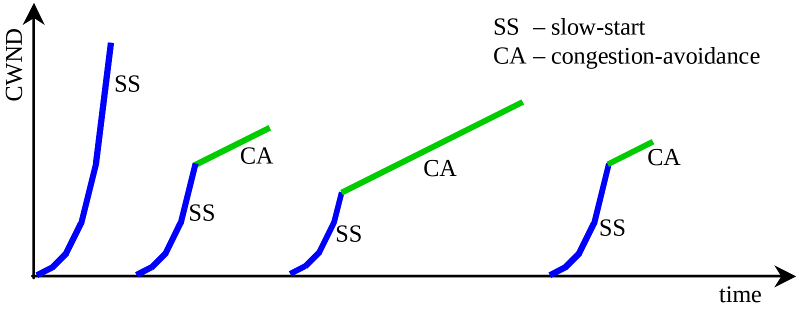

Figure 2.8: Schematic CWND control for TCP Tahoe.

Figure 2.8 shows the first mechanisms implemented for TCP congestion control, that is, slow-start and congestion-avoidance [41].

At the beginning of the data transfer in a TCP connection the primary goal is to use the available bandwidth as fast as possible without causing congestion. To reach this stable state the slow-start algorithm is applied. In order to avoid packet loss by sending data at higher rate then the network is capable TCP starts with a CWND of 1. For each ACK that is received from the network the size of CWND is increased by 1. This algorithm causes an exponential growth of CWND in dependence of the round trip time.

To avoid overloading the network and causing congestion, the growth of the CWND in the TCPs slow-start phase is interrupted when a packet loss is detected. For the next cycle, one-half of the current sender window size is stored in the variable slow-start threshold (SSTHRESH), which has a minimum of 2. TCP slow-starts again with a CWND of 1. When more than half of the data rate where the congestion originated is reached – indicated by a CWND size larger than the variable SSTHRESH – TCP enters the congestion-avoidance phase. In the congestion-avoidance phase the exponential growth of the CWND is reduced to a linear growth by increasing CWND by 1/CWND for every received ACK. While the slow-start algorithm is intended to get the self-clocked communication process running as fast as possible, the congestion-avoidance algorithm aims at slowly approaching the available bandwidth limit.

The TCP Tahoe definition implements additionally an algorithm called fast-retransmit, which is complemented in the TCP Reno version by the fast-recovery algorithm [42, 81]. The fast-retransmit algorithm detects lost packets before the retransmission timer expires. To achieve this, the receiver reacts on every packet that is received in the wrong order with an ACK of the last correct packet. Such an ACK is named duplicate ACK. The sender in turn concludes the packet loss from receiving multiple – usually 3 – duplicate ACKs, and retransmits the missing packet. As for any packet loss, the value SSTHRESH is set to CWND/2.

The fast-retransmit algorithm is initiated by a packet loss, and thus causes TCP to slow-start, even if the duplicate ACKs indicate that congestion was not the reason for the packet loss. Following upon the fast-retransmit, the fast-recovery algorithm takes into account that for each duplicate ACK a packet left the network and was transmitted. The CWND is divided in half and increased by the number of duplicate ACKs. As depicted in Figure 2.9, TCP continues with congestion-avoidance reducing the CWND to SSTHRESH when the first ACK for new data, sent after the packet loss, is received.

Finally, the New Reno version of TCP [32] speeds up the TCP behavior when multiple packets are lost in a transmission window. In regular TCP Reno the fast-retransmit algorithm retransmits the first missing packet and will receive a partial ACK, that is, an ACK indicating that some new data has arrived, but not acknowledging all outstanding data. This ACK will not trigger the fast-retransmit algorithm, since no duplicate ACKs are sent. As consequence the retransmission will be done only after a time-out. TCP New Reno concludes from such an partial ACK that still packets are missing and retransmits the missing packet. The fast-retransmit and fast-recovery status is maintained and the CWND is not reduced further. By this algorithm one missing packet is transmitted in every RTT. The SACK

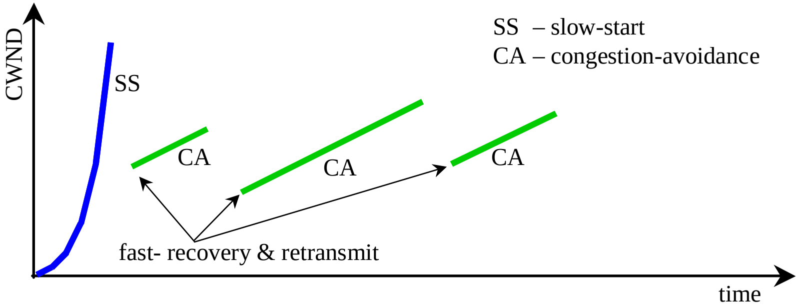

Figure 2.9: Schematic CWND control for TCP Reno.

[52, 34] extension of TCP allows to explicitly identify missing packets and to retransmit those packets within a RTT.

While the TCP versions Tahoe and Reno work with an additive increase and a multiplicative decrease to adapt to the congestion state of the network, TCP Vegas [12] estimates the available bandwidth from changes in the RTT. Further TCP Vegas changes the slow-start and fast-retransmit behavior of TCP Reno. Given sufficient buffers, TCP Vegas is reported to reduce packet loss and to increase the throughput. In comparison to the more aggressive startup behavior of TCP Reno, TCP Vegas is reported to suffer unfairness [53].

The development and optimization of TCP continues and several new algorithms are presented. Best known examples are the Random Early Discard (RED) [33] or the Explicit Congestion Notification (ECN) [75] mechanism for congestion avoidance. These mechanisms require not only changes in the algorithms implemented in end-systems, but also in all itermediate nodes. The majority of the servers, hosts and network devices in the Internet still use the Reno or New Reno version of TCP [63].

2.4.3 User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) [69] is a simple wrapper for the unreliable, connectionless IP datagram service on the transport layer. By extending the IP address with a port number UDP allows an end-to-end transport of datagrams with low complexity and overhead. Usually the source address and port is used to establish a full duplex communication.

The application has to provide session control and has to deal with effects like packet loss, reordering, duplication, or loss of connectivity. The low overhead qualifies UDP for the transport of real time application data, such as game or voice traffic. Since UDP does not implement any congestion control it might overload and starve other TCP connections. In order to allow the future transmission of TCP and UDP services on the same resources, UDP applications are expected to implement application level flow control mechanisms [48].

3 Internet Traffic Measurement

Several approaches for measuring Internet traffic are known and could be discerned with regard to the location and the level of detail of the measurement. On the user side application log files or proxy log files can be evaluated. On the server side server log files are captured, whereas packet traces are logged on the access line, backbone or network edge devices. On higher layer the evaluation of routing tables, router statistics or flow statistics is used to gain insight to Internet traffic characteristics [14]. After discussing the advantages and drawbacks of the different measurement methods we will illustrate the measurement environment and method used in this report and give an overview of the basic traffic data.

The evaluation of server log files can be used to create workload models of different applications, e.g. for WWW traffic [2, 51]. These files are easily obtained since the servers log the access data anyway. The major drawback of server log files is, that the evaluation of these files gives no feedback on the users behavior since a user may access different servers in one session. Thus, only a part of the users communication process is captured. Further, server logs are not able to provide information on the network state or network overhead.

Application log files and proxy log files are used to capture the characteristics of user traffic. Proxy cache servers automatically log data requests and give insight to the behavior of a user group [25]. In open environments, e.g. university networks, proxy servers are often not utilized due to out dating information. To capture application log files, normally applications have to be modified. These log files capture the behavior of single users [24, 21] and have been applied in the time of upcoming WWW services, when the code of the browser software was available. Since, today's most frequent used WWW client applications are commercial, it is not possible to modify these applications and, thus, no current application log file measurements are available.

Measurements on network edge devices, that is, access routers, allow traffic characterization at session level [28, 90, 91]. Similar to other measurement methods, such as flow statistics and evaluation of routing tables, that are described among others in [14], these methods in general allow not to characterize the behavior of single users but give valuable insight to the aggregated traffic profile.

The most detailed method for evaluating Internet traffic is to capture packet traces. Measurements in LAN environments allow to analyze the traffic transmitted, either extracting properties of different applications [8, 13, 47, 64], or evaluating the characteristics of the aggregate traffic stream [45]. Current packet trace measurements [46] at the Internet backbone allow due to the large traffic volume only short traffic snapshots, giving interesting insight to current trends in IP traffic patterns. While packet traces provide the most detailed information on the traffic characteristics, these traces suffer by difficulties caused by large data volumes and problems deriving the actual user behavior, that is, discerning user actions from software reactions.

In our approach, we combine a simultaneous measurement of the session log file of a dial-in router and the packet trace measured on the connection between the dial-in router and the Internet. This enables us to obtain packet level characteristics of IP traffic as well as statistics of the access speed.

3.1 Measurement Environments

In the following, we describe the origin of the data analyzed in this work. The main part of the traffic analysis is based on data measured at the dial-in Internet access of the computing center of the University of Würzburg as described in

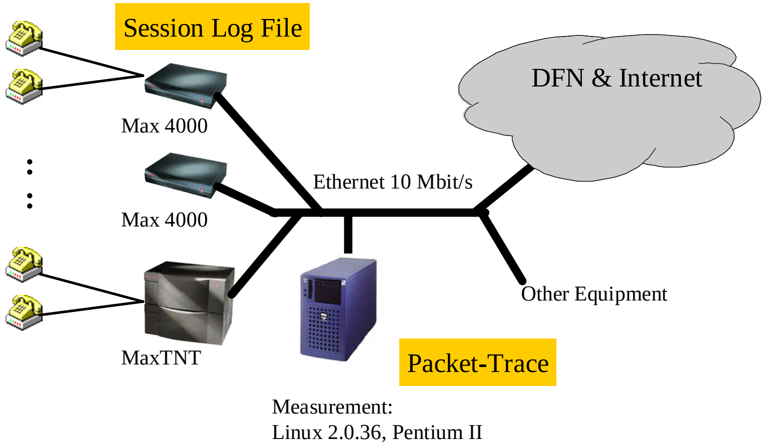

Figure 3.1: Measurement environment at the computing center of the University of Würzburg.

Section 3.1.1. For comparison reasons and to point out specific traffic properties related to other access technologies, we will refer to additional measurements described in Section 3.1.2.

3.1.1 Computing Center, University of Würzburg

The main measurement, also denoted as *dial-in* measurement was conducted during two weeks in February and March 1999 in the computing center of the University of Würzburg. About 6000 students and staff members are subscribed to the dial-in Internet access of the university. The access is implemented by three Ascend dial-in routers, enabling simultaneously 240 access lines. All lines facilitate digital ISDN access at 64 kbps, while 192 lines also facilitate analog modem access at speeds up to 56 kbps. The dial-in routers are connected by a 10 Mbps Ethernet bus to the German Research Network (DFN) and the Internet.

Figure 3.1 depicts the measurement configuration. Basic access data were logged by the routers, mainly the session start time, session duration, negotiated up- and downlink speed, data volume and the dynamically assigned IP address. In order to ensure privacy, all record entries allowing conclusions to single users were deleted from the log file.

To obtain detailed information of the user behavior, a sniffer measurement with the tool TCPDump [95] running on a Linux PC was made. All IP packet headers to and from the dial-in routers were logged. The analysis of this trace was performed with the help of several perl scripts and the tool TCPTrace [96]. The evaluation of the trace is based mainly on information contained in the IP and TCP header, such as time stamp, source- and destination address including port number, and the transport protocol used.

Aligning the router log file and the TCPDump trace allows to securely discern sessions, even in the case of fast IP address reuse. All traffic characteristics contained in a packet trace could be related to the dial-in speed of the modem used.

3.1.2 Comparison Measurement

In order to highlight characteristics of Internet traffic that are different to the dial-in traffic of the primary data set, we refer to the following data set. Further, we will contrast our results to the results of IP measurements in literature, and to own previously captured measurements [88, 89].

ADSL-Trial Münster

The ADSL trace was collected from May to December 1998 during an ADSL field trial. The PCs of 100 students were connected to the university's backbone in Münster, Germany. The downstream rate was 2.5 Mbps and the upstream rate was configured to 384 kbps. The Internet service was provided completely free of charge and the computers were connected to the Internet permanently, that is, no dial-up procedure was required. The users were observed with a modified version of TCPDump on a 10 Mbps Ethernet bus cyclically in 15 groups for one week.

In total 14 million IP packet headers belonging to HTTP, covering 480000 TCP connections, were collected. Since for these always-on lines no dial-up information could be collected, "sessions" were detected implicitly using an activity time-out of 20 minutes. A more detailed investigation of the ADSL trace is provided in [15, 16].

The ADSL trace was captured in a similar time period as the primary data set, but provides insight to user behavior not restricted by bandwidth or tariff issues.

3.2 Basic Measurement Evaluation

In the following chapter we describe the basic properties of the dial-in trace. During the measurement period approximately 62000 sessions were logged, the total data volume sums up to 82 GB. Figure 3.2 shows the number of users during the measurement period for each of the three dial-in routers. A periodic usage pattern in dependence of the daytime is observed. During the busy hour the MAX 4000 routers are loaded almost at their capacity limit of 60 simultaneous calls, while the MAX TNT router has a capacity of 120 simultaneous calls.

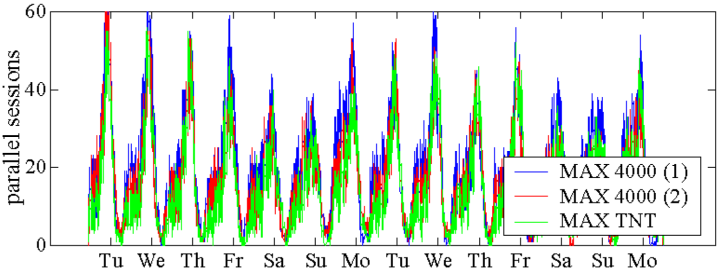

Figure 3.2: User activity during measurement period.

In Figure 3.3 the user activity for 24 hours is depicted. Obviously the users adapt to the phone tariff of the Deutsche Telekom AG by increasing activity at 6

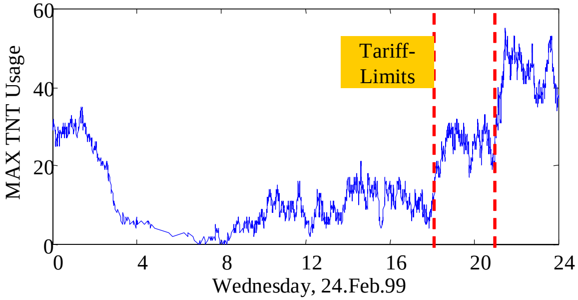

Figure 3.3: User activity during one day.

p.m and 9 p.m, that is, the starting time of cheaper phone tariffs.The identical user behavior was observed also in an 180 days measurement [28] at the University of Stuttgart, Germany in 1997.

3.2.1 Sessions Level

On topmost modeling level the user behavior is expressed by the session characteristics. The session duration describes the time the users are online and the session volume represents the amount of data that is transferred to and from a single user. This data contains not only the application data but also the overhead, that is, Ethernet, TCP and IP header. By this way of counting, also acknowledgements and synchronization packets are taken into account.

A session has an average duration of 15 minutes and a coefficient of variation (CoV, i.e. the standard deviation normalized by the mean) of 2.0. On weekdays the sessions are slightly shorter, whereas the sessions last in average 20% longer during the weekend. In Figure 3.4 the complementary cumulative probability distribution function (CCPDF) is depicted for the weekdays and the weekend. Both distributions have identical shape, indicating that the relation of short and long sessions is similar. As depicted in Figure 3.5, the duration of the sessions is

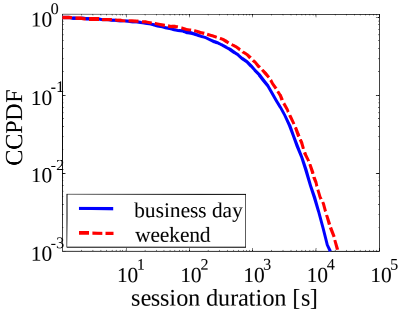

Figure 3.4: Session duration distribution during business days and weekend.

shorter during the business hours, extending in the evening and night. The CoV is reduced, but stays in the order of magnitude of the average CoV related to the full week. The exact numbers for the session duration and volume are listed in the Appendix, Table 8.1.

The session volume is heavily correlated to the session duration, indicated by a correlation coefficient of 0.7. In the average 1.2 MB of data are transferred. The session volume distribution, as depicted in Figure 3.6, exhibits a CoV of 3.3. Iden-

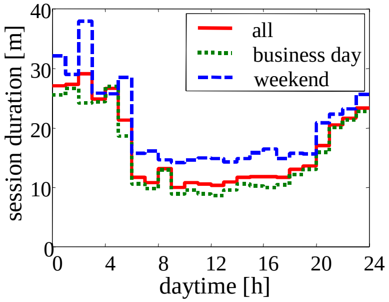

Figure 3.5: Session duration in dependence on daytime.

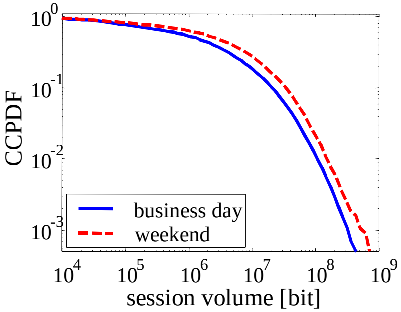

Figure 3.6: Session volume distribution on business days and weekend.

tical to the session duration, the hourly average session volume in Figure 3.7 is larger on the weekend and varies in dependence of the daytime.

While the activity period is described by the session volume and duration, the load imposed to the network depends on the inter-arrival time of the sessions, that is, the interval from the start of a session to the start of the next session. In the average, a new session starts every 19 seconds and the CoV of the corresponding distribution is 2.0. An inter-arrival time of 24 seconds reflects the reduced user ac-

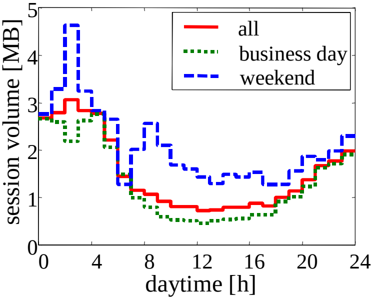

Figure 3.7: Session volume in dependence on daytime.

tivity on weekends, which is observed in Figure 3.2. The distribution of the inter-arrival time for business days and weekends, depicted in Figure 3.8, shows a heavy-tailed behavior in two orders of magnitude.

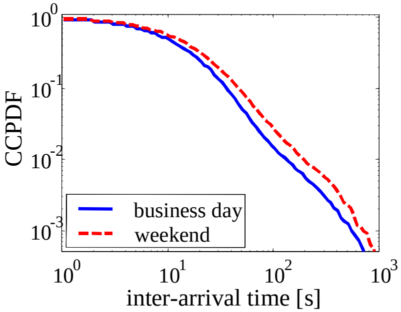

Figure 3.8: Session inter-arrival time during business days and weekend.

When the observation of the inter-arrival time is restricted to intervals depending on the daytime, the distributions show an exponential decay and have a CoV of 1.0. Figure 3.9 depicts this behavior for three one-hour intervals, that is, the exponential decay is reflected by a line in the semi logarithmic plot. The exact numbers are listed in the Appendix, Table 8.3.

3.2.2 Application Level

The user behavior is reflected by the volume and characteristics of the application generated traffic. In Table 3.1 and Table 3.2 we list the percentage of the most frequent used applications for TCP and UDP, respectively. We consider only applications with a share of more than 1% of the packets in relation to the total number of the protocols packets. The percentage of the traffic volume and the numbers for the upstream share of each application are also given. For each protocol a significant number of port combinations is not resolved; each of these port combinations

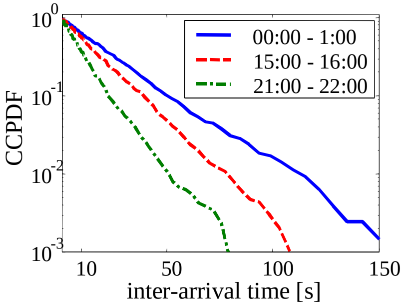

Figure 3.9: Session inter-arrival time for selected hours.

has a share of less than 1% of the protocols traffic and is in most cases not registered by the IANA and [79], respectively.

