

Modellierung von IP-basiertem Paketverkehr ausgewählter interaktiver Dienste

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Johannes Wolfgang Färber

geb. in München

Hauptberichter: Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Mitberichter: Prof. Dr. rer.nat. Hans-Peter Grossmann (Univ. Ulm)
Tag der Einreichung: 16. Dezember 2005
Tag der mündlichen Prüfung: 6. Juni 2007

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2007

für Giovanna

Kurzfassung

Modellierung von IP-basiertem Paketverkehr ausgewählter interaktiver Dienste

Moderne Kommunikationssysteme zeichnen sich durch eine hohe Komplexität aufgrund des Zusammenspiels vieler Protokolle und Komponenten aus. Ihr Verhalten kann nicht mehr eindeutig vorherberechnet werden. Um ihre Leistungsfähigkeit dennoch bewerten zu können, werden simulative Untersuchungen angestellt. Hierbei werden die wesentlichen Eigenschaften der Kommunikationssysteme in Modellen abgebildet, die dann unter verschiedenen Randbedingungen und bei unterschiedlicher Belastung beobachtet werden können. Aus den Zusammenhängen von Systemparametern und Systemverhalten können Erkenntnisse beispielsweise zur Optimierung der Systemkomponenten gewonnen werden. Für solche simulativen Untersuchungen muß durch Verkehrsmodelle eine definierte und wirklichkeitsnahe Systembelastung erzeugt werden, also der Kommunikationsverkehr, den das System bearbeiten muß.

In der vorliegenden Arbeit werden konkrete Kommunikationsdienste untersucht und mit dem Ziel modelliert, ihre typische Verkehrslast auf Paketschicht nachzubilden. Die Betrachtung einzelner Dienste erlaubt eine funktionale Modellierung der Kommunikationsvorgänge über alle Kommunikationsschichten, so daß auch Rückkopplungen und Abhängigkeiten berücksichtigt werden können. Ein wesentlicher Gesichtspunkt solcher Modelle ist die Suche nach invarianten Parametern, welche nicht von Ort und Zeitpunkt der Verkehrsbetrachtung abhängen. Kann ein Modell entsprechend unabhängig parametrisiert werden, dann kann es auch entsprechend flexibel eingesetzt werden. Die orts- und zeitabhängigen Anteile der Verkehrscharakteristik ergeben sich dann aus den in der Simulation berücksichtigten Rückkopplungen und Abhängigkeiten. Zur Identifizierung invarianter Parameter erfolgt eine Analyse der Kommunikationsvorgänge eines Dienstes. Durch Verkehrsmessungen kann dann das Verhalten der ausgewählten Parameter untersucht und statistisch beschrieben werden, so daß sie wieder in einem entsprechend aufgebauten Verkehrsmodell eingesetzt werden können.

Für diesen Modellierungsansatz wurden zwei interaktive Dienste ausgewählt, die für die Leistungsbewertung von IP-basierten Netzen von großer Bedeutung sind. Zunächst wird der Informationsabruf über das *World Wide Web* (WWW) untersucht, da dies einen Großteil des interaktiven Verkehrsaufkommens ausmacht. Als zweiter Dienst werden zunehmend populäre Computerspiele mit mehreren über Kommunikationsnetze verbundenen Teilnehmern betrachtet. Aus dem breiten Spektrum der Spielegenres werden dabei nur Reaktionsspiele untersucht, welche die größten Anforderungen an ein Kommunikationsnetz stellen.

Das WWW stellt einen typischen Abrufdienst bereit, der sich aufgrund der wechselnden Anforderungs- und Lesephasen durch eine große Verkehrsdynamik auszeichnet. Das Bündelverhalten wird durch den Umfang und Aufbau der WWW-Seiten, das Kommunikationsprotokoll HTTP (*Hypertext Transfer Protocol*) sowie das Verhalten von Server, Browser und Nutzer charakterisiert. WWW-Seiten bestehen aus einer Hauptseite und den darin referenzierten eingebetteten Objekten wie Bildern, Textrahmen oder Video-Clips. Die Hauptseite muß zuerst geladen werden (primärer Abruf), bevor die eingebetteten Objekte angefordert werden können (sekundäre Abrufe). Das HTTP setzt auf dem gesicherten Internet-Protokoll TCP auf und ist im Grunde ein einfaches Handshake-Protokoll, welches Anforderungen einzelner Objekte sowie die zugehörige Antwort definiert. Browser und Server sind meist so implementiert, daß HTTP-Objekte parallel über mehrere TCP-Verbindungen angefordert und übertragen werden können. In Abhängigkeit der Anwendungseinstellungen wird bei jedem Seitenabruf daher ein Verkehrsbüschel erzeugt, das mehr oder weniger aggressiv die verfügbare Bandbreite beansprucht. Ein WWW-Nutzer hat die Möglichkeit, abgesetzte Abrufe abzubrechen, so daß nicht nur seine Lesezeit sondern auch seine Ungeduld in die Verkehrscharakteristik einfließt.

Das in dieser Arbeit vorgeschlagene rückgekoppelte WWW-Verkehrsmodell bildet die genannten Abhängigkeiten nach. In Anlehnung an bestehende Verkehrsmodelle werden primäre und sekundäre Abrufe unterschieden und durch die Objektgröße charakterisiert. Sekundäre Abrufe hängen aber strikt vom Abschluß des primären Abrufs sowie der Anzahl erlaubter TCP-Verbindungen ab und nicht von einer festgelegten Zwischenankunftszeit. Primäre Abrufe werden erzeugt, wenn nach dem Laden der vollständigen Seite eine gewisse Lesezeit verstrichen ist oder wenn vor dem Laden der vollständigen Seite die Geduldsspanne des Nutzers abgelaufen ist. Die einzelnen TCP-Verbindungen werden mit ihren Flußkontrollmechanismen simuliert, so daß auch auf Transportschicht Rückkopplungen und Abhängigkeiten einfließen. Das Verkehrsmodell zeichnet sich also durch eine starke Abhängigkeit vom aktuellen Netzzustand aus, da von den Rückmeldungen auf Transportschicht, Kommunikationssteuerungsschicht und des Nutzers die Lasterzeugung maßgeblich bestimmt wird.

Als interaktiver Dienst muß das WWW gewisse zeitliche Anforderungen erfüllen. Allerdings können auch längere Wartezeiten toleriert werden. Für die Bewertung der Simulationsergebnisse ist dennoch die Wartezeit entscheidend, so daß für das vorgeschlagene Modell hauptsächlich die Seitenladezeit als Maß der Dienstgüte verwendet wird.

Die Parametrisierung des Modells basiert auf der Analyse der Protokoll- und Anwendungsoptionen (Protokollversion, Anzahl paralleler Verbindungen, interne Zeitüberwachung) sowie der statistischen Beschreibung der Verkehrsgrößen wie Objektgröße, Anzahl eingebetteter Objekte, Betrachtungsdauer oder Geduldsspanne. Durch Messungen können viele Verkehrsgrößen direkt beobachtet werden, einige Parameter treten aber nur indirekt auf oder sind nicht eindeutig unterscheidbar und müssen daher durch Schätzungen ermittelt werden. In diesen Fäl-

len werden entsprechende Schätzverfahren vorgestellt und bewertet. Die Modellparameter werden durch ihre empirisch bestimmten Verteilungsfunktionen beschrieben, die zur kompakteren und flexibleren Darstellung durch mathematische Funktionen angenähert werden.

Bei der simulativen Validierung des Verkehrsmodells zeigt sich, daß die geschätzten Modellparameter zu einem plausiblen Verkehrsverhalten führen. Im Vergleich mit SURGE, einem oft eingesetzten nicht rückgekoppelten WWW-Verkehrsmodell, zeigt sich, daß das reaktive Modell das Büschelverhalten des WWW-Verkehrs deutlich besser nachbilden kann. Da beim nicht-reaktiven Modell die Objekt-Abrufe über unabhängige Zwischenankunftszeiten initiiert werden, entstehen während des Ladevorgangs deutlich mehr Pausen, so daß die Seitenladezeit erheblich größer ist. Eine steigende Ende-zu-Ende-Verzögerung führt bei dem SURGE-Modell zu deutlich mehr Abbrüchen von Seitenladevorgängen und damit auch insgesamt zu deutlich kleineren WWW-Seiten als eigentlich vorgegeben sind. Bei Paketverlusten reagiert dagegen das reaktive Modell empfindlicher und erzeugt deutlich längere Seitenladezeiten, sofern die Geduldsspanne hoch genug gesetzt ist. Für eine detaillierte Betrachtung von Einzelverkehren oder der Bestimmung oder Optimierung der Dienstgüte (hier anhand der Seitenladezeit) in einem System eignet sich daher klar das reaktive Modell besser. Für die Lastuntersuchung von aggregiertem Verkehr, also der Überlagerung mehrere Quellen, kann aber besser ein einfacheres nicht-rückgekoppeltes Verkehrsmodell eingesetzt werden. Die Summenverkehre beider Modelle verhalten sich bei einer hohen Anzahl von Quellen in etwa gleich.

Online-Spiele werden den Dialogdiensten zugeordnet. Besonders die hier betrachteten Reaktionsspiele zeichnen sich durch die besonders hohen zeitlichen Anforderungen aus. Aufgrund der hohen Teilnehmerzahlen mit in der Regel begrenzter Bandbreite kann keine direkte Kommunikation zwischen allen Teilnehmern realisiert werden. Statt dessen hat sich eine Client-Server-Kommunikation durchgesetzt, bei welcher jeder Teilnehmer seine Aktionen einem meist dedizierten Server mitteilt, der für die Synchronisation des Spielgeschehens zuständig ist und alle Teilnehmer über die Änderungen im nächsten Zyklus informiert. Die gegenseitigen Informationsnachrichten müssen bei moderner Software nicht synchron erfolgen, sondern werden jeweils mit unabhängiger Rate erzeugt. Lediglich der Server synchronisiert seine Nachrichten an alle Spieler, so daß er büschelförmigen Verkehr generiert. Damit ergibt sich ein relativ regelmäßiger Verkehrsfluß, der phasenweise annähernd durch konstante Bitraten bzw. durch einen regelmäßigen Büschelverkehr beschrieben werden kann. Der Verkehrsfluß von Client zu Server kann zudem unabhängig von der Gegenrichtung betrachtet werden, so daß sich Spielerverkehr durch einfache unabhängige Verkehrsmodelle beschreiben läßt. Allerdings ändert sich die Anzahl der Teilnehmer dynamisch und damit ändert sich auch die Größe der Büschel sowie die Größe der Pakete, welche der Server aussendet. Abgesehen von dieser Abhängigkeit zwischen Diensteschicht und Transportschicht bestehen keine weiteren Abhängigkeiten. Wegen der großen Bedeutung der Antwortzeit wird UDP verwendet und auf den Einsatz von TCP mit seinen langwierigen Fehlerkorrekturmechanismen verzichtet. Die Fehler-

behandlung erfolgt dagegen in der Client-Applikation selbst, wo das Spielgeschehen durch Prädiktionsalgorithmen auch ohne Serverinformation fortgeführt werden kann, bis es wieder mit dem tatsächlichen Spielgeschehen resynchronisiert wird. Die Besonderheit des Dienstes liegt in seinen hohen Anforderungen an die Antwortzeit eines Kommunikationsnetzes. Es zeigt sich allerdings, daß neben der reinen Antwortzeit auch eine gewisse Regelmäßigkeit der Empfangsrate und trotz Fehlerbehandlung auch die Paketverluste (d.h. dem stockendem Spielgefühl nach einer Resynchronisierung) bei der Bewertung der erlebten Dienstgüte eine Rolle spielen. Diese Dienstgütekriterien werden zur Bewertung graphisch veranschaulicht und sie können auch von unidirektionalen Messungen abgeleitet werden.

Für die Parametrisierung des Modells müssen neben der Anzahl der Teilnehmer nur die Paketgrößen und die Paketzischenankunftszeiten erfaßt werden. Aufgrund der Dynamik der Spielphasen und der Statusabfragen potentieller Spieler können die tatsächlichen Teilnehmer einer Partie durch Netzmessungen nicht eindeutig bestimmt werden, so daß Schätzmechanismen zu ihrer Identifizierung eingesetzt werden. Die statistische Beschreibung der Verkehrsgrößen kann dann problemlos auch getrennt für Phasen gleichbleibender Teilnehmerzahl erfolgen. Die oben erwähnte Abhängigkeit der Paketgrößen des Servers kann durch eine lineare Anpassung der Parameter der Paketgrößenverteilung in Abhängigkeit der ermittelten oder vorgegebenen Anzahl der Teilnehmer nachgebildet werden.

Aufgrund fehlender Vergleichsmodelle und des gleichmäßigen Verkehrsverhaltens erfolgt die Bewertung des Netzspielesverkehrsmodells in einer exemplarischen Anwendung zur Dienstgütebewertung eines Systems bei unterschiedlicher Belastung. Dabei werden Verkehrseigenschaften ermittelt, welche Rückschlüsse über die vom Nutzer erfahrene Dienstgüte ermöglichen, bzw. welche die typische Netzbelastung auf Serverseite charakterisieren. Die graphische Darstellung der Dienstgütekriterien eignet sich dabei gut zur Bewertung des Systemverhaltens bei unterschiedlichen Randbedingungen.

Die Stärken des verfolgten Modellierungsansatzes liegen in der hohen Genauigkeit, mit welcher Einzelverkehre nachgebildet werden können sowie der direkten Beobachtung von Dienstgüteparametern. Im Falle von sehr sporadisch interaktiven Diensten wie dem WWW erzielt die Berücksichtigung der Rückkopplungen deutliche Vorteile für den Einsatz bei unterschiedlichen Netzauslastungen. Sobald aber stärker aggregierter Verkehr vieler Dienste und vieler Teilnehmer erzeugt werden soll, eignen sich besser Summenverkehrsmodelle, welche den Verkehr effizienter über abstraktere Vorschriften und eventuell auch auf höheren Abstraktionsebenen beschreiben. Aufgrund der schnellen Entwicklung der meisten populären Dienste ist eine schnelle Nachführung der Modellierung notwendig. Falls die Modellarchitektur nach wie vor gültig ist, müssen zumindest die Parameter regelmäßig neu bestimmt werden. Die individuelle Anpassung von Funktionen zur Parameterbeschreibung zeigt sich hierfür als zu komplex, so daß eine Automatisierung der Modellparametrisierung enorme Bedeutung gewinnen wird.

Modelling IP Based Packet Traffic of Selected Interactive Services

Today's communication networks are based on a large number of interacting protocols and components. This leads to a high level of complexity and a system behavior that is not strictly deterministic. In order to evaluate the system's performance, it is necessary to carry out simulation studies. For that, the main system characteristics are mapped to a system model, which is studied under different preconditions and workloads. From the dependencies between system parameters and the observed performance we can derive insights into the system behavior, which can, for example, be used for system optimization. This kind of simulative performance evaluation requires traffic models, which create well defined and realistic workload (i.e. the communication traffic to be processed by the system).

In this thesis, specific communication services are evaluated and modelled in order to describe their typical traffic load on packet level. Studying an individual service allows the building of a functional model over all communication layers – this also allows to take feedback loops and other dependencies into account. One main aspect of this kind of modelling is to find invariant model parameters that are independent of time and place of the traffic observation. If a model can be parameterized by such invariants, it can be deployed with a large flexibility. Those traffic characteristics, which depend on time and place, would automatically be generated when simulating the feedback loops and dependencies. In order to find invariant parameters, the service has to be analyzed carefully. The identified parameters can be derived from traffic measurements and characterized by statistical evaluation, which again allows the incorporation of them into a corresponding traffic model.

For this approach, two interactive services have been selected, which show a high impact on IP based networks. At first, information retrieval via the World Wide Web (WWW) is studied, as this traffic makes up for a large part of the total Internet traffic. The second service under study is computer network gaming, which shows fast growing popularity. From the large spectrum of computer game genres only fast action games are considered, as they have the hardest requirements on the communication network.

The WWW offers a typical information retrieval service. Based on the alternating request and reading phases, it is characterized by a large traffic dynamic. This burst behavior is caused by the size and structure of the web pages, by the communication protocol HTTP (Hypertext Transfer Protocol) and by the behavior of server, browser and user. Web pages are built of a main page, which refers to all embedded objects like images, text frames or video clips. The main page has to be loaded (primary retrieval) before the embedded objects can be requested (secondary retrievals). HTTP is built on top of TCP which provides data protection mechanisms. HTTP is a simple handshake protocol defining requests for individual objects and responses to such requests. In general, browser and server implementations allow the parallel retrieval of multiple HTTP objects over parallel TCP connections. Depending on the applica-

tion settings, a page request generates a series of automatic requests, which compete for the available bandwidth in a more or less aggressive way. As a WWW user can also abort running HTTP retrievals, not only his reading time and impatience influences the traffic characteristic.

In this thesis, all those dependencies are modelled in the suggested reactive WWW traffic model. Based on existing traffic models, primary and secondary retrievals are treated separately and characterized by their object size. However, secondary retrievals are only started after the completion of the corresponding primary retrieval and only if TCP connections are or become available. They do not depend on interarrival times. Primary retrievals are started after the completion of a page download plus a user reading time or if the user patience time ends before the completion of the page retrieval. As each TCP connection is simulated with its individual flow control mechanisms, feedback loops and dependencies are also modelled on transport layer. The traffic model is characterized by strong dependencies on the current network status, as workload generation depends on feedback mechanisms within the transport layer, the session layer and the user.

Although the WWW, as an interactive service, has certain timing requirements, larger response times can be tolerated as long as the retrieval service is functional. However, for the evaluation of the system performance, the response time is the most critical parameter. Therefore, the page retrieval time is chosen as the major quality of service (QoS) criteria.

The parameterization of the model is based on protocol and application analysis (protocol version, number of parallel connections, time out clocks) as well as on the statistical description of the traffic parameters as object size, number of embedded objects, reading time or impatience. Many traffic parameters can be easily derived by measurement, but some can not be observed directly. For these cases, estimation methods are used and evaluated. The model parameters are described by their probability distribution function. For a more compact and more flexible presentation, mathematical functions are approximated to those distribution functions.

The simulative validation of the traffic model reveals that the estimated model parameters lead to a reasonable behavior. Compared to SURGE, an often deployed WWW traffic model without using feedback, the reactive model reflects the burst characteristic of the WWW traffic much better. As the non-reactive model triggers object retrievals through independent interarrival times, a page download shows significantly more pauses and consequently a much higher page download time. If the end-to-end delay is increased, the SURGE model generates much more aborted page downloads, which leads to much smaller pages than anticipated in the model parameter description. With increasing packet loss, however, the reactive model reacts more sensitive and generates longer page download times if user patience is set high enough. For a detailed evaluation of effects of individual user traffic as well as for the determination and optimization of quality of service parameters (page download time) the reactive traffic

model is suited better. For performance evaluation of aggregated traffic (i.e. the superposition of multiple sources), a simpler non-reactive traffic model is suited better. For a high number of sources, the aggregated traffic of the two modelling approaches behave similarly.

Online gaming is a conversational service. The fast action games considered here are characterized by very high requirements to the response time of a network. As the number of participants is rather high and their connection bandwidth is limited, a direct communication between all players is not possible. Therefore a client/server architecture became standard for this kind of application. Every client reports its users actions to a dedicated server, which is responsible for the synchronization of the game state and informs all participants on the changes of the next cycle. These messages do not have to be sent synchronously, so every client can send them at an individual rate. Only the server synchronizes his messages to all participants by bursts of messages. The overall game traffic is characterized by a high degree of regularity, which can be described by almost constant bit rates or by regular packet bursts. Traffic flow from client to server can be treated independently from the other direction which allows the use of simple independent traffic models for game traffic. However, the number of participants changes dynamically, which also leads to a changing size of the burst length and the packet sizes sent out by the server. Besides this dependency between service layer and transport layer, no more dependencies have been observed. As a short response time is crucial for gaming, UDP is deployed instead of time critical TCP with its error correction mechanisms. Error correction is done within the application where the game state is predicted, if server information is lost until it can be re-synchronized with the reference game state. The most specific characteristic of fast action games form the requirements on the response time of a communication network. Apart from the pure response time, response time jitter and packet loss (i.e. a jumpy feeling after re-synchronization) also play a role when evaluating quality of service. In order to easily evaluate these QoS criteria, they are mapped to a graph. They can also be derived from unidirectional measurements.

In order to parameterize the model, only packet size and interarrival times as well as the number of participants have to be described. Because of the high session dynamic and status requests of many potential players, the number of active participants of a game can not easily be determined from network measurements. They are identified by estimation instead. The statistical description of the traffic parameters can be done separately for each phase with a different number of participants. The above mentioned dependency of server packet sizes can be described by a linear parameter adaptation of the approximated packet size distribution function depending on the observed number of active participants.

As no comparable game traffic models are available and as game traffic shows rather regular behavior, the assessment of the suggested traffic model is done within an example scenario in order to show its suitability to determine quality of service levels for online gaming in a system

under varying workload. The study reveals the typical game traffic characteristics, which allow conclusions on the observed quality of service. The QoS graph proves to be well suited for the evaluation of systems under different preconditions.

The general strength of the service specific modelling approach is found in the high level of detail of the modelled individual traffic. In addition, the possibility to directly observe QoS parameters is unique to that approach. In case of very sporadic and bursty traffic like in the WWW, the consideration of the feedback loop leads to significant advantages when simulating different workloads. However, if aggregated traffic from many services and many users should be generated, more comprehensive traffic models are better suited, which use abstract model parameters. Due to the fast development of popular services, their traffic models have to be adapted fast and steadily. If the basic model architecture is still valid, the parameter descriptions should be adapted regularly. An individual parameterization proves to be very complex, which indicates the need for automated model parameterization in the future.

Inhaltsverzeichnis

Kurzfassung	i
Inhaltsverzeichnis	ix
Abkürzungen	xiii
Formelzeichen	xv
1 Einleitung	1
1.1 Kommunikation über paketvermittelnde Netze	1
1.2 Übersicht über die Arbeit	3
2 Grundlagen zu Internetverkehr und Verkehrsbeschreibung	5
2.1 Paketbasierte Kommunikation im Internet	5
2.1.1 Entwicklung	5
2.1.2 Funktionsweise	6
2.1.3 Internet-Protokolle	8
2.1.3.1 Internet Protocol (IP)	8
2.1.3.2 Transmission Control Protocol (TCP)	9
2.1.3.3 User Datagram Protocol (UDP)	12
2.2 Dienste	13
2.2.1 Dienstverständnis	13
2.2.2 Internet-Dienste der Anwendungsschicht	13
2.2.3 Dienstgüte	15
2.3 Verkehrsmessung	17
2.3.1 Einordnung	17
2.3.2 Meßziel und Meßmethodik	18
2.3.3 PC-basierte Messung von IP-Paketverkehr	19
2.4 Verkehrsbeschreibung	22
2.4.1 Charakteristische Verkehrsgrößen	22
2.4.2 Statistische Beschreibung	25
2.4.3 Erzeugung von Verkehrsgrößen	31
2.4.4 Parameterschätzung für Verteilungen	32
2.4.4.1 Näherungsverfahren	33
2.4.4.2 QQ-Plot und PP-Plot	35
2.4.4.3 Erweiterte Näherungsverfahren	37

3	Verkehrsmodellierung des interaktiven Abrufdienstes WWW	41
3.1	Modellierung	41
3.1.1	Dienstmerkmale	41
3.1.2	Dienstgüte-Metriken für WWW-Verkehr	44
3.1.3	Bestehende Verkehrsmodelle	45
3.1.4	Hierarchische Modellierung	49
3.2	Empirische Bestimmung der Modellparameter	52
3.2.1	Messung	53
3.2.2	Parameterbestimmung	55
3.2.2.1	Erkennung der nutzererzeugten Anforderungen	56
3.2.2.2	Bestimmung der Nutzergeduld	60
3.2.2.3	Approximation der Modellparameter	62
3.2.2.4	Parameter der Nutzerschicht	62
3.2.2.5	Parameter der Dienstschicht	64
3.2.2.6	Parameter der Kommunikationssteuerungsschicht	68
3.2.3	Modellparameter	69
3.3	Bewertung des WWW-Verkehrsmodells	70
3.3.1	Randbedingungen der Simulation	70
3.3.1.1	Implementierung	70
3.3.1.2	Simulations-Szenarien	73
3.3.2	Bewertungskriterien	74
3.3.3	Bewertung des reaktiven WWW-Modells	77
3.3.3.1	Auswirkungen der Parameter-Schätzung auf die Modellstabilität	77
3.3.3.2	Auswirkungen der eingeführten Rückkopplung	81
3.3.3.3	Exemplarische Anwendung des Verkehrsmodells	85
4	Verkehrsmodellierung des verteilten Echtzeitdienstes Netzspiele	89
4.1	Modellierung	89
4.1.1	Einordnung	89
4.1.2	Eingrenzung	93
4.1.3	Dienstgüte-Metriken für Reaktionsspiele	94
4.1.4	Bestehende Verkehrscharakterisierungen	98
4.1.5	Client/Server-Modellierung	100
4.2	Empirische Bestimmung der Modellparameter	101
4.2.1	Messung	102
4.2.2	Parameterbestimmung	103
4.2.2.1	Paketgrößen des Servers	106
4.2.2.2	Paket-Zwischenankunftszeiten des Servers	107
4.2.2.3	Paket-Zwischenankunftszeiten der Clients	108
4.2.2.4	Paketgrößen der Clients	109
4.2.2.5	Teilnehmerverhalten	109
4.2.3	Modellparameter	112
4.3	Bewertung des Spieleverkehrsmodells	114

4.3.1	Bewertungsziel und Randbedingungen	114
4.3.2	Untersuchung der Verkehrscharakteristik	116
4.3.3	Dienstgütebewertung	119
5	Zusammenfassung und Ausblick	123
	Literaturverzeichnis	127
	Quellen im Internet	139
A	Ausgewählte Verteilungen zur Approximation von Modellparametern.	141
A.1	Hyperexponentielle Funktion n-ter Ordnung: $H_n(\pi, m_i)$	141
A.2	Lognormal-Verteilung: $L(\mu, \sigma)$	142
A.3	Pareto-Verteilung: $P(\alpha, k)$	143
A.4	Weibull-Verteilung $W(\alpha, \beta)$	144
A.5	Extremwert-Funktion (a, b)	145
B	Parameter der Verkehrsmodelle.	146
B.1	WWW-Verkehrsmodell	146
B.2	Netzspiele-Verkehrsmodell	148
C	Approximation von kombinierten Verteilungen	149
C.1	Weibull/Pareto	149
C.2	Lognormal/H2	151
C.3	Weibull / Lognormal	152

Abkürzungen

CCDF	Complementary Cumulative Distribution Function (komplementäre Verteilungsfunktion)
CS	Counter Strike, Half Life based FPS
FPS	First Person Shooter (Reaktionsspiel)
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IAT	Interarrival Time (ZAZ)
IP	Internet Protocol
IPX	Internetwork Packet Exchange (Novell)
LAN	Local Area Network
MMOG	Massive Multiplayer Online Game
PC	Personal Computer
PDF	Probability Density Function (Wahrscheinlichkeitsverteilungsfunktion)
PDU	Protocol Data Unit
QoS	Quality of Service
RTP	Real Time Protocol
RTS	Real Time Strategy Games
RTT	Round Trip Time
RWLG	Reaktiver WWW-Lastgenerator
SDU	Service Data Unit
SURGE	Scalable URL Reference Generator (Barford, Crovella)
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WSD	Web Server Delay
WWW	World Wide Web
ZAZ	Zwischenankunftszeit (IAT)

Formelzeichen

Allgemeine Formelzeichen

$F(x)$	Verteilungsfunktion $F(x) = P\{X \leq x\}$
$F^C(x)$	komplementäre Verteilungsfunktion $1 - F(x) = P\{X > x\}$
$f(x)$	Verteilungsdichtefunktion $f(x)$
$X(t)$	Stochastischer Prozeß
$\{X_t\}$	zeitdiskreter stochastischer Prozeß (Zeitreihe)
\bar{x}	Mittelwert der Zufallsvariablen X
σ, σ^2	Standardabweichung, Varianz
$\gamma(\tau)$	Autokovarianz in Abhängigkeit der Verschiebung τ
$\rho(\tau)$	Autokorrelationsfunktion in Abhängigkeit der Verschiebung τ

WWW-Verkehrsmodell

t_1	Antwortzeit (Ladedauer eines primären WWW-Abrufs)
t_2	Ladedauer eines sekundären WWW-Abrufs (eingebettetes Objekt)
t_A	Zwischenankunftszeit von primären HTTP-Anforderungen (Klicks)
t_B	Betrachtungsdauer einer fertig geladenen WWW-Seite
t_G	Geduldsspanne des WWW-Nutzers
t_{RTT}	Umlaufzeit (<i>Round Trip Time</i>)
t_W	absolute WWW-Seitenladedauer
t_{WSD}	Verzögerung des WWW-Servers (<i>Web Server Delay</i>)

r	Bandbreite (Bitrate) der Netzanbindung
s_2	Größe eines eingebetteten Objekts
s_W	Größe der WWW-Seite (Summe aller Objektgrößen)

Netzspiele-Verkehrsmodell

A_H	Verkehrsangebot des Hintergrundverkehrs
d, d'	Latenzzeit (<i>delay</i>), angepaßte Latenzzeit
h, h_S, h_C	Paket-Übertragungsdauer, gesendet von Server bzw. Client
N	Anzahl der Clients (Teilnehmer)
p	Paketverlustwahrscheinlichkeit
Q	Puffergröße
r, r_S, r_C	Paketrate (hier speziell: Senderate) von Server bzw. Client
T	Paketzwischenankunftszeit (hier am Empfänger)
T_S	Server-Verarbeitungszeit
τ	Signallaufzeit
ρ	Linkauslastung

Kapitel 1

Einleitung

1.1 Kommunikation über paketvermittelnde Netze

Der technische Aufbau von Kommunikationsnetzen hat sich in den letzten Jahrzehnten grundlegend verändert. Moderne Netze haben nicht mehr viel mit den klassischen Telefonnetzen gemeinsam, welche nach dem Prinzip der Durchschaltvermittlung arbeiten, d.h. daß für die Gesprächspartner eine dedizierte Leitung oder ein Kanal exklusiv reserviert wird. Selbst wenn dieser Kanal aufgrund häufiger Gesprächspausen nicht gut ausgenutzt wird, ist er belegt und kann nicht zur Übertragung anderer Kommunikationswünsche mitverwendet werden, die ggfs. auf eine freie Leitung warten müssen. Die Auslegung von solchen Netzen mittels klassischer Methoden der Nachrichtenverkehrstheorie (Erlang-Verlust-Formel) ist äußerst erfolgreich und führt zu sehr hoher Verfügbarkeit von Telefonnetzen. Diese hohe Verfügbarkeit, schnelle Reaktions- und Antwortzeiten sowie die gleichbleibende Gesprächsqualität von Telefonverbindungen sind heute zu einer Selbstverständlichkeit geworden.