TCP port	percentage of total TCP traffic		upstream percentage	
	packets	bytes	packets	bytes
HTTP	77.9	80.1	49.0	12.4
HTTPS	0.8	0.6	50.4	19.8
MAIL	5.6	5.6	47.2	32.5
FTP	2.8	3.0	44.1	14.3
TELNET	1.0	0.2	54.8	33.9
NEWS	1.0	1.2	45.2	6.5
other	10.9	9.3	-	-

Table 3.1: TCP application usage.

Almost all TCP traffic is attributed to WWW traffic, that is, traffic transmitted with the HTTP protocol. We distinguish in Table 3.1 HTTP, which takes 80% of all TCP traffic, and secure HTTP, which takes about 1% of the HTTP traffic and is often used for online banking or similar applications. With a share of 5.6%

and 2.8% for mail and FTP, these protocols have to be taken into account for the modeling of Internet traffic, while telnet and network news are of minor importance. In comparison to prior measurements in wide area networks [13], interactive remote access, that is, telnet and rlogin, lost its importance nowadays by a reducing the share from 45% to 1%. In current measurements [46] these services are even not named among the top 25 applications. The POP3 protocol, used for accessing mail on a local server, and SMTP used for sending mail to the server and relaying mail between servers, are subsumed under the keyword *mail*.

Games are the predominating origin of UDP traffic. At least 60% of all UDP packets are related to the game *Quake* (ports 27910, 27911, 27912, 27920) and *Battlenet* (port 6112). While games are dependent on user trends and change with the time, basic system services like DNS (port 53) show up in past and current measurements [13, 46].

UDP port	application	percentage of total UDP traffic		upstream percentage	
		packets	bytes	packets	bytes
6112	BattleNet	21.3	13.3	51.1	50.5
27910	Quake	17.6	16.3	72.7	49.7
53	DNS	13.3	16.6	51.4	26.5
27912	Quake	11.9	9.3	82.8	67.4
27920	Quake	6.9	5.7	73.8	55.2
137	NetBIOS	6.1	5.1	98.8	96.8
27911	Quake	4.9	3.9	77.8	60.8
4000	ICQ	1.6	1.0	53.7	53.5
other		16.5	28.8	-	-

Table 3.2: UDP application usage.

In the most cases UDP applications utilize the network symmetrically, that is, almost the same data volume is transmitted in up- and downstream direction. The exceptions are DNS, where answers cause three times the volume of the requests, and NetBIOS name service (port 137), where due to miss configuration only requests are sent. TCP applications normally receive more data from the Internet than are submitted to the network. This behavior is most obvious in the case of HTTP and FTP, where short requests cause large data downloads.

3.2.3 Transport Level

Protocol Analysis

Within the sessions the traffic is transferred by different protocols, according to the requirements of the applications. While TCP accounts with 92% of the transmitted packets for almost all measured traffic, also 7.3% of UDP and 0.6% ICMP packets are transmitted. In the average TCP packets are larger than UDP packets, and thus 97% of the traffic volume expressed in bytes belongs to the protocol TCP, cf. Table 3.3.

protocol	packets	volume [MB]	packets [%]	volume [%]
TCP	2.0480e+008	79962	91.94	96.90
UDP	1.6545e+007	2440	7.42	2.96
ICMP	1.3383e+006	102	0.60	0.12
other	7.9047e+004	13	0.04	0.02

Table 3.3: Protocol usage in measured trace.

While the number of TCP packets in upstream and downstream direction is almost identical, more than 86% of the TCP volume is transmitted downstream, that is, from the Internet to the user. For UDP the traffic is more balanced, 40% of the UDP packets transmitted downstream account for 60% of the traffic volume. The asymmetric profile of Internet traffic will be discussed more in detail in the context of applications and TCP in the following sections.

TCP

We examine the characteristics of the TCP connections, since 97% of the data traffic in the trace is transferred connection-oriented with the TCP protocol. The average TCP connection has a duration of 23.8 seconds and the corresponding dis-

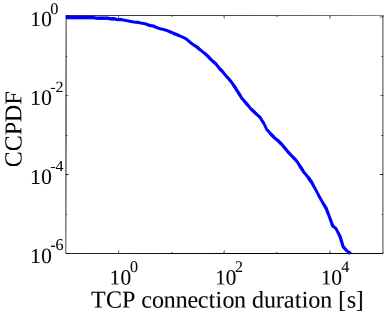

Figure 3.10: TCP connection duration.

tribution has a CoV of 4.3. The average data volume, including Ethernet, IP and TCP overhead, is 12.9 kB with a CoV of 12.7. In Figure 3.10 and Figure 3.11 the distributions of the TCP connection duration and volume in up- and downstream direction are depicted, respectively. The complementary distribution functions show in double logarithmic display a linear decay over at least three orders of magnitude, which indicates that these distributions are heavy-tailed [21, 45].

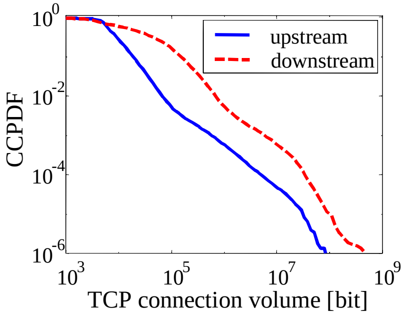

Figure 3.11: TCP connection volume in up- and downstream direction.

A feature of relevance for measuring heavy-tailed distributions is the fact, that depending on the steepness of the decay the variance and mean of the distribution becomes infinite. This property implies that the above mentioned mean values and variances of the measurement may increase when the number of samples is increased. A detailed description of heavy-tailed distributions and the estimation of their parameters will be given in Chapter 6.

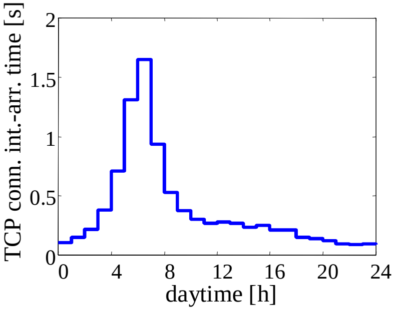

Figure 3.12: Average TCP connection inter-arrival time over daytime.

The inter-arrival time of the TCP connections in the aggregate traffic stream is inverse proportional to the number of parallel sessions in the system, that is, as depicted in Figure 3.12 short inter-arrival time in the evening and larger time between TCP connections in the morning hours. The average inter-arrival time of TCP connections is 0.2 seconds and the corresponding distribution has a CoV of 2.5. If the observation of the inter-arrival times is restricted to intervals of one hour, the arrival rate changes according to the session activity and the variance is reduced. Contrary to the session inter-arrival time the CoV of the TCP connection inter-arrival time is bigger than 1 and the tail of the distribution is not exponential, cf. Figure 3.13.

Due to the feedback mechanism of TCP, described in Section 2.4.2, at least for every second data packet a acknowledgement has to be transferred. Thus, the

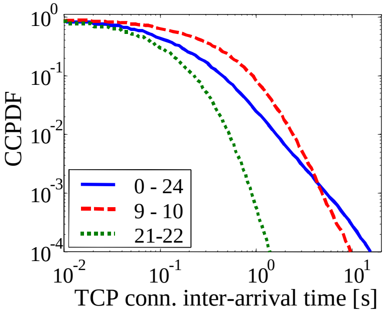

Figure 3.13: TCP connection inter-arrival-time distribution for selected hours.

number of packets in up- and downstream direction is correlated. Figure 3.14 depicts the scatter plot of the up-/downstream rate (volume) ratio over the volume transmitted in downstream direction. The intensity of the scatter points is reflected by the color intensity. The largest intensity is detected along the black line, which is expressed by the following equation, valid for HTTP connections [16]:

$$\frac{v_u}{v_d} \approx \frac{580\ \text{Byte}}{v_d} + \eta \cdot \frac{60\ \text{Byte}}{1500\ \text{Byte}}\,. \tag{3.1}$$

The relation of upstream volume v_u and downstream volume v_d is approximated by the ratio the acknowledgements, that is, 60 Bytes including Ethernet overhead, and the maximum packet size (1500 Bytes) for large connection volumes, where the parameter $1/\eta$ denotes the number of data packets acknowledged with a single acknowledgement. For low connection volume, the upstream share v_u is determined by the size of the request and the connection setup overhead which sums up to 580 Bytes in average.

Equation (3.1) is valid as well for connections that transmit data in upstream direction, but the directions in the equation have to be switched. In Figure 3.14 such connections appear in the upper middle of the plot and are not aligned to the black line.

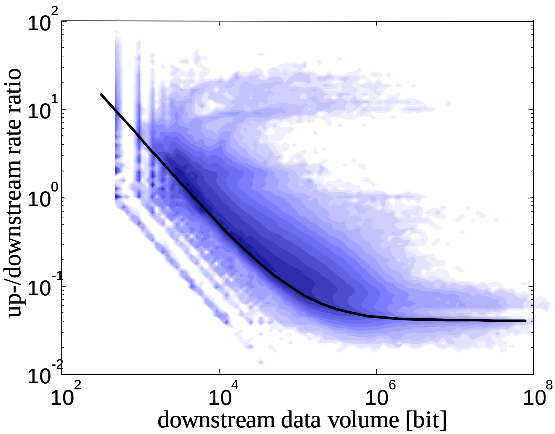

Figure 3.14: TCP connection up-/downstream correlation.

UDP

The protocol characteristics of UDP do not alter the application specific traffic properties. We will review some UDP applications, that is, DNS and game traffic in the next chapter.

4 Decomposing Applications into TCP-Connections

We identified the applications HTTP, FTP and mail in the previous chapter as the main sources of Internet traffic evaluated in the measured trace. Together these applications account for almost 90% of the measured traffic volume. In the following we will breakdown the traffic of these applications to TCP connections, identifying the protocol entities as described in Chapter 2. We will also examine Quake and DNS as representations of UDP traffic.

In common dial-in sessions the users restrict themselves not to a single application, but utilize a subset of the mentioned applications, e.g., a user is not only surfing in the WWW but also reads his e-mail. This application usage plays an important role for modeling and evaluating the data flow within a session. To reflect the application usage in reality, the services are evaluated under the condition that it is actually used within a session, that is, all distributions and values are evaluated only for sessions in which they are actually used.

4.1 Application Utilization

Before analyzing the main traffic sources in detail we summarize the application usage in session context. Table 4.1 lists the session division depending on application and service usage. We discern service utilization independent of other services in the same session, sessions that utilize only a single service and sessions that contain a single service in conjunction with system services, such as DNS or NetBIOS.

HTTP, utilized in 66% of the sessions is the most frequently used service, followed by reading mail with the POP3 protocol, which is used in 62% of the sessions. One or more mails are sent with the SMTP protocol in 20% of the sessions. Mail summarizes the usage of POP3 or SNMP, which is done in 65% of all sessions. FTP is utilized in only 6% of the sessions, while not in every session that contains FTP-control data also a FTP data download is done (2.1% of the sessions). In 82% of the sessions DNS is used and NetBIOS data is sent in 46% of the sessions. Quake is only measured in 0.2% of the sessions, but is mentioned here due to the largest average session volume.

service	percentage of sessions with		
	service	exactly one service	one service and DNS or NETBIOS
HTTP	66.2	2.76	16.26
FTP data	2.1	0.00	0.01
FTP control	5.9	0.00	0.02
FTP	5.9	0.03	0.17
SMTP	19.8	0.34	1.39
POP3	62.2	3.59	14.34
Mail	64.8	5.11	22.77
other TCP	25.6	0.24	1.21
Quake	0.2	0.00	0.00
DNS	82.4	1.13	-
NetBIOS	45.6	0.13	-
other UDP	23.1	0.05	0.61

Table 4.1: Correlation of common application/services.

Generally the services are not used on a stand-alone basis. Mail is used as single application in 5% of the sessions and HTTP in 3%. If we do not account DNS and NetBIOS, that is, data which is not controlled by the user, mail is used as single application in 23% of the sessions and HTTP in 16% of the sessions. The other services do not tend to be used stand-alone in significant numbers.

Some applications like FTP and mail utilize two services. FTP data is always used together with FTP control and sending mail with SMTP is combined in most cases with reading mail via POP3.

The volume of the applications used stand-alone (regardless of DNS and NetBIOS usage) as well as in combination with other applications is shown in Table 4.2. The volume of HTTP is reduced to the half if the service is not used in combination with other services. The usage of FTP nearly doubles, but since FTP is only used in a small fraction as stand-alone application this result is of little significance. The Mail usage does not change significantly. The volume dependence of the application usage is of importance when modeling high priced small bandwidth connections, such as wireless access networks in which a more focussed usage is expected.

volume of services in sessions	service usage [kB]	single service (incl. DNS/ NetBios) [kB]
HTTP	1354.8	759.5
FTP data	1643.8	n.a.
FTP control	8.1	n.a.
FTP	583.0	998.1
SMTP	90.0	88.1
POP3	66.2	62.6
Mail	91.1	76.6

Table 4.2: Volume correlation for single application/service usage.

In the following sections we will describe the components and characteristics of the different services in session context. We will use the term HTTP session to indicate a dial-in session in which HTTP is used. Alike, we will refer to FTP or mail sessions. All data in the next sections is evaluated under the condition that a service is actually used and could be unconditioned by the values given in Table 4.1.

4.2 HTTP

In an average HTTP session a volume of 1.35 MB of data is transferred in 128 TCP connections. According to the HTML protocol capabilities presented in Section 2.3.1, these HTTP/TCP connections can be grouped into web pages. Normally a web page consists of an initial file, which references inline objects, such as frames, tables and images. By requesting the initial file, all inline objects are transferred automatically. In the measurement evaluation context, we have no access to the content of the transferred documents. Thus, we define a web page as a set of HTTP connections, transferring continuously data. A time-out (gap) of more than one second indicates the start of a new page. Evaluating the measure-

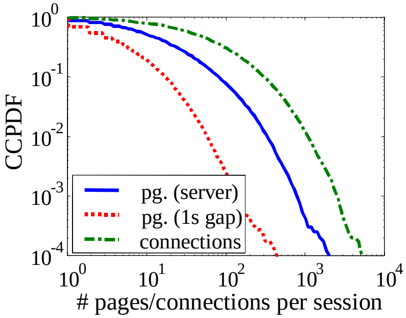

Figure 4.1: Number HTTP of connections and pages per session.

ment data it is not possible to detect parallel page downloads, that is, several active browser instants at the same time.

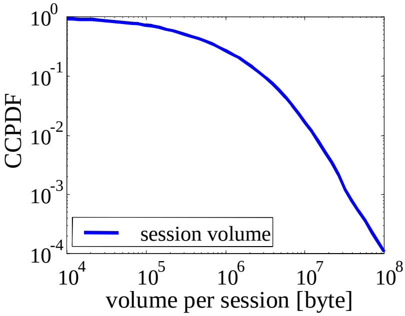

Figure 4.2: HTTP session volume.

A second web page definition, according to [47], not only discerns the pages by a one second gap but also assumes that all elements of a page are hosted on the same server. This definition, taking into account the locality of web pages provides a lower bound for the size of web pages.

4.2.1 HTTP Session and Page Volume Characteristics

Depending on the locality definition of a page, the average number of pages downloaded in a session differs. In the average 8.8 pages constructed of 14.5 TCP connections are downloaded in an a HTTP session, if the pages are discerned by time-out. When taking into account pages provided by a single server, 35 pages with 3.7 TCP connections each are contained in an average HTTP session.

The distributions of the HTTP session and page characteristics, i.e., the number of connections per session in Figure 4.1, the session volume in Figure 4.2, the number of connections per page and the page volume, Figure 4.3 and Figure 4.4, show a heavy-tailed behavior. According to this property the CoV of these distri-

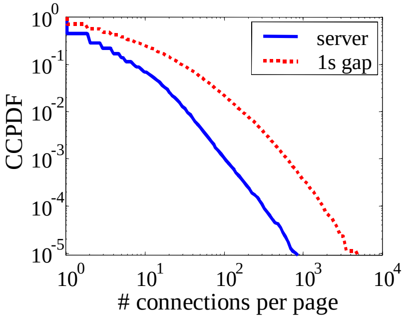

Figure 4.3: HTTP connections per page.

bution ranges from 2 to 3, with the exception of the distribution of the page volume, which has a CoV of 8.6 or 6.2, depending on the page definition, respectively. A summary of the HTTP session characteristics parameters is given in Table 4.3

.

page discrimination	by server and 1s gap		by 1s gap	
	mean	CoV	mean	CoV
session volume	1354.8 kB	2.7	~	~
connections per session	128.1	2.2	~	~
pages per session	34.8	2.4	8.8	2.0
page volume	38.9 kB	8.6	153.8 kB	6.2
connections per page	3.7	2.8	14.5	3.8

Table 4.3: HTTP session characteristics parameters.

4.2.2 HTTP Temporal Characteristics

A traffic characterization in our model is composed by the volume of the transferred objects and the temporal relation of the objects. In case of web pages, the

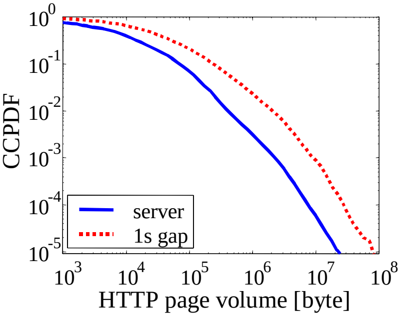

Figure 4.4: HTTP page volume.

temporal relation can be described by the inter-arrival time or the user-think time. The inter-arrival time describes the interval between the starting times of web pages. The user-think time describes the time from the end of a page download to the start of the following page download. When discerning the pages by a gap of 1 second and locality information, pages might overlap which is tallied as negative user-think time.

page discrimination	by server and 1s gap		by 1s gap	
	mean [s]	CoV	mean [s]	CoV
inter-arrival time	32.4	3.6	117.7	2.7
user-think time	33.1	4.3	39.7	4.5
negative user-think time	-49.8	2.4	-	-

Table 4.4: HTTP web page inter-arrival time and user-think time parameters.

Table 4.4 summarizes the averages and CoVs of the HTTP web page inter-arrival time and of the user-think time. The related distributions are depicted in Figure 4.5 and Figure 4.6 and show a heavy-tailed behavior over three orders of magnitude. The inter-arrival time of pages determined by locality information is less than one third of the pages discerned by a 1 second gap. This is explained by the higher number of smaller pages downloaded in a session.

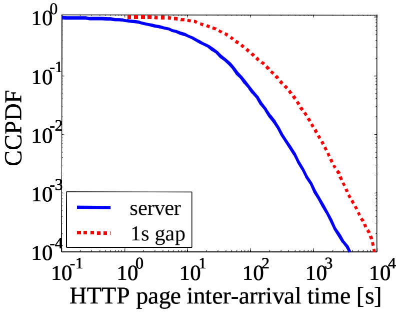

Figure 4.5: HTTP web page inter-arrival time.

The positive user-think time of pages according the locality definition is slightly shorter than the user-think time for sessions discerned by 1 second gap. This averages are computed only for positive values of the user-think time, that is, with reduced influence of overlapping pages. Some overlapping pages lead also to positive user-think time, for example, if several short pages are downloaded in parallel to a large page. A negative user-think time is measured for one half of the pages and sums up to a significant fraction of negative user-think time.

4.2.3 HTTP Initial Object Size

While a average HTTP/TCP connection has a data volume of 10.5 kB, the measurement shows differences in the connection volume when taking into account the position of the connection in relation to the HTTP page. We discern *initial connections*, i.e. the first connection of a page, *follow-on* connections which continue a page, and *single* connections, i.e. pages that contain only one connection. Single connections are not tallied in the category of initial connections.