Mit dem Aufkommen von Datenkommunikationsdiensten und der Vernetzung von Rechnern (insbesondere das Internet) wurden Kommunikationsnetze nach dem Prinzip der Paketvermittlung aufgebaut. Dabei werden Kommunikationsströme digitalisiert und auf Nachrichtenblöcke begrenzter Länge (Pakete) verteilt, die dann zur unabhängigen Übermittlung an das Netz gegeben werden. Das Kommunikationsnetz kann nun seine Kanalabschnitte mit der Vielfalt aller zu übertragenden Pakete deutlich besser auslasten. Der Empfänger muß die Pakete entgegennehmen und den ursprünglichen Kommunikationsstrom wieder herstellen. Dieses Prinzip erlaubt eine deutlich effektivere Nutzung der Infrastruktur, führt aber auch neue Probleme ein. Zunächst wird eine gewisse zeitliche Verzögerung bei der Digitalwandlung, Paketierung sowie der Pufferung und Analyse der Pakete in jedem Netzknoten in Kauf genommen. Außerdem können Pakete bei Überlast verloren gehen, so daß Schutz- und Korrekturmechanismen notwendig sind, um eine vergleichbar hohe Dienstgüte wie bei den Telefonnetzen zu erreichen. Da aber im Allgemeinen nicht nur ein Sprachdienst sondern eine große Vielzahl an Kommuni-

kationsdiensten durch dasselbe Netz vermittelt werden (Fax, Email, Dateiübertragung, Rechnerfernzugriff, ...), sollte eine ebenso verallgemeinerte Definition für den Begriff der Dienstgüte getroffen werden. Die aktuellen Eigenschaften des Netzes (Netzgüte) sind allerdings nicht mehr einfach auf die Anforderungen der Dienste abbildbar und die aktuelle Dienstgüte zwischen den Anwendungen zweier Endgeräte kann vom Netzbetreiber in der Regel nicht direkt beobachtet werden.

Für die Auslegung von Paketvermittlungsnetzen mit einer solchen Vielzahl an unterstützten Diensten sieht man sich zudem mit einem äußerst komplexen Verkehr konfrontiert. Die Überlagerung der unterschiedlichen Dienste führt zu einer enorm schwankenden Verkehrslast, die durch technisch bedingte, vom Netzzustand abhängige Rückkopplungen (Flußkontrolle) zusätzlich beeinflusst werden. Damit ist eine zuverlässige Vorhersage des Verkehrsverhaltens auch kurzfristig nicht mehr möglich. Um dennoch Aussagen zur Leistungsfähigkeit von Paketvermittlungsnetzen und zur Optimierung ihrer Parameter treffen zu können, wird das Verhalten abstrahierter Netzmodelle in Simulationen untersucht. Netzmodelle werden dabei durch Verkehrsmodelle mit unterschiedlichen Verkehrsmustern belastet. Diese Verkehrsmodelle können auf die Nachbildung des Gesamtverkehrs oder auf die Nachbildung von Teilverkehren ausgelegt sein. Je nach dem wie die Teilverkehre definiert werden, kann ihre Modellierung deutlich einfacher oder genauer erfolgen als die Modellierung eines komplexen Gesamtverkehrs.

Im Rahmen dieser Arbeit werden Teilverkehre anhand ihrer Dienstzugehörigkeit spezifiziert und modelliert. Durch Dienst- und Protokollanalyse, Messung und statistische Auswertung des Verkehrs können invariante Parameter benannt werden, die den Teilverkehr weitgehend unabhängig von Ort und Zeitpunkt der Messung beschreiben. Diese Invarianten werden zur Modellierung der dienstspezifischen Verkehre auf Paketebene eingesetzt, die in einem gewissen Rahmen entsprechend unabhängig von den Umständen einer Simulation zum Einsatz kommen können. Die Messung und Charakterisierung von dienstspezifischen Verkehren ist daher als zentrale Methode der Verkehrscharakterisierung zu betrachten.

In dieser Arbeit wurden zwei Dienste zur exakten Charakterisierung und Modellierung ausgewählt. Zum einen wird das Verhalten des World Wide Web-Verkehrs betrachtet, da dieser Dienst für einen Großteil des heutigen Datenaufkommens verantwortlich ist und weil er äußerst vielseitig auf Firmen- und Privatrechnern, über drahtlose Zugangsnetze und auch auf mobilen Endgeräten eingesetzt wird. Zum anderen wurden Netzspiele als interessanter Vertreter interaktiver verteilter Dienste identifiziert und zur weiteren Charakterisierung und Modellierung ausgewählt, weil sie einerseits besonders hohe Echtzeitanforderungen und andererseits großes Marktpotential aufweisen.

1.2 Übersicht über die Arbeit

Die Arbeit gliedert sich im Wesentlichen in zwei Hauptkapitel (Kapitel 3 und 4), welche jeweils die Verkehrsmodellierung, Parametrisierung und Bewertung für die beiden ausgewählten Dienste WWW und Netzspiele behandeln. Im Rahmen der Beschreibung der gewählten Verkehrsmodellierung und ihrer Parameter (Kapitel 3.1 und 4.1) werden einleitend grundlegende Dienstmerkmale und daraus abgeleitete Bewertungskriterien angeführt. Außerdem werden dort bestehende Modellierungen der betreffenden Dienste vorgestellt. Für die Parametrisierung der Modelle (Kapitel 3.2 und 4.2) wird auf Messungen und die entsprechende statistische Auswertung und Beschreibung der als Modellparameter ausgesuchten Verkehrsgrößen zurückgegriffen. Die Bewertung der Modelle (Kapitel 3.3 und 4.3) erfolgt abschließend anhand von Simulationen.

Den Hauptkapiteln ist ein Grundlagenkapitel (Kapitel 2) vorangestellt, in welchem zunächst das Umfeld der betrachteten Dienste beschrieben wird (Kapitel 2.1 und 2.2). Die Aspekte der Verkehrsmessung werden in Kapitel 2.3 behandelt und in Kapitel 2.4 werden schließlich die Grundlagen der Verkehrsbeschreibung dargestellt, auf welchen die Hauptkapitel aufsetzen.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick (Kapitel 5) auf die weitere Entwicklung der betrachteten Arbeitsgebiete.

Kapitel 2

Grundlagen zu Internetverkehr und Verkehrsbeschreibung

2.1 Paketbasierte Kommunikation im Internet

2.1.1 Entwicklung

Das Internet beruht auf dem experimentellen Weitverkehrs-Kommunikationsnetz ARPANET, das 1969 von der amerikanischen DARPA (*Defense Advanced Research Projects Agency*) ins Leben gerufen wurde. Das Ziel dieses Forschungsnetzes, war die Erprobung des Konzepts der Paketvermittlung für Datennetze, das 1964 von Paul Baran (*hot potato routing* oder *distributed adaptive message block switching*) [13] von der Rand Corporation, einer Ideenschmiede des amerikanischen Militärs und unabhängig davon 1965 von Donald Watt Davies [86,140] vom NPL (National Physical Lab, UK) vorgeschlagen wurde.

Das ARPANET verband zunächst Universitäten und Forschungseinrichtungen mit Hilfe des Protokolls NCP (*Network Control Protocol*). Parallel dazu wurde von Robert Kahn und Vint Cerf die TCP/IP Protokoll-Suite entwickelt, um viele proprietäre Rechnernetze (DEC, HP, IBM) über eine offene Netzarchitektur so zu verbinden, daß alle beteiligten Rechner ohne Kenntnis der Topologie miteinander kommunizieren konnten [43]. Durch diese dezentrale Organisation der Kommunikation wurde das Internet relativ robust gegen Ausfälle, da unabhängige Vermittlungsknoten (Router) die Aufgaben anderer Knoten übernehmen konnten. Das ARPANET wurde 1983 auf TCP/IP umgestellt. Die Koordination der Organisation und der Weiterentwicklungen des Internet wird heute durch die *Internet Engineering Task Force* (IETF) [34,173] wahrgenommen, einer offenen Vereinigung von Entwicklern, Betreibern, Herstellern und Forschern.

Insbesondere die Erfindung des World Wide Web (WWW) 1989 durch Tim Berners-Lee schuf die Grundlage für ein weltweites Informationsnetz, das durch die Entwicklung von leicht zu bedienenden WWW-Clients (Browsern) auch für Nicht-Experten zugänglich wurde. Damit wurde das Internet schnell auch zu einer kommerziellen Plattform für unterschiedlichste Informationsdienste und führte sein exponentielles Wachstum fort. Mitte 2002 wurden über 160 Millionen an das Internet angeschlossene Computer gezählt [175], wovon etwa 37 Millionen WWW-Inhalte bereithielten [177]. Die Kommunikation im WWW machte zu dieser Zeit etwa 75% des gesamten Internet-Verkehrs aus, gefolgt von E-Mail-Verkehr mit 15% Anteil [43].

Die Grundlage dieses Erfolgs liegt in der offenen und flexiblen Architektur des Internet, auf welche neue Dienste aufsetzen können. Im folgenden sollen die grundlegenden Charakteristiken dieser Architektur und die Vielfalt der darauf aufsetzenden Dienste aufgezeigt werden.

2.1.2 Funktionsweise

Anstatt eine durchgehende und exklusive Verbindung für eine Ende-zu-Ende-Kommunikation zu schalten, wie das bei klassischen leitungsvermittelnden Netzen getan wird, werden bei paketvermittelnden Netzen wie dem Internet digitalisierte Informationen auf Pakete verteilt, die dem Netz zur Übertragung an das Ziel übergeben werden. Netz-Ressourcen können nun durch Pakete unterschiedlicher Kommunikationsbeziehungen gemeinsam genutzt werden, so daß ein Multiplexgewinn erzielt werden kann. Die korrekte Wiederherstellung der zusammenhängenden Information aus den Teilm Informationen der Pakete erfordert zahlreiche Verarbeitungsschritte.

Um eine Verringerung der Komplexität und eine höhere Flexibilität zu erreichen, wird die Kommunikation in mehreren Schichten organisiert. In jeder Schicht sind Teilfunktionen eines Kommunikationsvorgangs realisiert, die als Dienstleistung der jeweils übergeordneten Schicht über sog. Dienstzugangspunkte zur Verfügung gestellt wird. Die Realisierungen einzelner Schichten eines End- oder Transitsystems werden Instanzen genannt und nehmen Dienste der untergeordneten Schichten in Anspruch. Die Kommunikationsabläufe zwischen Instanzen der gleichen Schicht in verschiedenen Systemen sind über Peer-to-Peer-Protokolle festgelegt.

Diese Architektur wurde durch die ISO (*International Standards Organization*) im ISO/OSI-Basisreferenzmodell formuliert, um eine Grundlage für Normen zur offenen Kommunikation (*Open Systems Interconnection*) zu schaffen (DIN ISO 7498 bzw. CCITT/ITU-T X.200). Die Architektur des Internet folgt ebenfalls dem Prinzip der Schichtung und ist auf die Schichten des OSI-Modells abbildbar (Bild 2.1).

Das TCP/IP-Referenzmodell gibt keinen verbindlichen Rahmen für die Umsetzung der untersten Schicht (*Host-to-network*) vor, jedoch sollen hier wie im ISO/OSI-Basisreferenzmodell die Aufgaben der Bitübertragungsschicht (Übertragung einzelner Bits auf einem Kanal) und

		OSI	Internet
Verarbeitung	7	Application	Application
Darstellung	6	Presentation	
Kommunikationssteuerung	5	Session	
Transport	4	Transport	Transport
Vermittlung	3	Network	Internet
Sicherung	2	Data Link	Host-to-network
Bitübertragung	1	Physical	

Bild 2.1: Das ISO/OSI Basisreferenzmodell und das TCP/IP-Referenzmodell (nach [148])

der Sicherungsschicht (Kanalzugriff, Rahmenerzeugung und -wiederherstellung, Fehlerkorrektur) wahrgenommen werden.

Die Internet-Schicht ist für die Übertragung von Paketen bis zum Zielsystem zuständig, schließt also deren Adressierung, Wegesuche und Vermittlung ein und entspricht damit der Vermittlungsschicht des ISO/OSI-Basisreferenzmodells. Während in letzterem aber sowohl verbindungslose als auch verbindungsorientierte Dienste vorgesehen sind, unterstützt die Internet-Schicht mit IP (*Internet Protocol*) nur die verbindungslose Paketvermittlung, d.h. Paketverluste werden nicht korrigiert. Eine zuverlässige Kommunikation muß daher in höheren Schichten realisiert werden. In der Transport-Schicht wird dies mit TCP (*Transmission Control Protocol*) erreicht, das durch Einsatz von Fehlererkennung, -korrektur und Reihenfolgekontrolle gesicherte Ende-zu-Ende-Verbindungen realisiert. Außerdem enthält es Mechanismen zur Flußkontrolle, um die Senderate bei Überlast-Situationen im Netz zu drosseln. Alternativ bietet die Transport-Schicht mit UDP (*User Datagram Protocol*) auch einen ungesicherten und verbindungslosen Kommunikationsdienst an, der im wesentlichen einen direkten Zugriff auf das IP-Protokoll erlaubt.

Die Kommunikation zwischen zwei Endgeräten findet in der Regel über mehrere Netzknoten statt, welche empfangene Informationen zum nächsten Kanalabschnitt weiterleiten. Je nach Aufgabenstellung werden zur Netzkopplung verschiedene Arten von Netzknoten eingesetzt. Bild 2.2 zeigt exemplarisch die Kopplung zweier Endsysteme über die unterschiedlichen Arten der Netzknoten. Für eine einfache und transparente Kopplung zweier gleichartiger Netze genügt beispielsweise ein *Repeater*. Wenn eine Nachricht an ein unbekanntes Netz adressiert ist, sorgt ein *Router* auf der Vermittlungsschicht für die korrekte Weiterleitung und wenn

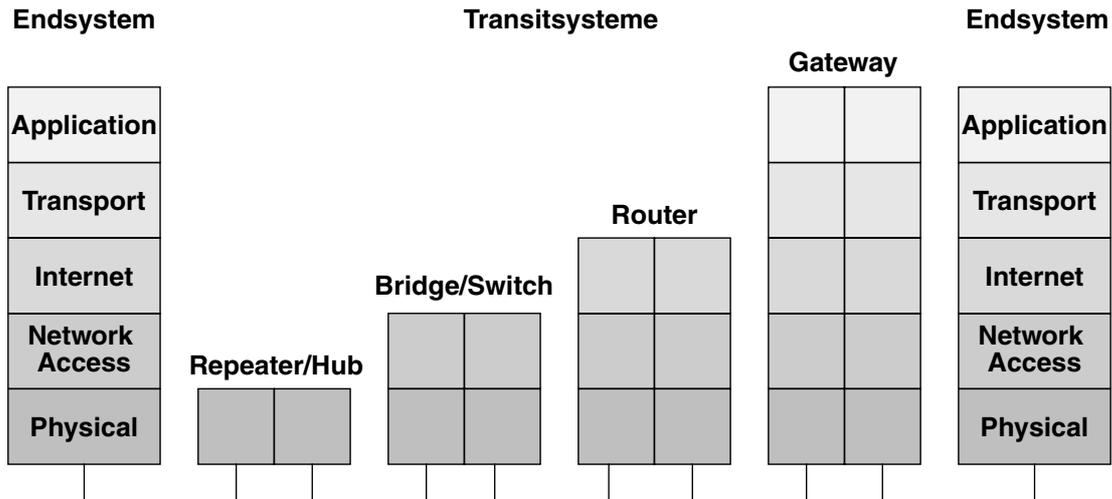


Bild 2.2: Netzkopplungen auf unterschiedlichen Schichten

Nachrichten von bestimmten Anwendungen ausgewertet werden sollen, wie beispielsweise bei *Firewalls* oder *Proxy-Caches*, muß ein *Gateway* den ankommenden Paketstrom bis zur Transport- oder Anwendungsschicht zusammensetzen, um entsprechende Funktionen ausführen zu können. In den Netzknoten sind alle Protokoll-Instanzen bis zur jeweils erreichten Schicht instanziiert, so daß die Kommunikationsvorgänge bis zu dieser Schicht abgeschlossen werden. Ein Netzknoten hat folglich keine Kenntnis der Kommunikationsbeziehungen auf höheren Schichten.

2.1.3 Internet-Protokolle

Die wichtigsten Protokolle, die zusammen mit TCP/IP zum Erfolg des Internet beigetragen haben, sind das *Border Gateway Protocol* (BGP) zum Austausch von Erreichbarkeits- und Routing-Informationen, das *Domain Name System* (DNS) zur Umsetzung von Adressnamen auf die numerischen IP-Adressen, die E-Mail Protokolle (POP3, IMAP und SMTP), Protokolle für Fernzugriff auf andere Rechner (TELNET), das *File Transfer Protocol* (FTP) zur Übertragung von Dateien und *Hyper Text Transfer Protocol* für das World Wide Web (HTTP) [43].

An dieser Stelle sollen die Protokolle IP, TCP und UDP etwas genauer betrachtet werden, da sie zur besonderen Charakteristik des zu modellierenden Paketverkehrs beitragen. Für genauere Erläuterungen sowie für eine umfassende Übersicht über TCP/IP-Protokolle sei beispielsweise auf [147, 148] verwiesen.

2.1.3.1 Internet Protocol (IP)

Die Hauptaufgabe des IP-Protokolls ist die Vermittlung von Paketen bis zum Empfänger. Die wichtigsten Felder des in Bild 2.3 gezeigten IP-Paketkopfes enthalten demnach die IP-Adressen für Sender und Empfänger (*Source Address*, *Destination Address*). Die *Header-Länge*

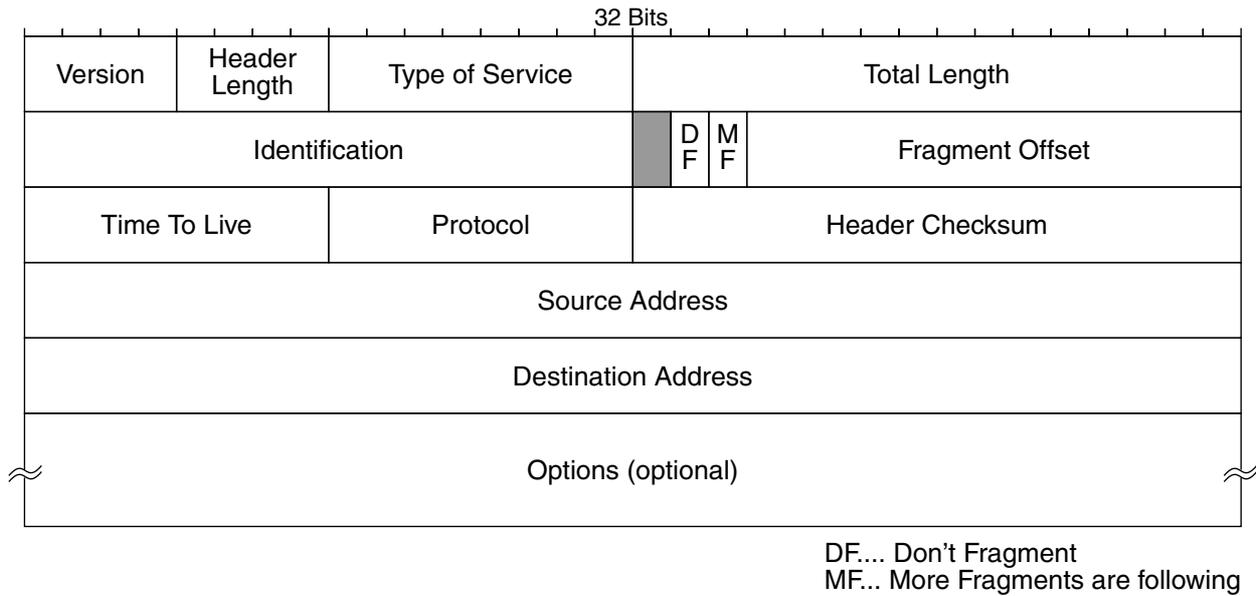


Bild 2.3: Die Kopffelder des IP-Protokolls

beträgt abhängig von den enthaltenen Optionen zwischen 20 und 60 Bytes. IP-Pakete selbst können eine maximale Länge von 64 kBytes haben, werden aber in der Regel auf die Größe der maximalen Paketgröße der Übertragungssysteme auf Sicherungsschicht beschränkt – meist wird die MTU (*maximum transmission unit*) von Ethernet mit 1500 Bytes verwendet.

Weitere Protokolle der Internet-Schicht sind ICMP, ARP, RARP und BOOTP. ICMP (*Internet Control Message Protocol*) dient zur Signalisierung von Fehlerzuständen und Adressinformationen sowie zum Testen von Erreichbarkeit. Das Protokoll ARP (*Address Resolution Protocol*) erlaubt die Umsetzung von IP-Adressen auf MAC-Adressen und die Protokolle RARP (*Reverse ARP*) und BOOTP (*BOOTstrap Protocol*) unterstützen die Zuweisung von IP-Adressen zu einem neu startenden Rechner. Weiterhin zählen die Routing-Protokolle OSPF und BGP zur Internet-Schicht.

2.1.3.2 Transmission Control Protocol (TCP)

Auf der Transportschicht wird zumeist das *Transmission Control Protocol* (TCP) eingesetzt, dessen Kopffelder in Bild 2.4 gezeigt werden. TCP ergänzt den unzuverlässigen Übertragungsdienst des IP-Protokolls so, daß alle Nachrichten vollständig und in richtiger Reihenfolge ausgeliefert werden. Um diese Funktionalität zu erreichen, unterstützt TCP eigene Ende-zu-Ende Verbindungen, die durch Port-Nummern bei Sender und Empfänger (*Source Port, Destination Port*) identifiziert werden – dies entspricht auch den Dienstzugangspunkten der Transportschicht. TCP-Segmente, die solchermaßen einer Verbindung zugeordnet werden, enthalten außerdem Reihenfolgennummern (*Sequence Number*), die eine Einordnung in den zu übertragenen Byte-Strom ermöglichen. Bei TCP werden keine Segmente sondern Bytes numeriert, so daß im Kopffeld entsprechend die Nummer des ersten Daten-Byte angegeben wird. Durch ein

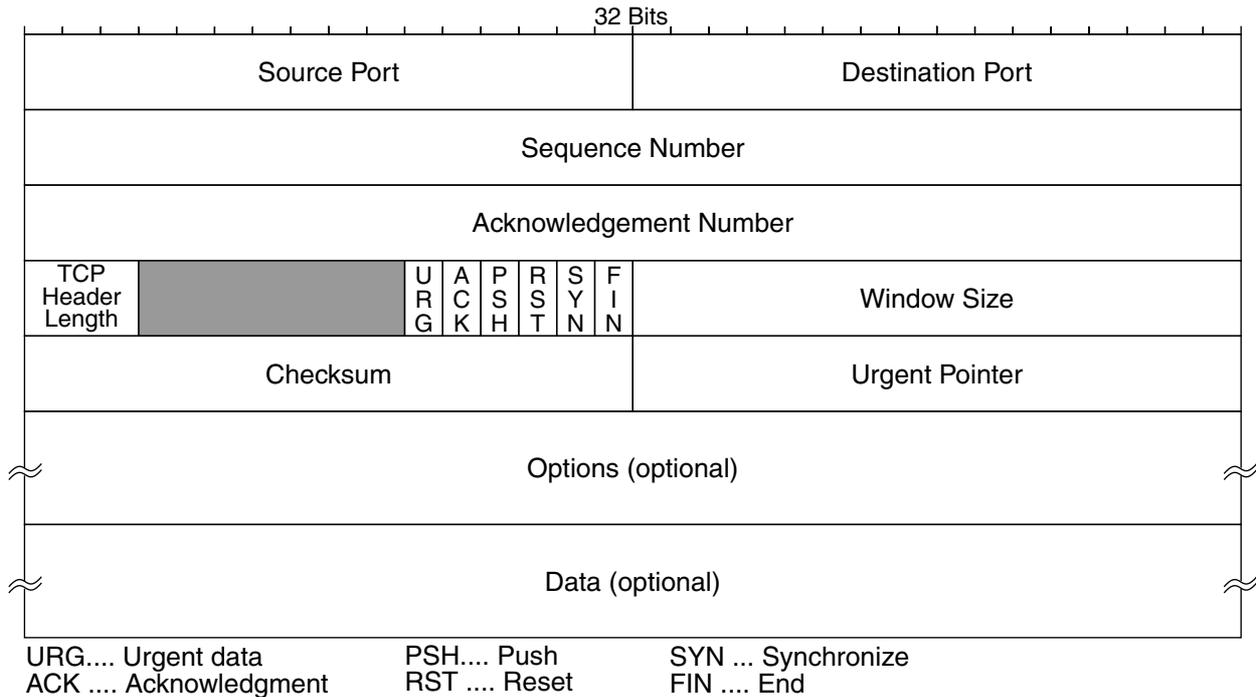


Bild 2.4: Die Kopffelder des TCP-Protokolls

eigenes Feld für eine Quittierungsnummer (*Acknowledgement Number*) können Quittungen per *Piggy Backing* gegeben werden, also zusammen mit einem normalen Daten-Segment. Bleibt eine Quittierung nach einem Paketverlust aus, wird dies durch eine Zeitüberwachung (*retransmission timer*) festgestellt. Da TCP ein Ende-zu-Ende-Protokoll für ein globales Kommunikationsnetz ist, kann keine genaue Aussage zu den zu erwartenden Umlaufzeiten getroffen werden, von denen aber die Dauer einer effektiven Zeitüberwachung abhängt. Auf Basis der Dauer jeder Quittierung führt TCP daher eine Schätzung der Umlaufzeit durch mit der wiederum die Dauer der Zeitüberwachung bestimmt wird.

Eine neue TCP-Verbindung muß auf Sender- und Empfängerseite zunächst in einen definierten Startzustand gebracht werden, d.h. es muß ein Verbindungsaufbau erfolgen. Wenn ein Endsystem TCP-Verbindungen akzeptiert, kann es von einem anderen Endsystem eine Verbindungsaufbau-Nachricht, die durch ein gesetztes SYN-Flag im Segmentkopf gekennzeichnet ist, empfangen. Die übertragene Reihenfolgennummer wird gespeichert und in der Antwort quittiert, die selbst wiederum eine Start-Reihenfolgennummer enthält und durch die Flags SYN und ACK gekennzeichnet ist. Erst mit der Quittung (ACK) durch den Initiator ist der Vorgang abgeschlossen. Bild 2.5 verdeutlicht den sogenannten 3-Wege-Handshake. Der Verbindungsabbau erfolgt für jede Übertragungsrichtung separat, d.h. der Datenfluß von B nach A kann in Bild 2.5 rechts auch nach Quittierung der ersten FIN-Nachricht weiterlaufen.

Neben der Bereitsstellung eines zuverlässigen Übertragungsdienstes übernimmt TCP auch Aufgaben der Flußkontrolle. Es verwendet einen Fenstermechanismus, der nur eine begrenzte Anzahl unquittierter Bytes zuläßt (*sliding window*). In jedem Segmentkopf wird die Fenster-

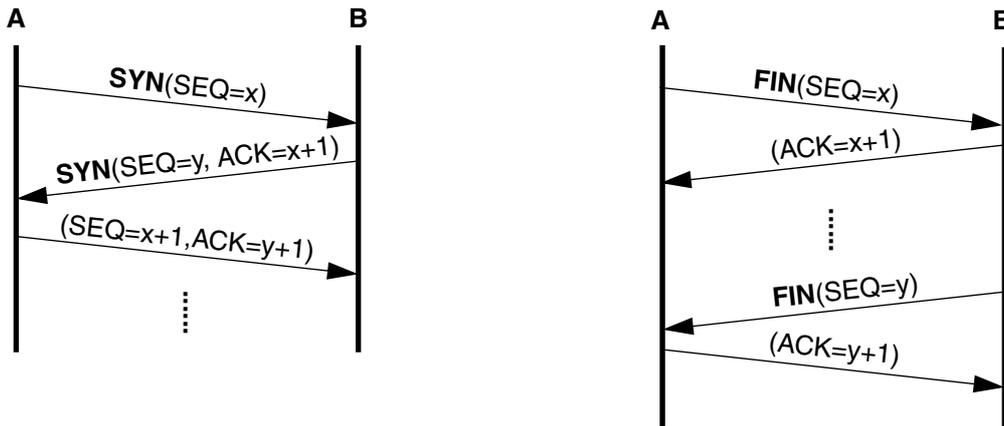


Bild 2.5: TCP-Verbindungsaufbau und -abbau (nach [148])

größe des Senders vom Empfänger vorgegeben (*window size*) – so kann ein Empfänger ohne ausreichenden Pufferplatz die Senderate drosseln oder auch ganz stoppen. Der Mechanismus kann zu einem Leistungseinbruch führen, wenn der Empfängerpuffer voll ist und der konsumierende Prozeß die Daten nur byteweise aus dem Puffer ausliest. Nach jedem ausgelesenen Byte würde der Empfänger dann die neue Fenstergröße von 1 Byte signalisieren, so daß der Sender eine Serie kleiner Segmente mit jeweils großem TCP/IP-Kopfteil aussendet (*silly window syndrome*). Durch Verhinderung der Signalisierung von zu kleinen Fenstergrößen kann das Problem behoben werden (Clark [50]). Ein ähnlicher Effekt kann auch auf Senderseite auftreten, wenn der erzeugende Prozeß Daten mit geringer Rate an TCP übergibt. Da TCP selbst den Aussendezeitpunkt bestimmen muß, kann es auch hier zu einer Serie kleiner ineffizienter Segmente kommen. Durch die Verhinderung von mehr als einem kleinen unquittierten Segment werden genügend Daten für eine effizientere Segmentgröße gesammelt (Nagle [129]).

TCP unterstützt außerdem einen zweiten Flußkontrollmechanismus (*congestion control*). Während der oben beschriebene Mechanismus bei Engpässen auf Seite des Empfängers greift, soll der zweite Mechanismus durch registrierte Paketverluste Überlastzustände im Netz erkennen und entsprechend die Senderate drosseln. Dies wird durch ein zweites Sendefenster (*congestion window*) erreicht – die tatsächlich verwendete Fenstergröße richtet sich nach dem kleineren Wert der beiden Fenster. Nach dem Verbindungsaufbau wird das *congestion window* so initialisiert, daß zunächst nur ein Segment gesendet werden kann. Mit jeder fehlerfreien Quittung wird das Sendefenster verdoppelt (*Slow Start*-Phase), bis ein Schwellwert erreicht wird, nach welchem das Sendefenster nur noch linear um eine maximale Segmentgröße erhöht wird (*congestion avoidance*-Phase). Stellt eine Zeitüberwachung fest, daß eine erwartete Quittung ausbleibt, wird angenommen, daß ein Segment aufgrund einer Überlastsituation im Netz verworfen wurde. Der Sender drosselt dann sofort seine Rate, indem er das *congestion window* auf eine Segmentgröße zurücksetzt und den Schwellwert halbiert, so daß die *congestion avoidance*-Phase früher beginnt. Ein TCP-Sender tastet sich somit ständig an eine optimale Senderate heran.

Insbesondere für die TCP-Flußkontrollmechanismen wurden zahlreiche Verbesserungen und Ergänzungen vorgeschlagen, die zum Teil in verschiedenen Varianten des TCP-Protokolls umgesetzt wurden. Die Leistungsfähigkeit wurde zunächst durch eine geringere Wartezeit vor einer Wiederholung von offensichtlich verlorenen Segmenten erhöht (*Fast Retransmit*). Geht ein einzelnes Segment verloren, können die nachfolgenden Segmente vom Empfänger nicht quittiert werden – stattdessen wird die zuletzt ausgesandte Quittierung wiederholt. Der Sender trifft die Entscheidung für ein wiederholtes Aussenden eines Segments vor Ablauf der normalen Zeitüberwachung, wenn er mehrere Quittierungen für das vorhergehende Segment erhalten hat. Diese Variante wird TCP Tahoe genannt [103]. Eine Ergänzung zu diesem *Fast Retransmit*-Algorithmus wurde im meist verwendeten TCP Reno mit dem *Fast Recovery*-Algorithmus [104] eingeführt, der nach einer *Fast Retransmit*-Wiederholung den Sender nicht zurück in die *Slow Start*-Phase, sondern gleich in die *congestion avoidance*-Phase springen läßt. Die Flußkontrollmechanismen sind ausführlicher in RFC 2581 [6] beschrieben. Es gibt sehr viele weitere Vorschläge zur Verbesserung der Leistungsfähigkeit wie beispielsweise SACK (*selective acknowledgements*) [73, 81, 120], New Reno [80, 73] oder TCP Vegas [31].

Insbesondere bei kurzen Verbindungen, wie sie häufig bei HTTP-Anforderungen vorkommen, erreicht TCP nur einen kleinen Datendurchsatz, wenn es beendet wird, bevor die *Slow Start*-Phase verlassen wird. Mit dem relativ langen Verbindungsaufbau ist bei kurzen Übertragungen außerdem die Wartezeit vergleichsweise lang. Zur Lösung dieses Problems wurde T/TCP zur Diskussion gestellt [32,33], welches eine Pufferung von TCP-Verbindungsdaten zu einem Zielrechner vorsieht, so daß neue Verbindungen mit verkürztem Verbindungsaufbau und ohne *Slow Start*-Phase beginnen können. Alternativ kann eine effektivere Nutzung von TCP auch einer übergeordneten Kommunikationssteuerung überlassen werden, welche die TCP-Verbindungen verwaltet und bei Bedarf offen lassen und wiederverwenden kann (*keep alive*) oder mehrere Übertragungen über dieselbe TCP-Verbindung parallel abwickeln kann (*pipelining*).

2.1.3.3 User Datagram Protocol (UDP)

Da die Sicherung einer Verbindung durch TCP nur Ende-zu-Ende erfolgt, also bei Wiederholungen die komplette Umlaufzeit gewartet werden muß bevor auch nachfolgende und bereits empfangene Segmente an die Anwendung ausgeliefert werden, ist das Protokoll beispielsweise für echtzeitkritische Kommunikation ungeeignet. Mit UDP (*User Datagram Protocol*) ist auf der Transportschicht auch ein verbindungsloses Protokoll definiert, das auf aufwändige Sicherungsmechanismen und Flußsteuerung verzichtet. Aus den in Bild 2.6 gezeigten Kopffeldern ist bereits die geringe Funktionalität ersichtlich – in den 8 Bytes des Kopfes sind lediglich je

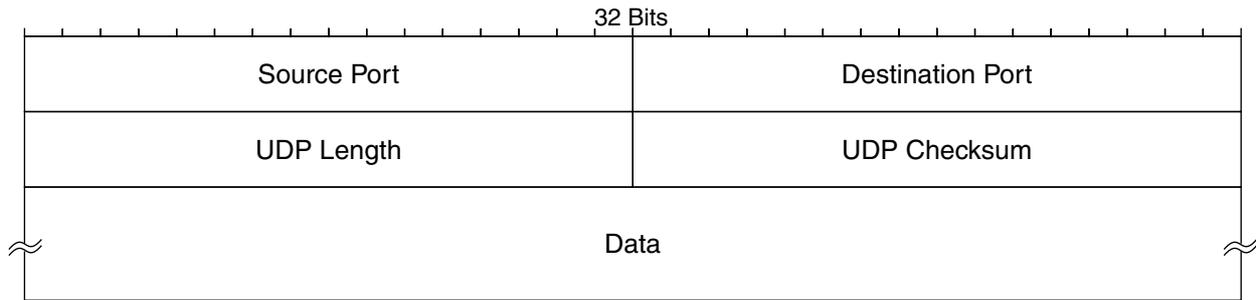


Bild 2.6: Die Kopffelder des UDP-Protokolls

eine Kennung für den Dienstzugangspunkt auf Sender- und Empfängerseite, die Segmentlänge sowie eine Prüfsumme für den Kopf vorgesehen.

2.2 Dienste

2.2.1 Dienstverständnis

Der Dienstbegriff wurde bereits im Rahmen des ISO/OSI-Basisreferenzmodells in Abschnitt 2.1.2 eingeführt. Unter einem Dienst wird die Bereitstellung einer Kommunikationsfunktion für die übergeordnete Schicht über einen Dienstzugangspunkt verstanden. So stellt beispielsweise das *Internet Protocol* IP den Instanzen der Transportschicht den Dienst der ungesicherten Paketvermittlung zum Zielknoten bereit. Die entsprechende Instanz wird am Zielknoten über den Dienstzugangspunkt identifiziert, der bei IP im Kopffeld "*Protocol*" vermerkt ist.

Im Rahmen dieser Arbeit bezieht sich der Dienstbegriff im folgenden allerdings nicht auf die Dienste der unteren Schichten, sondern auf die Dienste der Anwendungsschicht. Wenn von dienstspezifischem Verkehr gesprochen wird, so ist damit der Anteil des Internet-Verkehrs gemeint, der einem bestimmten Dienst der Anwendungsschicht zugeordnet werden kann, beispielsweise E-Mail-Verkehr oder HTTP-Verkehr.

2.2.2 Internet-Dienste der Anwendungsschicht

UDP und TCP stellen einen ungesicherten bzw. gesicherten Transportdienst für Instanzen der Anwendungsschicht bereit. Beide Protokolle nutzen Port-Nummern zur Kennzeichnung der Dienstzugangspunkte über welche sie von höheren Diensten in Anspruch genommen wurden. Diese Port-Nummern können im Prinzip frei gewählt werden, so daß auch neue Anwendungen ohne großen Aufwand eingeführt werden können. Es gibt aber einige Standarddienste, für welche reservierte Portnummern (*well known ports*) vorgesehen und in RFC1700 [137] beschrieben sind. Zur Veranschaulichung sind in Tabelle 2.1 einige Standarddienste aufgeführt.

Tabelle 2.1: Beispiele für reservierte Portnummern und ihnen zugeordnete Dienste

Port	Dienst
20	FTP-Datenverbindung
21	FTP-Steuerung
23	Telnet
25	SMTP (E-Mail)
53	Domain Name Server
80	WWW-Server
119	Netnews-Protokoll

Die unterschiedlichen Dienste zeichnen sich nicht nur durch unterschiedliche Funktionalität aus, sondern auch durch sehr unterschiedliches Kommunikationsverhalten. Während bei einer Datei-Übertragung mit FTP größere Datenmengen am Stück übertragen werden, werden beim Fernzugriff per Telnet nur einzelne Tastatureingaben, meist nur wenige Zeichen in eigenen TCP-Segmenten versendet. Die Dienste können hinsichtlich ihrer Kommunikationscharakteristik klassifiziert werden, so daß eine Bewertung und Behandlung ihrer Anforderungen vereinfacht wird. Im folgenden wird eine Unterscheidung von interaktiven Kommunikationsdiensten nach der ITU-T-Empfehlung I.211 [100] verwendet¹:

- Dialogdienste (*conversational services*) für Mensch-zu-Mensch- oder Mensch-zu-Maschine-Kommunikation mit echtzeitkritischen Anforderungen. Meist besteht aufgrund hoher Redundanz oder ständiger Neusynchronisierung keine Notwendigkeit, alle Daten vollständig zu übertragen, weshalb in der Regel auf UDP aufgesetzt wird.

Beispiele: VoIP, CSCW, Spiele, (WWW)

- Nachrichtendienste (*messaging services*) für die zeitlich verzögerte Kommunikation zwischen Menschen, bei welcher Nachrichten zwischengespeichert werden müssen. Da Nachrichten fehlerfrei übermittelt werden sollen, wird in der Regel TCP eingesetzt. Andernfalls muß die Anwendung für eine gesicherte Übertragung sorgen.

Beispiele: E-Mail, FTP, News, Diskussionsforen

- Abrufdienste (*retrieval services*) für Informationsdienste, die einem Nutzer bereitgestellte Nachrichten nur auf seinen Abruf hin übermitteln. Obwohl der Dienst interaktiv ist, bestehen nur relativ geringe Echtzeitanforderungen – kürzere Wartezeiten stören die Funktionalität des Dienstes nicht. Typischerweise wird auch hier TCP eingesetzt.

Beispiele: WWW, VoD, FTP, POP-Mail

¹ Es werden auch Verteildienste unterschieden (z.B. Rundfunk), die hier nicht weiter betrachtet werden.

2.2.3 Dienstgüte

Welche Anforderungen an das Kommunikationsnetz für einen Kommunikationsdienst relevant sind und inwiefern sie während der Dienstonutzung erfüllt werden, wird anhand der Dienstgüte beschrieben. Nach der Begriffsdefinition von ITG und ITU-T beschreibt die Dienstgüte die subjektive Zufriedenheit des Dienstonutzers:

ITG-Fachgruppe 5.2.1 [99]: *Gesamtheit der Qualitätsmerkmale eines Kommunikationsnetzes aus der Sicht der Benutzer eines betrachteten Dienstes.*

ITU-T Empfehlung E.800 [100]: *The collective effect of service performances which determine the degree of satisfaction of a user of the service.*

Diese sehr subjektive Definition hilft bei der technischen Bewertung der Dienstgüte und den nötigen Entscheidungsprozessen für eine Bereitstellung zufriedenstellender Dienstgüte zunächst nicht weiter. Im folgenden wird daher gemäß ITU-T I.350 [102] der Begriffs der Dienstgüte nur hinsichtlich an der Nutzerschnittstelle beobachtbarer Leistungsparameter verwendet. Damit kann die Dienstgüte durch eine Reihe von direkt beobachtbaren (primären) Parametern und daraus abgeleiteten Parametern beschrieben werden.

Für IP-Netze sind die Dienstgüteparameter der Transportschicht (nach [148])

- die akzeptable Verbindungsaufbau-Verzögerung,
- die Verbindungsaufbau-Fehlerwahrscheinlichkeit,
- der gewünschte Durchsatz,
- die Ende-zu-Ende-Verzögerung,
- die Nachrichten-Fehlerrate (im Verhältnis zu korrekt übertragenen Nachrichten),
- die Sicherheit vor unauthorisiertem Datenzugriff,
- die Nachrichten-Priorität gegenüber anderen Nachrichten eines Nutzers sowie
- die Verbindungsabbruchwahrscheinlichkeit (*resilience*).

Die tatsächlich vom Nutzer beobachtete Dienstgüte bezieht sich allerdings auf die Leistungsmerkmale des genutzten Dienstes, wie z.B. Zuverlässigkeit und Sicherheit bei der Zustellung von E-Mail-Nachrichten, Antwortzeit bei WWW-Abrufen oder Verzögerung und Sprachaussetzer bei der Sprachkommunikation. Diese Parameter sind zwar an der Schnittstelle des Dienstonutzers messbar, aber diese Schnittstelle ist Teil der Anwendungsschicht und liegt nicht mehr im Beobachtungsbereich eines Netzbetreibers, so daß dieser keine Aussagen zur tatsächlich erlebten Dienstgüte eines Dienstes treffen kann. Nur solche Parameter, die direkt auf Parameter der Transportschicht abgebildet werden, können vom Netzbetreiber auch beobachtet werden. Andere Parameter kann er nur durch Schätzverfahren mit einem gewissen Unsicherheitsfaktor gewinnen.

Die Beschreibung der Dienstgüte erlaubt nur Aussagen zu einer gewünschten oder beobachteten Dienstqualität. Ein Kommunikationsnetz, welches seinen Nutzern eine bestimmte Dienstgüte zusichern können soll, muß auch über Mechanismen verfügen, welche die Kommunikation der Dienste zuverlässig schützen. Dies schließt die Entscheidung, ob neue Kommunikationsbeziehungen akzeptiert werden können und die ständige Überwachung der Einhaltung der gewünschten Dienstgüteparameter aller aktiven Verbindungen ein. Beides erfordert eine genaue Spezifizierung der Dienstgüteparameter einer Verbindung, die in einem Verkehrsvertrag abgelegt wird. Alternativ zu einer solchen zugesicherten Dienstgüte, wie es beispielsweise in ATM-Netzen vorgesehen ist, kann ein Kommunikationsnetz auch nur eine relative Dienstgüte unterstützen – hierbei werden Pakete einer Verbindung mit Prioritäten versehen und relativ zu Verbindungsdaten anderer Prioritäten entsprechend anders behandelt (Beispiel: IP-Erweiterung DiffServ). Herkömmliche IP-Netze verzichten sogar ganz auf die Möglichkeit, feste Zusicherungen zu geben. Die erlebte Dienstgüte hängt dann aber von der Auslastung des Netzes ab.

In IP-Netzen ist die fehlende Unterstützung von Dienstgüte-Garantien nur zum Teil ein Problem für die Nutzer. Bei weniger zeitkritischen Nachrichten- und Abrufdiensten sorgen die TCP-Mechanismen für eine zuverlässige Übertragung – zeitliche Verzögerungen werden in Kauf genommen, halten sich bei typischen Netzauslastungen allerdings in akzeptablen Grenzen. Für die zunehmende Nutzung von Dialogdiensten wäre aber zumindest eine Priorisierung oder gar eine Ressourcen-Reservierung (RSVP, dedizierte IP-Netze) wünschenswert.

2.3 Verkehrsmessung

2.3.1 Einordnung

Verkehrsmessungen werden in Kommunikationsnetzen aus unterschiedlichsten Gründen durchgeführt. Sie dienen zunächst der Feststellung der aktuellen Auslastung der Ressourcen im laufenden Betrieb bzw. der Bestimmung der Netzgüte (Laufzeitmessung, Verlustmessung), um Engpässe und Leistungsreserven zu ermitteln und entsprechend darauf reagieren zu können. Dies kann langfristig durch den Aus- oder Umbau des Netzes geschehen oder kurzfristig durch eine automatische Lastverteilung bei der Pfadsuche (adaptives Routing). In komplexen, dezentral organisierten Netzen, wie dem Internet, werden Messungen auch zur Erfassung der gewachsenen Topologie eingesetzt [35,38,49]. Datenverkehr wird heute aber auch aus organisatorischen und rechtlichen Gründen aufgezeichnet, um Kommunikationsvorgänge nachvollziehbar zu machen oder als Grundlage zur Abrechnung einer Dienstnutzung.

Um Aussagen über zukünftige Anforderungen an Kommunikationsnetze treffen zu können, werden auch Verkehrsmessungen mit dem Ziel der Charakterisierung des Datenverkehrs durchgeführt. Die Analyse der aufgezeichneten Daten ermöglicht im einfachsten Falle, Trends zu erkennen und damit vorausschauende Maßnahmen zu ergreifen. Wenn Prognosen nicht so offensichtlich gestellt werden können, beispielsweise wenn das Verhalten neuer Systemkomponenten zu bewerten oder Dimensionierungsprobleme zu lösen sind, wird ein Modell des betrachteten Systems unter realistischer Last entweder analytisch oder simulativ getestet. Zur Erzeugung dieser Last werden Verkehrsmodelle benötigt, welche die durch Verkehrsmessungen ermittelte Charakteristik des realen Verkehrs nachbilden.

Die enormen Übertragungsraten in modernen Kommunikationsnetzen machen Verkehrsmessungen allerdings zu einer echten Herausforderung. Messungen an Hochgeschwindigkeitsverbindungen von 155Mbit/s (OC3), 622 Mbit/s (OC12), 2.5 Gbit/s (OC48) oder gar 9.6 Gbit/s (OC192) erfordern den Einsatz von Spezialhardware. Solche Messungen können in größerem Umfang nur von Netzbetreibern oder im Rahmen von großen Forschungsprojekten wie CAIDA (*Cooperative Association for Internet Data Analysis*) [166] und NLANR (*National Laboratory for Applied Network Research*) [176] durchgeführt werden. Die Messreihen, die von beiden Organisationen für Forschungszwecke zur Verfügung gestellt werden, sind entsprechend umfangreich – jeden Tag fallen bis zu 3.2 GByte komprimierte Daten an. Allein aufgrund der Datenmenge ist es nicht sinnvoll, allgemeine und vollständige Messungen durchzuführen, aus denen Daten für unterschiedlichste Auswertungen gezogen werden könnten. Messungen werden vielmehr spezifisch auf die Fragestellungen abgestimmt – in der Meßinfrastruktur von CAIDA werden Daten beispielsweise nur acht Mal am Tag für jeweils 90 Sekunden erfaßt, dafür aber langfristig und flächendeckend (in den USA), so daß globale Trends im Internet erkannt werden können – für eine Auswertung von Einzelnutzerverhalten

ist eine solche Messung beispielsweise zu kurz und auch an ungeeignetem Ort. Weiterführende Informationen zu Messungen im Internet können u.a. in [41,46,82,117,127,155] und insbesondere bei [166] und [176] gefunden werden.

2.3.2 Meßziel und Meßmethodik

Die im vorigen Abschnitt aufgeführte Klassifizierung der Meßziele erfordert jeweils eine unterschiedliche Vorgehensweisen bei der Messung. Zunächst müssen die relevanten Meßgrößen identifiziert werden. Diese umfassen Kenn- und Leistungsgrößen des Netzes (Erreichbarkeit, Topologie, Ende-zu-Ende-Verzögerung, Umlaufzeit, Link-Verlustwahrscheinlichkeit, Link-Auslastung) und Charakteristiken der Verkehrslast (Ankünfte und Größen bzw. Dauer von Paketen, Büscheln, Verbindungen, Dialogen, Sitzungen sowie aggregierten Verkehrseinheiten). Daraus können wiederum weitere Meßgrößen abgeleitet werden, wie z.B. der Ort eines Engpasses oder Antwortzeit eines Dialogdienstes.

Aus den Meßgrößen ergibt sich, ob eine passive Messung oder eine aktive Messung nötig ist. Passive Messungen zeichnen den Verkehr bzw. Ressourcenbelegungen auf, ohne selbst Datenverkehr zu erzeugen. Sie werden insbesondere zur Lastcharakterisierung verwendet. Aktive Messungen erzeugen dagegen Testpakete, die in das Netz eingespeist werden und entweder eine Antwort an das Meßgerät erzeugen (z.B. ping) oder von einem Empfänger ausgewertet werden. Eine Auswertung bezüglich der Laufzeit erfordert dabei eine aufwändige Zeit -Synchronisierung bei Sender und Empfänger [161].

Weiterhin muß festgelegt werden, zu welcher Zeit, an welchem Ort und bis zu welcher Protokollschicht bzw. auf welchem Abstraktionsgrad die Messung durchgeführt werden sollte. Meßzeitpunkt und Meßdauer müssen so gewählt werden, daß statistisch relevante Aussagen über die gemessenen Daten getroffen werden können, d.h. die Messung sollte ausreichend viele Meßgrößen enthalten (z.B. Verbindungsdauer). Gegebenenfalls muß die Messung zu definierten Zeitpunkten wiederholt werden, so daß mehrere vergleichbare Meßreihen zur Auswertung bereit stehen. Der Meßort muß natürlich die Sichtbarkeit der zu messende Größe gewährleisten. Ist eine Meßgröße am Meßort nicht direkt beobachtbar, so kann eventuell eine Schätzung basierend auf anderen Meßgrößen erfolgen [111,146].

Die Anforderungen des Meßziels sowie der abgeleiteten Meßumgebung bestimmen zudem, ob leistungsstarke hardware-basierte Meßgeräte benötigt werden oder ob kostengünstigere PC-basierte Software-Messungen mit geringerer Leistungsfähigkeit und Genauigkeit ausreichen. Oft genügt es auch, vorhandene Meßmöglichkeiten in Routern (z.B. Ciscos NetFlow) oder die Log-Funktion in Proxy-Caches oder HTTP-Servern zu verwenden.

Die in dieser Arbeit untersuchten Verkehrsmodelle sollen zur Leistungsbewertung von Kommunikationssystemen herangezogen werden. Der Abstraktionsgrad der Modelle soll die Last-

beschreibung auf Vermittlungsschicht erlauben, d.h. es soll eine Bewertung der Systemlast anhand des Verarbeitungsaufwands einzelner Pakete erfolgen. Hierfür werden Pakete durch ihre Größe und ihre Häufigkeit bzw. -Zwischenankunftszeit sowie durch ihren Routing-Kontext (Ziel- und Quelladresse) beschrieben. Da nur der Verkehr bestimmter Dienste modelliert wird, der für eine allgemeine Lastmodellierung mit anderen Paketverkehrsquellen überlagert werden muß, ist noch eine Zuordnung der Pakete zu ihrem Dialog nötig, beispielsweise durch eine Verbindungskennung (z.B. Port-Nummer).

Die Messungen zur Parametrisierung der Modelle muß daher auf Paketebene erfolgen. Pakete müssen nicht vollständig aufgezeichnet werden sondern es genügt, nur die wesentlichen Informationen des Paketkopfes zu erfassen (Adressen, Verbindungskennung). Für die detaillierte Auswertung von WWW-Verkehr ist allerdings auch die Kenntnis von HTTP-Kopfinformationen nötig. Nachdem HTTP ein Klartextprotokoll ist, das keine Bitcodierung sondern ASCII-Schlüsselworte verwendet, sind HTTP-Protokollköpfe relativ lang. Da die Modelle das Nutzerverhalten auf Sitzungsebene beschreiben sollen, muß die Messung lange genug sein, um mehrere vollständige Nutzersitzungen aufzuzeichnen. Selbst bei intensiv genutzten interaktiven Diensten darf angenommen werden, daß Nutzer im Normalfall einen üblichen Schlafrhythmus haben und eine Sitzung daher kaum länger als 20 Stunden dauert. Der Meßort sollte nahe am zu modellierenden Teilnehmer liegen, also am Endbenutzer. Um möglichst viele Nutzer zu beobachten, empfiehlt sich eine Messung im Zugangnetz eines Internet-Diensteanbieters.

Insbesondere die erfaßten Daten des WWW-Verkehrs enthalten sensible Nutzerinformationen wie Passwörter, besuchte WWW-Adressen oder Sitzungsdauer. Um Mißbrauch auszuschließen, müssen die erfaßten Daten anonymisiert werden, so daß weder Inhalte noch IP-Adressen rekonstruiert werden können. Allerdings sollen der Typ der WWW-Objekte und der Zusammenhang auch von längeren Sitzungen eines Nutzers erhalten bleiben. Für die vorliegende Arbeit wurden IP-Adressen einer Meßreihe durch eine fortlaufende Identifikations-Nummer ersetzt und die Zuordnungstabellen anschließend gelöscht. Die WWW-Adressen und Optionen wurden vollständig gelöscht und lediglich für erkannte Objekttypen wurde stattdessen eine Typ-Kennung eingefügt.

2.3.3 PC-basierte Messung von IP-Paketverkehr

PC-basierte Messungen sind insbesondere aufgrund der guten Verfügbarkeit und der hohen Speicherkapazität von PCs interessant. Im Vergleich zu Spezialgeräten liegt ihr Nachteil in relativ ungenauen Zeitstempeln.

Die Genauigkeit von PC-basierten Messungen hängt von der verwendeten Hardware (Netzkarte, Festplatte, Bussysteme, Prozessorleistung, Speicherorganisation), dem Betriebssystem und der Erfassungsoftware ab. In [12] wird die Meßgenauigkeit von Linux-Systemen bei Verwendung der Meß-Software *tcpdump* an einer Ethernet-Schnittstelle untersucht. Ethernet-Rah-

men werden zunächst vollständig von der Netzkarte empfangen bevor ein Interrupt (Betriebssystem-Unterbrechung) abgesetzt wird. Das Betriebssystem ruft daraufhin die entsprechende Interrupt-Routine für die Netzkarte (Treiber) auf, welche den Rahmen in eine Datenstruktur kopiert und mit einem erst zu diesem Zeitpunkt angeforderten Zeitstempel versieht. Sollten mehrere Rahmen während einer Interrupt-Routine eintreffen, werden sie erst nach deren Beendigung gemeinsam behandelt. Die Zeitstempel entsprechen dann entweder dem Kopierzeitpunkt innerhalb der Interrupt-Routine oder dem Aufrufzeitpunkt der Interrupt-Routine. Die Zeitstempel stimmen also nicht exakt mit dem Zeitpunkt des Eintreffens überein. Die ermittelte Ungenauigkeit entspricht in etwa der Dauer einer Interrupt-Routine und wird mit 10 µs als Richtwert angegeben. Bei Vergleichen mit baugleichen Meß-PCs ähnlicher Konfiguration wurde eine Abweichung der gleichzeitig gemessenen Zeitstempel um +/- 5 µs beobachtet.