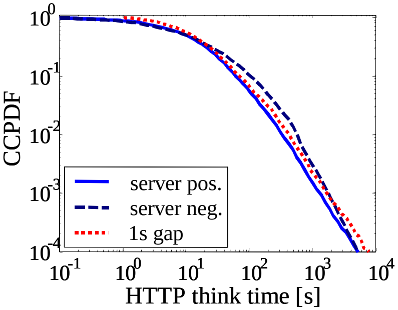

Figure 4.6: HTTP user-think time.

Table 4.5 summarizes the characteristics of the HTTP connection volume distribution functions shown in Figure 4.7. The average initial connection of a page is larger than the follow-on connections, in which usually icons, i.e. small pictures, are transferred. For pages that are discerned by time-out and single server, 55.7% of all pages consist of single connections, which are of similar average size as the follow-on connections but have a significantly higher variance.

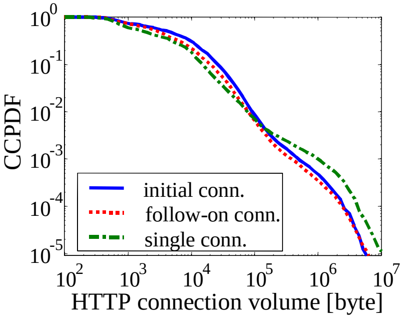

Figure 4.7: HTTP connection volume.

If the pages are discerned by time-out only, 30% of the pages consist of a single connection. These single connection pages are 50% larger in the average than the single connection pages for the pages discerned by server for two reasons. First, small inline objects like advertisement banners, are hosted by different servers and account in this page definition as follow-on connections. Secondly, counting these otherwise denoted single connections as follow-on connection, gives more weight to large downloads, e.g. images or files.

TCP connection	per server and 1s time-out		1s time-out	
	mean	CoV	mean	CoV
initial	13.1 kB	4.8	16.3 kB	6.6
follow-on	10.1 kB	5.8	10.2 kB	6.6
single	10.6 kB	11.4	15.4 kB	10.3

Table 4.5: initial request / follow-on downloads.

As depicted in Figure 4.8, the data rate is not correlated to the HTTP page volume, but large pages are not downloaded at low data rates due to user impatience.

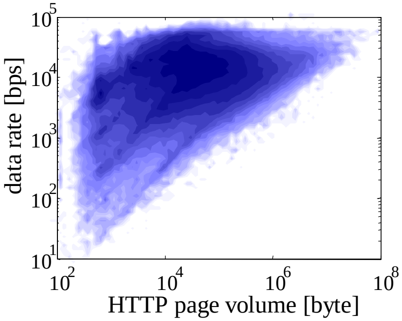

Figure 4.8: HTTP volume – data rate correlation for pages discerned by 1s time-out and locality.

4.3 FTP

The FTP protocol is used for data transfer in the Internet. It utilizes two different ports for transmitting control information and for actually transferring data. While the FTP control protocol is utilized in 5.7% of all sessions, data is transferred with FTP in only 2% of all sessions. Due to their negligible volume the control connections are from the modeling viewpoint of minor interest, but are also listed in Table 4.6 and shown in the following figures for completeness.

4.3.1 FTP Session and Connection Volume

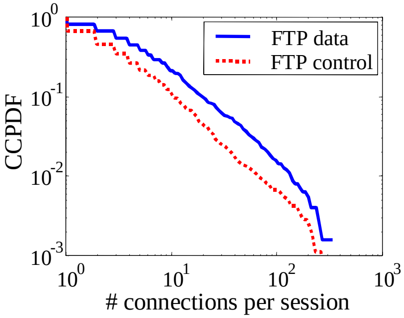

Figure 4.9: Number of FTP connections per session.

Under the condition that data is transferred, an average FTP data session is with a volume of 1.6 MB larger than an average HTTP session, but contains only 11 connections in the average. The distributions of the number of FTP connections and volume per session are depicted in Figure 4.9 and Figure 4.10, respectively. The corresponding distribution parameters are summarized in Table 4.6.

As consequence of number of connections per session and the session volume, the average FTP data connection has a volume of 150 kB. The FTP data con-

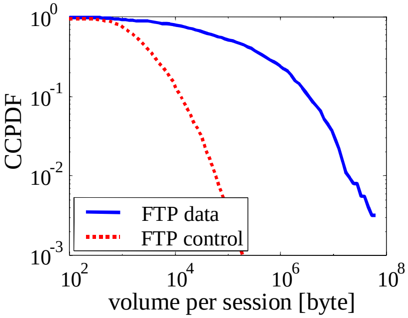

Figure 4.10: FTP session volume.

FTP	session			
	# connections	CoV	volume	CoV
control	6.0	3.0	8.1 kB	12.7
data	10.9	2.5	1649 kB	3.5

Table 4.6: FTP session characteristics distribution parameters.

nection volume distribution depicted in Figure 4.11 does not show the heavy-tailed behavior that is observed with HTTP connections. The complementary distribution function decays linearly up to a connection volume 1 MB, and then descends steeper due to the effect that FTP is preferred for larger volumes combined with a maximum observed file size.

4.3.2 FTP Connection Inter-Arrival Time

For FTP control connections a mean inter-arrival time of 169 seconds is obtained from the measurement data. These control connections represent the frame in which the FTP data connections are generated. The data connections are seen with a inter-arrival time of 58 seconds which is shorter than the average FTP data con-

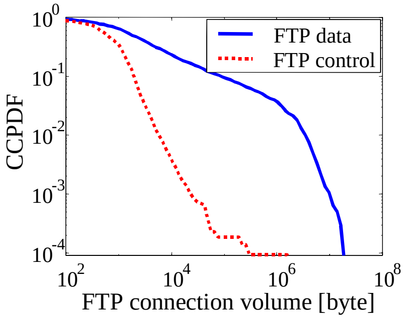

Figure 4.11: FTP connection volume.

nection duration of 64.7 seconds. This discrepancy to the FTP model, that allows exactly one control connection and one data connection is resolved by the fact that the last connection in a FTP session has – with 521 kB and 240 seconds – a significant larger volume and duration than an average connection. The connections for which we can compute the inter-arrival time have a duration of 47.0 s and a volume of 113.4 kB. The inter-arrival time is composed of a major part by the transfer duration and of a smaller fraction by the user interaction, that is, the time the user needs to initiate a download or to request a listing.

The FTP inter-arrival times, summarized in Table 4.7, have a large coefficient of variation. Even so, the according distributions depicted in Figure 4.12 show no distinct heavy-tailed pattern.

FTP	TCP connection			
	volume	CoV	inter-arrival time	CoV
control	1.3 kB	21.0	169.1 s	3.3
data	151 kB	5.5	58.3 s	4.8

Table 4.7: FTP connection characteristics distribution parameters.

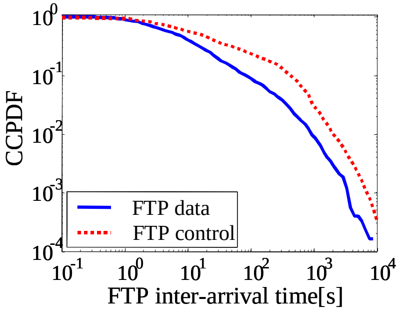

Figure 4.12: FTP connection inter-arrival time.

The data rate of FTP data connections is slightly correlated to the connection volume. As depicted in Figure 4.13 the data rate grows with increasing connection volume. The low rate obtained for small connection volume is caused by the initial TCP slow start phase, even if the servers accessed offer higher data throughput.

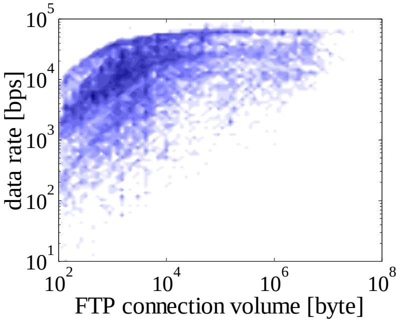

Figure 4.13: FTP data rate to connection volume correlation.

4.4 Mail

Besides WWW browsing, mail is the most frequently used application. In more than 63% of all sessions either mail is retrieved or sent. Two protocols are related in the evaluated data to the mail application – SMTP for sending mail and POP3 for retrieving mail. According to the intention of these protocols the traffic volume is highly asymmetric.

Even if the total mail volume of an average session does not justify the detailed modeling of mail traffic from a performance viewpoint in broadband scenarios, the modeling of mail gains importance in a low-bandwidth mobile access network scenario.

4.4.1 Mail Session and Connection Volume

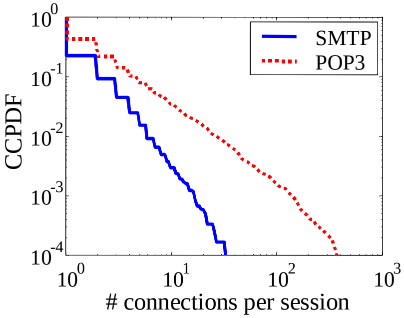

Figure 4.14: Number of connections per mail session.

Figure 4.14 shows the distribution functions of the number of connections per mail session. While the distribution function of the number of connections per session shows a linear decay over two orders of magnitude, this property is found only partially in the volume distribution, that is depicted in Figure 4.15. In the average 1.5 SMTP connections are used to send mail within a session and 3 POP3 connections are used to read mail from the server. The volume of an average POP3 session is with 66 kB in the same order of magnitude as the volume of an SMTP session with 90 kB. The parameters of the corresponding mail session characteristics are summarized in Table 4.8.

Mail	session			
	# connections	CoV	volume	CoV
SMTP	1.5	0.97	89.9 kB	13.5
POP3	2.9	3.7	66.3 kB	14.5

Table 4.8: Mail session distribution characteristics.

In common e-mail clients, the first POP3 connection is usually triggered by the start of the client and could be followed by periodic “mail-check” connections. SMTP connections are usually initiated by the user after editing a message. As consequence of these “mail-check” connections the number of POP3 connections in an average session is larger than the number of SMTP sessions, but the volume of a single POP3 connection is smaller in the average and has higher variance than a SMTP connection.

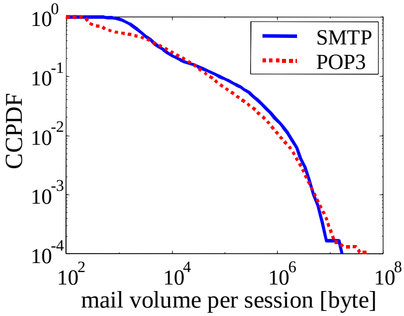

Figure 4.15: Mail session volume.

The distribution of the POP3 connection volume in Figure 4.16 shows a significant share of connections smaller than 300 Bytes. These connections represent mail check attempts, where no new mail is found for download. For the remainder of the mail connections, that is, connections in which actually mail is transferred, the distributions for POP3 and SMTP connections and have similar shape. A summary of the distribution parameters for the mail connection size can be found in Table 4.9.

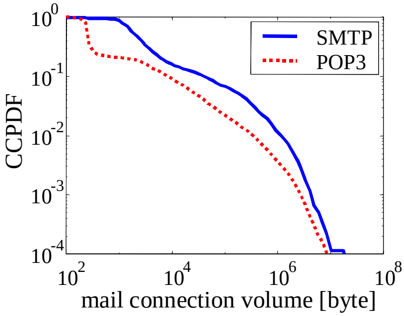

Figure 4.16: Mail connection volume.

4.4.2 Mail Connection Inter-Arrival Time

Mail	TCP connection			
	volume	CoV	inter-arrival time	CoV
SMTP	62.1 kB	13.8	304.8 s	3.0
POP3	22.5 kB	24.7	172.3 s	3.0

Table 4.9: Mail connection distribution characteristics.

The inter-arrival time of mail connections is of low significance due to the small number of mail connections in an average session. The numeric average values

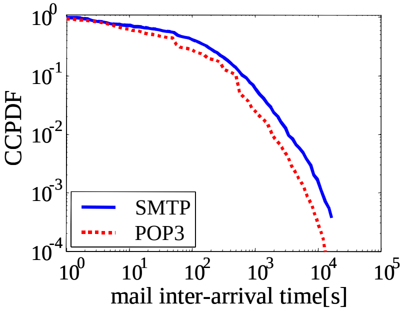

Figure 4.17: Mail connection inter-arrival time.

are summarized in Table 4.9. For POP3 connections the irregularity in the distribution at 60 seconds observed in Figure 4.17 is of interest. This indicates an interval for regular automatic mail checks, which would normally be done in an LAN scenario. The remaining curve is dominated by the user interaction, that is, the time to write multiple e-mails or to check repeatedly for e-mail.

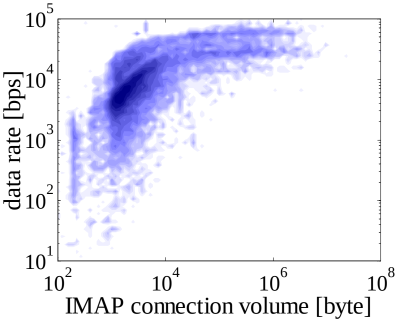

Figure 4.18: SMTP data rate – volume correlation.

The data rate obtained for mail connections is correlated to the connection volume according to the TCP rate control. For increasing data volume, as well for SMTP as for POP3 connection the data rate increases to the limit given by the access speed. Comparing Figure 4.18 and Figure 4.19, the “mail-check” connections of the POP3 protocol are an exception to this correlation pattern, since for these connections the size is constant, but the transaction time depends on user interaction for providing log-in data.

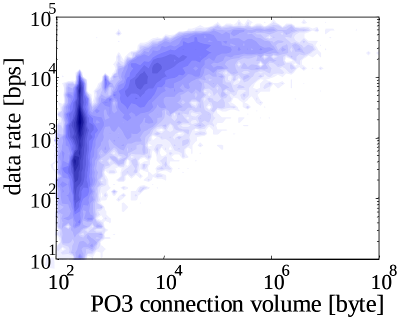

Figure 4.19: POP3 data rate – volume correlation.

4.5 DNS

The Domain Name Service (DNS) is utilized in more than 80% of the sessions to resolve mnemonic device names to IP addresses. As such the DNS service is applied before the establishment of TCP connections or UDP flows and adds serial delay to connections [17]. Thus, DNS is an important component to the Internet usability even if the transfer volume in the measurement accounted for only 0.4% of the total traffic volume.

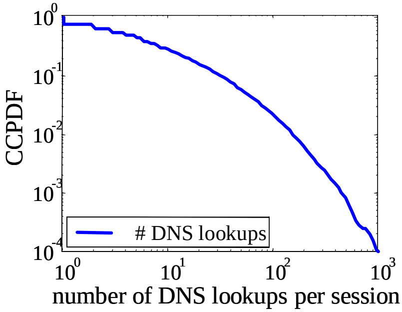

Figure 4.20: Number of DNS lookups per session.

In an average session that utilizes DNS 14.6 DNS lookups with a coefficient of variation of 2.6 are performed. As depicted in Figure 4.20 the number of DNS lookups per session ranges from 1 to 1000. An IP address has to be resolved every 9th connection. For HTTP traffic this value indicates a DNS lookup before every second page, when discerning the pages by 1s gap and locality, c.f. Section 4.2.

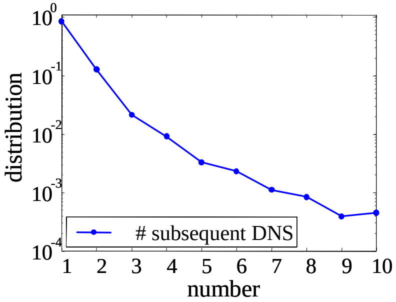

Figure 4.21: Subsequent DNS lookups before connection establishment.

Normally a DNS lookup is done by sending a query to a sever and awaiting the response, which are transferred in a single UDP packet, each. The query requires an average volume of 79 Bytes with a CoV of 0.1 and the response 228 Bytes with a CoV of 0.3. In some cases several query/response pairs are required to do the DNS lookup – the corresponding distribution is shown in Figure 4.21. In the average 1.3 query/response pairs with a CoV of 1.1 are required for a DNS lookup. This can be explained by DNS lookups to non existing domain names, either caused by input failure or browsers that try to complete domain names (so called ".com completion").

DNS lookup delay	delay [s]	CoV
single DNS	1.0	9.1
subsequent DNS	1.9	9.3
DNS to connection	1.2	10.9
total DNS delay	3.1	7.1

Table 4.10: DNS lookup delay summary.

Figure 4.22 summarizes the delay introduced by DNS lookups. The left graph shows the time distribution between DNS request and response and the distribution for the time from the first request to the last response, that is the DNS lookup delay for subsequent DNS. The average single DNS lookup takes 1.0 seconds, the duration of subsequent DNS lookups is 1.9 seconds. Both delays have a CoV larger than 9.1, which indicates that some very large values influence the average. More than 40% of the DNS lookups introduce a delay of less than 200 ms, but 10% of the DNS lookups have a delay of more than 3 seconds.

The second component that adds to the delay before establishing a connection is the time from the receipt of the DNS lookup response to the establishment of the connection. The graph in Figure 4.23 depicts the time between the receipt of a DNS reply to the establishment of a connection and the time required for the

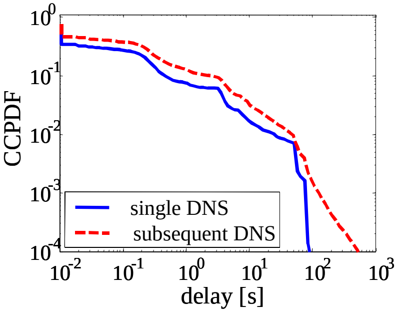

Figure 4.22: DNS delays: Time for single and subsequent DNS lookups.

complete DNS lookup to the connection establishment. The average times and coefficients of variation are summarized in Table 4.10.

Note, that the measurement influence the results for the DNS lookup performance in two ways. First, the measurement takes place before the bottleneck of the network scenario. The hosts perceive some additional delay resulting from the transmission of the queries and results over the modem line. Second, the measure-

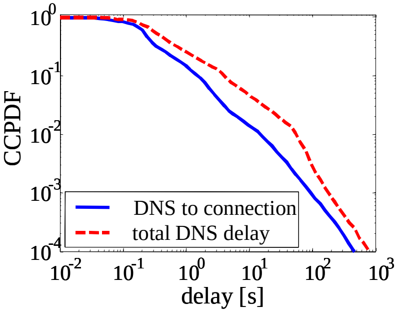

Figure 4.23: Time from receipt of DNS query to connection opening.

ment did not record the success of DNS lookups. The delay between a DNS lookup and the establishment of a connection, as well as the number of subsequent DNS query/response pairs contains unresolved requests. The graph, c.f. Figure 4.23, does not allow to discern successful DNS lookups that are followed by connection establishment and the sum of unsuccessful and successful DNS lookups.

4.6 Game Traffic – Quake

As an example for game traffic characterization we concentrate on the measured “quake” data. In the trace, at least 60% of the measured UDP packets are related to game traffic – the share of quake is at 2/3 of the game traffic. In relation to the all measured traffic, online games account for 4% of all packets. This value holds not only for Internet access traffic, but was also found typical for Internet backbone traffic [46].

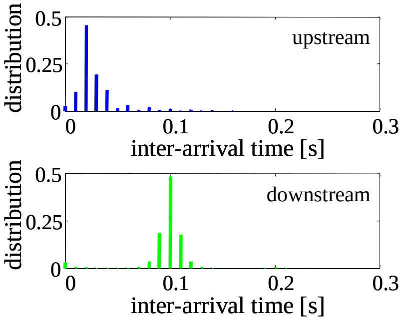

Figure 4.24: Network game traffic: Quake packet inter-arrival time.

Game traffic, especially first person shooters such as quake, half-life or counter strike are considered to be extremely delay-sensitive. As such, those games require a similar network quality as voice or interactive video conferencing [10, 27].

Quake follows a strict client/server traffic model. The complete game scenario is managed at the server site. The server sends the state information individually to all clients. The clients render the game situation and give user interaction feedback, that is the players movement and status, to the server. The packet rate may depend on the clients capabilities. Since the server has to update all subscribed clients, its aggregated traffic profile is bursty [27].