Insbesondere bei hoher Last, d.h. bei großen Rahmenlängen und Dauerbelastung kommt es auch zu Paketverlusten. Der Typ der Netzkarte sowie die Qualität der zugehörigen Treiber-Software spielen bei der Leistungsfähigkeit eine entscheidende Rolle. Ein Vergleich von zwei exemplarischen Produkten (Intel und Tulip) zeigt deutliche Unterschiede. Wo das erste System bereits Verluste über 35% verbucht, bleibt das zweite noch unter 0.1%! Tabelle 2.2 zeigt, wie groß die Unterschiede bei hoher Systemlast sein können. Es sei allerdings angemerkt, daß diese Ergebnisse nur bei einer Dauerbelastung (über 400000 Pakete mit einer Paketrate von 2000/s) beobachtet wurden – bei kürzerer Belastung werden durch Pufferung deutlich weniger Paketverluste erzielt.

Tabelle 2.2: Vergleich der Leistungsfähigkeit verschiedener Ethernet-Netzarten

Netzarte		Intel EtherExpressPro 100	Tulip-Chip
Aufzeichnen der ersten 68 Bytes bei 2000 Paketen/s	kurze Pakete (~1.5 Mbit/s)	verlustfrei	verlustfrei
	lange Pakete (bis 24 Mbit/s)	Paketverluste ab 400 Bytes: 2%	
Aufzeichnen der voll- en Paketlänge bei 2000 Paketen/s	kurze Pakete (~1.5 Mbit/s)	starke Paketverluste	wenig Paketverluste ab 700 Bytes: 0.04% bei 1500 Bytes: 0.07%
	lange Pakete (bis 24 Mbit/s)	linear ansteigend bis 1500 Bytes: 37%	
Aufzeichnen der ersten 68 Bytes bei 12000 Paketen/s	kurze Pakete (Spieleverkehr)	–	wenig Paketverluste 0.006%

Um die Genauigkeit von PC-basierten Messungen zu verbessern, wurden bereits zahlreiche hardwarebasierte Erweiterungen vorgeschlagen, mit denen die Zeitauflösung auf $0.1 \mu\text{s}$ erhöht werden kann [124,143]. Die in dieser Arbeit verwendeten Messungen wurden mit dem Ziel der Verkehrsanalyse von interaktiven Diensten durchgeführt. Bei diesen Diensten ist eine zeitliche Auflösung der Paketankünfte von $10 \mu\text{s}$ ausreichend genau, so daß auf eine Hardwareunterstützung des Mess-Systems verzichtet werden konnte.

2.4 Verkehrsbeschreibung

In der Telekommunikation wird Verkehr allgemein als "Vorgänge der Inanspruchnahmen von Betriebsmitteln zur Übermittlung von Nachrichten" definiert [99]. Erst in Verbindung mit einer konkreten Betrachtung läßt sich auch eine konkrete quantitative Beschreibungsform festlegen. Im Falle der Leistungsbewertung von Kommunikationssystemen bauen darauf wiederum System- und Verkehrsmodelle auf.

In diesem Abschnitt werden zunächst unterschiedliche Betrachtungsmöglichkeiten für Nachrichtenverkehr aufgezeigt, um daraus geeignete Beschreibungsformen in Form einer Auswahl der relevanten Verkehrsgrößen ableiten zu können. Anschließend werden Grundlagen der statistischen Beschreibung dieser Verkehrsgrößen dargelegt. Abschließend werden die Erzeugung der Verkehrsgrößen und die dafür später eingesetzten Approximationsmethoden diskutiert.

2.4.1 Charakteristische Verkehrsgrößen

Gemäß der oben genannten allgemeinen Definition von Verkehr muß eine Verkehrsbeschreibung alle Aspekte enthalten, die in irgendeiner Weise zur Belegung von Betriebsmitteln führen können. Grundsätzlich müßte eine allgemeine Verkehrsbeschreibung demnach folgende Informationen enthalten:

- **Verkehrsaufkommen:** Die Ankunftszeitpunkte und die Dauer bzw. Größe der Verkehrseinheiten (z.B. Anrufe, Pakete), um die Belegung der Betriebsmittel (Kanäle, Warteschlangen) zu bestimmen.
- **Verarbeitungsinformationen:** Wenn Teile der Kommunikation unterschiedlich behandelt werden sollen, müssen jeweils die Verbindungskennung (Adressen, Portnummern) und Klassifizierungen bekannt sein.
- **Nachrichteninhalte:** Wenn im Rahmen der Nachrichtenübertragung auch deren Inhalte ausgewertet werden müssen, z.B. mit dem Ziel der Komprimierung oder Filterung, müssen entsprechende Inhaltsbeschreibungen oder die vollständigen Inhalte zur Bestimmung der Arbeitslast vorliegen.

Der zuletztgenannte Aspekt impliziert eine nahezu vollständige Erfassung einer Kommunikation und taugt allenfalls zur Untersuchung von Spezialfragen. Für eine effiziente Charakterisierung von Verkehr ist eine abstrahierte Beschreibung sinnvoll, die sich auf das Verkehrsaufkommen und ggfs. auf einige Verarbeitungsinformationen beschränkt.

Eine Verkehrsbeschreibung bezieht sich außerdem auf eine bestimmte Aktivitätsebene, durch welche die oben genannten Verkehrseinheiten festgelegt sind. Bild 2.7 zeigt eine Unterscheidung in Aktivitätsebenen, welche die ITG in [99] erarbeitet hat. Ob Sitzungen, Kommunikationsver-

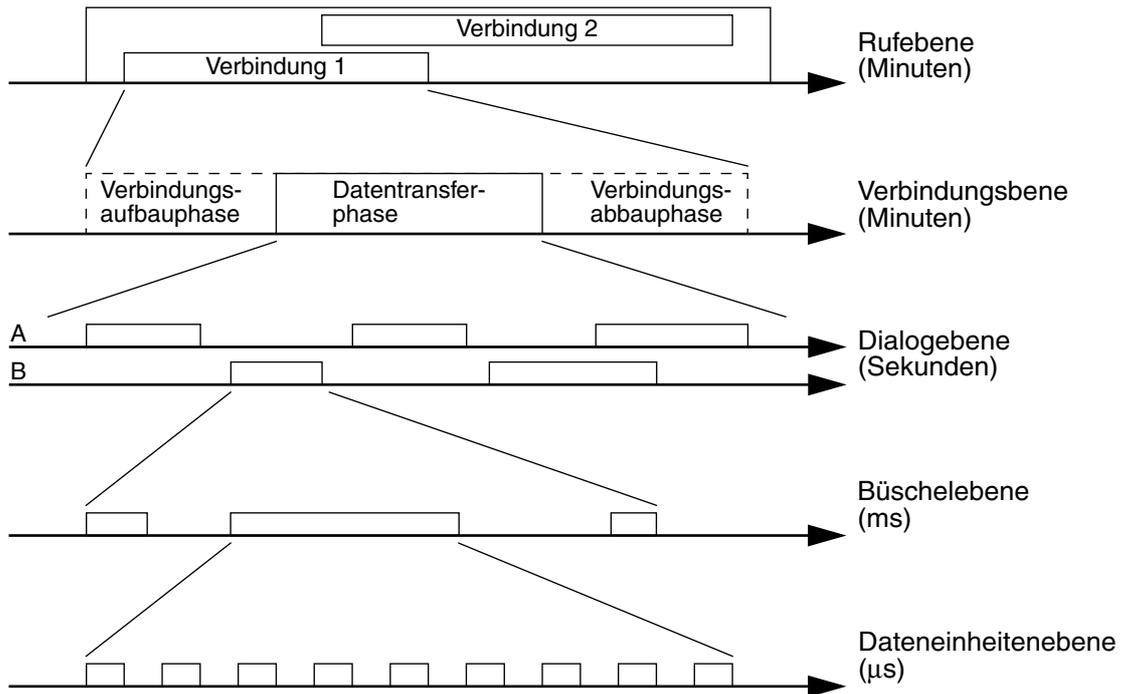


Bild 2.7: Aktivitätsebenen der Kommunikation nach der ITG-Begriffsdefinition [99]

bindungen, Dialoge, Bündel, oder Dateneinheiten betrachtet werden, hängt von der Problemstellung und dem gewünschten Abstraktionsgrad ab. Bei der Verkehrsbeschreibung können auch höhere Aktivitätsebenen mit einbezogen werden.

Bei Internetverkehr ist die Unterscheidung zwischen Dialog (zeitliche Abfolge der Interaktionen zwischen zwei Partnern) und Verbindung (zeitliche Abfolge des Auf- und Abbaus von Verbindungen) nicht so eindeutig wie in Bild 2.7 dargestellt. Dialoge können durchaus auch über mehrere unterschiedliche Verbindungen laufen, also oberhalb der Verbindungsebene liegen, wie z.B. in [46] dargestellt. Dies ist beispielsweise der Fall, wenn bei interaktiven Diensten unterschiedlich wichtige Informationsteile über unterschiedliche Verbindungsarten (UDP und TCP) übertragen werden oder wenn Web-Browser mehrere TCP-Verbindungen parallel einsetzen, um Inhalte schneller bzw. von unterschiedlichen Servern zu laden.

Definitionsgemäß bezieht sich eine Verkehrsbeschreibung immer auf bestimmte Betriebsmittel, d.h. der Verkehr wird an einem bestimmten Punkt der Kommunikationsinfrastruktur betrachtet. An diesem Bezugspunkt setzt sich der Verkehr im Allgemeinen aus zahlreichen unterschiedlichen Kommunikationsbeziehungen zusammen. Die Charakteristik dieses aggregierten Verkehrs kann zusammenfassend beschrieben werden (Summenverkehr) oder aus Beschreibungen seiner Teilverkehre (Flows, Verkehr einzelner Teilnehmer oder Dienste) zusammengesetzt werden. Bild 2.8 veranschaulicht den Einfluß des Betrachtungspunktes auf Umfang und Art des Verkehrs.

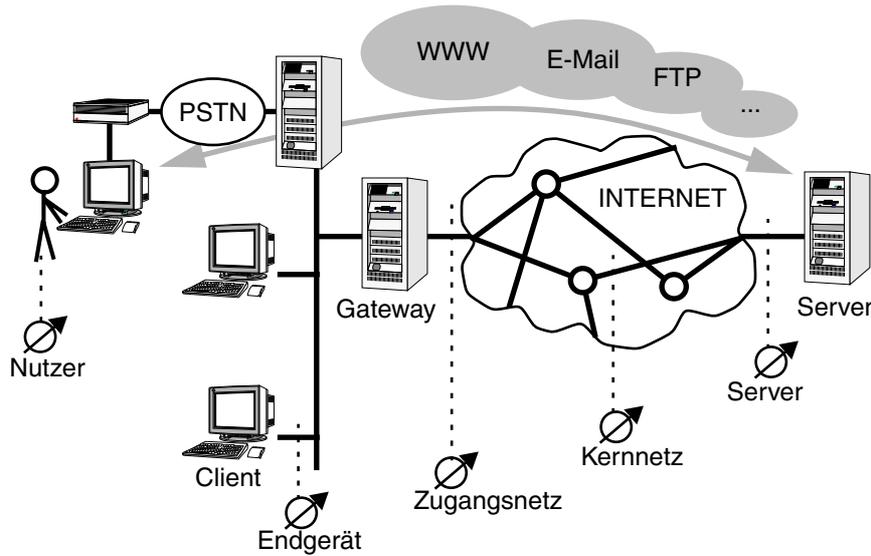


Bild 2.8: Unterscheidung des Verkehrs bzgl. Betrachtungsort und Diensten

Teilverkehre können oft anschaulicher und genauer beschrieben werden und ihre Verwendung erlaubt eine größere Flexibilität bei der Verkehrsbeschreibung. Insbesondere die Betrachtung von Verkehren bestimmter Dienste hat den Vorteil, daß die einfacheren Kommunikationsflüsse auf höheren Protokollschichten beschrieben werden können. Durch Berücksichtigung der Protokollfunktionalitäten ergibt sich indirekt der Verkehr auf tieferen Schichten. Damit entfällt eine direkte Beschreibung auf der angestrebten Aktivitätsebene, welche die teils komplexen Verkehrscharakteristiken, die durch die eingesetzten Protokolle erzeugt werden, ebenfalls einschließen müßte. Teilverkehre sollten so allgemeingültig wie möglich beschrieben werden, um sie leichter einem gewünschten Kontext (Betrachtungsort, -zeit und Lastzustand) anpassen zu können. Die Suche nach Invarianten in der Verkehrsbeschreibung führt auf die Untersuchung einzelner Dienste und ihrer charakteristischen, immer wiederkehrenden Verkehrsgrößen wie z.B. typische Tagesverkehrskurven, Sitzungsankünfte nach einem Poissonprozeß oder Größenverteilung von übertragenen oder angebotenen HTTP-Objekten (siehe auch [8,135]).

Eine zusammenfassende Verkehrsbeschreibung ist zumeist weniger umfangreich aber auch weniger anschaulich. Sie ist bei Verkehren mit sehr hohem Aggregationsgrad oder auch zur Trendanalyse sinnvoll. Sie bezieht sich aber nur direkt auf die angestrebte Aktivitätsebene und muß für jeden Betrachtungsort und dessen Lastzustand gesondert erfolgen.

Eine indirekte Verkehrsbeschreibung, die hierarchisch über mehrere Aktivitätsebenen erfolgt, ist besonders als Basis für die Modellierung von interaktiven Kommunikationsbeziehungen interessant, wie sie auch in der vorliegenden Arbeit vorgenommen wird. Sie erlaubt nämlich die Nachbildung von Rückkopplungseffekten, wie sie in jedem Sicherungsprotokoll zu finden sind (Quittierungsmechanismen, Fehlerbehandlung, Datenflußsteuerung) und sie ermöglicht die Ende-zu-Ende-Betrachtung der Dienstgüte.

Die Zielsetzung der vorliegenden Arbeit ist die Verkehrsmodellierung hinsichtlich der Leistungsuntersuchung von IP-basierten Kommunikationssystemen auf Paketebene. Da dienstspezifischer Ende-zu-Ende-Verkehr betrachtet wird, sind Informationen zum Routing jedes Paketes nur in Spezialbetrachtungen erforderlich (Verbindungen zu mehreren unterschiedlich lokalisierten Servern) und müssen in erster Näherung nicht betrachtet werden. Für die Leistungsbeurteilung der IP-basierten Systeme müssen auch keine Inhalte ausgewertet werden. Damit genügt es, den Verkehr nur in seinem Umfang zu beschreiben, also die Häufigkeit und die Größe von Nachrichteneinheiten. Damit kann die Belegung der zu untersuchenden Betriebsmittel bestimmt und bewertet werden.

2.4.2 Statistische Beschreibung

Zufallsvariable

Die charakteristischen Verkehrsgrößen (hier Paketgrößen bzw. Paketankünfte) treten in einem Beobachtungszeitraum jeweils als Datenreihe auf, welche als Ergebnis eines Zufallsexperiments aufgefaßt werden kann. Die zugrundeliegenden statistischen Eigenschaften werden dann mit Hilfe von Zufallsvariablen beschrieben. Eine Zufallsvariable X kann bei jeder Realisierung einen Wert aus einem bekannten diskreten oder einem kontinuierlichen Wertebereich annehmen, der von unkontrollierbaren zufälligen Umständen abhängt und nicht vorhersagbar ist. Jede mögliche Realisierung x erfolgt mit einer bestimmten Wahrscheinlichkeit, welche durch die Verteilungsdichtefunktion $f(x)$ bzw. die Verteilungsfunktion $F(x) = P\{X \leq x\}$ der Zufallsvariablen festgelegt ist. Die Verteilungsfunktion ist dabei die Summenfunktion der Verteilungsdichtefunktion. In Bild 2.9 sind qualitative Verläufe der beiden Funktionstypen für kontinuierliche und diskrete Zufallsvariablen angegeben.

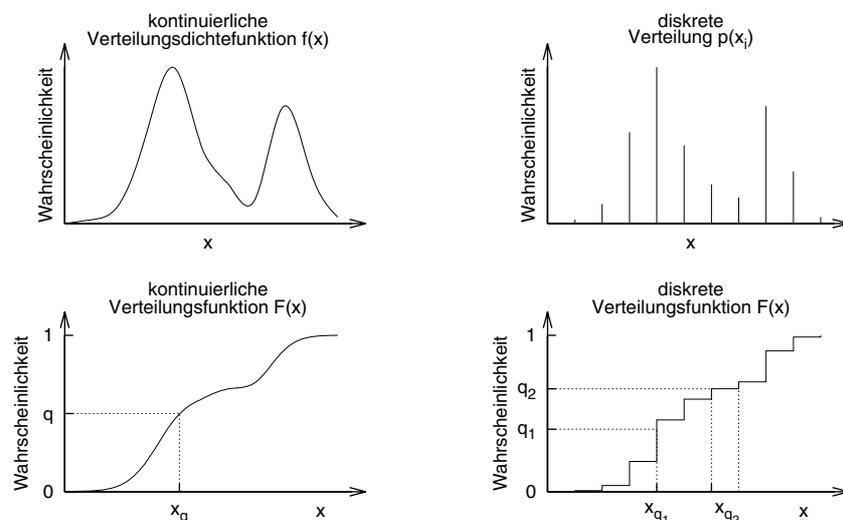


Bild 2.9: Verteilungsdichtefunktion und Verteilungsfunktion

Momente und Quantile

Eine Zufallsvariable ist durch ihre Verteilungsfunktion vollständig beschrieben. Sie kann aber zur schnelleren und anschaulicheren Interpretation durch Maßzahlen ergänzt werden. Die häufigsten Maßzahlen sind der arithmetische Mittelwert aus allen möglichen, mit ihrer Wahrscheinlichkeit gewichteten, Realisierungen \bar{x} (Erwartungswert, $E[X]$) und die Standardabweichung σ , welche die mittlere Abweichung vom Erwartungswert beschreibt (mittlerer Fehler). Anstelle der Standardabweichung wird oft die einfacher handhabbare Varianz σ^2 ($VAR[X]$) verwendet oder der dimensionslose Variationskoeffizient $c = \sigma/\bar{x}$. Sie sind Sonderfälle der Momente einer Verteilung. Das k -te gewöhnliche Moment m_k und das k -te zentrale Moment μ_k sind dabei folgendermaßen definiert:

$$m_k = E[X^k] = \int_0^{\infty} x^k f(x) dx \quad (2.1)$$

$$\mu_k = E[(X - m_1)^k] = \int_0^{\infty} (x - m_1)^k f(x) dx \quad (2.2)$$

Erwartungswert und Varianz entsprechen dem ersten gewöhnlichen Moment $m_1 = E[X]$ und dem zweiten zentralen Moment $\mu_2 = VAR[X]$.

Weitere Maßzahlen für die Eigenschaften einer Verteilung sind die Quantile x_q , welche den Wertebereich der Zufallsgröße angeben, für welchen $P\{X \leq x_q\} \leq q$ und $P\{X > x_q\} \leq 1 - q$ gilt, d.h. Zufallswerte, die kleiner als das Quantil x_q sind, treten nur mit der Wahrscheinlichkeit q auf. Bei streng monoton steigender Verteilungsfunktion ist diese Definition für das Quantil x_q eindeutig und die zugehörige Wahrscheinlichkeit kann, wie in Bild 2.9 unten links gezeigt, direkt an der Verteilung abgelesen werden. Andernfalls bezeichnet x_q einen Wertebereich (Bild 2.9 unten rechts). Das 50%-Quantil $x_{0.5}$ wird auch Median genannt und bestimmt den Wert, der in der Mitte einer sortierten Liste aller Realisierungen liegt. Der Median wird oft aufgeführt, weil er unempfindlich gegenüber Ausreißern ist.

Heavy-Tail-Verteilung

Im Rahmen von Verkehrsuntersuchungen werden zunehmend Verteilungen mit der *Heavy-Tail*-Eigenschaft gefunden. Eine *Heavy-Tail*-Verteilung besitzt einen signifikanten Anteil von sehr großen Zufallsgrößen, d.h. die Wahrscheinlichkeit für deren Auftreten ist nicht vernachlässigbar. Man spricht von einer *Heavy-Tail*-Verteilung, wenn ihre komplementäre Verteilung $F^C(x) = 1 - F(x)$ nach dem Potenzgesetz

$$F^C(x) \sim x^{-\alpha} \text{ für } x \rightarrow \infty, 0 < \alpha < 2 \quad (2.3)$$

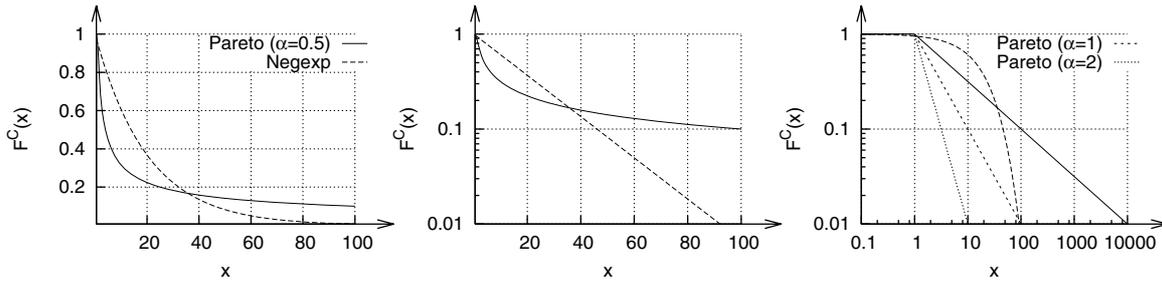


Bild 2.10: Lineare, halb- und doppelt-logarithmische Darstellung einer negativ-exponentiellen Verteilung und einer Pareto Verteilung (*Heavy-Tail*)

abfällt. Das bedeutet, daß die Momente der Verteilung gegen unendlich divergieren können. Insbesondere sind für $\alpha \leq 2$ die Varianz und für $\alpha \leq 1$ auch der Mittelwert undefiniert. Für die Darstellung von Verteilungsfunktionen mit *Heavy-Tail* empfiehlt es sich, anstatt $F(x)$ die komplementäre Verteilungsfunktion (*complementary cumulative distribution function, ccdf*) $F^C(x)$ aufzutragen. Bild 2.10 zeigt, wie dann das *Tail*-Verhalten durch logarithmische bzw. doppelt-logarithmische Darstellung veranschaulicht werden kann. Dabei wird der Kurvenverlauf des *Tails* zu Lasten des Kopf- und Kernbereichs der Funktion hervorgehoben.

Stochastische Prozesse

Zur Beschreibung von zufallsabhängigen Abläufen in Systemen ist eine Zufallsvariable nicht geeignet, da sie in der Regel nur für einen Systemzustand zu einem bestimmten Zeitpunkt t gilt. In einem zeitlichen Verlauf ist die Unabhängigkeit aufeinanderfolgender Zufallsexperimente nicht mehr unbedingt gegeben. Daher werden zur Beschreibung stochastische Prozesse herangezogen. In einem stochastischen Prozeß sind Zufallsvariablen für jeden möglichen Zeitpunkt definiert. Werden nur diskrete Zeitpunkte betrachtet, spricht man von einem zeitdiskreten stochastischen Prozeß $\{X_t\} = \{X_1, X_2, \dots\}$ (Zeitreihe), andernfalls von einem zeitkontinuierlichen stochastischen Prozeß $X(t)$.

Die Kenngrößen eines stochastischen Prozesses werden entsprechend über die Kenngrößen seiner Zufallsvariablen gebildet, insbesondere der Erwartungswert $\overline{X(t)} = E[X(t)]$ und die Varianz $VAR[X(t)] = E[(X(t) - \overline{X})^2]$. Er wird vollständig von seiner n -dimensionalen Verteilungsfunktion charakterisiert:

$$F_{\underline{X}}(\underline{x}, t) = F_{X_1 \dots X_n}(x_1, \dots, x_n, t_1, \dots, t_n) = P\{X(t_1) \leq x_1, \dots, X(t_n) \leq x_n\} \quad (2.4)$$

Ein Prozeß heißt stationär, wenn er seine statistischen Eigenschaften im zeitlichen Verlauf nicht ändert, d.h. seine Verteilungsfunktion ist invariant gegenüber Zeitverschiebungen τ .

$$F_{\underline{X}}(\underline{x}, t + \tau) = F_{\underline{X}}(\underline{x}, t) \quad (2.5)$$

Ein Prozeß heißt schwach stationär, wenn sein Erwartungswert zeitinvariant, also konstant ist und seine Autokorrelation nur von der Zeitverschiebung τ der Betrachtungszeitpunkte $X(t)$ und $X(t+\tau)$, nicht aber von t abhängt.

Korrelation

Wenn Abhängigkeiten in einer Zeitreihe bestehen, können diese mit Hilfe der Autokorrelation $R_{XX}(t_1, t_2) = E[X(t_1)X(t_2)]$ bestimmt werden. Die Autokovarianz $\gamma(t_1, t_2)$ gibt die Ähnlichkeit zweier Realisierungen des Prozesses $X(t)$ zu unterschiedlichen Zeitpunkten t_1 und t_2 an.

$$\gamma(t_1, t_2) = E[X(t_1)X(t_2)] - E[X(t_1)] E[X(t_2)] \quad (2.6)$$

Bei unkorrelierten Zufallsgrößen ergibt die Autokovarianz Null. Durch Normierung auf die Standardabweichungen der beiden Zufallsbetrachtungen $X(t_1)$ und $X(t_2)$ ergibt sich der Autokorrelationskoeffizient $\rho(t_1, t_2)$:

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\text{VAR}[X(t_1)]} \sqrt{\text{VAR}[X(t_2)]}} \quad , \rho \in [-1, 1] \quad (2.7)$$

Bei stationären Prozessen ist die Autokorrelation unabhängig von der Zeit t , so daß der Autokorrelationskoeffizient nur noch von der Verschiebung τ abhängt:

$$\rho(t, t + \tau) = \rho(\tau) = \frac{\gamma(\tau)}{\text{VAR}[X(t)]} \quad (2.8)$$

Die Autokorrelationsfunktion zeigt den Autokorrelationskoeffizienten ρ in Abhängigkeit der Verschiebung τ . Die Größe des Absolutwertes ist ein Maß für die Stärke der linearen Abhängigkeit. Bei einem statistisch unabhängigen Prozeß wird für die Autokorrelationsfunktion stets Null erwartet. Bei empirischen Daten wird sich aber immer ein Wert ungleich Null ergeben, der sich bei zunehmender Verschiebung τ allerdings sehr schnell gegen Null bewegt. Auch bei linearer (Kurzzeit-)Korrelation geht die Autokorrelationsfunktion langsam aber stetig gegen Null und zwar mit $\rho(\tau) \sim a^{|\tau|}$ für $\tau \rightarrow \infty$ und $0 < a < 1$ [109].

Langzeitkorrelation und Selbstähnlichkeit

Bei Prozessen, deren Autokorrelationsfunktion langsam abnimmt, bestehen offensichtlich auch Abhängigkeiten zwischen weit auseinanderliegenden Ereignissen. Man spricht von einer Langzeitkorrelation (*long range dependence*, LRD), wenn die Autokorrelationsfunktion

$$\rho(\tau) \sim |\tau|^{-\beta} \quad \text{für } \tau \rightarrow \infty \text{ und } 0 < \beta < 1 \quad (2.9)$$

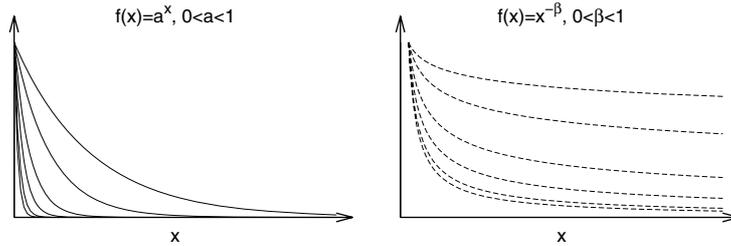


Bild 2.11: Qualitativer Verlauf der Autokorrelationsfunktion bei Kurzzeit- und Langzeitkorrelation

d.h., daß die Summe der Autokovarianzen divergiert. Bild 2.11 illustriert den Verlauf der Autokorrelationsfunktion bei kurz- und langzeitabhängigen Prozessen.

Für einen zeitdiskreten schwach stationären Prozeß $\{X_t\} = \{X_1, X_2, \dots\}$ mit Mittelwert \bar{x} und Varianz σ^2 ist ein aggregierter Prozeß $\{X_t^{(m)}\}$ definiert als

$$X_t^{(m)} = \frac{(X(tm - m + 1) + \dots + X(tm))}{m} \quad \text{mit } m \in N \quad (2.10)$$

Der aggregierte Prozeß kann als blockweise Zusammenfassung der Realisierungen des Originalprozesses verstanden werden. Mit zunehmender Blockgröße m würde bei kurzzeitkorreliertem Basisprozeß die Autokorrelationsfunktion des aggregierten Prozesses $\rho^{(m)}(\tau)$ verschwinden. Weist der Basisprozeß aber Langzeitkorrelation auf, trifft dies nicht zu, d.h. trotz beliebig großer Blockbildung schwanken die Blockmittelwerte unvermindert stark. Der Basisprozeß heißt dann exakt selbstähnlich zweiter Ordnung, wenn

$$\rho^{(m)}(\tau) = \rho(\tau) \quad (2.11)$$

und asymptotisch selbstähnlich zweiter Ordnung, wenn diese Bedingung nur für große Blockgrößen $m \rightarrow \infty$ zutrifft. Für die Varianz des aggregierten Prozesses eines langzeitkorrelierten Basisprozesses gilt

$$\text{VAR}[\{X_t^{(m)}\}] \approx m^{2H-2} \sigma^2 \quad (2.12)$$

wobei H der Hurst-Parameter $0.5 < H < 1$ ist. Der Hurst-Parameter ist ein Maß für den Grad der Selbstähnlichkeit. Für $H = 0.5$ ist der Basisprozeß nicht selbstähnlich (die Varianz seines aggregierten Prozesses verschwindet unabhängig von τ). Für $H = 1$ ist der Basisprozeß stark selbstähnlich (die Varianz seines aggregierten Prozesses ist 1 unabhängig von τ). Der Hurst-Parameter ist bestimmt durch $H = 1 - \beta/2$ (vgl. Gleichung (2.9)).

Die Beziehung in Gleichung (2.12) führt auf ein graphisches Verfahren zur Abschätzung des Hurst-Parameters, den Varianz-Zeit-Graph (*variance time plot*). Ausgehend von einer Stichprobe $\{x_i\}$, der ein vermuteter selbstähnlicher Prozeß zugrundliegt, wird dabei die Zeitreihe des zugehörigen aggregierten Prozesses $\{x_i^{(m)}\}$ durch blockweise Mittelwertbildung berech-

net. Bei hinreichend großer Blocklänge m muß dann für die Varianzen der erzeugten Zeitreihen gelten

$$\text{VAR}[\{x_i^{(m)}\}] \approx m^{2H-2} \text{VAR}[\{x_i\}] \quad (2.13)$$

$$\log \frac{\text{VAR}[\{x_i^{(m)}\}]}{\text{VAR}[\{x_i\}]} = \log(\text{VAR}[\{x_i^{(m)}\}]) - \log(\text{VAR}[\{x_i\}]) = (2H - 2) \log m \quad (2.14)$$

Durch Auftragen der Differenz der Logarithmen der Varianzen über $\log m$ erhält man eine Kurve, deren Steigung ein direktes Maß für den Hurst-Parameter ist. Bei der Bestimmung der Steigungsgeraden (Regressionsgeraden) werden allerdings weder sehr kleine noch sehr große m berücksichtigt.

Wavelets

Wavelets sind im Grunde eine Weiterentwicklung der Fourier-Transformation. Die Fourier-Transformation definiert eine alternative Darstellung von Zeitsignalen durch die Funktionen Sinus und Kosinus, d.h. sie zerlegt das Signal in seine Frequenzen und transformiert es damit vom Zeitbereich in den Frequenzbereich. In der Praxis wird sie auf begrenzte Abschnitte des Signals im Zeitbereich angewandt, um eine abschnittsweise Transformation durchzuführen, die das Signalverhalten zu bestimmten Zeitpunkten weniger fehleranfällig beschreibt – je kleiner das Fenster, also je genauer die zeitliche Auflösung, desto größer ist dann aber auch der Bereich großer Frequenzen, der nicht mehr darin enthalten und folglich auch nicht erfaßt werden kann.

Die Wavelet-Transformation verwendet anstelle von Sinus und Kosinus eine ganze Familie von periodischen Funktionen, die Wavelets genannt werden, die im Gegensatz zu Sinus und Kosinus aber begrenzt sind bzw. an den Rändern gegen Null gehen, z.B. durch eine einhüllende Gaußkurve. Die Transformation erfolgt dabei mehrfach für verschiedene Skalierungen. Dabei wird ausgehend von einem sog. Mutter-Wavelet ein formgleiches aber gedehntes Abbild verwendet. Damit haben die eingesetzten Wavelets unterschiedliche Grundfrequenzen und unterschiedliche Breite, was zur Signalanalyse in unterschiedlichen Frequenz- und Zeitbereichen führt. Das Ergebnis ist eine Schar von Funktionen, die für jede Skalierung durch Koeffizienten eindeutig bestimmt sind. Somit sind die Wavelet-Koeffizienten sowohl von der Skalierung a als auch von der Position der Wavelets b abhängig, d.h. ein Signal $f(t)$ wird in eine Funktion $c(a, b)$ zweier Variablen transformiert. Die Rekonstruktion des Zeitsignals erfolgt aus den Koeffizienten der Wavelet-Transformationen, ist allerdings aufwändiger und hängt von den Eigenschaften der gewählten Wavelets ab [97].

Wavelets sind äußerst vielseitig einsetzbar und haben sich als Bindeglied zwischen den unterschiedlichsten Forschungsdisziplinen erwiesen (Bildverarbeitung, Signalverarbeitung, Mecha-

nik, Geologie, Astronomie....). Auch bei der Betrachtung selbstähnlicher Prozesse wurden Wavelets eingesetzt. Selbstähnliche (fraktale) Prozesse sind durch ihre besondere Büschelförmigkeit charakterisiert. Der Grad der Büschelförmigkeit kann als mittlere Energie eines Signals (Frequenz) betrachtet werden. Für jede Skalierung j kann ein eigener mittlerer Energiewert E_j berechnet werden. In [3] verwenden Abry und Veitch die Koeffizienten der Wavelet-Analyse zur Bestimmung des Hurstparameters der zugrundeliegenden Verkehrsdaten. Das Verfahren wird als robust gegen Instationarität bezeichnet und erfordert keine Annahmen über den zugrundeliegenden Modellprozeß. In [74] wird ein Kaskadenmodell vorgeschlagen, das aus den Koeffizienten der Wavelet-Analyse einer Datenreihe eine neue Datenreihe beliebig vorgegebener Länge mit den gleichen (selbstähnlichen) Eigenschaften erzeugt.

2.4.3 Erzeugung von Verkehrsgrößen

Für die Modellierung von Verkehren müssen die charakteristischen Verkehrsgrößen so erzeugt werden, daß ihre zuvor beobachteten bzw. festgelegten statistischen Eigenschaften beibehalten werden. Im einfachsten Fall geschieht dies jeweils durch einen Erneuerungsprozeß, der nur durch seine Verteilungsfunktion $F(x)$ beschrieben wird. Durch Bildung eines gleichverteilten Zufallswertes p zwischen 0 und 1 kann mit der Umkehrfunktion $F^{-1}(p)$ eine Realisierung x der nach $F(x)$ verteilten Zufallsgröße erzeugt werden. Durch Wiederholung entsteht so eine Reihe unabhängiger identisch verteilter Zufallsvariablen (*independent and identically distributed, iid*), die dann als Werte der Verkehrsgröße in das Systemmodell eingespielt werden.

Wenn für die spezifizierten Verkehrsgrößen die Annahme der Unabhängigkeit und Gleichverteilung nicht gegeben ist, kann die Modellierung oft durch Einführung von Zuständen und der Betrachtung stationärer Teilphasen dennoch auf Erneuerungsprozesse zurückgeführt werden. Dieser Ansatz ist oft sehr erfolgreich, da viele natürliche Prozesse durch Poissonprozesse beschrieben werden können (beispielsweise Zwischenankünfte und Dauer von Telefongesprächen). Der Poissonprozeß ist ein Erneuerungsprozeß mit negativ exponentieller Verteilung und verfügt damit über die Eigenschaft der Gedächtnisfreiheit, d.h. seine Entwicklung hängt zu keinem Zeitpunkt von seinem bisherigen Verlauf (Gedächtnis) ab.

Die in der klassischen Telekommunikation gerne getroffenen Annahme von Poissonprozessen ist für die Charakterisierung des Summenverkehrs in paketbasierten Kommunikationsnetzen nicht mehr zulässig. Sowohl in lokalen als auch in Weitverkehrsdatennetzen wurden Langzeitabhängigkeiten und damit selbstähnlicher Verkehr beobachtet. Es wurde eine alternative Modellierungsvorschrift zur Erzeugung von asymptotisch selbstähnlichem Verkehr vorgeschlagen, bei welcher hinreichend viele On-Off-Quellen überlagert werden, deren On- bzw. der Off-Phasendauer einer *Heavy-Tail*-Verteilung folgt [61,113,133,156]. Bei entsprechender Parametrisierung weist die Pareto-Verteilung diese Eigenschaft auf, die häufig im sogenannten M/Pareto-Modell eingesetzt wird [26]. Über diesen Modellierungsansatz wird auch das Auf-

treten von selbstähnlichem Internet-Verkehr erklärt, da die Verteilung von typischerweise übertragenen Dateigrößen, die für die Dauer einer Aktivitätsphase verantwortlich sind, *Heavy-Tails* aufweisen [57,61].

Die Verteilungsfunktion ist eine elementare Beschreibungsform zur Erzeugung von Verkehrsgrößen. Sie enthält alle statistischen Informationen und gibt auch dann noch über das Verhalten Aufschluß, wenn wichtige Kenngrößen wie Mittelwert oder Varianz unendliche Werte annehmen. Wenn Verteilungen durch mathematische Funktionen abgebildet werden können, ist ihre Darstellung durch wenige Parameter möglich. Durch Angabe von Funktionstyp und den ersten Momenten einer Verteilung (insbesondere genügen oft Mittelwert und Varianz) lassen sich so die elementaren Prozesse effizient und eindeutig definieren.

Wenn mathematische Funktionen zur Beschreibung von gemessenen Systemen herangezogen werden sollen, müssen anhand der gemessenen Verteilungen geeignete Funktionen bestimmt und deren Parameter geschätzt werden (siehe Abschnitt 2.4.4). Alternativ zu diesem zeitaufwändigen Prozeß wird auch die direkte Verwendung der empirische Verteilungsdaten vorgeschlagen [111]. Empirische Verteilungen eignen sich zur schnellen Parametrisierung bestehender Modelle, sind allerdings nicht flexibel und nur tabellarisch darstellbar.

Noch weiter geht der Ansatz, auf jegliche Abstraktion eines gemessenen Verkehrs zu verzichten und ganze Meßreihen direkt als Eingangsgröße für Systemmodelle zu verwenden. Allerdings setzt dies hinreichend lange Messungen voraus und eignet sich nicht für die mit dieser Arbeit beabsichtigten Leistungsuntersuchungen. Für die Untersuchung von Caching-Verfahren ist diese Methode allerdings hilfreich.

Insbesondere zur Beschreibung von selbstähnlichem Verkehr werden mit der Fraktalen Brownschen Bewegung (FBM) oder dem Einsatz von wavelet-basierten Kaskadenmodellen auch grundsätzlich andere Modellierungsansätze verfolgt. An dieser Stelle wird auf diese Verfahren aber nicht weiter eingegangen, weil sie auf die Modellierung von aggregiertem Verkehr abzielen und nicht auf die Charakterisierung dienstspezifischer Merkmale.

2.4.4 Parameterschätzung für Verteilungen

Wenn unabhängige Zufallsprozesse für die Erzeugung der charakteristischen Verkehrsgrößen definiert und ihre statistischen Merkmale bestimmt sind, können Sie durch ihre inverse empirische Verteilungsfunktion nachgebildet werden. Die empirische Verteilungsfunktion ist aber nur umständlich als Tabelle von Klassenwahrscheinlichkeiten in die Modellierung zu integrieren und erschwert die Weiterverwendung durch Dritte. Daher wird in der Regel eine mathematische Beschreibung der Verteilungsfunktionen bevorzugt, welche die wesentlichen Eigenschaften des Zufallsprozesses abbildet.

Zunächst muß eine geeignete mathematische Funktion ausgewählt werden, welche die Grundcharakteristik der empirischen Verteilung aufweist. Im Anhang A werden einige häufig verwendete Funktionen mit ihrem charakteristischen Kurvenverlauf, ihren Eigenschaften und Parametern gezeigt. Es ist vorteilhaft, Funktionen mit wenigen Parametern zu wählen, um möglichst wenige Freiheitsgrade zu erhalten, die zusätzliche Randbedingungen erfordern. Im zweiten Schritt werden die Parameter der gewählten Verteilung (Schätzfunktion) bestimmt. Für diese Approximation der mathematischen Funktionen an die empirischen Verteilungen gibt es unterschiedliche Verfahren.

2.4.4.1 Näherungsverfahren

Momentenmethode

Bei der Momentenmethode werden die ersten n Momente der empirischen Verteilung berechnet. Die n Parameter der Schätzfunktion werden nun so bestimmt, daß diese dieselben ersten n Momente aufweist. Die Methode ist dann nicht anwendbar, wenn die Momente der Stichprobe nicht zuverlässig bestimmt werden können. Das ist insbesondere bei Zufallsreihen mit selbstähnlichen Eigenschaften der Fall und äußert sich darin, daß die Momente auch bei zunehmender Beobachtungsdauer auf keinen stabilen Wert konvergieren.

Maximum-Likelihood-Methode

Die Maximum-Likelihood-Methode definiert die Likelihood-Funktion L , welche die Wahrscheinlichkeit ausdrückt, daß die beobachtete Stichprobe $\xi_1, \xi_2, \dots, \xi_n$ bei der zugrundeliegenden Verteilung $F(x)$ herauskommt. Die Likelihood-Funktion enthält auch die Parameter der Schätzfunktion, die so bestimmt werden sollen, daß L maximal wird.

Für eine diskrete Zufallsvariable X mit dem Wertevorrat $W = \{x_1, \dots, x_n\}$ sei die Wahrscheinlichkeitsverteilung $f(x_k) = P\{X = x_k\}$ bekannt, nicht aber deren m Parameter $\vartheta_1, \dots, \vartheta_m$, also

$$P\{X = x_k\} = f(x_k, \vartheta_1, \dots, \vartheta_m). \quad (2.15)$$

Die Wahrscheinlichkeit, daß genau die vorgegebene Stichprobe gezogen wurde, ist

$$L(\xi_1, \dots, \xi_n, \vartheta_1, \dots, \vartheta_m) = \prod_{k=1}^n f(\xi_k, \vartheta_1, \dots, \vartheta_m). \quad (2.16)$$

Es werden nun diejenigen $\vartheta_1, \dots, \vartheta_m$ gewählt, für welche L maximal wird. Dies kann durch Nullsetzen der partiellen Ableitung nach jedem Schätzparameter und der Lösung des entsprechenden Gleichungssystem erreicht werden. Alternativ und insbesondere bei großen Stichproben sinnvoll wird $\ln(L)$ partiell nach jedem Schätzparameter abgeleitet und zu Null gesetzt. Die

logarithmierte Likelihood-Funktion kann als Summe einfacher bestimmt werden und hat ihren Maximalwert an der selben Stelle [22,107,153].

$$\ln(L(\xi_1, \dots, \xi_n, \vartheta_1, \dots, \vartheta_m)) = \sum_{k=1}^n \ln(f(\xi_k, \vartheta_1, \dots, \vartheta_m)) \quad (2.17)$$

Die Lösungsgleichungen der Maximum-Likelihood-Methode müssen allerdings für jeden Schätzfunktionstyp gesondert bestimmt werden. Nur für wenige Funktionen können knappe Lösungsformeln bestimmt werden – oft müssen aufwändige numerische Lösungsverfahren verwendet werden.

Minimierung der Fehlerquadratsumme

Die Methode der Minimierung der Fehlerquadrate (*least square*) beruht auf der Abweichung der geschätzten Funktionswerte $\hat{y}_i = F(x_i)$ von den empirischen Funktionswerten y_i , die quadriert und aufsummiert werden. Diese Summe aller Fehlerquadrate Q ist abhängig von den m Parametern der Schätzfunktion $\vartheta_1, \dots, \vartheta_m$ und soll über deren Änderung minimiert werden.

$$Q = \sum_{i=1}^n (y_i - F(x_i, \vartheta_1, \dots, \vartheta_m))^2 \quad (2.18)$$

Sind die Längen der (symmetrischen) Fehlerbalken der empirischen Funktionswerte bekannt, kann die gebildete Differenz noch mit dem Kehrwert der jeweiligen Länge gewichtet werden. Bei einer linearen Verteilungsfunktion kann Gleichung (2.18) analytisch minimiert werden, indem nach jedem ϑ_i partiell abgeleitet und zu Null gesetzt wird. Im nichtlinearen Fall werden iterative Verfahren wie beispielsweise das Levenberg-Marquardt-Verfahren angewendet. Dabei wird nach jeder Iteration ein Erfolgswert λ erhöht oder verringert und abhängig davon ein Verschiebungsvektor bestimmt, der ausgehend vom aktuellen Parametervektor den Parametersatz für die nächste Iteration berechnet. Das Verfahren konvergiert, wenn eine Akzeptanzschwelle von Q unterschritten wird, oder nur minimale Änderungen zwischen zwei Iterationen stattfinden. Das Verfahren findet schnell ein Minimum, das allerdings nicht unbedingt das globale Minimum ist.

Der große Vorteil bei der Minimierung der Fehlerquadrate liegt in ihrem breiten Anwendungsbereich, da sie auf fast jede Verteilung angewandt werden kann. Die Methode ist aber sehr empfindlich gegenüber den Anfangswerten, die aus einer groben Schätzung gewonnen werden müssen [39,169,180].

FW-Methode (Feldmann/Willinger)

In [76] beschreiben Feldmann und Willinger eine rekursive abschnittsweise Annäherung einer Hyperexponentiellen Verteilung hoher Ordnung (z.B. $n = 20$) an bestehende mathematische

Verteilungsfunktionen mit *Heavy-Tail*, wie die Weibull- oder Pareto-Verteilung. Die Methode kann auch auf empirische Verteilungen angewandt werden [1]. Die Ordnung n der Hyperexponentiellen Verteilung beschreibt die Anzahl der enthaltenen Exponentialfunktionen. Ihre komplementäre Verteilungsfunktion ist

$$F^C(x) = \sum_{i=1}^n p_i e^{-\lambda_i x}. \quad (2.19)$$

Es werde angenommen, daß die Größen λ_i geordnet seien ($\lambda_1 < \dots < \lambda_n$). Große i entsprechen einem kleinen Erwartungswert der Exponentialfunktion und ihr Kurvenverlauf fällt schneller ab. Wenn λ_{i+1} ausreichend größer als λ_i ist, kann bei der Überlagerung der Anteil der Teilfunktionen mit $i > 1$ vernachlässigt werden. Die Methode erfordert, daß zunächst der Bereich gewählt wird, über den die Näherung erfolgen soll – außerhalb dieser Grenzen ist das Ergebnis nicht mehr genau. Der Wertebereich wird dann in Abschnitte zerlegt, wobei das Verhältnis benachbarter Grenzen ausreichend groß sein muß, also z.B. $c_i/c_{i+1} = 10$. Mit Hilfe des Grenzwertes c_i und eines Stützwertes innerhalb des Abschnitts i bei bc_i mit $1 < b < c_i/c_{i+1}$ können dann die Parameter der Teil-Exponentialfunktion i bestimmt werden (unter Berücksichtigung aller Ergebnisse $1, \dots, i-1$ durch Bildung einer neuen Zielverteilung $F_i^C(xc_i)$).

$$\lambda_1 = \frac{1}{(b-1)c_1} \ln\left(\frac{F^C(c_1)}{F^C(bc_1)}\right) \text{ und } p_1 = F^C(c_1) e^{\lambda_1 c_1} \quad (2.20)$$

$$\lambda_i = \frac{1}{(b-1)c_i} \ln\left(\frac{F_i^C(c_i)}{F_i^C(bc_i)}\right) \text{ und } p_i = F_i^C(c_i) e^{\lambda_i c_i} \text{ für } 1 < i < k \quad (2.21)$$

$$\lambda_k = \frac{1}{c_k} \ln\left(\frac{p_k}{F_k^C(c_k)}\right) \text{ und } p_k = 1 - \sum_{j=1}^{k-1} p_j \quad (2.22)$$

$$\text{mit } F_i^C(xc_i) = F_{i-1}^C(xc_i) - \sum_{j=1}^{i-1} p_j e^{-\lambda_j x c_i} \quad (2.23)$$

Bei halblogarithmischer Darstellung kann das Verfahren als abschnittsweise lineare Anpassung betrachtet werden. Das Verfahren erzielt bei $n = 20$ i.d.R. sehr gute Ergebnisse und zeigt, daß teilweise schwer zu behandelnde *Heavy-Tail*-Verteilungen auf einfache negativ exponentielle Verteilungen zurückgeführt werden können.

2.4.4.2 QQ-Plot und PP-Plot

Für *Heavy-Tail*-Verteilungen ist insbesondere eine graphische Bewertung der Güte einer Approximation interessant, da sie Aufschluß über die Näherung über den gesamten Verlauf der

Verteilung erlauben. Neben der einfachen Darstellung der empirischen sowie der angenäherten Verteilungsfunktion in linearem, einfach logarithmischen und doppelt logarithmischen Maßstab gibt es mit dem QQ-Plot (*quantile-quantile plot*) und dem PP-Plot (*probability-probability plot*) weitere sehr anschauliche Bewertungsverfahren. Insbesondere erlauben beide Methoden die Lokalisierung von signifikanten Abweichungen.

Wie in Abschnitt 2.4.2 bereits beschrieben, wird unter einem q -Quantil x_q derjenige Parameterwert verstanden, der mit der Wahrscheinlichkeit q nicht überschritten wird, also $q = P\{X \leq x_q\} = F(x_q)$. Dies entspricht praktisch der Definition der Verteilungsfunktion und demnach wird ein Quantil zu einer vorgegebenen Wahrscheinlichkeit q über die invertierte Verteilungsfunktion bestimmt. Da die Invertierung mathematisch nicht immer möglich ist, kommen hier auch Schätzverfahren zum Einsatz.

Bei einem QQ-Plot werden die q -Quantile der theoretischen Verteilung über die zugehörigen q -Quantile der empirischen Verteilung aufgetragen. Bei optimaler Anpassung sollte sich die Winkelhalbierende ergeben, die als Referenz mit eingezeichnet wird. QQ-Plots eignen sich gut zur Beurteilung der Randwerte.

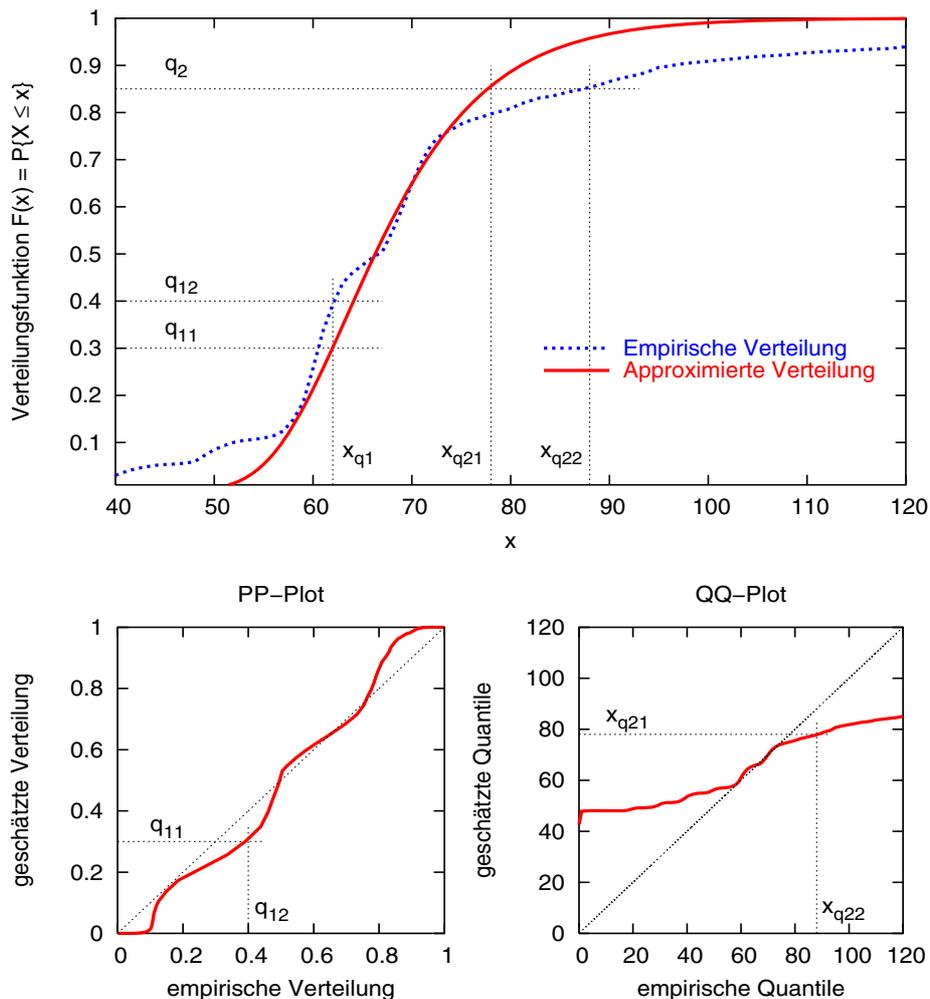


Bild 2.12: Qualitative Veranschaulichung von PP-Plot und QQ-Plot

Bei PP-Plots werden die theoretischen Unterschreitungswahrscheinlichkeiten q für Zufallsvariablenwerte x über die empirischen Unterschreitungswahrscheinlichkeiten aufgetragen. Auch hier sollte sich bei optimaler Approximation eine Winkelhalbierende ergeben. PP-Plots eignen sich gut, um die Anpassung in der Mittellage zu beurteilen.

Bild 2.12 veranschaulicht diese beiden Darstellungsformen anhand einer qualitativen Approximation. Hinsichtlich der Erzeugung von Zufallsvariablen anhand einer approximierten Verteilungsfunktion ist die größere Bedeutung des QQ-Plots offensichtlich.

2.4.4.3 Erweiterte Näherungsverfahren

Approximation von kombinierten Verteilungen

Gerade bei der Verkehrscharakterisierung werden einige Verkehrsgrößen beobachtet, deren Verteilungen *Heavy Tails* aufweisen. Damit ist die Wahrscheinlichkeit für das Auftreten sehr großer Parameterwerte signifikant hoch, so daß der Verteilungsschwanz – je nach Bedeutung der Verkehrsgröße – eine bedeutende Charakteristik des Verkehrsverhalten beschreibt.

In Relation zum Kopfbereich wird der *Tail*-Bereich allerdings von relativ wenigen Meßpunkten mit kleinen absoluten Wahrscheinlichkeiten charakterisiert, so daß übliche Approximationsmethoden wie Minimierung der Fehlerquadrate und die Maximum Likelihood-Methode entsprechend wenig Gewicht auf diesen Teil legen und den *Tail*-Verlauf meist sehr schlecht nachbilden. Auch die Momentenmethode ist ungeeignet, da der *Tail*-Verlauf durch Momente nur sehr schwer zu erfassen ist. Zudem existieren im schlimmsten Fall gar keine höheren Momente von geeigneten Approximationsfunktionen mehr. Auch werden oft begrenzte Werte, also abgeschnittene Verteilungen betrachtet, deren Momentenberechnung entsprechend komplizierter wird.

Zudem weisen die häufig verwendeten mathematischen Verteilungen, meist nur wenige Parameter und damit auch eine zu unflexible Kurvenform auf, um sowohl den Verlauf der Kopfteils für kleine Werte als auch den Verlauf des Schwanzteils für große Werte gut zu beschreiben.

Wenn eine genaue Nachbildung des *Tail*-Verlaufes erforderlich ist, bietet sich die Verwendung von abschnittswisen Funktionen zur genauen Approximation der Teilbereiche an, wie dies beispielsweise in [28] genutzt wurde. Die Anzahl der Abschnitte ist im Prinzip beliebig groß, oft genügen aber zwei Teilfunktionen zur Beschreibung des Normalbereichs und des *Tail*-Bereichs.

An dieser Stelle wird eine einfache methodische Vorgehensweise vorgestellt, nach welcher einige der Approximationen in Kapitel 3.2 erfolgen. Die Methodik folgt den grundsätzlichen Anforderungen der gewünschten Approximation: im Kopfbereich soll die grundsätzliche Charakteristik der Verkehrsgröße beschrieben werden und im *Tail*-Bereich soll der Verlauf der Verteilungsfunktion möglichst genau nachgebildet werden.