The user view of quake is recorded in the measurement. That is, the data to and from a single client is evaluated. In an average Quake session 6.9 MB of data including 4.8MB of UDP – the actual data related to Quake – are transferred in 36 minutes. The session duration is more than double compared to an average session and varies less, indicated by a CoV of 1.2 compared to a CoV of 2.0 for average sessions.

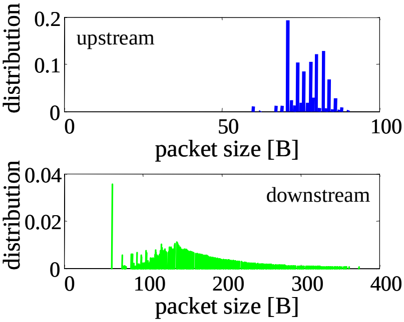

Figure 4.25: Network game traffic: Quake packet size.

The part of the session relevant for modelling the time critical segment is the activity phase, in which the actual game takes place. This phase is characterized by the transfer of UDP packets. Figure 4.24 shows the timing of the UDP packet transfer during activity phases and Figure 4.25 depicts the size of the transmitted

UDP datagrams. In upstream direction, that is, from client to server, on average a packet of 77 Bytes is transmitted every 30 ms. The variation is low, with a CoV of 0.03 regarding packet size and 1.0 regarding inter-arrival time. In downstream direction an average packet size of 192 Bytes with CoV of 0.62 is observed. These packets are transferred every 100 ms with a slight jitter, leading to a CoV of 0.39 in the packer inter-arrival time. The packet size and inter-arrival time of the quake data produces a continuous date rate of 20 kbps upstream and 15 kbps downstream that can usually be provided by dial-in modems.

5 Influence of the Access Speed on the Traffic Characteristics

In the previous chapters we studied the protocol characteristics and properties of the measured Internet traffic. The trace evaluated was created by dial-in modem users accessing the Internet with differing access speeds. To model future access network technologies such as GPRS, UMTS or optical media, the dependence of the traffic profile from the access speed is analyzed. To evaluate Internet usage trends with regard to increasing access-speed, we relate our results to results derived from traces with faster access speed, i.e., we take into account the traffic profile of an ADSL trace.

5.1 Measuring the Access Speed

In general, packet traces contain no information about the equipment used to access the network. Information about the access speed can be derived in two ways.

The direct way is to measure the traffic in an environment with known infrastructure, where the access is provided by a particular technology. For the cited ADSL trace [15, 16], the participants took part at a field trial with controlled access technology.

The trace we refer mainly in this work (c.f. Section 3.1.1) was recorded at an Internet access based on dial-in routers. These routers are centrally managed with authentication servers, such as TACACS [31] or RADIUS [80]. The authentication servers provide detailed log-files that contain a protocol of the authentication, authorization and accounting process. Besides the information about the session duration and the assigned IP address we evaluated the up- and downstream access speed information.

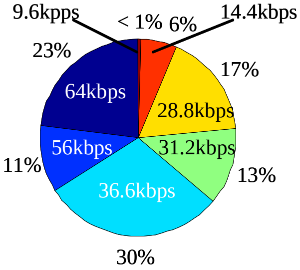

Figure 5.1: Percentage of modem classes in the measurement.

The pie chart in Figure 5.1 shows the share of modem classes in the trace. The main part of the users accessing the Internet in a speed range from 28.8 kbps to 33.6 kbps. One third of the sessions are connections with up to date technology, that is ISDN (64 kbps) or 56 kbps modems. Note that the later class of modem connects asymmetrically, upstream with 33.6 kbps and downstream depending on the quality of the line up to 56 kbps. As shown in the dial-in speed histogram in Figure 5.2 the actual connection speed of 56 kbps modems is between 40 kbps and 56 kbps. Only 7% of the sessions stem from slow 9.6 kbps and 14.4 kbps modems. Since the characteristics of the 28.8 kbps, 31.3 kbps and 33.6 kbps connections show no significant difference in the following graphs, we will integrate the data in a class denoted as 30 kbps modems.

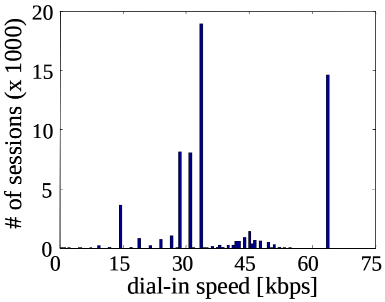

Figure 5.2: Histogram of dial-in speeds.

5.2 Session Characteristics

The following section evaluates the session characteristics in dependence of the access speed. The related analysis of the trace was presented in Section 3.2.1. We will look at the distribution of the session duration and volume.

5.2.1 Session Volume and Duration

Table 5.1 shows the average session volume and duration in dependence of the modem speed. The average dial-in session has a duration of 15 minutes and a volume of 1.2 MB of data.

The two slowest modem classes have a significantly shorter duration and lower transfer volume than the remaining classes. While the average session duration of the dial-up modems increases only slightly with the access speed the data volume raises proportionally to the modem speed. The coefficient of variation is nearly independent of the modem speed.

The session duration of the ADSL connections is significantly larger than that of dial-up modems, which originates from the free access. The data volume also

		Duration		Volume	
Speed	#Sessions	Mean [s]	CoV	Mean [kB]	CoV
9.6 kbps	273	536	1.93	241	2.68
14.4 kbps	3624	644	2.20	540	2.85
28.8 kbps	10602	885	1.97	1133	3.31
31.3 kbps	7969	904	1.89	1135	2.80
33.6 kbps	18538	896	1.91	1137	2.68
56 kbps	6809	964	1.85	1578	2.44
64 kbps	14268	979	2.14	1866	3.70
ADSL-up	3551	3376	2.33	1933	19.15
ADSL-down				10020	9.56

Table 5.1: Average and variation of session volume and duration.

increases, but less than expected by the access speedup to a downstream rate of 2.5 Mbps and the extended connection duration. The high bandwidth and session duration of ADSL connections causes also very high variations in the data volume transmitted in the sessions

The session duration distribution is depicted in Figure 5.3. The complementary cumulative probability distribution function shows in double logarithmic

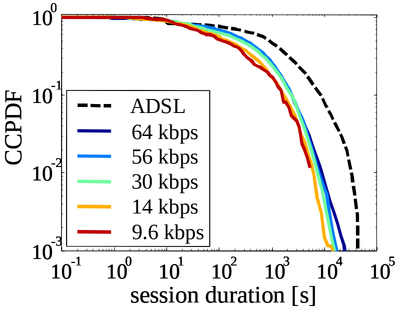

Figure 5.3: Access speed dependent session duration distribution.

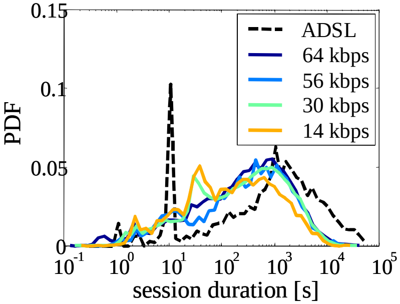

Figure 5.4: Session duration distribution.

plotting that the shape of the distribution is identical for all modem speeds. The probability density function depicted in Figure 5.4 shows two peaks, which suggest to separately analyze short sessions. We will see later that the main application in these short sessions is mail. While a first accumulation of dial-up sessions is observed at a duration of 60 seconds, the peak for ADSL connections is observed at 10 seconds. This is explained by the fact that a simple check for e-mail

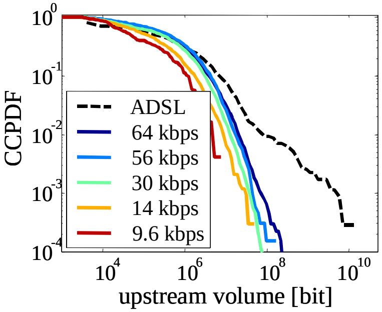

Figure 5.5: Upstream session volume.

could be done very quickly if no dial-in procedure is required, that is, for ADSL modems. About 30% of the sessions do not last longer than 100s.

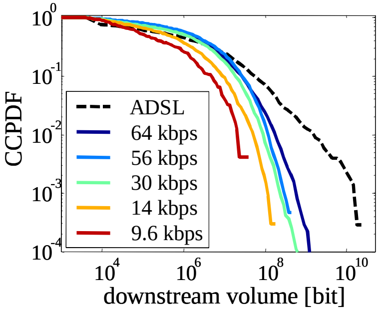

Figure 5.6: Downstream session volume.

Besides the session duration the data volume transferred is used for the characterization of sessions. The up- and downstream volume distribution is shown in Figure 5.5 and Figure 5.6, respectively. For all modem speeds – including ADSL connections – the curves exhibit a similar shape, while the data amount increases with the modem speed. The tail of the distributions decays linearly in double logarithmic representation, which may be an indication of the distributions being heavy-tailed. The ratio of down- and upstream session volume ranges from 4 to 7.

5.2.2 Application Usage

A gain in the access speed allows the comfortable use of new and different services. While it might be acceptable to read Mail at a speed of 14.4 kbps, the speed will not allow the reasonable use of interactive video telephony. To evaluate the service usage dependence on the access speed, we list the volume share of the services used in our measurement in Table 5.2. In the dial-in trace, 80% of the traffic is caused by HTTP. Mail is the preferred application of users with slow 9.6 kbps modems. The percentage of mail reduces with increasing modem speed, while services like FTP or Games are utilized more often. As mentioned in Section 4.6, Games have stringent requirements on the network quality and, thus run only with faster access technology. FTP is used typically for larger downloads (c.f., Section 4.3) which are not practicable for slow access speed.

Speed [kbps]	Total Data Volume [GB]	HTTP	Mail	FTP	Telnet	News	DNS	Games	Other TCP	Other UDP
9.6	0.07	55.03	29.88	0.02	0.31	0.00	0.87	0.00	13.44	0.45
14.4	1.9	80.41	9.27	1.13	0.47	1.32	0.69	0.00	6.29	0.42
28.8	12.0	80.19	9.89	0.98	0.27	2.36	0.42	0.24	4.61	1.04
31.3	9.9	79.18	5.89	1.96	0.08	0.83	0.41	0.43	10.35	0.86
33.6	21.0	83.12	5.42	2.23	0.31	0.49	0.43	0.12	7.28	0.61
56	10.7	79.97	4.58	2.85	0.15	0.55	0.39	0.00	10.49	1.03
64	26.6	74.02	3.29	5.01	0.23	1.50	0.28	3.05	11.51	1.11
ADSL	42.5	12.69	0.71	30.7	0.11	0.30	0.08	0.0	55.4	

Table 5.2: Volume percentage of services.

The ADSL trace exhibits a completely different picture. The main part of the traffic is caused by FTP and a video on demand application (denoted in Table 5.2 as “other TCP/UDP”). Part of the FTP traffic was also related to this video application, since the video server was partially maintained via FTP.

The statistic in Table 5.2 could be somewhat misleading, since one might think that ISDN users read less Mail than users with slower modems or ADSL users are surfing less than users with dial-up modems. Figure 5.7 shows the average data volume per session for the services HTTP, Mail, FTP, and “other TCP/UDP”. While the data volume of HTTP and FTP increase with modem speed, the volume of Mail is independent of modem speed. The reason for this property is that, with the exception of mailing lists, the usage of mail requires writing and sending messages. This limits the volume of mail independently of modem speed. As explained before, the high FTP and “other TCP/UDP” usage of ADSL users is related to additional offered services, that is, video on demand.

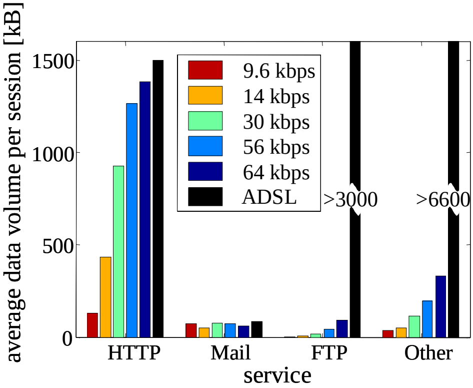

Figure 5.7: Average data volume per service (HTTP, Mail, FTP) and session.

As indicated in Figure 5.3, around 30% of the sessions last less than 100 seconds. Comparing the services used in these sessions with the average of all sessions the role of HTTP and Mail are found to be exchanged. Mail volume ranges from 80% to 50% in these short sessions. This results from the fact that users check mail or download mail and then read mail off-line. This could also imply the first peak in the session duration distribution.

About 10% of the TCP port numbers are used too rarely to show up in this statistic and could not be assigned to known services. Compared to investigations carried out at the same time in universities LANs [92], where traffic from Real Audio radio transmission were an issue, no evidence for Real Audio streaming was found in the dial-in measurement. The unidentified UDP port numbers take at most one percent of the traffic volume, thus, it could be assumed that the application Real Audio is yet not attractive to modem users that are charged on usage time.

5.3 Application Characteristics

Table 5.3 lists, depending of the access speed, the percentage and application volume of sessions in which one of the main applications HTTP, Mail or FTP is used. Compared to Figure 5.7 all results in this section are conditioned by the fact that an application is actually used. The usage of HTTP grows slightly but the volume of HTTP sessions increases significantly with the access speed. An identical increase of the usage percentage and volume can be seen for FTP. The Mail usage reduces slightly with increasing access speed, and the volume keeps a constant trend. Some exceptions are seen in the trace, that are probably related to the heavy-tailed nature of the distributions contained in the trace. For example, for 28.8 kbps modems some very large e-mails were captured, confirmed by the CoV of 18 and a 90% volume quantile of 79kB that is even less than the average volume.

	Application Usage					
Speed	HTTP		FTP		Mail	
	%	Vol.[kB]	%	Vol. [kB]	%	Vol. [kB]
9.6	29.5	413	0.8	0.6	74.6	95
14.4	54.3	727	3.9	143	73.3	58
28.8	68.5	1170	4.6	187	67.4	142
31.2	67.4	1170	4.2	482	68.3	79
33.6	65.9	1261	5.1	437	68.8	75
56	72.5	1521	8.4	470	68.5	90
64	72.4	1744	9.0	973	59.3	91

Table 5.3: Application usage and volume in dependence of access speed.

In the following, the Tables 5.4, 5.5 and 5.6 list the decomposition of the HTTP, Mail and FTP sessions into pages respectively TCP connections depending on the access speed. In some cases the numbers do not reflect the trends exactly – a observation that is also based on the heavy-tailed nature of the described distributions. For such distributions, the mean value is infinite, which implies that mean value estimated from finite samples varies heavily and may increase with the number of observations.

	SMTP				POP3			
Speed	Session Volume [kB]	# Conn.	Conn. Volume [kB]	IAT [s]	Session Volume [kB]	# Conn.	Conn. Volume [kB]	IAT [s]
9.6	15	1.0	15	292	103	1.7	60	185
14.4	52	1.4	38	234	41	1.8	23	207
28.8	136	1.4	97	231	104	2.4	43	207
32.2	72	1.4	51	242	59	2.6	23	194
36.6	72	1.5	49	269	55	2.5	22	190
56	113	1.4	80	242	59	5.2	11	89
64	91	1.5	59	489	68	3.4	20	206

Table 5.4: Speed dependent SNMP and POP3 decomposition.

For SMTP and POP3 used for reading e-mail, the variations found are not related to the access speed. The number of SMTP connections in an average session is constant and differences in the session volume are related to the average volume of single SMTP connections. For the POP3 protocol a similar behavior is observed, with the exception of the 56kbps modem class. In this category we observe a significantly higher number of connections in an average session with average volume. The connections are smaller and show a shorter inter-arrival time, which in summary leads to a session volume comparable to that of other modem speeds.

The increase of the HTTP session size is caused likewise by an increase of the average number of pages per session, number of connections per page and average volume of the pages and TCP connections. The observation holds for both web page definitions, c.f., Section 4.2.1. The inter-arrival time for pages discerned by locality definition reduces with increasing access speed. For pages discerned by 1s gap, the inter-arrival time does not depend on the access speed, instead the

		HTTP: pages discerned by 1s gap					
Speed	Session Vol. [kB]	# Pages	Page Vol. [kB]	# Conn./ Page	IAT [s]	UTT [s]	UTT (neg.) [s]
9.6	413	6.2	59	7.2	123	39	-
14.4	727	7.4	98	11.6	126	38	-
28.8	1170	8.0	147	14.8	120	37	-
32.2	1170	9.3	125	12.9	110	37	-
36.6	1261	9.7	130	13.6	113	37	-
56	1521	9.1	166	15.6	116	38	-
64	1744	8.1	214	16.7	127	48	-
		HTTP: pages discerned by locality and 1s gap					
Speed	Session Vol. [kB]	# Pages	Page Vol. [kB]	# Conn./ Page	IAT [s]	UTT [s]	UTT neg. [s]
9.6	413	15.0	24	3.0	50	36	57
14.4	727	23.3	31	3.7	42	35	68
28.8	1170	31.1	38	3.8	34	32	55
32.2	1170	35.0	33	3.4	32	32	53
36.6	1261	35.9	35	3.7	33	32	48
56	1521	37.8	40	3.8	30	31	47
64	1744	37.0	47	3.7	30	36	46

Table 5.5: Speed dependent HTTP decomposition.

average number of connections per page increases more than that of pages discerned by locality.

FTP data sessions show comparable behavior as the HTTP sessions. The average volume of the FTP sessions increases with the access speed. The reasons are found in an increase of both, the average volume of single connections as well as the average number of connections within a session.

	FTP data				FTP control			
Speed	Session Volume [kB]	# Conn.	Conn. Volume [kB]	IAT [s]	Session Volume [kB]	# Conn.	Conn. Volume [kB]	IAT [s]
9.6	-	-	-	-	-	-	-	-
14.4	425	7.0	61	37	4.1	2.8	1.5	279
28.8	678	18.5	37	21	5.3	4.2	1.3	185
32.2	1762	8.9	198	77	4.9	5.6	0.9	230
36.6	1104	9.2	120	68	5.6	5.3	1.1	206
56	1467	9.1	160	68	17.6	6.4	2.7	178
64	2462	11.6	212	65	7.9	7.8	1.0	130

Table 5.6: Speed dependent FTP decomposition.

The decomposition of the most frequent used applications gives no single explanation for the increasing session volume. The growth in not only related to the fact that more objects, i.e. web pages, files, are downloaded, but also the size and complexity of the objects grow.

5.4 HTTP connection characteristics

Since 80% of the traced traffic volume of dial-in modems are HTTP connections and the same trend was found for FTP connections, we concentrate in the following on characteristics of HTTP connections.

As shown in Section 3.2 the user activity changes in dependence of the daytime and the tariff structure of the German Telekom AG. This user activity is also reflected in the achievable average data rate during the day. Figure 5.8 and Figure 5.9 depict the TCP connection data rate in up- and downstream direction, respectively. The largest data rate for dial-in modems is obtained between midnight and 8 a.m, while the data rate of ADSL connections is reduced only during the local business hours. For dial-in modems, the obtained data rate does not increase during the maximum activity period, even when the backbone load of the bearer DFN network [93] decreases in the evening. Nevertheless, the modem

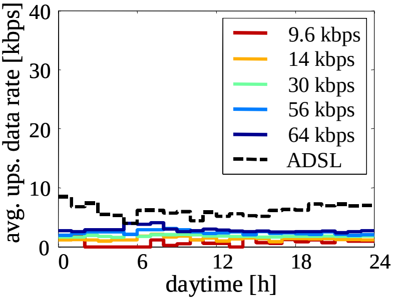

Figure 5.8: Upstream TCP-connection data rate in dependence of daytime.

speed is reflected in the obtained data rate, but the gain for ADSL modems is less than expected by the 2.5 Mbps downstream rate. Further discrimination of the connection destinations in conjunction with a data rate analysis could give valuable hints for network bottleneck detection.

The TCP-connection volume distribution depicted in Figure 5.10 and Figure 5.11 for up- and downstream direction, exhibits a similar shape for all access speeds. Also the shape of the volume distribution in up- and downstream di-

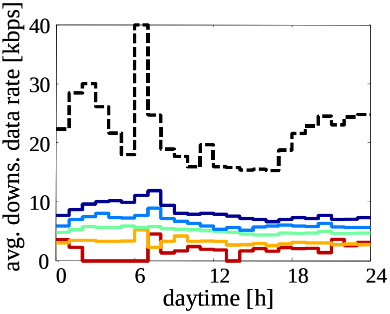

Figure 5.9: Downstream TCP-connection data rate in dependence of daytime.