- Grobe Auswahl der Teilbereiche: falls der Verlauf der empirischen Verteilungsfunktion nicht offensichtlich eine andere Aufteilung vorgibt, sollen zu dem Kopfbereich alle Werte kleiner dem 90%-Quantil $x < x_{0,9}$ zählen. Die Werte oberhalb des 90%-Quantils $x \geq x_{0,9}$ zählen dann zum *Tail*-Bereich.
- Auswahl der Approximationsfunktionen: die Approximation erfordert zunächst die Auswahl von geeigneten Funktionen für die Teilabschnitte. Dabei hilft die Betrachtung der empirischen Verteilung als Wahrscheinlichkeitsdichte $f(x)$ und als komplementäre Verteilungsfunktion $F^C(x)$ und der Vergleich mit dem charakteristischen Verlauf von bekannten Funktionen, wie sie auch in Anhang A aufgeführt sind. Die logarithmische oder doppelt logarithmische Darstellung von $F^C(x)$ zeigt den Verlauf des *Tails* besonders gut.
- Approximation des Kopfbereiches: als Approximationsmethode, die eine einfache Begrenzung des betrachteten Wertebereichs zulässt, wird das Verfahren der Minimierung der Fehlerquadrate gewählt. Die Approximation wird der Einfachheit halber bereits auf die komplementäre Verteilungsfunktion angewandt.
- Approximation des *Tail*-Bereiches: um den Verlauf der komplementären Verteilungsfunktion auch für sehr kleine Wahrscheinlichkeiten gut nachzubilden, werden die Wahrscheinlichkeitswerte vor der Approximation durch Minimierung der Fehlerquadrate logarithmiert $\log F^C(x)$. So wie die logarithmische Darstellung den *Tail*-Verlauf durch Gewichtung kleiner Wahrscheinlichkeiten gut hervorhebt, wird auch die Approximation kleine Wahrscheinlichkeiten stärker gewichten und den *Tail*-Verlauf gut nachbilden.
- Wahl des Übergangspunktes: Werden die beiden Näherungsfunktionen unabhängig voneinander bestimmt, treffen sie sich im Allgemeinen nicht genau an der Grenze der Teilbereiche. Die zusammengesetzte Verteilungsfunktion weist einen positiven oder einen für Verteilungsfunktionen nicht definierten negativen Sprung auf (vgl. Bild 2.13). Falls in der

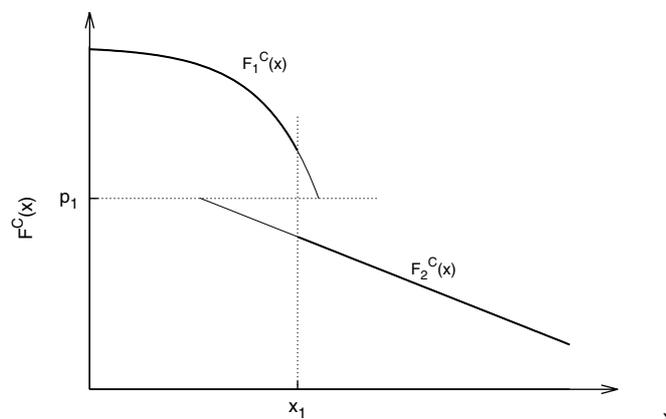


Bild 2.13: Zusammensetzen in Teilbereichen approximierter Verteilungsfunktionen

Nähe der Bereichsgrenze ein Schnittpunkt existiert, soll dieser als Übergangspunkt (x_1, p_1)

gewählt werden – damit verschiebt sich die Grenze zwischen Kopf- und Schwanzbereich. Andernfalls soll der Punkt gewählt werden, an dem sich die Funktionen am nächsten kommen (Minimum der absoluten Differenz beider Teilfunktionen). Ist die Wahl nicht eindeutig, wird am eingangs gewählten 90%-Quantil festgehalten, also $(x_{0,9}, 0.9)$.

- Anpassung der Approximation im Schwanzbereich: Falls kein Schnittpunkt als Übergangspunkt gewählt wird, muß die Näherungsfunktion für den Schwanzbereich so angepaßt werden, daß sie durch den Übergangspunkt verläuft, also $F(x_1) = p_1$. Damit wird die Näherungsfunktion um einen Freiheitsgrad reduziert, der nötigenfalls durch Einführung einer Verschiebung geschaffen werden kann. Nochmalige Approximation der so eingeschränkten Funktion führt zum Endergebnis für den *Tail*-Bereich. Die Zusammensetzung beider Näherungsfunktionen führt zu einer stetigen, wenn auch im Normalfall in x_1 nicht differenzierbaren Verteilungsfunktion.

Die Vorgehensweise kann durch Iteration ohne großen Aufwand auf mehrere Teilbereiche erweitert werden. Allerdings setzt sie die mathematische Umformbarkeit zumindest einer Teilfunktion voraus, so daß sie in (x_1, p_1) nach einem ihrer Parameter aufgelöst werden kann, der nun in Abhängigkeit der anderen Parameter verwendet wird. Beispielsweise für die Lognormal-Funktion kann bereits keine geschlossene Form der Verteilungsfunktion angegeben werden. Für die in Kapitel 3.2 gewählten Kombinationen von Verteilungsfunktion werden die notwendigen Umformung der Näherungsfunktionen für den *Tail*-Bereich in Anhang C.1 (Weibull/Pareto), C.2 (Lognormal/H2) und C.3 (Weibull/Lognormal) gezeigt.

Abgeschnittene Funktionen

In der Praxis sind viele Verkehrsgrößen begrenzt aber mathematische Verteilungsfunktionen sind meist unbegrenzt, d.h. Zufallsgrößen können unendlich groß werden. Die Begrenzung des Ergebnisbereichs kann durch die Verwendung von abgeschnittenen Verteilungen erreicht werden [150]. Hierfür muß das Abschneiden der Verteilungsdichtefunktion und das Abschneiden der Verteilungsfunktion unterschieden werden. Im ersten Fall werden Zufallswerte, die eine Obergrenze überschreiten abgelehnt und eine erneute Zufallszahlenerzeugung durchgeführt, d.h. die Restwahrscheinlichkeit wird gleichmäßig auf die abgeschnittene Verteilung verteilt. Im zweiten Fall wird im Falle der Überschreitung der Wert der Obergrenze als Zufallszahl verwendet. Hier werden also alle Ereignisse der Restwahrscheinlichkeit auf die Obergrenze kumuliert. In beiden Fällen ändern sich die Momente und auch die Form der Verteilungsfunktionen. Die Approximation sollte daher mit den abgeschnittenen mathematischen Funktionen erfolgen.

Kapitel 3

Verkehrsmodellierung des interaktiven Abrufdienstes WWW

3.1 Modellierung

3.1.1 Dienstmerkmale

Das World Wide Web (WWW) ist eine Entwicklung des CERN, das 1989 ursprünglich als internes Informationssystem konzipiert war und sich auf Grund seiner einfachen Benutzung zum populärsten Dienst des Internet entwickelt hat [148]. Die Nutzbarkeit des WWW wurde durch Suchdienste erweitert, welche die effektive weltweite Recherche nach Stichworten ermöglichen. Seit 1994 wird seine Entwicklung vom World Wide Web Consortium (W3C) organisiert. Das WWW basiert auf der Inhaltsbeschreibungssprache HTML (*Hypertext Markup Language*) sowie dem Übertragungsprotokoll HTTP (*Hypertext Transfer Protocol*). Über graphische Benutzerprogramme (Browser) können Inhalte angefordert und betrachtet werden.

HTML definiert die Inhalte und deren Darstellung und ermöglicht die Verknüpfung von Schlüsselstellen innerhalb eines HTML-Dokuments mit Verweisen auf weitere HTML-Dokumente. Referenzierte Dokumente können dabei sowohl Text- als auch Bild-, Musik- oder Videodateien sein, die von modernen Browsern wiedergegeben werden. Die im Browser dargestellten WWW-Seiten bestehen aus einem Rahmen-Dokument (Basisdokument), in welchem weitere Dokumente über Referenzen eingebettet sein können, die als Teil des Basisdokuments dargestellt werden sollen und automatisch nachgeladen werden. Dieses automatisierte Laden von Daten ohne Zutun des Nutzers prägt die Verkehrscharakteristik im World Wide Web entscheidend. HTML wurde zunächst von der IETF in RFC 1866 [24] spezifiziert, aber inzwischen von den Empfehlungen HTML 3.2 bzw. HTML 4.0 des W3C ersetzt und wird dem-

nächst durch flexiblere und auf XML (*Extensible Markup Language*) basierte Sprachkonzepte ergänzt.

HTTP ist ein einfaches Abfrageprotokoll, das auf die Internet-Protokolle TCP und IP aufsetzt und auf dem Client/Server-Kommunikationsmodell beruht. Im Prinzip unterscheidet es nur die beiden Nachrichtentypen Anforderung (*Request*) und Antwort (*Response*). Anfragen werden stets vom Client (*Browser*) an den Server gesendet und enthalten eine weltweit eindeutige Referenz (URI, *Uniform Resource Identifier*) des gewünschten Dokuments. Der Server antwortet entweder mit dem angeforderten Dokument oder mit einer Fehlermeldung. Die Abfolge von Anforderung und Antwort sei im folgenden als HTTP-Abwurf bezeichnet. Eine HTTP-Kommunikation umfaßt immer einen vom Nutzer erzeugten primären Abruf des Basisdokuments und eventuell automatisch erzeugte sekundäre Abrufe der eingebetteten Objekte. Bild 3.1 zeigt den schematischen Ablauf einer solchen Kommunikation sowie die verwendeten Begriffe.

Die weit verbreitete ursprüngliche Version der Protokolls (HTTP/1.0) ist eine einfache und grundlegende Spezifikation und wurde in RFC 1945 [25] definiert. Aufgrund von Leistungsschwächen wurde HTTP/1.1 formuliert (RFC 2616, [79]). Weiterhin wird an einem Rahmenwerk für flexiblere Erweiterungen des Protokolls gearbeitet.

HTTP-Nachrichten enthalten immer die Protokollversion und die angeforderte URI bzw. den Antwort-Status und gegebenenfalls die Antwortdaten. Es gibt zahlreiche weitere Kopffelder für Metainformationen (Inhaltstyp, Applikationstyp, Cache-Informationen, usw.), die aber optional sind.

Für die Betrachtung des WWW-Verkehrs sind die folgenden Besonderheiten des HTTP-Protokolls sowie die Verwendung der TCP/IP-Kommunikation relevant.

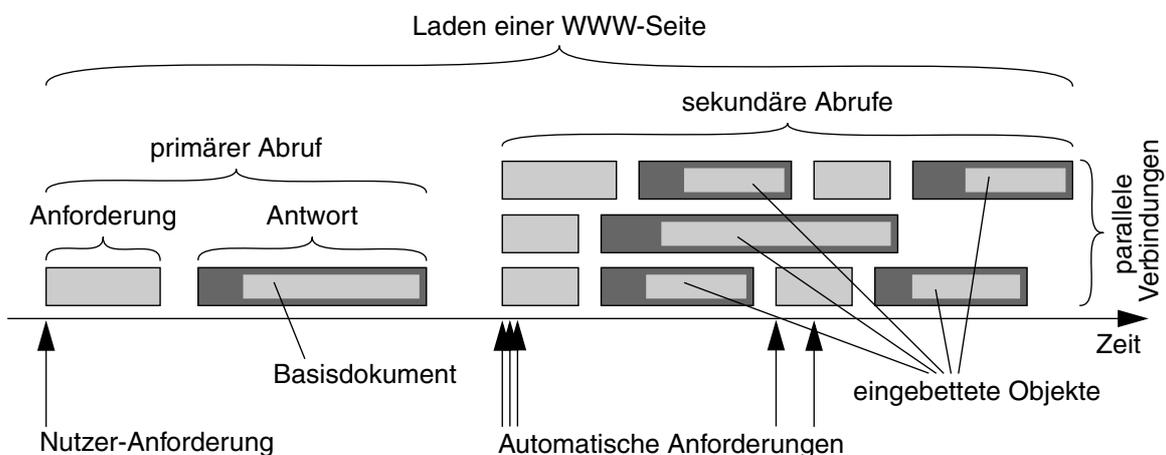


Bild 3.1: Schematischer Ablauf einer HTTP-Kommunikation und Begriffserklärung

- Nutzung von parallelen TCP-Verbindungen

Für HTTP-Verkehrsbetrachtungen ist insbesondere die Nutzung paralleler TCP-Verbindungen durch die HTTP-Browser von Bedeutung. Durch die *Slow Start*-Phase benötigt TCP einige Zeit, bis eine Datenübertragung mit akzeptabler mittlerer Bitrate erfolgt. Da viele HTML-Dokumente relativ klein sind und zumindest in HTTP/1.0 jeweils über eine eigene TCP-Verbindung übertragen werden, läuft das Protokoll zum Großteil in dieser Startphase ab und nicht in der operativen Phase (*congestion avoidance*). Um die Übertragung einer WWW-Seite zu beschleunigen, setzen WWW-Browser in der Regel mehrere TCP-Verbindungen ein, um eingebettete Objekte parallel zu übertragen. In [79] wird diese Vorgehensweise auch empfohlen - allerdings mit nur maximal zwei parallelen Verbindungen und nicht, wie von vielen Browsern verwendet, 4-6 parallele Verbindungen pro WWW-Server [7,152]. Die Nutzung paralleler Verbindungen führt durch mehrfachen Verbindungsaufbau und Verbindungsverwaltung zu signifikantem Mehraufwand und auch zu einem aggressiveren Verhalten des TCP-Verkehrs – Paketverluste, die eigentlich zur Halbierung des Sendefensters führen würden, resultieren im aggregierten Verkehr von N parallelen TCP-Verbindungen nur zu einer Reduzierung um $1/(2N)$ ([7,157]). Aus Sicht des Netzes wäre also eine alternative Beschleunigung der Übertragung wünschenswert. Laut [7] nutzen nur die Hälfte aller Clients überhaupt parallele Verbindungen.

- Persistente Verbindungen

Persistente Verbindungen wurden als optionales aber nicht offizielles Kopffeld ("Connection: keep alive") bereits bei HTTP/1.0 verwendet und als sinnvolle Erweiterung in HTTP/1.1 aufgenommen. Damit wird signalisiert, daß die bestehende TCP-Verbindung nach abgeschlossenem HTTP-Abruf nicht abgebaut, sondern gehalten werden soll. Durch diese Weiterverwendung der TCP-Verbindung bleiben bereits erreichte Sendefenstergrößen erhalten, und es wird die Ineffizienz der Übertragung durch die *Slow Start*-Phase verringert. Außerdem entfällt natürlich der TCP-Verbindungsaufbau für jeden neuen Abruf. In [7] wird darauf hingewiesen, daß persistente Verbindungen oft ungenutzt offengehalten werden und damit Ressourcen des WWW-Servers blockieren. Aus diesem Grund werden solche Verbindungen nach einiger Zeit beendet (sowohl Server- als auch Client-seitig [152]) oder von manchen Servern gar nicht erst akzeptiert [145].

- Pipelining

Eine Neuerung in HTTP/1.1 ermöglicht es, innerhalb einer persistenten Verbindung mehrere Anforderungen abzusetzen, ohne jeweils die Antwort abwarten zu müssen. Die dadurch verursachte Verschachtelung der Abrufe erhöht die Auslastung einer TCP-Verbindung und verringert die Reaktionszeit des WWW-Servers. Pipelining wird allerdings nur von etwa 30% aller WWW-Server unterstützt [157] und kann selbst dann nicht immer eingesetzt werden, da zur Unterscheidung der sequentiell eintreffenden Antworten im Antwort-Kopf

genaue Angaben zu den Antwort-Längen stehen müssen, die aber bei den immer häufiger verwendeten dynamisch erzeugten Dokumenten nicht direkt verfügbar sind.

- Caching

Das Zwischenspeichern von HTML-Dokumenten beschleunigt die Ladezeiten einer WWW-Seite enorm. Caching findet dabei im Browser, im Zugangnetz (Proxy-Server) und auch im Server selbst statt. Trotz der zunehmenden Verwendung von dynamisch erzeugten Inhalten, die nicht zwischengespeichert werden können, erreichen Proxy-Server (abhängig von Cache-Größe, Nutzeranzahl und Ersetzungsstrategie) Trefferraten von 50% und mehr [158]. Die hohen Trefferraten werden auf die beobachtete Verteilung der Beliebtheit von HTML-Dokumenten nach dem Gesetz von Zipf zurückgeführt [37,57], welches besagt, daß die Wahrscheinlichkeit, daß das n -beliebteste Dokument abgerufen wird, proportional zu $1/n$ ist¹.

3.1.2 Dienstgüte-Metriken für WWW-Verkehr

Für die Leistungsbewertung eines Systems bezüglich des Dienstes WWW werden Metriken benötigt, die quantitative Aussagen über die erreichte Dienstgüte erlauben. Die klassischen Metriken auf Paketschicht (Verlustwahrscheinlichkeit, Verzögerung und Verzögerungsschwankung) sind hierfür ungeeignet. Vielmehr muß die vom Nutzer empfundene Dienstgüte beschrieben werden, d.h. die Metriken müssen auf Beobachtungen auf der WWW-Dienstschicht beruhen und damit das Ende-zu-Ende-Verhalten der Kommunikation beschreiben. Die möglichen Kriterien für eine solche Bewertung basieren also auf den für den Nutzer erkennbaren Rückmeldungen, die bereits aus Bild 3.1 (prinzipieller Ablauf eines WWW-Seitenabrufs) hervorgehen (s.a. Tabelle 3.1).

Für den Nutzer ist in erster Linie die schnelle Darstellung der angeforderten Seite entscheidend. Als Hauptkriterium soll dabei die WWW-Seitenladedauer t_W betrachtet werden. Da WWW-Browser die abgerufenen HTML-Basisdokumente bereits darstellen können, wenn eingebettete Objekte noch geladen werden, ist auch die Dauer des primären Abrufs ein Dienstgütemerkmal. Die WWW-Seite kann bereits genutzt werden, ohne das komplette Laden abzuwarten. Die Dauer des primären Abrufs wird im folgenden als Antwortzeit t_1 bezeichnet. Ein unterstützendes Kriterium, das dem Nutzer allerdings nicht unmittelbar ersichtlich ist, ist die Dauer der sekundären Abrufe t_2 (Ladezeit der eingebetteten Objekte), die ähnlich der Antwortzeit einen Eindruck von der Schnelligkeit des Netzes vermittelt.

¹ Das Gesetz von Zipf bezog sich ursprünglich nur auf Worthäufigkeiten in einem Text.

Tabelle 3.1: Dienstgüte-Kriterien für den WWW-Dienst.

Dienstgüte-Kriterium	Bezeichnung
Seitenladedauer	t_W
Antwortzeit (Ladezeit primärer Abruf)	t_1
Ladezeit sekundärer Abrufe	t_2
Abbruchwahrscheinlichkeit (Ungeduld)	p_U

3.1.3 Bestehende Verkehrsmodelle

Die Modellierung von WWW-Verkehr ist fast so alt wie das WWW selbst. Es gibt zahlreiche Ansätze mit Schwerpunkten auf der Nachbildung der Netzlast (so wie in dieser Arbeit), der Last eines WWW-Servers oder eines Proxy-Caches.

In [113,133] wurde erkannt, daß sich Ankünfte bei Datenverkehren nicht mehr mit klassischen Poisson-Verteilungen beschreiben lassen. Obwohl für das Nutzerverhalten nach wie vor Poisson-Ankünfte angenommen werden können (Sitzungen), sind auf tieferen Kommunikationsschichten komplexere Verkehrsmuster zu beobachten, die auch fraktale Eigenschaften (Selbstähnlichkeit) besitzen. Für die Modellierung von Verkehren mit asymptotisch selbstähnlichen Eigenschaften wurde ein M/G/∞-Modell vorgeschlagen, wobei die Bediendauer einer Verteilung mit *Heavy Tail* gehorcht (z.B. Pareto oder Weibull), d.h. die komplementäre Verteilungsfunktion $F^C(x)$ fällt bei großen Werten von x proportional zu x^{-a} , $a < 2$ (Def. nach [113]). Das Modell stützt sich auf Beobachtungen von *Heavy-Tail*-Verteilungen für Dateigrößen in Rechner- und Kommunikationssystemen, welche die Dauer einer Datenübertragung bestimmen [61]. Alternativ können auch fraktale Prozesse zur Erzeugung selbstähnlicher Muster verwendet werden. In [74] weisen Feldmann, Gilbert und Willinger darauf hin, daß Internet-Verkehr sogar multifraktale Eigenschaften hat, d.h. der Grad der Selbstähnlichkeit ist für unterschiedliche Skalierungsfaktoren verschieden. Zur Erzeugung solcher Verkehrsmuster wurde ein Kaskadenmodell vorgeschlagen, das eine vorgegebene Anzahl von Datenwerten durch wiederholte Teilung und Gewichtung einer Anfangsmasse nach bestimmten Regeln erzeugt (multiplikativer Prozeß).

Die genannten Modellierungsansätze sind abstrakte aber effektive Methoden zur direkten Erzeugung eines Paketmusters. Werden sie für die Simulation eingesetzt, unterscheiden sie sich im Prinzip nicht vom oft angewandten Wiedereinspielen zuvor aufgezeichneter Meßreihen. In beiden Fällen wird dem untersuchten System eine feste Last auferlegt, die nicht auf den Systemzustand reagieren kann. Sollen aber Effekte untersucht werden, die sich über den Zeitbereich erstrecken, in welchem sich die TCP/IP-Flußkontrolle auswirkt, so ist die Modellierung der TCP/IP-Flußkontrollmechanismen notwendig. Die Beschreibung der Arbeitslast muß

dann auf höheren Schichten ansetzen [8,89,96,135]. Hierarchische Modelle, die aber ohne Rückkopplungsmechanismen wieder direkte Paketankünfte und Paketgrößen erzeugen [4,136,142], bilden zwar charakteristische Paketmuster nach, können aber ebenfalls nicht auf den Systemzustand reagieren.

Eine Übersicht über hierarchische WWW-Verkehrsmodelle und ihre Parameter ist in [42] zu finden. Die meisten Ansätze konzentrieren sich auf die Nachbildung der HTTP-Kommunikation, d.h. die Erzeugung von zu übertragenden Dateien, ohne aber auf die Transportschicht einzugehen. In [57] werden im Jahre 1995 beispielsweise Größe und Typ angeforderten HTTP-Dateien sowie ihre Beliebtheit analysiert. Diese Analyse wird in [60,61] fortgeführt.

In [62] beschreibt Deng im Jahre 1996 WWW-Dokument-Ankünfte eines Clients anhand eines ON/OFF-Modells mit Verteilungen für ON-Dauer, OFF-Dauer und Anforderungs-Zwischenankunftszeit während der ON-Phase. Für die Größe der Dokumente wird auf die in [60] gefundene Pareto-Verteilung verwiesen (siehe auch Tabelle 3.2). Die Arbeit versteht sich als Ergänzung zu [60], wo keine Angaben zur Zwischenankunftszeit (ZAZ) der HTTP-Anforderungen gemacht werden.

Tabelle 3.2: Parameter des WWW-Verkehrsmodells nach Deng

Modell-Parameter	Beschreibung
Größe HTTP-Anforderung [Bytes]	vernachlässigt
Größe HTTP-Antwort [Bytes]	Pareto: $\alpha=1.0$, $k=1000$ (nach [60])
ON-Dauer [s]	Weibull: $\alpha=0.91 - 0.77$; $\beta=e^{4.4} - e^{4.6}$
OFF-Dauer [s]	Pareto: $\alpha=0.9-0.58$
Anforderungs-ZAZ im ON-Zustand [s]	Weibull: $\alpha=0.5$; $\beta=1.5$

In [118] präsentierte Bruce Mah im Jahre 1997 ein einfaches Modell für den Verkehr eines WWW-Browsers, das auf der Auswertung von gemessenem IP-Verkehr basiert und die Grundlage für einige spätere WWW-Verkehrsmodelle bildet (z.B. in [111,112]). Unter Verwendung einer Heuristik, die in Abschnitt 3.2.2 näher beschrieben wird, werden Modellparameter der HTTP-Schicht aus der Paketmessung geschätzt. Der WWW-Verkehr wird im Wesentlichen durch die Größe der HTML-Dokumente, die Anzahl der eingebetteten Objekte und die Betrachtungsdauer nach dem Laden einer vollständigen Seite beschrieben. Über die Startzeitpunkte der HTTP-Abrufe von eingebetteten Objekten wird in [118] nichts gesagt, so daß angenommen wird, daß diese streng sequentiell erfolgen. Da spätere WWW-Browser auch parallele TCP-Verbindungen unterstützen, muß das Modell entsprechend ergänzt werden. Tabelle 3.3 zeigt die bekannt gemachten Parameter des Modells. Die Parameter-Beschreibung in [118] erfolgt leider anhand von empirischen Verteilungsfunktionen, so daß die in Tabelle 3.3 wiedergegebene mathematische Beschreibung unvollständig ist. Mah unterscheidet zusätzlich jeweils

primäre und sekundäre Anforderungs- und Antwortgrößen, die sich allerdings unwesentlich unterscheiden. Das Modell wurde von Henderson und Sahouria für die Nutzung durch das Simulationswerkzeug *ns-2* angepaßt [178].

Tabelle 3.3: Parameter des WWW-Verkehrsmodells nach Mah

Modell-Parameter	Beschreibung
Größe HTTP-Anforderung	Bimodal: ~82% 250 Bytes, ~18%: 1kByte
Größe HTTP-Antwort [Bytes]	Pareto: $\alpha=1.04 - 1.14$, $k=30 - 80$ (empirische VF)
Anzahl eingebetteter Objekte	$m=2.8 - 3.2$; median=1 (empirische VF)
Betrachtungsdauer [s]	median=15 (empirische VF)
Seitenabrufe pro Server	mean=4; median=2 (empirische VF)
Beliebtheit einzelner WWW-Server	Zipf-Law

In [106] ergänzen Khaunte und Limb die Ergebnisse von Deng und Mah und bieten eine mathematische Modellierung der Parameter an. Außerdem wird die Notwendigkeit der Modellierung der TCP/IP-Kontrollpakete erkannt (Verbindungsaufbau). Allerdings wird die Erzeugung dieser Pakete direkt modelliert anstatt TCP/IP-Flußkontrollmechanismen nachzubilden.

In [48] wird ein sehr detailliertes WWW-Verkehrsmodell für Clients vorgestellt. Choi und Limb berücksichtigen erstmals, daß sekundäre Abrufe erst nach Abschluß der primären Abrufe stattfinden. Die Zwischenankunftszeit der sekundären Abrufe sowie die Betrachtungsdauer folgen dabei einem Erneuerungsprozeß. Parallele Verbindungen oder Abhängigkeiten zwischen sekundären Abrufen werden aber nicht berücksichtigt. Die Bestimmung der festgelegten Parameter (siehe Tabelle 3.4) erfolgt auch hier mit einem Schätzalgorithmus, der in Abschnitt 3.2.2 vorgestellt wird.

Tabelle 3.4: Parameter des WWW-Verkehrsmodells nach Choi und Limb.

Modell-Parameter	Beschreibung
Größe HTTP-Anforderung [Bytes]	Lognormal: mean=360.4, stddev=106.5
Größe HTTP-Antwort [Bytes]	primär: Lognormal: mean=10710, stddev=25032 sekundär: Lognormal: mean=7758, stddev=126168
Anzahl eingebetteter Objekte	Gamma: mean=5.55, stddev=11.4
Anforderungs-ZAZ im ON-Zustand [s]	Gamma: mean=0.86, stddev=0.187
Betrachtungsdauer [s]	Weibull: mean=39.5, stddev=92.6
Auswertedauer der primären Antw. [s]	Gamma: mean=0.13, stddev=0.187

In [14,15] stellen Barford und Crovella ein WWW-Verkehrsmodell namens SURGE (*Scalable URL Reference Generator*) vor. Das Modell beschreibt Zugriffe mehrerer Nutzer auf WWW-

Seiten eines Servers mit einem ON/OFF-Modell je Nutzer nach dem Vorbild von Deng [62]. WWW-Dokumente und Dokumentgrößen werden allerdings nicht zufällig bei jeder Anforderung erzeugt, sondern aus einem Satz von WWW-Dokumenten ausgewählt, der in der Initialisierungsphase nach Größen-Verteilungen der auf dem Server gespeicherten Dateien und der im Netz übertragenen Dateien erzeugt wird. Die Auswahl erfolgt unter Berücksichtigung der Beliebtheit einzelner WWW-Dokumente eines Servers nach Zipfs Gesetz und der empirisch beobachteten temporalen Lokalität einzelner Dokumente zur Beschreibung von Caching. Elemente einer WWW-Seite werden während der ON-Phase mit kurzen Pausen (sog. active OFF) übertragen, d.h. SURGE geht von sequentiellen HTTP-Abrufen nach HTTP/1.0 ohne parallele Verbindungen aus. In [18] wird SURGE aber für die Unterstützung von HTTP/1.1 erweitert. Die OFF-Phase ist die Zeit zwischen dem Abruf zweier Seiten, entspricht also der Betrachtungsdauer (siehe Tabelle 3.5). Zusammen mit Mah [118] legt SURGE die Grundlage der heutigen WWW-Verkehrsmodelle [112,146].

Tabelle 3.5: Parameter des WWW-Verkehrsmodells nach Barford und Crovella (SURGE)

Modell-Parameter	Beschreibung
Größe HTTP-Anforderung	-
Größe der Dateien auf dem Server [kBytes]	<i>Body</i> : Lognormal: $\mu = 9.357$, $\sigma = 1.318$ <i>Tail</i> : Pareto: $\alpha=1.1$, $k=133$
Größe der HTTP-Antwort [Bytes]	Pareto: $\alpha=1.0$, $k=1000$ Bytes
Anzahl eingebetteter Objekte	Pareto: $\alpha=2.43$, $k=1$
Anforderungs-ZAZ im ON-Zustand [s]	Weibull: $\alpha = 1.46$, $\beta = 0.382$
Betrachtungsdauer (inactive OFF) [s]	Pareto: $\alpha=1.5$, $k=1$
Seitenabrufe pro Server	Lognormal: $\mu = 1.5$, $\sigma = 0.8$
Beliebtheit einzeln. WWW-Dokumente	Zipf-Law

In [75] verwenden Feldmann, Gilbert, Huang und Willinger ein Modell für das Netzsimulationswerkzeug *ns-2* [178] auf Grundlage von SURGE. In [151] präsentiert Wallerich das WWW-Verkehrsmodell *nsweb*, das ebenfalls auf SURGE basiert und für die Nutzung durch *ns-2* gedacht ist. Das Modell unterstützt HTTP/1.1 mit persistenten Verbindungen und Pipelining.

In [114,115] präsentieren Liu, Niclause und Jalpa-Villanueva einen ähnlichen Ansatz zur Erzeugung von WWW-Anforderungen an einen einzelnen WWW-Server. Allerdings wird hier kein Verkehrsmodell sondern ein Benchmark-Tool behandelt, d.h. es werden reale Anforderungen an einen realen WWW-Server abgesetzt. Die Last wird entsprechend auf der Nutzerschicht erzeugt. Sie wird durch die Zeitpunkte der Nutzeranforderungen mittels Erneuerungsprozessen für Sitzungspausen, Betrachtungsdauer und der Anzahl der betrachteten Seiten pro Sitzung (Sitzungsdauer) charakterisiert (siehe Tabelle 3.6).

Tabelle 3.6: Parameter des WWW-Verkehrsmodells nach Liu, Niclausse, Jalpa-Villanueva.

Modell-Parameter	Beschreibung
Sitzungspause [s]	Negativ Exponentiell: mean=100-2000
Anzahl eingebetteter Objekte	Inverse Gauss: mean=5, stddev=1.28
Anforderungs-ZAZ im ON-Zustand [s]	Lognormal: mean=35.8, stddev=56.4

Neben WWW-Verkehr von Client und Server werden auch Lastcharakterisierungen von WWW-Proxy-Servern durchgeführt [2,9]. Dabei konzentrieren sich die Studien auf Verkehrs-Beschreibungen zur Untersuchung von Cache-Leistungsfähigkeit. Bezüglich der Modellierung der Netzverkehrslast sind hier insbesondere die Größenverteilungen der übertragenen Dateien interessant (Lognormal mit $\mu=11.68$ und $\sigma=2.24$ für Body und *Heavy Tail* mit $\alpha=1.5$).

3.1.4 Hierarchische Modellierung

WWW-Paketverkehr wird von vielen Faktoren bestimmt. Die grundlegende Kommunikation wird durch das Nutzerverhalten ausgelöst und durch seine Interaktion mit dem Dienst weitergeführt. Die abgefragten Informationen und deren Darstellung werden durch den Dienst WWW mit seiner Beschreibungssprache HTML bestimmt. Das übertragene Datenvolumen ist durch die Dateigrößen der HTML-Dokumente bestimmt und die Abruf-Folge ist durch die Struktur der seiteninternen HTML-Referenzen vorgegeben. WWW-Browser setzen für die Übertragung meist parallele TCP-Verbindungen ein, in welchen die Dateien nach den Regeln der LAN-Protokolle (in der Regel Ethernet) und des Transportprotokolls TCP/IP übertragen werden. Das bedeutet, daß die Daten in Pakete entsprechend der maximal zulässigen Segmentgröße aufgeteilt (im Falle von Ethernet 1460 Bytes) und um die TCP/IP-Kopffelder erweitert werden. Unter Berücksichtigung der TCP-Fenstergrößen und Quittierungen werden sie dann übertragen und gegebenenfalls wiederholt.

Im Rahmen dieser Arbeit wird ein Verkehrsmodell vorgestellt, daß diese hierarchischen Vorgänge nachbildet, sich also prinzipiell an die oben aufgezeigten Ansätze anlehnt. Seine Besonderheit liegt in der durchgehenden Berücksichtigung der Rückkopplungseffekte zwischen Nutzerverhalten und den darunterliegenden Schichten bis zur Transportschicht. Neben der Anschaulichkeit, welche die Implementierung und die Parametrisierung vereinfacht, erlaubt ein hierarchischer Ansatz auch eine größere Flexibilität durch Erweiterung oder Abstraktion von Modulen sowie die Berücksichtigung von Rückkopplungseffekten auf allen Hierarchie-Schichten. Allerdings führt eine so detaillierte Modellierung auch zu einer sehr umfangreichen Beschreibung mit zahlreichen Parametern, die aufgrund der raschen Änderungen der Dienstcharakteristik des WWW regelmäßig nachgeführt werden sollten.

Das hierarchisch-rückgekoppelte WWW-Verkehrsmodell erzeugt Paketverkehr über mehrere Schichten (siehe auch Bild 3.2):

- auf **Nutzerschicht** werden Seitenaufrufe generiert, welche die Erzeugung der eigentlichen Arbeitslast auf der Dienstschicht auslösen. Der Zeitpunkt der Seitenaufrufe ist abhängig von der Geduld der Nutzers und der mittleren Betrachtungsdauer einer fertig geladenen WWW-Seite.
- auf **Dienstschicht** werden charakteristische HTML-Dokumentgrößen sowie die zugehörigen HTTP-Anforderungen und -Antworten erzeugt. HTML-Referenz-Abhängigkeiten und HTTP-Protokoll legen zudem noch die Randbedingungen für die Implementierung dieser Schicht sowie der Nutzung der TCP-Verbindungen fest.
- auf **Kommunikationssteuerungsschicht** werden TCP-Verbindungen initiiert und im Falle persistenter Verbindungen weiterverwendet. Die erteilte Auftragslast kann auch über mehrere parallele Verbindungen abgewickelt werden. Die Client- und Server-Anwendungen legen auch fest, welche HTTP-Version eingesetzt werden kann.
- in der **Transportschicht** werden die zu übertragenen Daten jeder Verbindung in Pakete zerlegt und über TCP/IP zuverlässig übertragen. Das schließt neben eventuellem Verbindungsaufbau auch Quittierungs- und Fenstermechanismen ein.

Die Parameter für diesen Modellansatz können aus Bild 3.2 direkt abgeleitet werden und sind auch in Tabelle 3.7 angegeben. Auf Nutzerschicht wird lediglich der Zeitpunkt des nächsten Seitenabrufs festgelegt. Solange der Ladevorgang der vorigen Seite noch nicht abgeschlossen ist, kann er nach Ablauf einer Geduldsspanne abgebrochen werden. In diesem Fall wird sofort ein weiterer Seitenabruf erzeugt. Nach dem vollständigen Laden einer Seite bestimmt die Betrachtungsdauer, wann der nächste Seitenabruf erfolgt. Die Initiierung der Kommunikation auf Nutzerschicht wird also durch die Modell-Parameter Geduldsspanne und Betrachtungsdauer charakterisiert. Ergänzend kann hier noch eine Unterscheidung zwischen verschiedenen WWW-Servern und ihrer jeweiligen Besuchsdauer getroffen werden, wie es in den vorgestellten Modellansätzen teilweise vorgesehen ist. Auch die Gesamtsitzungsdauer kann hier begrenzt werden.

Auf Dienstschicht wird die eigentliche Arbeitslast erzeugt, die durch die Anzahl der eingebetteten Objekte sowie die Größen von HTTP-Anforderungen und HTTP-Antworten charakterisiert ist. Primäre und sekundäre Elementgrößen können dabei unterschieden und ihre Abhängigkeit berücksichtigt werden.

Auf der Schicht der Kommunikationssteuerung sind die meisten Modell-Parameter durch die Verwendung bestimmter WWW-Browser oder WWW-Server festgelegt. Sie ändern sich nur bei einem Wechsel des besuchten WWW-Servers (beispielsweise wenn dadurch HTTP/1.1 nicht mehr unterstützt wird und HTTP/1.0 eingesetzt werden muß). Auf Seite des Servers muß allerdings noch die Verzögerung der Auslieferung einer Antwort berücksichtigt werden, die je

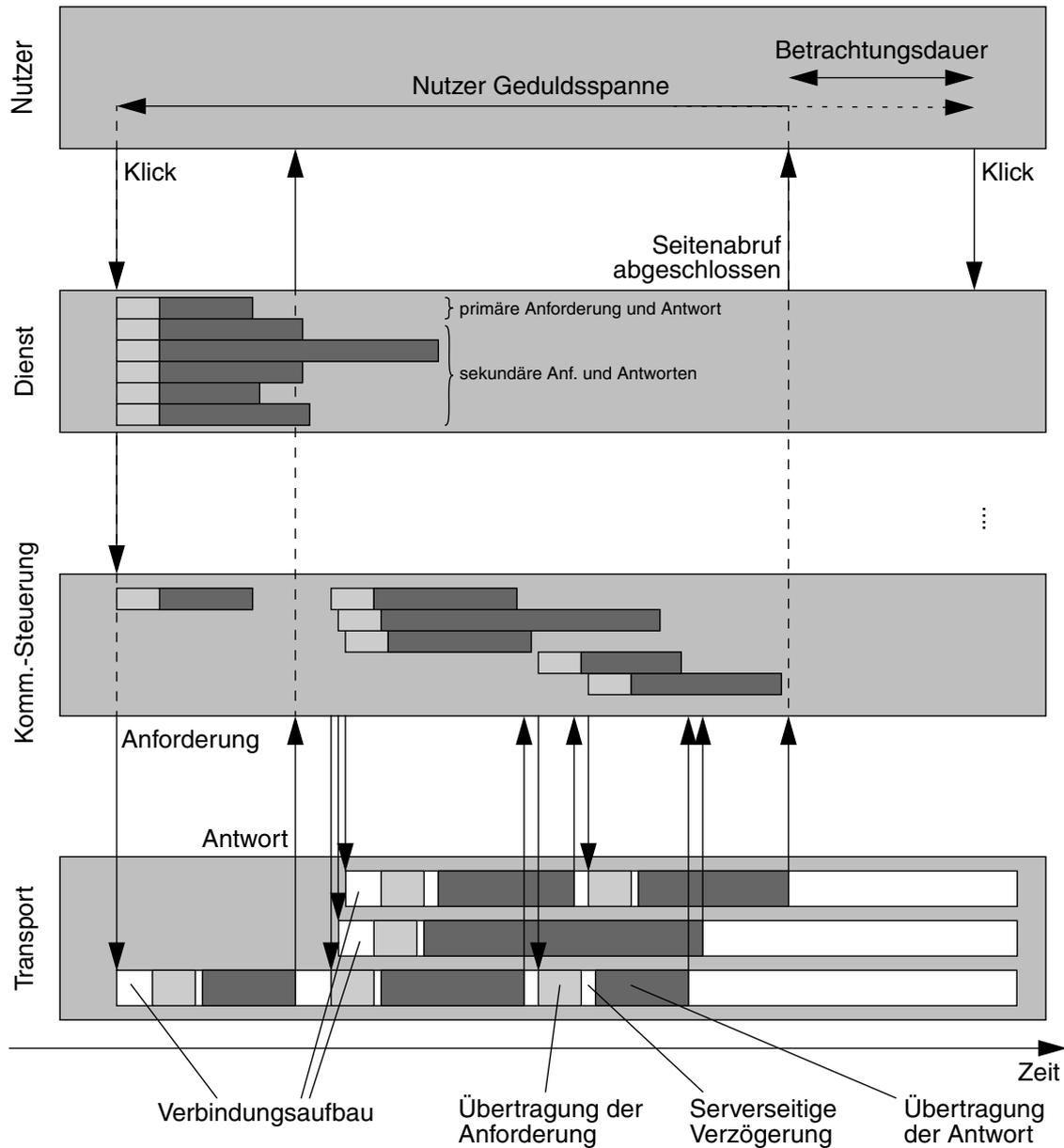


Bild 3.2: Nachbildung eines WWW-Seitenabrufs über mehrere Kommunikationsschichten

nach Last und Server-Architektur variiert. Insbesondere bei dynamisch erzeugten Seiten ist diese Verzögerung signifikant. Interne Verzögerungen auf Seiten des Clients sollen an dieser Stelle vernachlässigt werden (Dokumentbereitstellung, Dokumentbearbeitung, Darstellung oder Protokollverarbeitung).

Auf Transportschicht sind zahlreiche Parameter der TCP/IP-Protokolle zu berücksichtigen. Wenn keine Leistungsuntersuchung des TCP/IP-Protokolles erfolgen soll, werden feste Stan-

dardeinstellungen vorgeschlagen, wie beispielsweise eine maximale Segmentgröße von 1460 Bytes oder eine Empfangspuffergröße von 65 kBytes.

Tabelle 3.7: Parameter des WWW-Verkehrsmodells

Nutzer	Dienst	Kommunikationssteuerung
<ul style="list-style-type: none">• Sitzungsdauer• Server-Sitzungsdauer• Nutzergeduld• Betrachtungsdauer	<ul style="list-style-type: none">• Anforderungsgröße• Basis-Dokumentgröße• Größe v. eingeb. Objekten• Anzahl eingeb. Objekte	<ul style="list-style-type: none">• HTTP Version• Anzahl paralleler Verb.• Verb.-Zeitüberwachung• Interne Verzögerungen

Die Modellierung zeichnet sich durch Rückkopplungen auf allen Schichten sowie durch die Nachbildung der strukturellen Zusammenhänge von primären und sekundären Abrufen über parallele Verbindungen aus. Dadurch wird zum einen die nötige Flexibilität für einen allgemeinen Einsatz des Modells erreicht, und zum anderen werden die durch die Hierarchien implizierten Korrelationen und Verkehrsbüschel nachgebildet. Durch die Betrachtung der Nutzer- und Dienstsicht erlaubt das Modell zudem, Dienstgüte-Metriken für WWW zu erfassen und Systeme entsprechend simulativ zu bewerten.

3.2 Empirische Bestimmung der Modellparameter

Das in Kapitel 3.1.4 vorgeschlagene Modell für WWW-Verkehr wird von vier Verkehrsschichten (Nutzer, Dienst, Kommunikationssteuerung, Transport) charakterisiert, die weitgehend unabhängig voneinander modelliert und auch parametrisiert werden. Die Erzeugung von charakteristischen Werten für jeden Parameter wird auf Erneuerungsprozesse zurückgeführt, welche durch ihre Verteilung vollständig beschrieben sind.

Viele Parameter sind aus Verkehrsmessungen direkt bestimmbar, aber einige Parameter der höheren Schichten können beispielsweise nur aus Sicht der Anwendung beobachtet werden. Die Erfassung von statistisch relevanten Meßreihen für solche Parameter erfordert dann den Einsatz von WWW-Browsern, die um eine Protokollierfunktion erweitert wurden. Die Messungen müßten zudem in hinreichend grossen Feldversuchen durchgeführt werden, was aus organisatorischen und datenschutzrechtlichen Gründen schwierig ist und deren Ergebnisse entsprechend umständlich reproduzierbar wären.

Alternativ soll hier eine einfache Meßmöglichkeit am Zugangs- oder Kernnetz auf IP-Paket-schicht genutzt werden, welche eine umfangreiche eigene Messung sowie die Verwendung von zahlreichen verfügbaren Meßergebnissen anderer Forscher erlaubt. Um diejenigen Parameter zu bestimmen, die aus solchen Messungen nicht direkt beobachtet werden können, werden heuristische Schätzverfahren eingesetzt, die auf der Analyse der Kommunikationsvorgänge eines Dienstes und der Rekonstruktion der Abläufe auf höheren Schichten beruhen. Solche

Schätzverfahren wurden bereits eingesetzt [45,48,111,118], allerdings ohne eine grundlegende Genauigkeitsbewertung vorzunehmen.

In diesem Abschnitt wird beschrieben, wie die Parameter des WWW-Verkehrsmodells bestimmt werden. Im Abschnitt 3.2.1 werden die Randbedingungen der Messung der Verkehrsgrößen erläutert. Abschnitt 3.2.2 zeigt, wie aus den gemessenen Verkehrsgrößen die Modell-Parameter bestimmt werden – dies schließt insbesondere auch die Schätzung einiger Parameter mit ein. In Abschnitt 3.2.3 wird schließlich die statistische Auswertung aller Modellparameter mit vorgeschlagenen Näherungen präsentiert.

3.2.1 Messung

Mit der Parametrisierung des WWW-Verkehrsmodells als Meßziel ergeben sich die in Tabelle 3.7 aufgeführten Parameter als zu erfassende Größen. Aufgrund der Vorgabe einer netzseitigen Messung können einige Parameter allerdings nicht direkt beobachtet werden. Zudem muß die Messung unter eingeschränkten Randbedingungen erfolgen.

Um die Nutzer-Parameter erfassen zu können, müssen Nutzer durch eine eindeutige IP-Adresse unterscheidbar sein, d.h. sie dürfen nicht über einen Proxy-Server oder eine Firewall kommunizieren, welche die Kommunikation mehrerer Nutzer zusammenfaßt und als einzige IP-Adresse eines Teilnetzes sichtbar ist. Wenn die Sitzungsdauer sowie das Nutzerverhalten innerhalb einer Sitzung erkannt werden sollen, muß der Meßpunkt eine vollständige Sicht auf den vom Nutzer erzeugten Verkehr erlauben. Damit muß eine entsprechende Messung im Zugangnetz erfolgen. Messungen im Kernnetz erlauben im Allgemeinen nur eine teilweise Sicht auf die Kommunikation eines WWW-Nutzers.

Die Parameter der Dienstschicht (Größen und Anzahl der HTTP-Objekte) sind im Prinzip aus einer Paketmessung herauslesbar, wenn das darüberliegende Protokoll HTTP ebenfalls ausgewertet wird. Die Messung muß daher neben den TCP/IP-Paketköpfen noch ausreichend viele TCP-SDU-Daten (*Service Data Unit*) erfassen, um auch HTTP-Kopfinformationen einzuschließen. Aufgrund sehr dynamischer HTTP-Felder und ihrer dynamischen Verwendung empfiehlt es sich, mindestens 600 Bytes eines jeden Pakets aufzuzeichnen.

Die bei der Modellierung getroffene Unterscheidung in primäre und sekundäre Anforderungen kann allerdings auch auf HTTP-Schicht nicht erkannt werden. Eine solche Unterscheidung ist ausschließlich bei Beobachtungen auf Anwendungsschicht möglich. Durch den Einsatz von Heuristiken können allerdings relativ gute Schätzungen des Nutzerverhaltens erreicht werden (siehe Abschnitt 3.2.2), so daß die netzseitige Messung eine Parametrisierung des WWW-Verkehrsmodells erlaubt.

Es wurden Messungen auf Paketschicht an der Außenanbindung einiger studentischen Wohnheime der Universität Stuttgart durchgeführt. Das von einem eingetragenen Verein selbst reali-

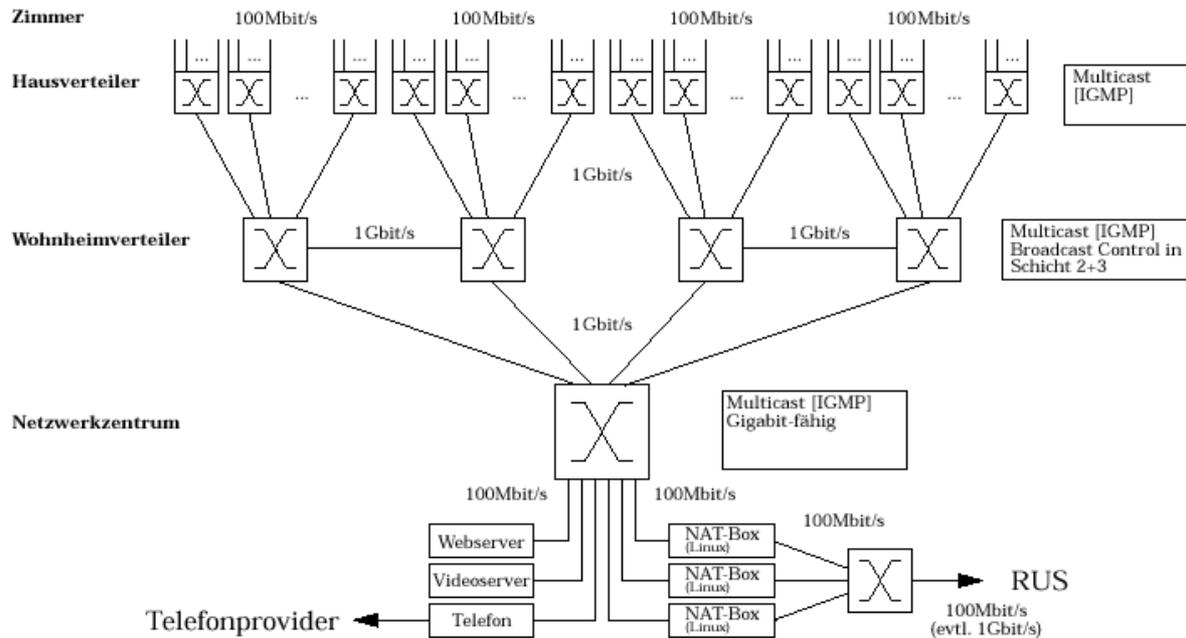


Bild 3.3: Hochgeschwindigkeitsnetz des SELFNET e.V. (Stand 2000) mit Anbindung an das Rechenzentrum der Universität Stuttgart (RUS).

sierte und verwaltete SELFNET ist ein modernes Hochgeschwindigkeitsnetz, das den Teilnehmern den Zugang mit einer Bitrate von 100 Mbit/s gewährt und intern mit mehreren 1GB/s Abschnitten operiert. Die Außenanbindung erfolgt mit 100 Mbit/s über das Rechenzentrum der Universität. Bild 3.3 zeigt eine Übersicht über den damaligen Stand des SELFNET.

Die Messung erfolgte an einem Monitor-Ausgang des Routers, auf den sowohl der eingehende als auch der ausgehende Verkehr kopiert wurde und so von einem Linux-PC mit der Meß-Software *ethereal* aufgezeichnet werden konnte. Um auch die Informationen der HTTP-Kopffelder mit sehr umfangreichen Optionen zu erfassen, wurden lange Pakete erst ab 600 Bytes abgeschnitten. In der Zeit vom 29.11.2000, 14:00 bis 30.11.2000, 14:00 wurden 26 GBytes in 60 Millionen IP Paketen von über 200 Teilnehmern aufgezeichnet.

Aufgrund der Erfassung der Paketinhalte ist für die Weiterverarbeitung außerhalb des SELFNET eine umfangreiche Anonymisierung der Daten erforderlich, die eine Vorabauswertung auf dem Meßrechner bedingt. Dabei werden für WWW-Verkehr irrelevante Pakete entfernt. Um die IP-Adressen so zu verfälschen, daß kein Rückschluß auf den Eigentümer möglich ist, aber Pakete dennoch dem richtigen (anonymen) Nutzer zugerechnet werden können, werden einfache Zuordnungslisten verwendet, die nach der Auswertung einer Serie gelöscht werden. Desweiteren werden nur solche IP-, TCP- und HTTP-Kopffelder übernommen, die eindeutig erkannt und über Filterregeln bearbeitet werden. Nicht erkannte Felder werden gelöscht. Insbesondere die Erfassung der URI, die aufgrund der gewünschten Erkennung des übertragenen Dateityps nötig ist, stellt ein kritisches Feld dar und muß auf die Datei-Endung begrenzt werden. Tabelle 3.8 zeigt exemplarisch die erfaßten Kopffelder des WWW-Verkehrs.

Tabelle 3.8: Erfasste Kopffelder des WWW-Verkehrs

Beispiel einer HTTP-Anforderung	Beispiel einer HTTP-Antwort
<p>Frame 94 Arrival Time: Nov 29, 2000 14:12:30.5423 Time delta from previous packet: 0.013084 sec</p> <p>Ethernet II Type: IP (0x0800)</p> <p>Internet Protocol Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00) Total Length: 374 Flags: 0x04 Fragment offset: 0 Time to live: 128 Protocol: TCP (0x06) Header checksum: 0x5f4c (correct) Source: 2.2.1.2 Destination: 5.7.9.11</p> <p>Transmission Control Protocol Source port: 1494 (1494) Destination port: 80 (80) Header length: 20 bytes Flags: 0x0018 (PSH, ACK) Window size: 17520</p> <p>Hypertext Transfer Protocol GET http://&.gif HTTP/1.0 User-Agent: Mozilla/4.6 [de] (WinNT; I)</p>	<p>Frame 127 Arrival Time: Nov 29, 2000 14:12:30.8316 Time delta from previous packet: 0.007255 sec</p> <p>Ethernet II Type: IP (0x0800)</p> <p>Internet Protocol Version: 4 Header length: 20 bytes Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00) Total Length: 1500 Flags: 0x04 Fragment offset: 0 Time to live: 42 Protocol: TCP (0x06) Header checksum: 0x561a (correct) Source: 5.7.9.11 Destination: 2.2.1.2</p> <p>Transmission Control Protocol Source port: 80 (80) Destination port: 1494 (1494) Header length: 20 bytes Flags: 0x0010 (ACK) Window size: 17520</p> <p>Hypertext Transfer Protocol HTTP/1.1 200 OK Server: Apache/1.3.11 (Unix) PHP/3.0.12 Content-Length: 37785 Content-Type: image/gif</p>

3.2.2 Parameterbestimmung

Das Ziel der Parametrisierung ist es, für jeden Modell-Parameter, der in Abschnitt 3.1 festgelegt wurde (siehe Tabelle 3.7) eine stochastische Beschreibung zu finden, um typischen WWW-Verkehr zu erzeugen. Ausgangsbasis ist dabei die Messung auf Paketschicht, deren Ergebnisse zunächst analysiert und interpretiert werden müssen. Abschließend folgt die statistische Auswertung und Beschreibung der ermittelten Parameter.

Aus den IP-Paketen, die bei der Messung erfaßt wurden, lassen sich über die IP-Adressen und TCP-Port-Nummern zunächst WWW-Kommunikationsflüsse einzelner Nutzer identifizieren, also auch die Sitzungsdauer eines Nutzers bzw. dessen Verweildauer auf einem Server.

Die zusätzlichen erfaßten HTTP-Kopfinformationen erlauben eine genaue Identifizierung der Ankunftszeitpunkte von HTTP-Anforderungen und HTTP-Antworten und eine Bestimmung der Protokollversion, der Art der Anforderungen (Text, Bild, Video, ...) sowie deren Erfolgsstatus (HTTP-Status-Codes). Da innerhalb der TCP-Verbindungen keine überlappenden Anforderungen beobachtet wurden (*HTTP-Pipelining*), kann der zeitliche Ablauf des HTTP-

Protokolls für jede TCP-Verbindung eines jeden Nutzers nachvollzogen werden. Aufgrund unzuverlässiger Verwendung der Größenangaben im HTTP-Kopffeld "Content-Length" empfiehlt es sich, für die Größenbestimmung von Anforderungen und Antworten die übertragenen TCP-Nutzbytes zu bestimmen. Paketwiederholungen sind durch Analyse der TCP-Sequenznummern erkennbar. Die Länge der HTTP-Header wird dabei mit einbezogen.

Für die Parametrisierung des WWW-Verkehrsmodells muß neben diesen Größen auch das Verhalten des Nutzers bestimmt werden, d.h. die primären HTTP-Anforderungen müssen identifiziert werden. Wie in Abschnitt 3.2.1 bereits angedeutet wurde, besteht keine Möglichkeit, das Nutzerverhalten unterhalb der Anwendungsschicht direkt zu beobachten, da das HTTP-Protokoll nutzererzeugten Anforderungen nicht gegenüber den automatisch erzeugten Anforderungen nach eingebetteten Objekten kennzeichnet. Durch Analyse der zeitlichen, der kontextuellen und der inhaltlichen Beziehungen innerhalb der HTTP-Kommunikation können aber Schätzungen des Nutzerverhaltens getroffen werden. Da von dieser Bestimmung weitere wesentliche Modell-Parameter und Bewertungsgrößen abhängen, nämlich die Anzahl der eingebetteten Objekte, die Betrachtungsdauer, die Seitenladezeit und die Anforderungs-Zwischenankunftszeit, ist die Schätzmethodik von besonderer Bedeutung.