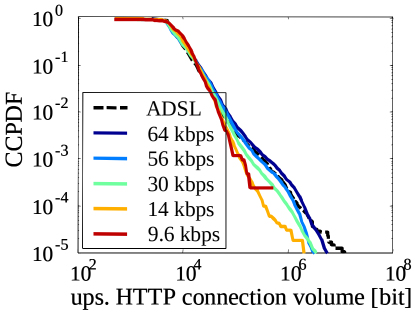

Figure 5.10: Upstream TCP-connection volume distribution.

rection is similar, showing a heavy-tailed decay over four orders of magnitude. The downstream volume is about one order of magnitude larger than the upstream volume. The volume increases only slightly with the access speed.

The shape of the data rate distributions is independent of the access speed but differs for up- and downstream direction, as shown in Figure 5.12 and Figure 5.13. In general the downstream rate is one order of magnitude larger than

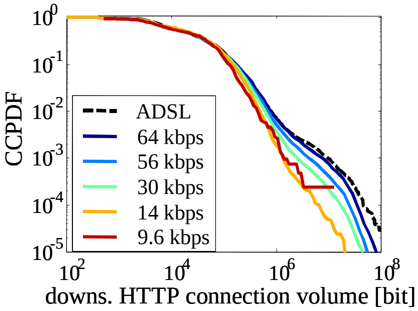

Figure 5.11: Downstream TCP-connection volume distribution.

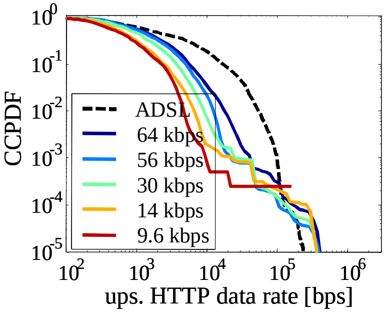

Figure 5.12: Upstream TCP data rate distribution.

the upstream rate, which follows directly from the TCP-connection volume distribution. Some connections reach upstream data rates that are 10 times higher than the modem speed. This could happen if a connection setup is requested with a SYN-packet and refused with a Reset-packet. In this case, the modem speed does not affect the measurement.

The throughput of TCP-connections depends on a closed-loop control cycle. Therefore, the up- and downstream data rate is correlated. Figure 5.14 shows the

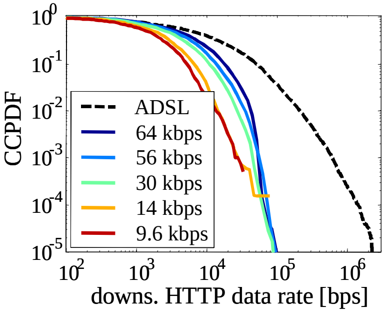

Figure 5.13: Downstream TCP data rate distribution

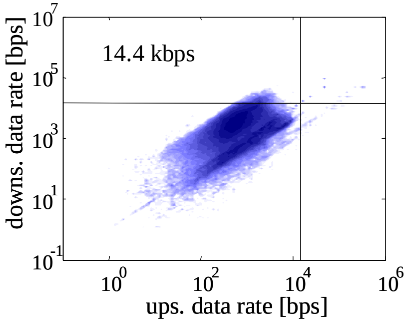

Figure 5.14: TCP data rate correlation for 14.4kbps.

correlation of up- and downstream data rates for 14 kbps. The corresponding chart for ADSL access speed is depicted in Figure 5.15. The figures for the remaining modem speeds are omitted since no additional information is contained. The range of up- and downstream ratio is similar for all modem speeds, but of course the faster modems capture a wider range of data rates.

If up- and downstream data rate were highly correlated, the plot in Figure 5.15 would be a diagonal line.

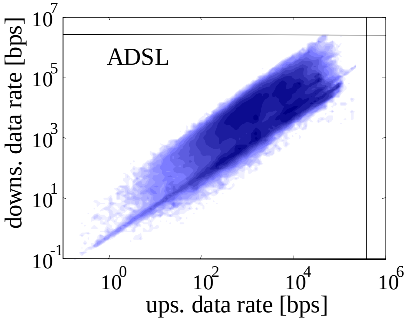

Figure 5.15: TCP data rate correlation for ADSL.

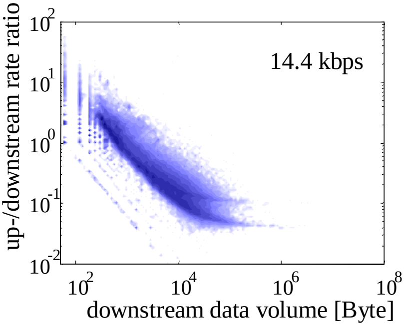

Figure 5.16: HTTP connection up-/downstream rate to connection volume correlation for 14.4kbps modem class.

The up- and downstream rate ratio varies mainly over 2 orders of magnitude, as shown in Figure 5.16, where the rate ratio is rendered over the downstream data volume for 14.4 kbps modems. Again, the figures look similar for the investigated modem speeds, but for faster modems larger connections are measured (c.f. Figure 5.17). For small downstream data volume the upstream rate is higher than

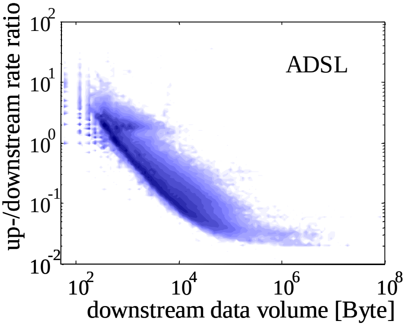

Figure 5.17: HTTP connection up-/downstream rate to connection volume correlation for ADSL access.

the downstream rate. For large connections a limit is reached at 1/50, since two data packets of 1500 Bytes could be acknowledged with one packet of 60 Bytes. The graphs are determined by the basic TCP acknowledgement mechanism, described in Section 3.2.3. This mechanism implies, that for fast transmission of HTTP data, where about 50% of the downstream connection volumes are less than 3000 Bytes, a symmetric connection is preferable.

6 Parametrization

In the previous chapters we described the components of Internet traffic on different levels of the protocol hierarchy. To utilize these results a description of the traffic in form of distributions that can be used for random number generation is required. We will review which distributions are suitable to reflect the properties of Internet traffic. We aim to identify these distributions and to give parameters for the relevant components of the Internet traffic model.

6.1 Heavy-Tailed and Subexponential Distributions

Heavy-tailed and subexponential distributions are often used for modelling data sets with high variance, where extremely large values compared to the sample mean occur. This kind of data is seen for computer network data, e.g. in file size distributions of web pages and file servers [2, 21], and CPU time usage [38]. Heavy-tails are also reported to be present in data sets related to financial and insurance data [50, 36]. The auto-correlation of computer network and video traffic shows a heavy-tailed behavior [35, 45], a phenomenon often referred to as self-similarity.

Many of the distributions mentioned in this work have the property that the tails of these distributions decrease more slowly than an exponential tail. In a plot on double logarithmic axes, the tail of the complementary cumulative probability density function (CCPDF) appears to be a straight line which is an indication that these distributions are heavy-tailed or subexponential, respectively.

In the following we will show the mathematical background of heavy-tailed distributions and discuss the difference of heavy-tailed and subexponential distributions.

Let X be a random variable with CPDF $F(x) = P[X \leq x]$ and a complementary CPDF (CCPDF) $\bar{F}(x) = P[X > x] = 1 - P[X \leq x]$. The distribution $F(x)$ is heavy-tailed if

$$\bar{F}(x) \sim cx^{-\alpha}, \qquad 0 < \alpha < 2, 0 < c . \tag{6.1}$$

The notation $a(x) \sim b(x)$ stands for $\lim_{x \to \infty} a(x)/b(x) = 1$. Heavy-tailed distributions have infinite variance and for $\alpha \leq 1$ the distribution has an infinite mean.

When rewriting Equation (6.1) as

$$\lim_{x \to \infty} \frac{d\log\bar{F}(x)}{d\log x} = -\alpha \tag{6.2}$$

the mathematical foundation for the characterization of heavy-tailed distributions as straight line in double logarithmic representation becomes obvious. For finite samples the linear decay can be seen only over some orders of magnitude. Further the empirical sample mean and variance are finite, but may grow with increasing sample size. Subexponential distributions have also a tail that decays slower than any exponential tail, but opposed to heavy-tailed distributions these distributions have finite mean and variance. A examples for heavy-tailed distributions is the Pareto distribution, and for subexponential distributions the lognormal or Weibull distribution.

6.2 Estimation of the Heavy-Tailed Index

The identification of the presence of a heavy-tail in a measured sample is of importance for the modeling and evaluation of the measurement. On one hand, the heavy-tailed property limits the set of distributions suitable for modeling. On the other hand, heavy-tails imply that the estimation of the samples moments, that can be infinite, may not be consistent when increasing the sample size. Especially if the sample is drawn from a distribution with infinite mean, e.g. Pareto with $a < 1$, the empirical mean and variance of the sample are finite but increase with the size of the sample.

In the previous chapters we applied the simplest estimator for heavy-tails in distributions, that is, the visual inspection of the plot of the CCPDF in a double logarithmic graph. The distribution is said to be heavy-tailed if the tail decreases in a straight line. According to Equation (6.2) the gradient of the linear part of the tail is denoted as heavy-tailed index α.

In the following we will describe two more mathematical approaches to determine the heavy-tailed property of a distribution, the Hill and the scaling estimator.

6.2.1 Hill Estimator

The Hill estimator [40] evaluates the heavy-tailed index in dependency of the k largest elements in the sample. These are described by the sample's order statistics $x_{(i)}$, where the sample items are sorted by size, that is, $x_{(1)} \leq \ldots \leq x_{(n)}$. The Hill estimator is defined as follows:

$$H_{k,n} = \left(\frac{1}{k} \sum_{i=0}^{k-1} (\log(x_{(n-i)}) - \log(x_{n-k})) \right)^{-1} . \tag{6.3}$$

The Hill estimator is applied plotting the estimator against increasing values of k. The estimate of α is given when the estimator converges to a consistent value.

For both, the visual estimation of the heavy-tail and the Hill estimator, the quality of the result depends on the proper choice of the beginning of the tail. For measured data sets the distinction between the body and the tail of the distribution is not obvious. The problem is demonstrated in Figure 6.1 for the empirical distri-

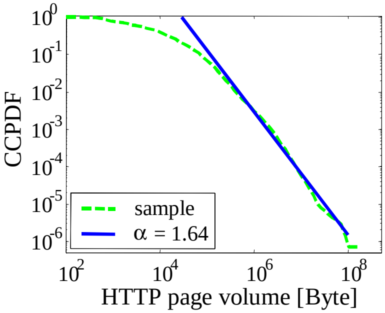

Figure 6.1: Tail behavior for HTTP page volume.

bution of web pages. The blue line represents the linear least-square fit for pages larger than 100 kByte. The limit of 100 kByte was determined by optical estimation of the start of the tail. A selection of a start value smaller than 100 kByte, e.g. 50 kByte – which is also correct according the optical estimate – would result in a less steep gradient of the line.

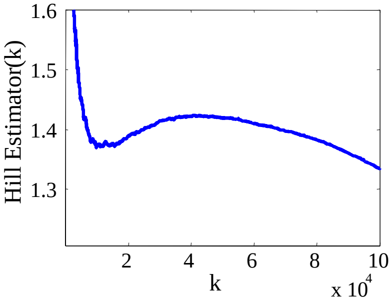

Figure 6.2: Hill estimator for the HTTP page volume distribution.

The same difficulties occur, when applying the Hill estimator. Figure 6.2 illustrates the Hill estimator applied to the HTTP page volume data set for a wider range of the parameter k as usual. Normally when increasing k until the estimator converges, the estimator would stop at a k between 10000 and 20000. These values relate to a tail beginning with elements larger than 350 kByte. When the estimation is continued, the estimator increases for values of k ranging from 20000 to 40000, which represents elements larger then 200 KByte in the tail of the distribution. A further increase of k leads to a decay of the estimator.

The interpretation of Figure 6.2 gives an estimate of α between 1.37 and 1.42, where the estimator stays constant for a reasonable number of samples. It allows not to determine where the tail of the distribution unambiguously begins.

6.2.2 Scaling Estimator

A method to determine the heavy-tailed index of distributions without specifying the beginning of the tail is introduced with the scaling estimator [23]. This estimator determines automatically the region of the distributions tail where a power-law behavior is found. The estimator gives no proof of heavy-tailed behavior, but facilitates the decision based on a graphical inspection of the distributions region where heavy-tailed behavior is detected.

The Scaling Property

The scaling estimator is based on a formulation of the central limit theorem for distributions with infinite variance. In the classical central limit theorem, the distribution of the sum of independent random variables with finite variance converges to a normal distribution. For heavy-tailed random variables with index α the sum converges asymptotically to a stable distribution with identical index α. The effect, which can be observed also in empirical data sets, is the basis for the scaling estimator.

To apply the estimator the following operations and measurements are applied to the data set. The aggregation $X^{(m)}$ of the data set X is defined by the sum on non-overlapping blocks of the data sets elements x_i:

$$x_i^{(m)} = \sum_{j=(i-1)m+1}^{im} x_j \,. \tag{6.4}$$

For two different aggregations $X^{(m_1)}, X^{(m_2)}$ with $m_1 < m_2$ of the same data set, the tail behavior is illustrated in Figure 6.3. For any point $(\log x_1, \log(P[X^{(m_1)} > x_1]))$ in the tail of $X^{(m_1)}$ the horizontal distance δ and the vertical distance τ to the tail of $X^{(m_2)}$ is measured.

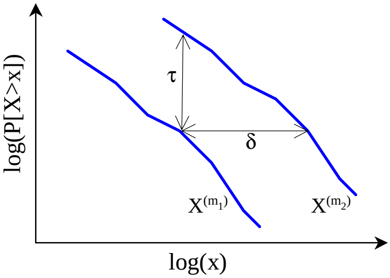

Figure 6.3: Measurements used for the scaling estimator.

The quantities δ and τ have the following properties when measured on a data set distributed according a strictly α-stable, $0 < \alpha \leq 2$, distribution. The properties hold also for the tails of arbitrary heavy-tailed distributed data sets, since these converge asymptotically to α-stable distributions when being aggregated.

The horizontal distance δ computes, using the scaling property, to:

$$\delta = \frac{1}{\alpha} \log \frac{m_2}{m_1} \,. \tag{6.5}$$

For perfectly scaling distributions, such as the α-stable or normal distribution, Equation (6.5) holds for arbitrary points in the distribution, i.e., any point of the distribution could be used for measuring the distance δ and to obtain an estimate of α. When applying the estimator to empirical distributions with asymptotic power tails, the range of points where the estimator is evaluated has to be restricted. For this means, the vertical distance τ is evaluated. For elements in the tail of heavy-tailed distributions, c.f., Equation (6.1), the following relation, based on the scaling property, is derived [23]:

$$\tau = \log\frac{m_2}{m_1}. \tag{6.6}$$

The scaling property holds also for the normal distribution ($\alpha = 2$), to which the aggregation of finite variance variables tend to converge. Since the estimator is applied to the tail of the distribution, there might be few data points to allow for a large aggregation. Thus, the estimate might be not reliable, but this case can be identified by a visual inspection of the results.

Application of the Scaling Estimator

Before the scaling estimator is applied to a measured sample the sample mean is subtracted from every element of the sample. The aggregation step size, that is, the quotient of two neighboring aggregations is set to $(m_{i+1}/m_i) = 2$. The number of aggregations is set to 10, yielding the aggregated data sets $X^{(2)}, X^{(4)}, \ldots, X^{(512)}$. For each data set the CCPDF in double logarithmic representation is derived.

In the next step, the values of δ and τ in relation to the next higher aggregation level are measured for each point in the tail of the CCPDF plots. With the term “tail” we refer to the upper 90th percentile of the sample. The scaling parameter $\hat{\alpha}$ is estimated for every point in the tail by the following relation, derived from Equation (6.5):

$$\hat{\alpha} = \log\frac{m_{i+1}}{m_i}/\delta. \tag{6.7}$$

An estimate $\hat{\alpha}$ is accepted if the corresponding vertical distance τ is within a relative error θ of less than 10% from the theoretical vertical distance, as defined in Equation (6.6). The estimate $\hat{\alpha}$ is accepted if

$$\left|\tau - \log\frac{m_{i+1}}{m_i}\right| < \theta \cdot \log\frac{m_{i+1}}{m_i}\,. \tag{6.8}$$

The final estimate of the scaling parameter α is given by the average of the accepted estimates $\hat{\alpha}$ from all aggregations of the data set. The algorithm will lead to an estimate even if only few data points lead to accepted estimates. Thus, a visual inspection is required to decide whether the scaling is seen over some orders of magnitude, which indicates a heavy-tailed behavior. If the sample scales only locally, the sample is classified as not heavy-tailed.

6.2.3 Results of the Heavy-Tailed Index Estimation

In the following we demonstrate the evaluation of the scaling estimator and compare the results to the results of the Hill estimation. The evaluation is done exemplary on three samples, that show different behaviors. The results for the modeling relevant samples are summarized in Table 6.1.

Example: HTTP Inter-Arrival Time – No Evidence for Heavy-Tail

The scaling estimator of the HTTP inter-arrival time is depicted in Figure 6.4. The aggregations are shown in the left graph in increasing order from left to right. Ac-

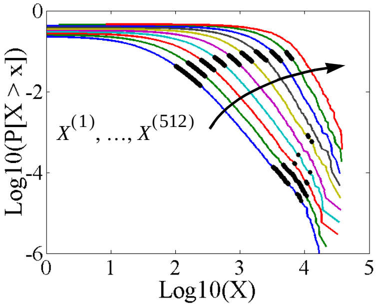

Figure 6.4: Scaling estimator for the HTTP inter-arrival time.

cepted estimates are indicated by bold points. The estimator shows valid estimates only at the beginning and at the last portion of the tail. Thus, the scaling property is not found in a significant part of the tail. The Hill estimator, shown in the right graph of Figure 6.5, does not converge, either. As conclusion, the HTTP inter-arrival time distribution is not heavy-tailed.

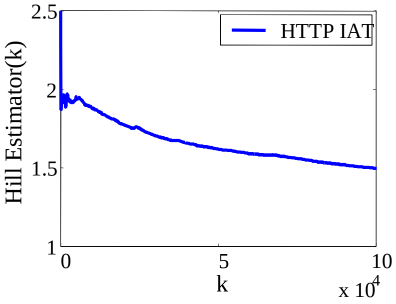

Figure 6.5: Hill estimator for the HTTP inter-arrival time.

Example: HTTP Page Volume – Strong Evidence for Heavy-Tail

Figure 6.6 and Figure 6.7 show the scaling and Hill estimator for the measured HTTP page volume, respectively. The scaling estimator is bases on accepted trial estimates found in the tail of the empirical distribution over three orders of magnitude. The scaling property is present in a large area of the aggregated data sets. The evidence of a heavy-tail by the scaling estimator is confirmed by the convergence of the Hill estimator.

The scaling estimator derives an α value of 1.25, while the Hill estimation converges at 1.4. The difference stems from the fact, that the scaling estimator takes the end of the tail, where only few samples are measured, not into account.

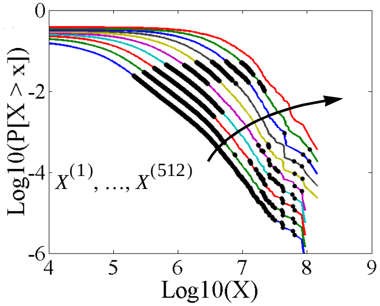

Figure 6.6: Scaling estimator for the HTTP page volume.

Example: Session Volume – Weak Evidence for Heavy-Tail

In difference to the previous mentioned examples, the tail estimation for the session volume gives evidence for and against being heavy-tailed. As depicted in Figure 6.8 the Hill-estimator does not converge. While the Hill-estimator does not indicate a heavy-tailed behavior, the scaling-estimator in Figure 6.9 shows accepted estimates over two orders of magnitude for the first aggregations of the

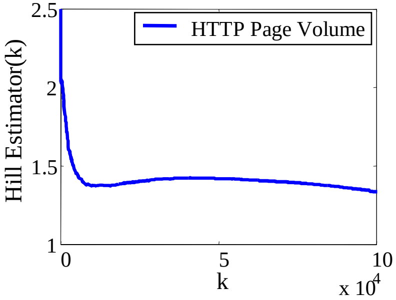

Figure 6.7: Hill estimator for the HTTP page volume.

data set. The size of the sample does not allow for reliable results for the higher aggregations of the data set.

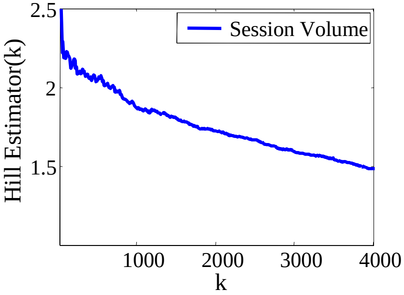

Figure 6.8: Hill estimator for the session volume.

Interpreted from the facts in Figure 6.9 the session volume shows no heavy-tailed behavior. But the scaling behavior at the first aggregation levels indicates that a larger sample may continue the scaling also for higher levels of aggregation.

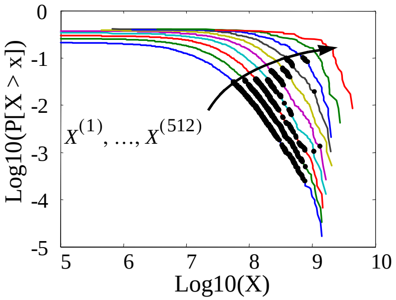

Figure 6.9: Scaling estimator for the session volume.

Summarized Tail Estimation Results

The following Table 6.1 summarizes the results of the tail estimation. We have evaluated both, the scaling and Hill estimator. A result based on the scaling method is given for every sample, the validity of the estimation, based on the visual inspection, is shown in the last column. The graphs for the Hill estimation and visual inspection of the scaling estimator are given in the Appendix B.

sample type		scaling estimate	Hill estimate	heavy-tailed?
session	inter-arrival time	0.99	1.55	no
	volume	1.71	-	no
	duration	2.02	-	no
HTTP by locality and 1s gap	inter-arrival time	1.22	-	no
	conn. per page	1.66	-	no
	page volume	1.25	1.4	yes
	initial object size	1.15	1.5	yes
	follow-on object size	1.19	1.7	no
	user-think time	1.32	1.7	yes
HTTP, by 1s gap	inter-arrival time	1.35	-	no
	conn. per page	1.57	-	no
	page volume	1.24	-	yes
	initial object size	1.15	1.5	yes
	follow-on object size	1.23	1.7	no
	user-think time	1.31	1.6	yes
FTP control	inter-arrival time	1.26	-	no
	volume	1.39	-	no
FTP data	inter-arrival time	1.24	-	no
	volume	1.31	-	no
Mail SMTP	inter-arrival time	1.40	-	no
	volume	1.54	-	no
Mail POP3	inter-arrival time	1.26	-	no
	volume	1.21	0.7	yes

Table 6.1: Tail estimation summary.

A general observation of the tail estimation is that the scaling- and Hill estimator result in different numerical estimates. In most cases the Hill-estimator gives a larger estimate of α since it takes all values of the tail of the distributions into account, whereas the scaling-estimator does not take into account under-represented values at the end of the tail.

6.3 Distribution Fitting

This sections aims at giving an applicable set of distributions and parameters that can be used to model Internet traffic. The presented distributions are selected to keep the model simple and applicable for analysis and simulation purposes. We estimate the parameters for the sampled model components with regard to the most often cited distributions for Internet traffic, i.e., for the Pareto, lognormal, Weibull and gamma distributions. An overview of WWW traffic models utilizing these distributions is given in [87].

A more accurate modeling of Internet traffic is possible at the cost of higher complexity distributional models. Recently, several combined models have been presented, such as the combination of a lognormal and Pareto distribution for modeling the body and tail of a sample [3], respectively. The double Pareto lognormal distribution and multiple application of this model are shown to improve the matching accuracy of file size distributions [76, 39]. Non parametric approaches such as kernel estimates have been presented to work with high accuracy on small data sets [44]. These more accurate models may lead to a reduced applicability if several components have to correspond in an Internet traffic model.

Before discussing the parameter estimates, we will present the distributions and the way the parameters were estimated.

6.3.1 Distributions for Modeling Internet Traffic

Pareto Distribution

The Pareto distribution is heavy-tailed, respectively follows a power law over its entire range. The PDF of the Pareto distribution is given by:

$$f(x) = ak^{\alpha}x^{-\alpha-1}, \qquad 0 < k \le x\,. \tag{6.9}$$

The CPDF is defined by

$$F(x) = 1 - (k/x)^{a} \tag{6.10}$$

where the constant k represents the smallest possible value of the random variable and α denotes the slope of the tail in double logarithmic representation.

The parameters α and k are determined by a least square estimation of a line to the empirical double logarithmic CPDF of the data set. This estimation utilizes the fact that the Pareto distribution is heavy-tailed for the entire range of the distribution.

Lognormal Distribution

The PDF of the Lognormal distribution is given by:

$$f(x) = \begin{cases} \dfrac{1}{x\sqrt{2\pi b}}\exp\left(\dfrac{-(\log(x) - a)^2}{2b}\right) & \text{for } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{6.11}$$

The mean of the Lognormal distribution computes as $\exp(a + b/2)$ and the variance is defined as $\exp((2a + b) \cdot (\exp(b) - 1))$. A closed form for the Lognormal distribution and its CPDF is not available.

For a sample the parameters a and b of the Lognormal distribution are estimated as follows: $\hat{a} = (\sum_{i=1}^{N}\log(x_i))/N$ and $\hat{b} = \left(\sum_{i=1}^{N}(\log x_i - \hat{a})^2\right)/N$.

Weibull Distribution

The Weibull PDF is determined by the parameters a and b. For $b = 1$ the Weibull distribution is identical to the exponential distribution:

$$f(x) = abx^{b-1}\exp(-ax^b) \qquad x > 0 . \tag{6.12}$$

The CPDF is defined by

$$F(x) = 1 - \exp(-ax^b) . \tag{6.13}$$

The parameters are estimated with the software package MATLAB [94], that provides a maximum likelihood estimator for Weibull distributions.

Gamma Distribution

The gamma distribution is also used in publications for the modeling of Internet traffic [18, 87]. The PDF of the gamma distribution is defined as follows:

$$f(x) = \frac{1}{b^a\Gamma(a)}x^{a-1}\exp\left(-\frac{x}{b}\right) , \tag{6.14}$$

where $\Gamma(a) = \int_0^\infty x^{a-1}e^{-x}dx$ for $a > 0$. If a is an integer, the distribution is also named Erlang distribution. The CPDF is expressed by the incomplete gamma function and has to be evaluated numerically.

Similar to the Weibull distribution, the parameters of the gamma distribution were estimated with the software package MATLAB. Some of the data sets did not match to the gamma distribution.

6.3.2 Evaluation of the Parametrization: QQ Plot

A common method to compare distributions is the QQ plot. If $F(x)$ denotes the distribution function of a continuos random variable, the q-quantile, $0 < q < 1$, of $F(x)$ is given by the value x_q that resolves $F(x_q) = q$. A QQ plot compares two distributions by plotting the quantiles x_{q_i} for $0 < q_1 < q_2 < \ldots < q_n < 1$ of the two distributions against each other. When the distributions are almost identical or, in the case of empirical distributions, similar, the plot will be approximately linear with a slope of 1.

To accommodate for the characteristics of the data samples in this work, we have adopted the QQ plot accordingly. Since some of the data sets are too large to plot every possible quantile, we decided to use a set q_i where the resolution increases at the tail of the distribution. Since the range of the values in most of the samples ranges over several orders of magnitude, the QQ plots are depicted in double logarithmic representation.

The modifications keep the basic properties of QQ plots, but allow to inspect the matching of the distributions for small quantiles, as well as for quantiles in the tail region.

6.3.3 Parametrization Results

In this section, we present some examples for the evaluation of the parametrization and give an tabular overview of the suggested parameters for the measured data sets.

Fitting of not Heavy-Tailed Data Set: Session Duration

The session duration was classified as data set which is not heavy-tailed in the strict sense. As depicted in Figure 6.10, the QQ plot of the fitted distributions, i.e. Pareto, lognormal, Weibull and gamma confirms this classification. The Weibull distribution fits best to the grey diagonal line, which represents the exact matching to the empirical data set. The lognormal distribution fits the body of the distribution well, but does not fit at the tail of the distribution. The Pareto distribution does not fit at all. The gamma distribution, is similar to the Weibull distribution, but does not fit as good as the Weibull distribution. The latter observation was seen not only for this data set, but seems to hold for all observed data sets for which a valid gamma parametrization was found.

Fitting of Heavy-Tailed Data Set: HTTP Initial Object Size

Figure 6.11 shows the QQ plot for the parametrization of the HTTP initial object size for web pages that are discriminated by 1 second gap. While the Weibull distribution does not fit to the samples distribution, the Pareto and Lognormal distri-

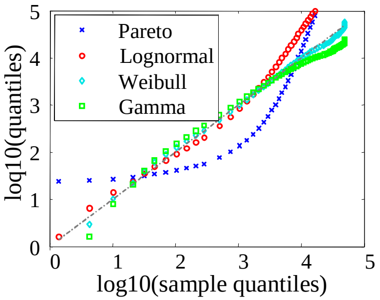

Figure 6.10: Parametrization of session duration.

butions fit to the data set. The Pareto distribution fits the tail of the empirical data and the discrepancy at the body of the distribution is not too big. The lognormal distribution fits well to the body of the data set, but does not represent the tail behavior. Thus the Pareto distribution would be first choice for modeling the heavy-tailed HTTP initial object size.

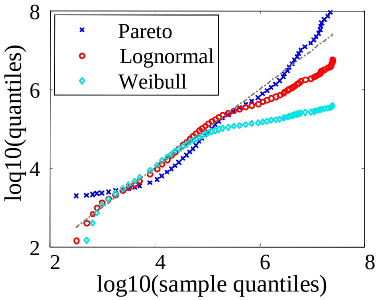

Figure 6.11: Parametrization of the HTTP initial object size.

Are Heavy-Tailed Data Samples Fitted Optimal with Heavy-Tailed Distributions?

In the previous two examples the data sets were fitted best by distributions that belong to the type the data set was classified, i.e., subexponential respectively heavy-tailed. Measured data sets are always limited in the number of samples. Thus, we can test for properties like infinite variance or scaling behavior, but the empirical mean and variance of the data set is finite and a small uncertainty of the test results remains.

The HTTP page volume data set is classified by the scaling-estimator as heavy-tailed. The QQ plot for the fitted Pareto, lognormal and Weibull distribution in Figure 6.12, shows that the best fit is obtained with the lognormal distribution. Nevertheless, the Pareto distribution leads to an acceptable model that emphasizes the heavy-tail. The effect is also observed with the HTTP user-think time data sets.

Figure 6.13 shows the reverse effect. The heavy-tailed property was not found in the HTTP connections per page data set, but the heavy-tailed Pareto distribution models the sample better than the other distributions.

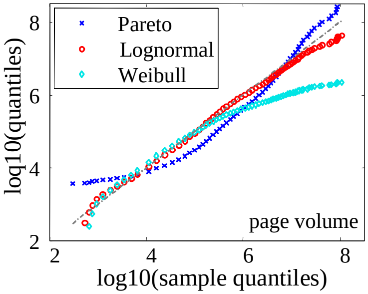

Figure 6.12: Parametrization of the HTTP page volume data set.

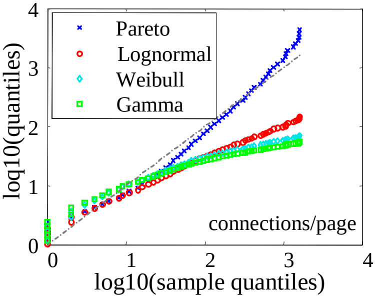

Figure 6.13: Parametrization of the HTTP connections per page data set.

Summarized Parametrization Results

The chapter is concluded with a summary of the parametrization results. The Table 6.2 gives the type and parameters for the distribution that fits best to the sample. If the heavy-tailed property of the selected distribution and sample does not match, we give an alternative parameter set. Since, the lognormal distribution fitted the heavy-tailed data sets – especially with regard to the body of the distribution – quite well, we will give this alternative also in the cases where the best fit was obtained with the Pareto distribution. The use of a distribution with finite variance shows better convergence in simulations and can be reasonable if the study aims a short-range effects. The results base also on QQ plots.

sample type		best fit by QQ plot	alternative
session	inter-arrival time	Lognormal a = 2.29, b = 1.16	-
	volume	Weibull a = 0.0011, b = 0.45	-
	duration	Weibull a = 0.028, b = 0.57	-
HTTP by locality and 1s gap	inter-arrival time	Weibull a = 0.17, b = 0.60	-
	conn. per page	Pareto a = 1.76, k = 1.68	Lognormal a = 0.65, b = 0.92
	page volume	Lognormal a = 8.78, b = 1.86	Pareto a = 1.19, k = 3587
	initial object size	Pareto a = 1.26, k = 1780	Lognormal a = 8.04, b = 1.51
	follow-on object size	Lognormal a = 8.08, b = 1.48	-
	user-think time	Lognormal a = 2.09, b = 1.91	Pareto a = 0.99, b = 1.07
HTTP, by 1s gap	inter-arrival time	Lognormal a = 3.81, b = 1.31	-
	conn. per page	Lognormal a = 1.44, b = 1.37	-
	page volume	Lognormal a = 10.0, b = 1.94	Pareto a = 1.01, b = 6755
	initial object size	Pareto a = 1.16, k = 1940	Lognormal a = 8.45, b = 1.50
	follow-on object size	Lognormal a = 8.04, b = 1.48	-
	user-think time	Lognormal a = 2.60, b = 1.33	Pareto a = 1.29, k = 6.30

Table 6.2: Parametrization results.

sample type		best fit by QQ plot	alternative
FTP control	inter-arrival time	Lognormal a = 2.91, b = 2.32	-
	volume	Lognormal a = 6.32, b = 1.28	-
FTP data	inter-arrival time	Lognormal a = 2.18, b = 1.75	-
	volume	Lognormal a = 7.97, b = 2.53	-
Mail SMTP	inter-arrival time	Lognormal a = 3.91, b = 2.12	-
	volume	Pareto a = 0.83, k = 1280	Lognormal a = 8.16, b = 1.77
Mail POP3	inter-arrival time	Lognormal a = 3.22, b = 2.22	-
	volume	Pareto a = 0.81, k = 354	Lognormal a = 6.38, b = 1.77

Table 6.2: Parametrization results.

7 Summary

This monograph was concerned with the characterization and modeling of Internet traffic from a users perspective.

For this purpose, a flexible multi-layer model was presented, which is capable to describe Internet user traffic on session, application and connection layer. The session layer represents the user activity period in the system. During this period the user works with one or several different applications, described in the application layer. These applications in turn transmit data over possibly parallel TCP connections. The communication structure of the applications is derived in a review of the Internet protocol stack. These communication structures range from the serial exchange of data in a single connection, e.g. when mail is retrieved, up to several parallel TCP connections to different servers, when a single web page is downloaded.

The developed model is based on the evaluation of an Internet traffic measurement at a dial-in access of the University of Würzburg. The Internet access was charged by a time-based tariff, which was found to impact the user activity with regard to the frequency of sessions and the session duration. The main traffic sources were the applications HTTP with 80% of the total traffic volume, followed by mail, and FTP with 6% and 3%, respectively. This traffic mix was also reported in other backbone measurements at this the time.

A thorough analysis of the application utilization showed that in most sessions, except protocol supporting services such as DNS, a mix of different applications was used. The volume of the applications was found to be larger when used in combination with other applications. Breaking down the application traffic to the TCP connection level unveiled the characteristics of the application protocols. The description communication structure is given in terms of connection volume and inter-arrival time. For the characterization of the HTTP protocol the connections were grouped to pages, that represent the data traffic transferred upon a single user interaction.

The distributions describing the model elements were found to have high variance and a tail that decreases more slowly than exponential tails. The variance of distributions, of such samples are drawn from, can be finite variance or infinite. The distributions are called subexponential or heavy-tailed, respectively. To test for the heavy-tailed property the Hill and scaling estimator were applied to the data samples. The mail volume, HTTP page size, initial object size and user-think time were estimated heavy-tailed. The use of such distributions in simulations requires special attention with regard to the stability of the simulation [22].

The samples were fitted to distributions and a parametric description was obtained. The heavy-tailed samples were fitted to heavy-tailed as well as finite variance distributions, where the matching accuracy was even better in some cases. Since finite samples always contain a remaining uncertainty when estimating infinite properties, the user has the choice to utilize finite variance distributions for simulations when not aiming at long-range dependent properties of the traffic.

As continuation of the parametrization process an application of the theory of Generalized Pareto Distributions [26, 77] is a promising approach for further research.

Together with the packet trace a access log file was evaluated to obtain the dial-in speed of the sessions. Further, the measurement results were compared to a measurement of an ADSL access network. For larger access speed, the session duration and volume increased. Especially for HTTP data, i.e, web page downloads, the volume increased proportionally with the modem speed. A saturation was reached for the ADSL access, since web page download is limited by user interaction. Faster access speed was also seen to enable new services. Online games were only detected for higher access speeds and for the ADSL users a video server was implemented.

Invariant to the access speed was the volume of e-mail transferred in an average session. The distribution of the volume of TCP-connections was invariant with regard to the body of the distribution, but with higher access speed large connections were seen, that contributed to the tail of the distribution.

The presented characterization and model of Internet user traffic was derived from a measurement at a modem dial-in access network with a time-based tariff system. On one hand, the trend for fixed-network Internet access is heading towards ADSL with flat-rate tariff. This configuration allows for new services, i.e. file-sharing, which takes at the beginning of 2003 up to 40% in backbone networks. On the other hand, the majority of the Internet users still accesses the Internet over dial-in lines. For this reason, web content is still developed to be accessible on slow lines. Emerging technologies such as GRPS and UMTS propel the convergence of wireless and data networks. These new networks provide similar data rates and the access is limited by tariff as seen in the investigated scenario. Thus, the presented model is applicable for investigations in the user access line and the concentrating access network.

Appendix

A Session Characteristics

This section lists the main characteristics of the measured session in hourly resolution. The characteristics show a strong dependence on the daytime, which is especially useful for the activity modeling.

A.1 Session Duration

starting time	session duration					
	all		weekday		weekend	
	[s]	CoV	[s]	CoV	[s]	CoV
0:00	1622	1.6	1533	1.6	1925	1.6
1:00	1637	1.6	1600	1.6	1738	1.6
2:00	1745	1.7	1452	1.8	2273	1.6
3:00	1493	1.5	1465	1.5	1549	1.5
4:00	1601	1.3	1620	1.4	1545	1.1
5:00	1277	1.6	1115	1.6	1711	1.6
6:00	702	1.7	638	1.9	945	1.2
7:00	651	2.4	590	2.6	967	1.5

Table 8.1: Session duration in dependence of daytime.

starting time	session duration					
	all		weekday		weekend	
	[s]	CoV	[s]	CoV	[s]	CoV
8:00	789	2.7	773	2.8	875	2.0
9:00	604	2.0	537	1.9	850	2.0
10:00	647	2.1	572	2.1	879	2.0
11:00	632	1.9	535	1.8	893	1.8
12:00	619	1.9	519	1.7	888	1.9
13:00	654	2.1	576	1.9	853	2.3
14:00	701	2.1	633	2.0	888	2.2
15:00	707	2.0	616	1.9	946	1.9
16:00	708	1.9	602	1.8	987	1.8
17:00	705	2.1	629	2.4	892	1.7
18:00	782	2.1	732	1.9	945	2.3
19:00	817	1.9	779	2.0	939	1.7
20:00	1020	1.9	953	1.8	1251	1.9
21:00	1232	1.8	1204	1.9	1337	1.7
22:00	1298	1.7	1274	1.7	1392	1.6
23:00	1396	1.6	1367	1.6	1537	1.6

Table 8.1: Session duration in dependence of daytime.

A.2 Session Volume

starting time	session volume					
	all		weekday		weekend	
	[kB]	CoV	[kB]	CoV	[kB]	CoV
0:00	2677	2.6	2654	2.7	2754	2.1
1:00	2782	2.2	2593	2.2	3289	2.2
2:00	3062	2.1	2190	2.0	4627	1.9
3:00	2824	2.2	2622	2.1	3239	2.4

Table 8.2: Session volume in dependence of daytime.

starting time	session volume					
	all		weekday		weekend	
	[kB]	CoV	[kB]	CoV	[kB]	CoV
4:00	2774	1.4	2755	1.3	2832	1.5
5:00	2219	2.2	2058	2.5	2653	1.5
6:00	1443	3.0	1485	3.2	1283	1.6
7:00	1153	2.5	986	2.5	2019	2.3
8:00	1070	4.6	792	2.7	2563	4.4
9:00	917	5.8	596	3.8	2106	5.0
10:00	814	3.9	532	2.6	1692	3.5
11:00	814	3.0	519	3.2	1606	2.3
12:00	726	2.8	460	2.5	1438	2.3
13:00	736	3.5	514	2.8	1302	3.3
14:00	797	4.4	544	2.5	1495	4.3
15:00	795	3.0	554	2.8	1429	2.6
16:00	882	3.4	636	3.3	1534	2.9
17:00	825	2.9	639	3.4	1281	2.2
18:00	1006	2.8	918	2.6	1288	3.1
19:00	1143	3.3	1014	3.5	1559	2.8
20:00	1380	3.1	1238	3.1	1869	3.0
21:00	1666	2.9	1632	3.1	1795	2.1
22:00	1765	3.5	1710	3.8	1981	2.4
23:00	1980	2.4	1913	2.4	2304	2.0

Table 8.2: Session volume in dependence of daytime.

A.3 Session Inter-Arrival Time

starting time	session inter arrival time					
	all		weekday		weekend	
	[s]	CoV	[s]	CoV	[s]	CoV
0:00	22	1.0	20	1.0	31	1.0
1:00	37	1.0	35	1.0	42	1.0
2:00	65	1.1	69	1.1	56	1.0
3:00	135	1.1	140	1.1	127	1.0
4:00	279	1.0	265	1.1	324	0.9
5:00	331	1.0	310	1.1	388	0.8
6:00	183	1.0	155	1.0	287	0.9
7:00	60	1.1	50	1.1	112	0.9
8:00	33	1.2	27	1.0	64	1.1
9:00	23	1.0	21	1.0	34	1.0
10:00	18	1.0	17	1.0	23	1.0
11:00	17	1.0	16	1.0	18	1.0
12:00	17	1.0	17	1.0	18	0.9
13:00	16	1.0	16	1.0	16	1.0
14:00	16	1.0	15	1.0	17	1.1
15:00	17	1.0	16	1.0	17	1.0
16:00	15	1.0	14	1.0	15	1.0
17:00	14	1.0	14	1.0	14	1.1
18:00	11	1.0	10	1.0	13	1.0
19:00	10	1.0	10	1.0	13	1.0
20:00	11	1.1	10	1.1	14	1.1
21:00	10	1.1	9	1.0	14	1.1
22:00	12	1.1	10	1.1	16	1.0
23:00	15	1.6	13	1.1	26	1.8

Table 8.3: Session inter-arrival time in dependence of daytime.

B Tail Estimation

To improve the readability of the document, the graphs for the tail-estimation and distribution fitting of the measured Internet traffic characteristics were shown only partially in the related section. The complete overview of the estimators is given in the following,

B.1 Session Estimators

Session Inter-Arrival Time

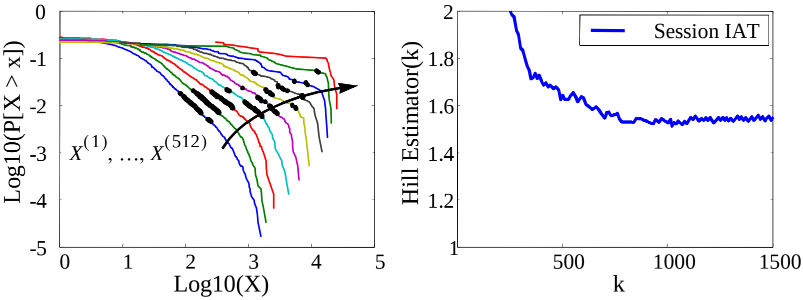

Figure B.1: Scaling and Hill estimator of the session inter-arrival time.

Session Volume

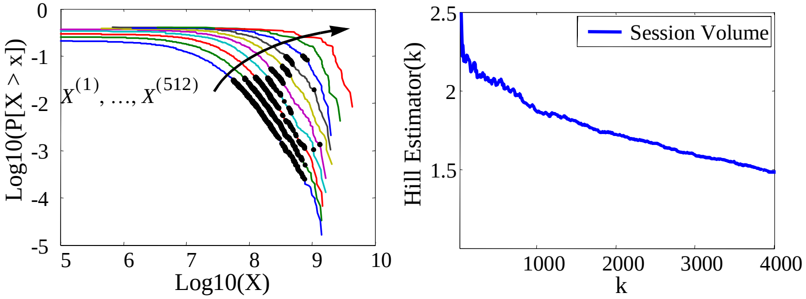

Figure B.2: Scaling and Hill estimator of the session volume.

Session Duration

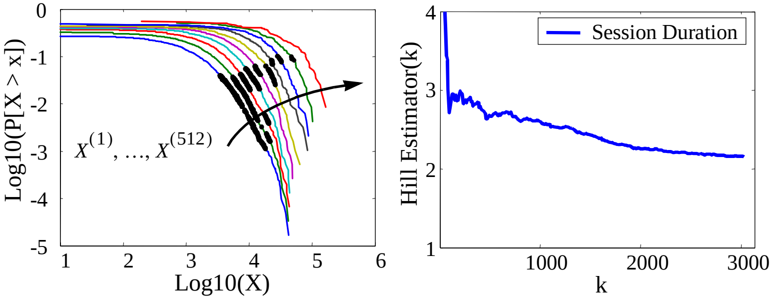

Figure B.3: Scaling and Hill estimator of the session duration.

B.2 Web Page Estimators, by locality and 1s gap

HTTP Inter-Arrival Time

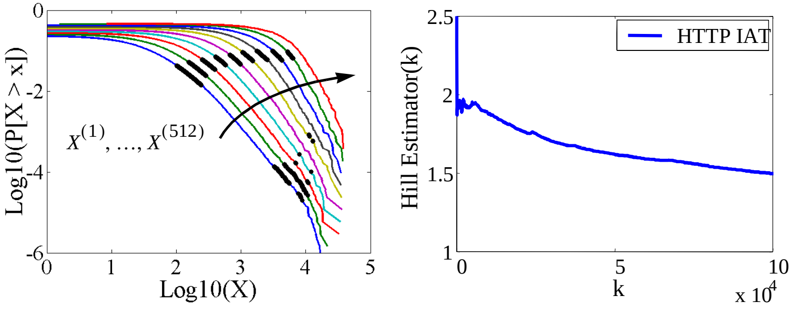

Figure B.4: Scaling and Hill estimator of the HTTP inter-arrival time, pages discerned by locality and 1s gap.

HTTP Connections Per Page

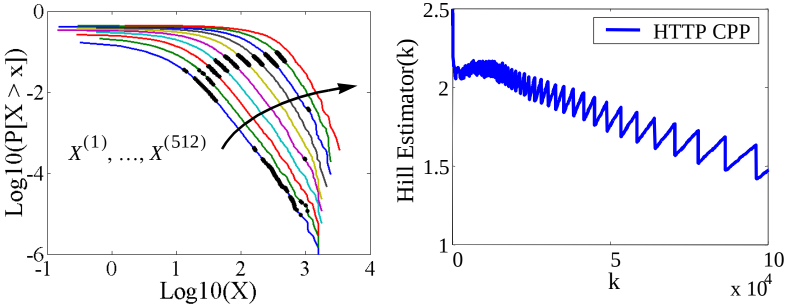

Figure B.5: Scaling and Hill estimator of the number of connections per HTTP page, pages discerned by locality and 1s gap.

HTTP Page Volume

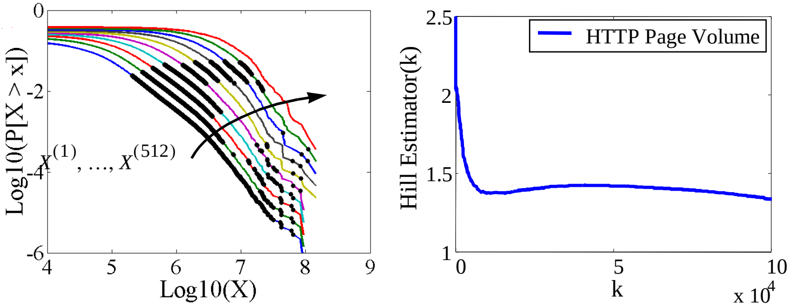

Figure B.6: Scaling and Hill estimator of the HTTP page volume, pages discerned by locality and 1s gap.

HTTP Initial Object Size

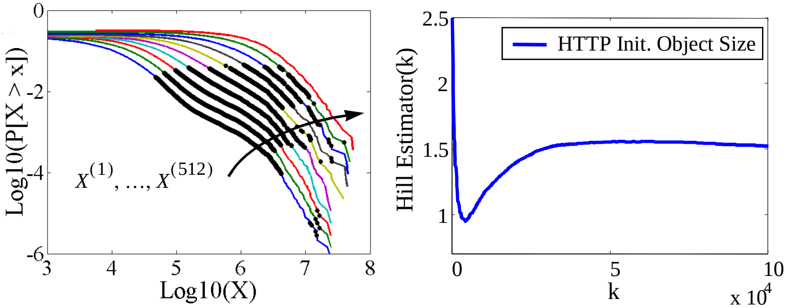

Figure B.7: Scaling and Hill estimator of the HTTP initial object size, pages discerned by locality and 1s gap.

HTTP Follow-On Object Size

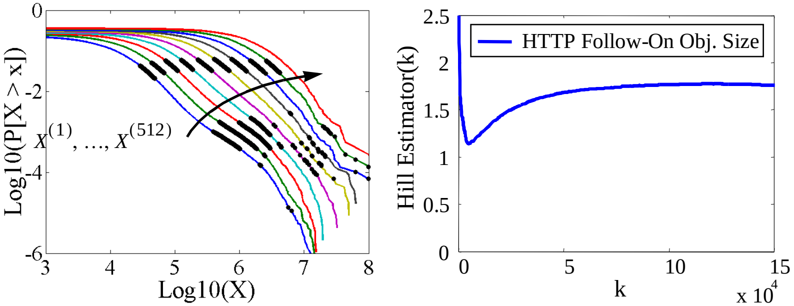

Figure B.8: Scaling and Hill estimator of the HTTP follow-on object size, pages discerned by locality and 1s gap.

HTTP User-Think Time

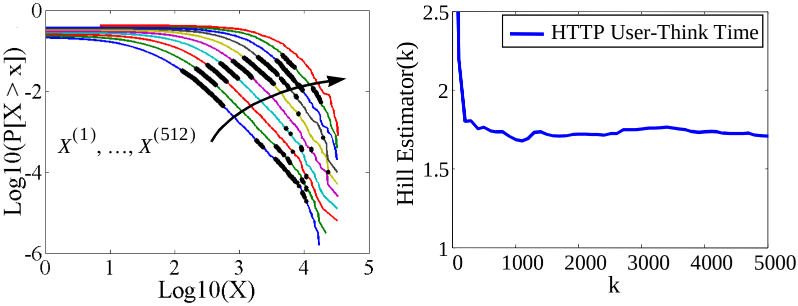

Figure B.9: Scaling and Hill estimator of the HTTP user-think time, pages discerned by locality and 1s gap.

B.3 Web Page Estimators, by 1s gap

HTTP Inter-Arrival Time

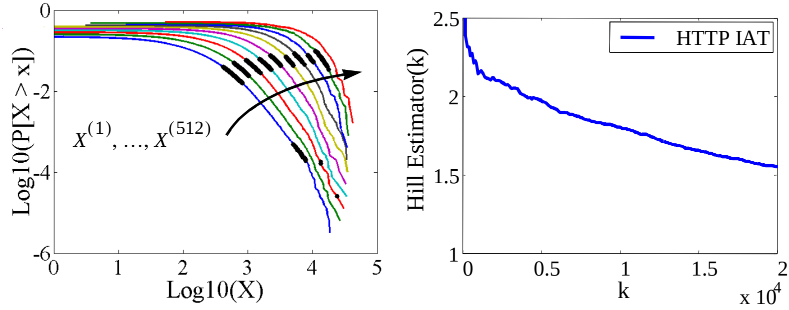

Figure B.10: Scaling and Hill estimator of the HTTP inter-arrival time, pages discerned 1s gap.

HTTP Connections Per Page

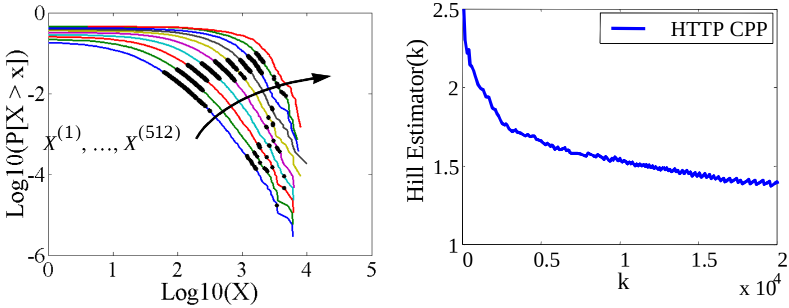

Figure B.11: Scaling and Hill estimator of the number of connections per HTTP page, pages discerned 1s gap.

HTTP Page Volume

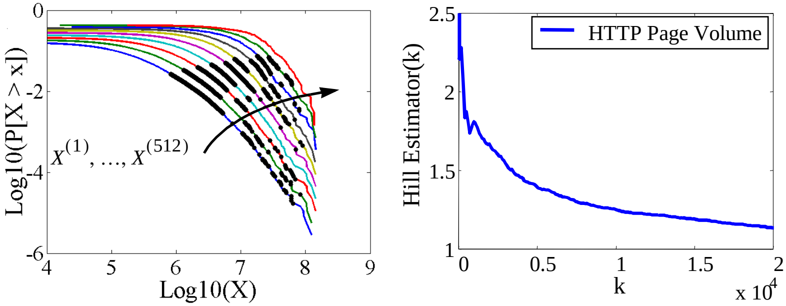

Figure B.12: Scaling and Hill estimator of the HTTP page volume, pages discerned by 1s gap.

HTTP Initial Object Size

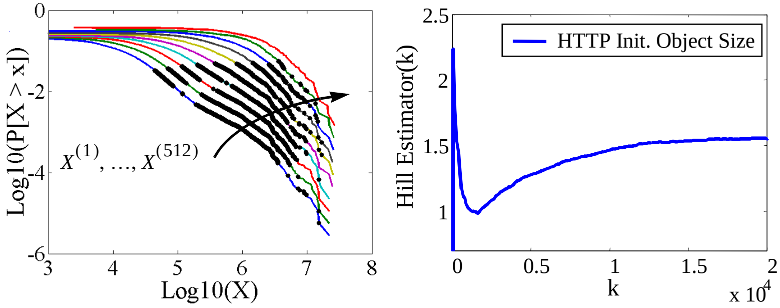

Figure B.13: Scaling and Hill estimator of the HTTP initial object size, pages discerned by 1s gap.

HTTP Follow-On Object Size

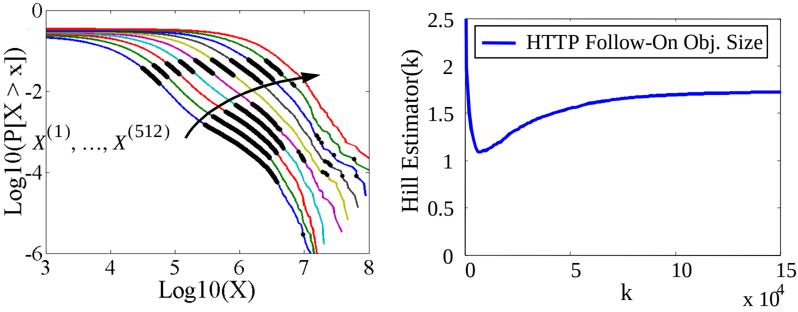

Figure B.14: Scaling and Hill estimator of the HTTP follow-on object size, pages discerned by 1s gap.

HTTP User-Think Time

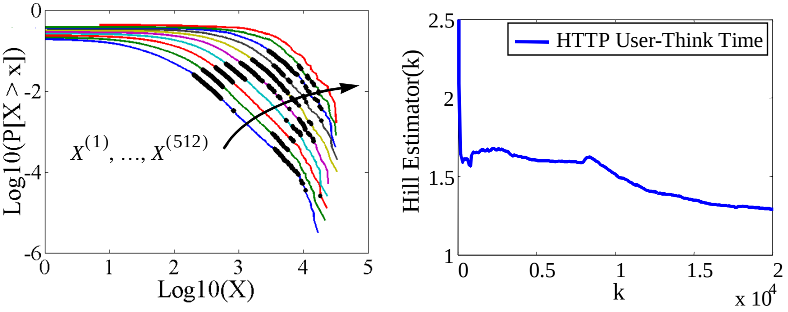

Figure B.15: Scaling and Hill estimator of the HTTP user-think time, pages discerned by 1s gap.

B.4 FTP Estimators

FTP Control Inter-Arrival Time

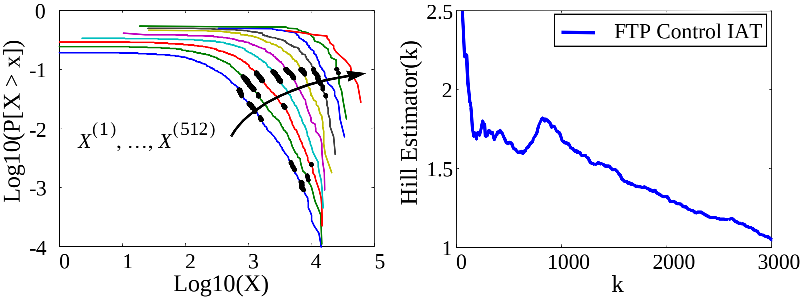

Figure B.16: Scaling and Hill estimator of the FTP control inter-arrival time.

FTP Control Volume

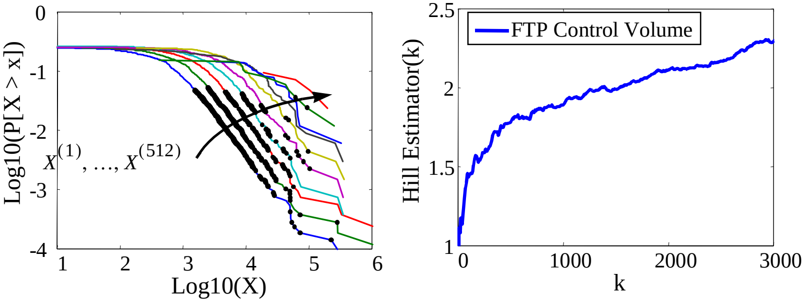

Figure B.17: Scaling and Hill estimator of the FTP control volume.

FTP Data Inter-Arrival Time

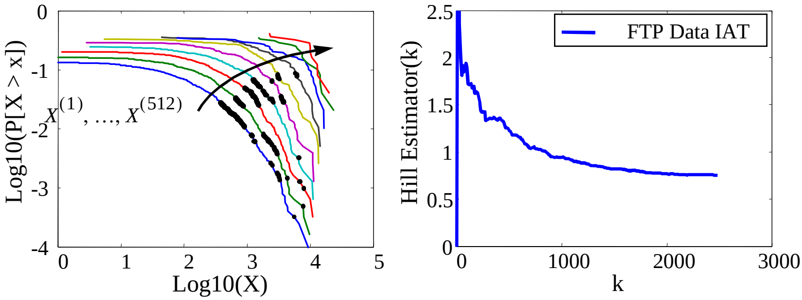

Figure B.18: Scaling and Hill estimator of the FTP data inter-arrival time.

FTP Data Volume

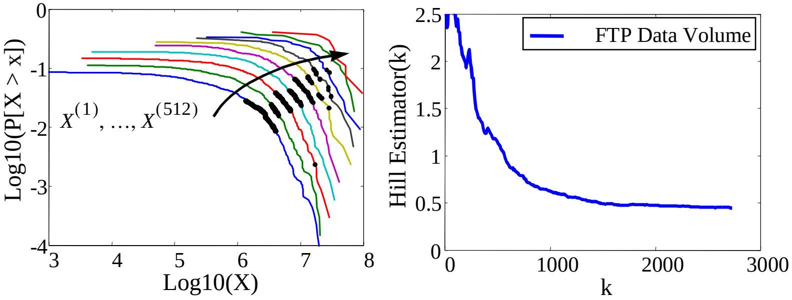

Figure B.19: Scaling and Hill estimator of the FTP data volume.

B.5 Mail Estimators

SMTP Inter-Arrival Time

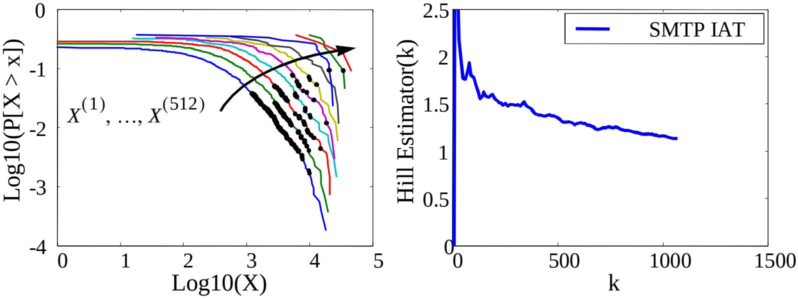

Figure B.20: Scaling and Hill estimator of the SMTP inter-arrival time.

SMTP Volume

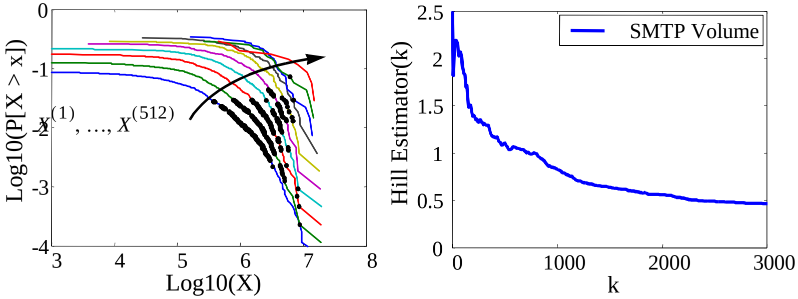

Figure B.21: Scaling and Hill estimator of the SMTP volume.

POP3 Inter-Arrival Time

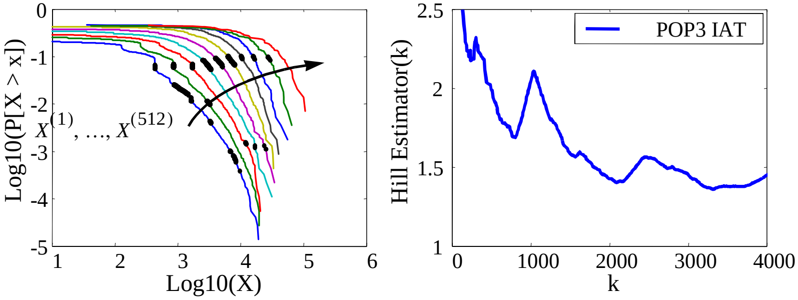

Figure B.22: Scaling and Hill estimator of the POP3 inter-arrival time.

POP3 Volume

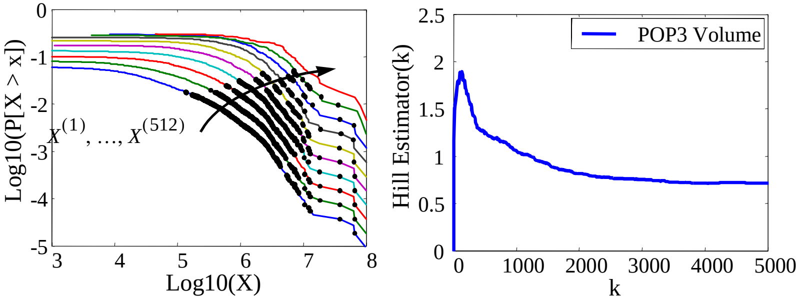

Figure B.23: Scaling and Hill estimator of the POP3 volume.

List of Figures

Bibliography

[1] H. Akimaru and K. Kawashima. *Teletraffic: Theory and Applications*. 2nd Edition, Springer, 1999.

[2] M. F. Arlitt and C. L. Williamson. *Web Server Workload Characterization: The Search for Invariants.* In Proc. of the 1996 ACM SIGMETRICS, May 1996, Philadelphia, PA, USA.

[3] P. Barford, A. Bestavros, A. Bradley, and M. E. Crovella. *Changes in Web Client Access Patterns: Characteristics and Caching Implications*. World Wide Web Journal, Special Issue on Characterization and Performance Evaluation, Vol. 2, 1999.

[4] T. Berners-Lee, R. Cailliau, and J.-F. Groff. *The World-Wide Web*. Computer Networks and ISDN Systems, 25, NorthHolland, 1992.

[5] T. Berners-Lee and D. Connolly. *Hypertext Markup Language - 2.0*. RFC 1866, November 1995.

[6] T. Berners-Lee, R. Fielding, and H. Frystyk. *Hypertext Transfer Protocol - HTTP/1.0.* RFC 1945, May 1996.

[7] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax.* RFC 2396, August 1998.

[8] V. Bolotin, J. Coombs-Reyes, D. Heyman, Y. Levi, and D. Liu. *IP Traffic Characterization for Planning and Control.* In Proc. of the ITC 16, June 1999, Edinburgh, UK.

[9] S. Bonham, D. Grossman, W. Portnoy, and K. Tam. *Quake: An Example Multi-User Network Application - Problems and Solutions in Distributed Interactive Simulation.* Technical Report, University of Washington, Seattle, USA. May 2000.

[10] M. S. Borella. *Source Models of Network Game Traffic.* Computer Communications, Vol. 23, No. 4, February 2000.

[11] N. Borenstein and N. Freed. *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies.* RFC 1521, September 1993.

[12] L. Brakmo and L. Peterson. *TCP Vegas: End to End Congestion Avoidance on a Global Internet.* IEEE Journal on Selected Areas in Communication, Vol. 13, No. 8, October 1995.

[13] R. Caceres, P. B. Danzig, S. Jamin, and D. J. Mitzel. *Characteristics of Wide-Area TCP/IP Conversations.* In Proc. of the ACM SIGCOMM ’91. November 1991, Zurich, Swiss.

[14] R. Caceres, N. G. Duffield, A. Feldmann, et al. *Measurement and Analysis of IP Network Usage and Behavior.* IEEE Communications Magazine, Vol. 38 No. 5, May 2000.

[15] J. Charzinski. *HTTP/TCP Connection and Flow Characteristics.* Performance Evaluation, Vol. 42 No. 2-3, September 2000.

[16] J. Charzinski. *Internet Client Traffic Measurement and Characterization Results.* In Proc. of the ISSLS 2000, June 2000, Stockholm, Sweden.

[17] J. Charzinski. *Web Performance in Practice - Why We are Waiting.* AEÜ International Journal of Electronic Communication, Vol. 55 No. 1, 1-1, 2001.

[18] H.-K. Choi and J. O. Limb. *A Behavioral Model of Web Traffic.* In Proc. of the 7th International Conference on Network Protocols (ICNP’99), October 1999, Toronto, Canada.

[19] D. Connolly and L. Masinter. *The 'text/html' Media Type.* RFC 2854, June 2000.

[20] M. Crispin. *Internet Message Access Protocol, Version 4 (IMAP4).* RFC 1730. December 1994.

[21] M. E. Crovella and A. Bestavros. *Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes.* In Proc. of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, May 1996, Philadelphia, PA, USA.

[22] M. E. Crovella and L. Lipski. *Long-Lasting Transient Conditions in Simulations with Heavy-Tailed Workloads.* In Proc. of the 1997 Winter Simulation Conference, December 1997, Atlanta, USA.

[23] M. E. Crovella and M. S. Taqqu. *Estimating the Heavy Tail Index from Scaling Properties.* Methodology and Computing in Applied Probability, Vol. 1 No. 1, 1999.

[24] C. R. Cunha, A. Bestavros, and M. E. Crovella. *Characteristics of WWW Client-based Traces.* Technical Report TR-95-010, Boston University Computer Science Department, June 1995.

[25] B. D. Davison. *A Survey of Proxy Cache Evaluation Techniques.* In Proc. of the Fourth International Web Caching Workshop, April 1999, San Diego, CA,USA.

[26] M. Falk, R.-D. Reiss, and J. Hüsler. *Laws of Small Numbers: Extremes and Rare Events.* DMV-Seminar 23, Birkhäuser, 1994.

[27] J. Färber. *Network Game Traffic Modelling.* In Proc. of NetGames 2002, April 2002, Braunschweig, Germany.

[28] J. Färber, S. Bodamer, and J. Charzinski. *Measurement and Modelling of Internet Traffic at Access Networks.* In Proc. of the EUNICE '98 - Open European Summer School on Network Management and Operation. August 1998, München, Germany.

[29] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. *Hypertext Transfer Protocol -- HTTP/1.1.* RFC 2068 January 1997.

[30] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol - HTTP/1.1.* RFC 2616, June 1999.

[31] C. Finseth. *An Access Control Protocol, Sometimes Called TACACS.* RFC 1492, July 1993.

[32] S. Floyd and T. Henderson. *The NewReno Modification to TCP's Fast Recovery Algorithm.* RFC 2582, April 1999.

[33] S. Floyd and V. Jacobson. *Random Early Detection Gateways for Congestion Avoidance.* IEEE / ACM Transactions on Networking, Vol. 1, No. 4, August 1993.

[34] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky. *An Extension to the Selective Acknowledgement (SACK) Option for TCP.* RFC 2883, July 2000.

[35] M. W. Garrett and W. Willinger. *Analysis, Modeling and Generation of Self-Similar VBR Video Traffic*. In Proc. of the ACM SIGCOMM '94, August 1994, London, UK.

[36] C. M. Goldie and C. Klüppelberg. *Subexponential Distributions.* In ed. R. L. Adler, R. Feldman and, M. S. Taqqu. *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Birkhäuser, Boston, 1998.

[37] IEEE 802 LAN/MAN Standards Committee, 802.3 Ethernet Working Group. *Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.* available at http://ieee.org.

[38] M. Harchol-Balter and A. Downey. *Exploiting Process Lifetime Distributions for Dynamic Load Balancing*. In Proc. of the ACM SIGMETRICS '96, May 1996, Philadelphia, USA.

[39] F. Hernández-Campos, J. S. Marron, G. Samorodnitsky, and F. D. Smith. *Variable Heavy Tailed Durations in Internet Traffic, Part I: Understanding Heavy Tails.* In Proc. of 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), October 2002, Fort Worth, TX, USA

[40] B. M. Hill. *A Simple General Approach to Inference about the Tail of a Distribution.* The Annals of Statistics, No. 3, 1975.

[41] V. Jacobson. *Congestion Avoidance and Control*. In Proc. of ACM SIGCOMM ’88, August 1988, Stanford, USA.

[42] V. Jacobson. *Modified TCP Congestion Avoidance Algorithm.* end2end-interest mailing list, April 30, 1990. ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail.

[43] B. Kantor and P. Lapsley. *Network News Transfer Protocol.* RFC 977, February 1986.

[44] U. R. Krieger, N. M. Markovitch, and N. Vicari. *Analysis of World Wide Web Traffic by Nonparametric Estimation Techniques.* In Proc. of the International Conference on the Performance and QoS of Next Generation Networking, November 2000, Naogya, Japan.

[45] W. E. Leland, W. Willinger, M. S. Taqqu, and D. V. Willson. *On the Self-Similar Nature of Ethernet Traffic.* In Proc. of the ACM SIGCOMM'93, September 1993, San Francisco, CA, USA.

[46] S. McCreary and kc claffy. *Trends in Wide Area IP Traffic Patterns: A View from Ames Internet eXchange.* In Proc. of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling, September 2000, Monterey, CA, USA.

[47] B. A. Mah. *An Empirical Model of HTTP Network Traffic.* In Proc. of the IEEE INFOCOM '97, April 1997, Kobe, Japan.

[48] J. Mahdavi and S. Floyd, *TCP-Friendly Unicast Rate-Based Flow Control.* Technical note sent to the end2end-interest mailing list, January 8, 1997. http://www.psc.edu/networking/papers/tcp_friendly.html.

[49] G. Malkin. *RIP Version 2.* RFC 2453, November 1998.

[50] B. B. Mandelbrot. *Scaling in Financial Prices: I. Tails and Dependance.* Quantitative Finance, Vol. 1, 2001.

[51] S. Manley and M. Seltzer. *Web Facts and Fantasy.* In Proc. of the 1997 USENIX Symposium on Internet Technologies and Systems. December 1997, Monterey, CA, USA.

[52] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *TCP Selective Acknowledgement Options.* RFC 2018, October 1996.

[53] J. Mo, R. J. La, V. Anantharam, and J. Walrand. *Analysis and Comparison of TCP Reno and Vegas.* In Proc. of the IEEE INFOCOM'99. March 1999, New York, USA.

[54] P. V. Mockapetris. *Domain names - concepts and facilities.* RFC 1034, November 1987.

[55] P. V. Mockapetris. *Domain names - implementation and specification.* RFC 1035, November 1987.

[56] S. Molnar and T. D. Dang. *Scaling Analysis of IP Traffic Components.* In Proc. of the 13th ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, September 2000, Monterey, CA, USA.

[57] J. Moy. *OSPF Version 2.* RFC 2328, April 1998.

[58] M. W. Murhammer, O. Atakan, S. Bretz, L. R. Pugh, K. Suzuki, and D. H. Wood. *TCP/IP Tutorial and Technical Overview. IBM Redbook, October 1998.*

[59] J. Myers and M. Rose. *Post Office Protocol, Version 3 (POP3).* RFC 1939, May 1996.

[60] E. Nebel and L. Masinter. *Form-based File Upload in HTML.* RFC 1867, November 1995.

[61] NetBIOS Working Group. *Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods*. RFC 1001, March 1987.

[62] H. F. Nielsen, J. Gettys, A. Baird-Smith, E. Prudhommeaux, H. W. Lie, and C. Lilley. *Network Performance Effects of HTTP/1.1.* In Proc. of ACM SIGCOMM'97, August 1997, Cannes, France.

[63] J. Padhye and S. Floyd. *On Inferring TCP Behavior.* In Proc.of ACM SIGCOMM’01, September 2001, San Diego, USA.

[64] V. Paxson. *Empirically-Derived Analytic Models of Wide-Area TCP Connections.* IEEE/ACM Transactions on Networking, 2(4), August 1994.

[65] D. Perkins. *Point-to-Point Protocol for the transmission of multi-protocol datagrams over Point-to-Point links.* RFC 1171, July 1990.

[66] D. C. Plummer. *Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware.* RFC 826, November 1982.

[67] J. Postel. *Internet Control Message Protocol.* RFC 792, September 1981.

[68] J. Postel. *Simple Mail Transfer Protocol (SMTP).* RFC 821, August 1982.

[69] J. Postel. *User Datagram Protocol (UDP).* RFC 768, August 1980.

[70] J. Postel (Editor). *Internet Protocol (IP).* RFC 791, September 1981.

[71] J. Postel (Editor). *Transmission Control Protocol (TCP).* RFC 793, September 1981.

[72] J. Postel and J. K. Reynolds. *File Transfer Protocol (FTP).* RFC 959, October 1985.

[73] J. Postel, J. K. Reynolds. *Telnet Protocol Specification.* RFC 854, May 1983.

[74] D. Raggett. *HTML Tables.* RFC 1942, May 1996.

[75] K. Ramakrishnan and S. Floyd. *A Proposal to add Explicit Congestion Notification (ECN) to IP.* RFC 2481, January 1999.

[76] W. J. Reed and M. Jorgensen. *The double Pareto-lognormal distribution - A new parametric model for size distribution.* Submitted to Com. Stats -Theory & Methods, 2002.

[77] R.-D. Reiss and M. Thomas. *Statistical Analysis of Extreme Values.* 2nd ed. Birkhäuser, Basel, 2001.

[78] Y. Rekhter, T. Li. *A Border Gateway Protocol 4 (BGP-4).* RFC 1771 March 1995.

[79] J. Reynolds and J. Postel. *Assigned Numbers.* RFC 1700, October 1994. Now maintained at www.iana.org as living document.

[80] C. Rigney, A. Rubens, W. Simpson, and S. Willens. *Remote Authentication Dial In User Service (RADIUS).* RFC 2138, April 1997.

[81] W. Stevens. *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms.* RFC 2001, January 1997.

[82] W. R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols.* Addison Wesley Longman Inc., 13th printing, 1999.

[83] M. Subramaniam. *Network Management: An Introduction to Principles and Practice.* Addison-Wesley, 2000.

[84] A. S. Tanenbaum. *Computer Networks.* Prentice Hall, 3rd edition, 1996.

[85] K. Thompson, G. J. Miller, and R. Wilder. *Wide-Area Internet Traffic Patterns and Characteristics.* IEEE Network, 11(6), November/December 1997.

[86] J. Touch, J. Heidemann, and K. Obraczka. *Analysis of HTTP Performance.* USC/ISI Research Report 98-463, December 1998.

[87] P. Tran-Gia, D. Staehle, and K. Leibnitz. *Source Traffic Modeling of Wireless Applications*. International Journal of Electronics and Communications (AEÜ). Vol. 55, No. 1, 2001.

[88] N. Vicari. *Measurement and Modeling of WWW-Sessions.* Institute of Computer Science, University of Würzburg, Technical Report No. 184, September 1997.

[89] N. Vicari. *Models of WWW-Traffic: a Comparison of Pareto and Logarithmic Histogram Models.* In Proc. of the 5th Polish Teletraffic Symposium/ Project COPERNICUS 1463 ATMiN - Closing Seminar, April 1998, Warsaw, Poland.

[90] N. Vicari and S. Köhler. *Measuring Internet User Traffic Behavior Dependent on Access Speed.* In Proc. of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling, September 2000, Monterey, USA.

[91] N. Vicari, S. Köhler and J. Charzinski. *The Dependence of Internet User Characteristics on Access Speed.* In Proc. of the LCN'2000, November 2000, Tampa, USA.

[92] A. Wolman, G. Voelker, N. Sharma, N. Cardwell, M. Brown, T. Landray, D. Pinnel, A. Karlin, and H. Levy. *Organization-Based Analysis of Web Object Sharing and Caching.* In Proc. of the 2nd USENIX Conference on Internet Systems and Technologies (USITS), October 1999, Boulder, USA.

[93] DFN. *B-Win Statistics.*
http://www-win.rrze.uni-erlangen.de/cgi-bin/auswahl-linkstat.pl.

[94] MATLAB: Statistics Toolbox.
http://www.mathworks.com/access/helpdesk/help/toolbox/stats/stats.shtml.

[95] TCPDump, available via http://ftp.ee.lbl.gov/.

[96] TCPTrace, available via
http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html.

ISSN 1432 - 8801