In Abschnitt 3.2.2.1 werden daher verschiedene bisher eingesetzte Heuristiken zur Schätzung vorgestellt und bewertet. Eine weitere Schwierigkeit liegt in der Bestimmung der Nutzergeduld. Obwohl sie nicht direkt gemessen werden kann, kann sie anhand der gemessenen Größen abgeschätzt werden, wie in Abschnitt 3.2.2.2 gezeigt wird. Nach der Identifizierung aller Modell-Parameter folgt die statistische Beschreibung dieser Zufallsgrößen. In Abschnitt 3.2.2.3 werden die Ergebnisse dieser Auswertung dargestellt sowie mathematische Approximationen dieser Modellgrößen vorgeschlagen.

3.2.2.1 Erkennung der nutzererzeugten Anforderungen

Um aus den erkannten HTTP-Anforderungen und HTTP-Antworten die vom Nutzer erzeugten Anforderungen zu bestimmen, wurden mehrere Verfahren vorgeschlagen, die in Tabelle 3.9 zusammengefaßt sind. Sie beruhen auf der Analyse zeitlicher, kontextueller oder inhaltlicher Zusammenhänge oder einer Kombination daraus.

Rein zeitliche Aspekte beziehen sich auf die Ankunfts- und Endezeitpunkte der HTTP-Nachrichten und erfordern den geringsten Aufwand. Unter dem Kontext einer HTTP-Nachricht wird ihr Status im Vergleich zu andern HTTP-Abrufen verstanden, d.h. es wird betrachtet, ob auf weiteren TCP-Verbindungen noch Antworten anderer HTTP-Abrufe erwartet werden. Nur wenn keine Antwort mehr aussteht, darf eine abgeschlossene Seiten-Übertragung angenommen werden. Die Analyse der einzelnen TCP-Verbindungen ist etwas aufwändiger, kommt aber ohne Betrachtung der HTTP-Kopffelder aus. Bei der inhaltlichen Analyse müssen die HTTP-Kopffelder ausgewertet werden, um durch Vergleich mit der typischen WWW-Seiten-

struktur (Basis-Dokumente sind vom Typ HTML) und HTTP-Protokollabläufen (Status-Codes) auf Nutzeraktivität zu schließen.

Tabelle 3.9: Übersicht über die betrachteten Heuristiken

Heuristik	Kriterien	
Zeitlich	$T_A > 1s$	
Zeitlich & Kontextuell [45,118]	$T_R > 1s$	
Inhaltlich [48]	Typ von Anforderung und Antwort = html Status Code = 200 (OK)	
Kombiniert [160] (ein Kriterium in nur einer Zeile muß erfüllt sein)	Typ der Antwort = pdf ps doc ppt zip tar gz rar	
	$T_R > 1s$	
	Typ der Antwort = html text unknown query	$T_A > 1s$
		$T_R > 0.05s$

- Zeitbasierte Heuristik

Bei der einfachsten Heuristik wird lediglich die Zwischenankunftszeit der HTTP-Anforderungen einer Client-Server-Beziehung betrachtet. Übersteigt diese Zeit einen Grenzwert (1s), wird davon ausgegangen, daß die Seiten-Übertragung abgeschlossen ist und die nächste Anforderung durch eine Aktion des Nutzers ausgelöst wird. Aufgrund langer Ladezeiten ist diese Methode unsicher und wurde nur zu Vergleichszwecken eingesetzt.

- Zeit- und kontextbasierte Heuristik

Eine alternative Methode betrachtet die Ruhezeit einer Client-Server-Kommunikation. Übersteigt diese Zeit einen Grenzwert (1s) wird die nächste folgende HTTP-Anforderung als primär betrachtet. Die Ruhezeit wird dabei vom Ankunftszeitpunkt des letzten IP-Paketes einer HTTP-Antwort bis zur Ankunft der nächsten HTTP-Anforderung definiert. Diese Übertragungspausen werden nur dann als Ruhezeiten betrachtet, wenn keine weiteren HTTP-Antworten auf parallelen TCP-Verbindungen erwartet werden, d.h. es gibt keine unvollständigen HTTP-Abrufe mehr. Diese Methode wurde zunächst von Mah [118] und in ähnlicher Form von Charzinski [45] zur Auswertung von WWW-Messungen eingesetzt.

- Inhaltsbasierte Heuristik

In [48] stellten Choi und Limb eine Heuristik vor, welche ausschließlich inhaltliche Aspekte von HTTP-Kopffeldern betrachtet. Danach muß eine primäre Anforderung zum einen vom Typ "text/html" sein und die zugehörige Antwort ebenfalls, das heißt das referenzierte Basis-Objekt muß HTML-Code enthalten und zum anderen muß die Antwort mit dem Sta-

tus-Code 200 (OK) eintreffen, d.h. Fehlermeldungen wie z.B. nicht gefundene Referenzen oder Verweise auf verschobene Referenzen führen dazu, daß die Anforderung als sekundär gekennzeichnet wird.

- Kombinierte Heuristik

In [160] wird eine Kombination und Erweiterung der oben genannten Methoden vorgeschlagen. Die Methode ist sehr komplex, führt aber zu den besten Ergebnissen. Eine Anforderung wird danach als primär angenommen,

- wenn der Typ des angeforderten Objekts typischerweise nicht eingebettet ist, wie z.B. Textverarbeitungsdokumente (pdf, ps, doc, ppt, zip, tar, gz, rar), oder
- wenn die Ruhezeit zwischen Antwortende und Anfrageankunft länger als 1 Sekunde dauert (das entspricht der Heuristik nach Mah und Charzinski), oder
- wenn das angeforderte Objekt vom Typ html, text, unbekannt oder eine Abfrage (*query*) ist und die Anforderungs-Zwischenankunftszeit eine Sekunde überschreitet (das bedeutet, daß Anforderungen nach solchen Dateien auch bei kürzeren Ruhepausen als nutzererzeugt interpretiert werden) oder
- wenn das angeforderte Objekt vom Typ html, text, unbekannt oder eine Abfrage (*query*) ist und die Ruhezeit die angenommene anwendungsinterne Auswertungsdauer (*Parsing*) von 5 ms überschreitet (das bedeutet, daß Anforderungen nach solchen Dateien auch bei schnell aufeinanderfolgenden Anforderungen als nutzererzeugt gelten, außer wenn sie sehr schnell abgesetzt werden).

Die letzten beiden Bedingungen decken lediglich Spezialfälle ab.

Eine Bewertung aller Heuristiken in [160] zeigt, daß alle genannten Ansätze zu guten Ergebnissen führen. Aber durch systematische Fehler werden teilweise Fehlentscheidungen getroffen, die signifikante Folgefehler nach sich ziehen.

Die Bewertung erfolgt exemplarisch anhand der Messung von automatisch erzeugtem WWW-Verkehr sowohl auf IP-Schicht als auch auf Anwendungsschicht (siehe Bild 3.4). Die Messung auf der IP-Schicht ("IP-Messung") erfolgt auf dieselbe Weise, wie auch die zur Parametrisierung des Modells verwendeten Daten erfaßt wurden, also mittels einer einfachen Paketerfassungs-Anwendung auf einem PC (*tcpdump* oder *ethereal*). Die genannten Heuristiken werden auf diese Meßreihe angewendet, um die Schätzungen der Modell-Parameter zu erhalten. Gleichzeitig protokolliert die Anwendung (*Surf-Roboter*) jede HTTP-Anforderung mit und markiert die als "Nutzer" erzeugten Anforderungen. Alternativ könnte das reale Nutzerverhalten auch von einem modifizierten WWW-Browser mitprotokolliert werden. Damit können wiederum die Modell-Parameter, basierend auf dem tatsächlichen Nutzerverhalten, ermittelt und mit den geschätzten Parametern verglichen werden.

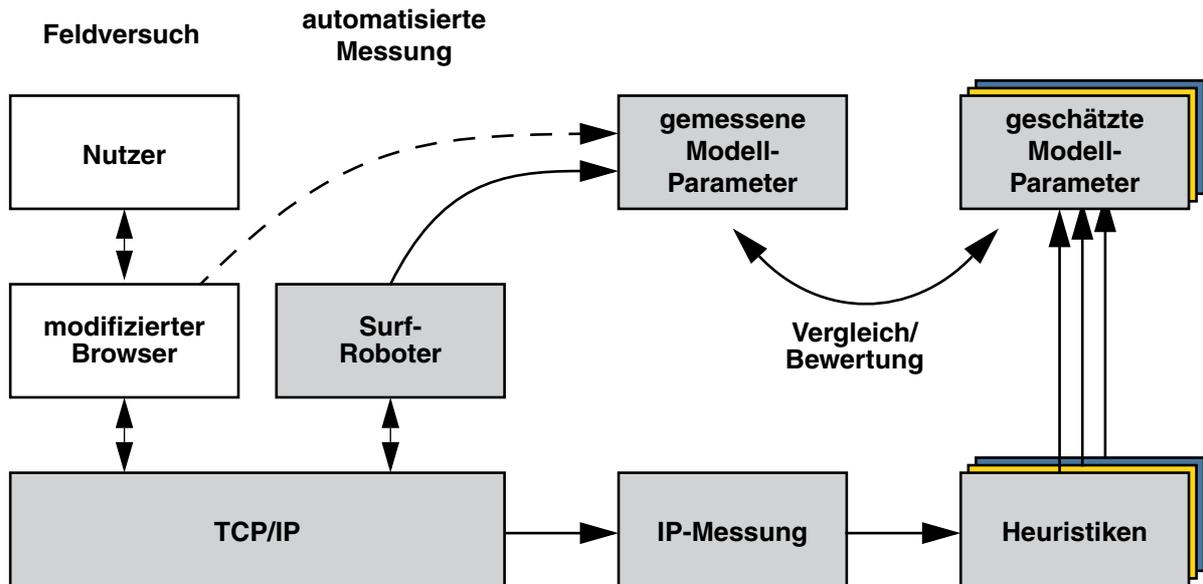


Bild 3.4: Bewertung der Heuristiken durch Vergleich mit paralleler Aufzeichnung des realen Nutzerverhaltens.

Der automatisch erzeugte Verkehr muß dabei nicht nutzertypisch sein, sondern lediglich die speziellen Fälle abdecken, welche zu systematischen Fehlern der Heuristiken führen können. Er dient lediglich der Bewertung der Genauigkeit und Stabilität der Heuristiken.

Für die Bewertung ist nicht nur die richtige oder falsche Erkennung von Nutzer-Anforderungen entscheidend, sondern deren Auswirkungen auf die daraus resultierenden Modellparameter. Für zwei exemplarische Modell-Parameter zeigt Bild 3.5 die komplementären Verteilungsfunktionen, die auf den untersuchten Heuristiken sowie den Referenzdaten basieren, im Vergleich.

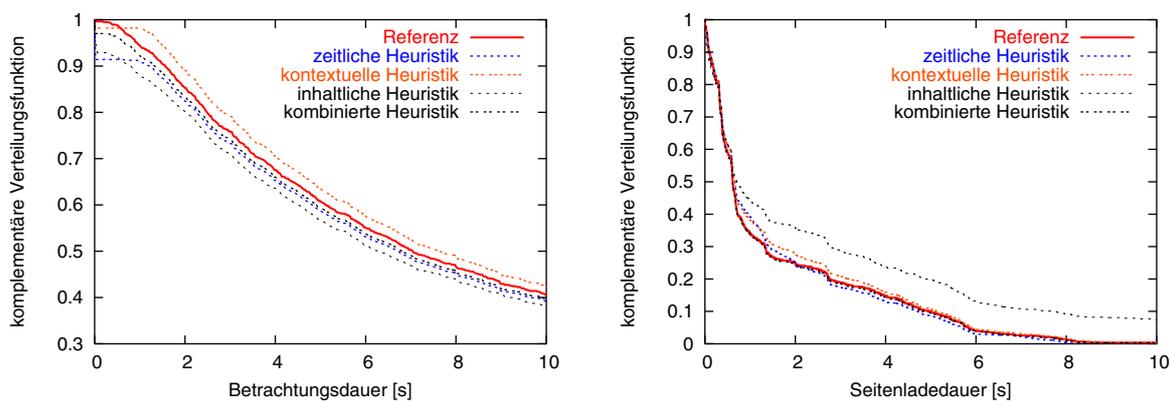


Bild 3.5: Vergleich der Ergebnisse der untersuchten Heuristiken zu Identifizierung nutzerzeugter Anforderungen anhand der davon abhängigen Modell-Parameter.

Die inhaltsbasierte Heuristik führt zu den schwerwiegendsten Fehlern, da deren zugrundeliegende Annahmen der HTTP-Kommunikation heute nicht mehr ganz zutreffen. Die automati-

sche Weiterleitung von WWW-Start-Seiten zur besseren Wartbarkeit oder Lastverteilung führt regelmäßig zu den Fehlermeldungen 301 ("Moved Permanently") oder 302 ("Moved Temporarily") und automatischem Anfragen der darin enthaltenen Referenz. Die falsche Zuordnung des primären HTTP-Abrufs zu den sekundären HTTP-Abrufen der vorigen WWW-Seite führt zu Fehlern in der Anzahl der eingebetteten Objekte sowie der Seitenladedauer und der Betrachtungsdauer.

Die rein zeitliche Heuristik ist nur als Vergleichs-Methode untersucht worden, verhält sich allerdings deutlich stabiler und erzeugt gute Ergebnisse. Die zeit- und kontextbezogene Heuristik führt zu sehr guten Ergebnissen. Am besten schneidet die kombinierte Heuristik ab, die aber deutlich komplexer ist als alle anderen. Wenn die Anwendung der Heuristik allerdings nach der Messung erfolgen kann, ist die Komplexität nicht mehr entscheidend. Als Ergebnis dieser Bewertung wird die kombinierte Heuristik zur weiteren Auswertung eingesetzt.

3.2.2.2 Bestimmung der Nutzergeduld

Die Modellierung der Nutzergeduld schafft eine Abbruchmöglichkeit im Falle von sehr langen Ladezeiten, die im Modell nicht direkt erzeugt werden, sondern durch Rückkopplung und Verhalten des untersuchten Netzes von außen bestimmt ist.

Eine Geduldsspanne, nach welcher ein Nutzer eine Transaktion abbricht, ist nicht direkt beobachtbar und damit auch nicht meßbar, da sie nicht immer in Erscheinung tritt. Es genügt nicht, nur die Fälle zu identifizieren, in welchen Ungeduld auftritt (beispielsweise durch Rückschluß aufgrund sehr kurz aufeinander folgenden oder gar sich überschneidenden Endezeitpunkt der Seitenübertragung und der Anforderung der nächsten Seite), da die resultierende Ungeduldsspanne als Modell-Parameter nicht verwendbar ist.

Eine mögliche Abschätzung kann aber durch Kenntnis der Betrachtungsdauer t_B sowie die Messung der WWW-Seiten-Ladedauer t_W und der Zwischenankunftszeit der Nutzer-Anforderungen t_A erfolgen. Die Anforderungs-Zwischenankunftszeit t_A ergibt sich wie folgt:

$$t_A = \begin{cases} t_W + t_B & \text{für } t_G > t_W \\ t_G & \text{sonst} \end{cases} \quad (3.1)$$

Die Zusammenhänge dieser Zeiten sind auch in Bild 3.6 dargestellt. Über die vom WWW-Modell erzeugten Zufallszeiten Geduld (t_G) und Betrachtungsdauer (t_B) sowie die Rückmeldung der erfolgten Seitenübertragung ergibt sich im Normalfall (bei ausreichender Geduldsspanne) eine neue Anforderung nach erfolgtem Laden und Betrachten der Webseite. Im Ungeduldsfall (d.h. bei zu langer Ladezeit) erfolgt nach der abgelaufenen Geduldsspanne ein Abbruch und eine erneute Nutzer-Anforderung.

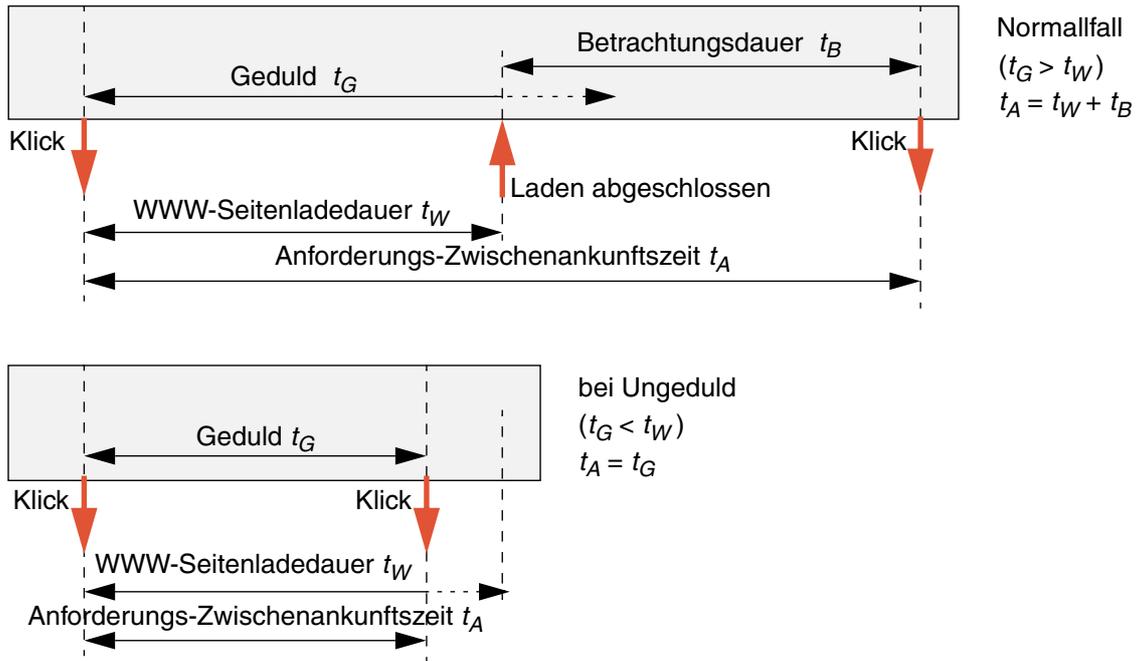


Bild 3.6: Resultierende Anforderungs-Zwischenankunftszeit im Normalfall (oben) und bei Abbruch aufgrund der Ungeduld des Nutzers (unten).

Wenn alle anderen Größen in Gleichung (3.1) bekannt sind, kann die Verteilung für die Geduldsspanne so abgeschätzt werden, daß wieder die Charakteristik der tatsächlich beobachteten Anforderungs-Zwischenankunftszeit erreicht wird. Vereinfachend kann zudem angenommen werden, daß im allgemeinen Fall die Betrachtungsdauer deutlich länger ist als die Ladezeit, also den dominanten Anteil an der Verteilung von t_A beiträgt. Tatsächlich liegen die beobachteten Verteilungen für t_A und t_B sehr eng beieinander. Die Geduldsspanne richtet sich ebenfalls nach der Ladezeit, da der Nutzer eine gewisse Erwartung für diese Wartezeit hat. Damit bleibt die Betrachtungsdauer in jedem Fall die dominante Größe, da Ungeduld in der Regel selten eintritt. Tritt sie auf, führt sie abhängig von ihrem Mittelwert zu einer Verschiebung der t_A -Verteilung nach unten, d.h. die Wahrscheinlichkeit für lange Anforderungs-Zwischenankunftszeiten sinkt. Dieses Verhalten wird von simulativen Untersuchungen bestätigt.

Die Ergebnisse in Bild 3.7 zeigen, daß die Geduldsspanne so parametrisiert werden kann, daß die simulativ bestimmte Verteilung für die Anforderungs-Zwischenankunftszeit annähernd deckungsgleich zur empirisch beobachteten Verteilung ist. Dieses Ergebnis wird mit einer negativ-exponentiell verteilten Geduldsspanne mit einem Mittelwert von 100s erreicht. Die genaue Übereinstimmung des charakteristischen Verlaufs ist dabei durch die dominante Betrachtungsdauer zu erklären. Der Geduldsspannenparameter dient lediglich zur Anpassung der Verschiebung.

Der Faktor der Nutzer-Geduld wird hier stark vereinfacht beschrieben. Eine detailliertere Modellierung, welche die Verteilung der Geduldsspanne abhängig von der Seitengröße

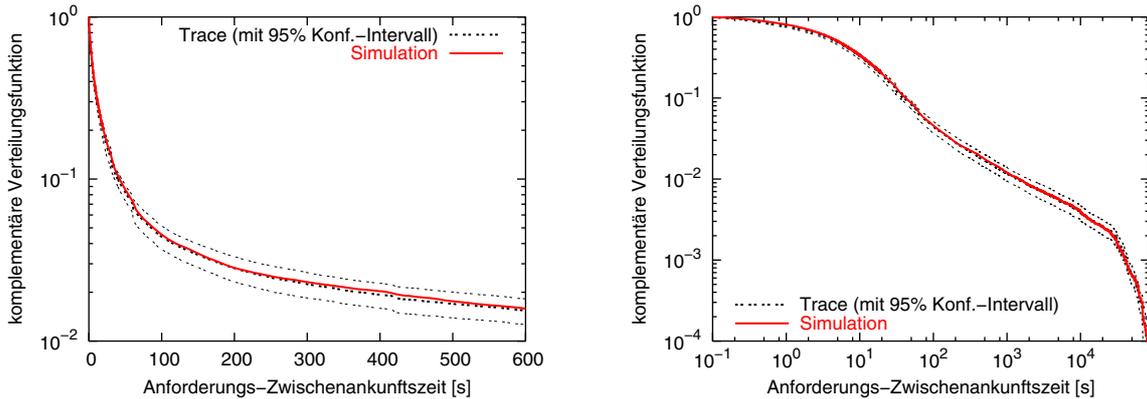


Bild 3.7: Anforderungs-Zwischenankunftszeit bei Messung und Simulation mit negativ-exponentiell verteilter Geduldsspanne ($E[T_G]=100s$).

(Anzahl der eingebetteten Objekte) sowie der Netzanbindung (Bandbreite) dynamisch ermittelt, wäre möglich, aber sehr komplex.

Kommerzielle Web-Entwickler setzen oft die 8-Sekunden-Regel [179] ein, nach der die typische Nutzergeduld zwischen 8 und 64 Sekunden beträgt. In [189] wurden Untersuchungen zu typischen Ladezeiten der populärsten Webseiten mit dem Ergebnis durchgeführt, daß bei einer Anschlußrate von 28 kbit/s die Ladezeiten der meisten Seiten etwa 30s betragen. Diese Werte haben sicherlich nur vorübergehende Gültigkeit, zeigen aber, daß auch andere Untersuchungen zu Geduldsspannen in der hier gefundenen Größenordnung geführt haben.

3.2.2.3 Approximation der Modellparameter

Nach der Identifizierung der nutzererzeugten HTTP-Anforderungen können alle wesentlichen Modellparameter, die in Tabelle 3.7 festgelegt wurden, bestimmt und statistisch ausgewertet werden. Die Auswertung konzentriert sich auf die Verteilungsfunktion, da bei vielen Parametern sogenannte *Heavy Tails* erwartet werden, d.h. es besteht eine signifikante Wahrscheinlichkeit für extrem große Werte. Somit ist die Aussagekraft von Mittelwerten und Variationskoeffizienten gering. Wie bereits in Kapitel 2.4.4.3 erklärt wird, können Verkehrsgrößen mit Verteilungen mit *Heavy Tail* das Verkehrsverhalten signifikant beeinflussen und sollten daher bei der Approximation berücksichtigt werden. Die Verteilungsfunktionen werden im folgenden komplementär in halb-logarithmischer oder doppelt-logarithmischer Darstellung gezeigt, da der Verlauf des *Tails* so besonders gut zu erkennen ist.

3.2.2.4 Parameter der Nutzerschicht

Die wichtigsten Modell-Parameter der Nutzerschicht sind die Nutzergeduld, die bereits in Abschnitt 3.2.2.2 bestimmt und mit einer Näherung beschrieben wurde, sowie die Betrachtungsdauer einer geladenen Seite.

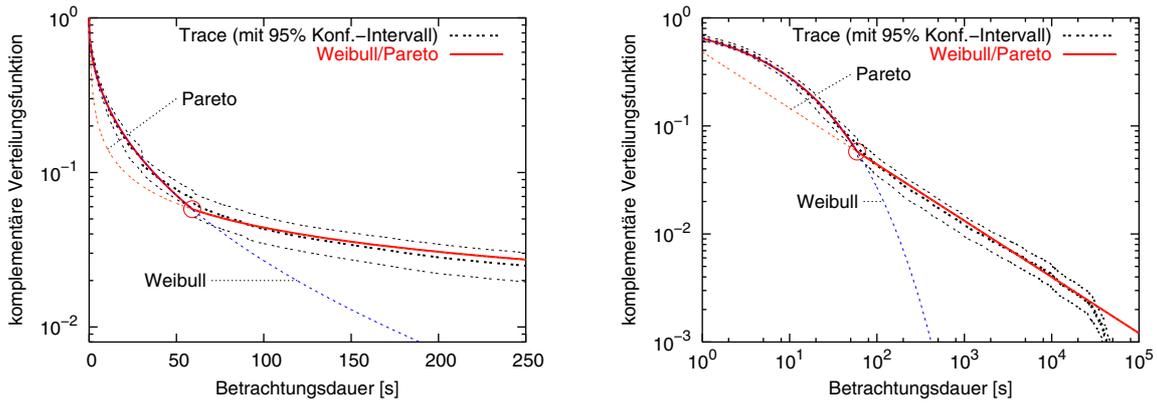


Bild 3.8: Approximation der Betrachtungsdauer.

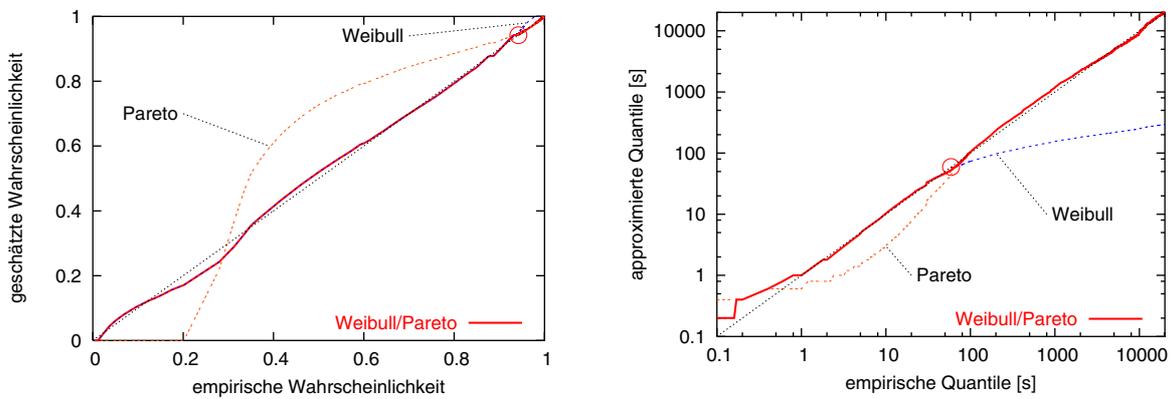


Bild 3.9: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

Die beobachteten Zeiten vom Abschluß eines Seitenladevorgangs bis zur nächsten Nutzeranforderung erstrecken sich über mehrere Größenordnungen. Da das Ende einer Sitzung nicht erkennbar ist, umfaßt die Betrachtungsdauer auch Pausen zwischen zwei Sitzungen. Folglich werden extrem kurze Wartezeiten im Sekundenbereich genauso wie stunden- oder tagelangen Pausen beobachtet. Eine einfache Beschreibung des charakteristischen Verlaufs der Verteilungsfunktion (siehe Bild 3.8) läßt sich am ehesten mit einer Pareto-Verteilung erreichen, die das *Tail*-Verhalten gut nachbildet. Im Bereich kleiner Werte weicht diese Näherung aber deutlich ab. Daher empfiehlt sich eine Kombination aus Weibull für den Kopf und Pareto für den Schwanz der Verteilung. Der Übergang zwischen beiden Funktion erfolgt dabei gemäß der in Kapitel 2.4.4.3 vorgestellten Approximationsmethode genau am Schnittpunkt der beiden, sich kreuzenden Verteilungsfunktionen (in Bild 3.8 gekennzeichnet).

Die Genauigkeit dieses Ansatzes kann anhand von PP- und QQ-Plots beurteilt werden (Bild 3.9). Der lineare PP-Plot zeigt deutlich die Abweichung der Pareto-Lösung für kleine Wahrscheinlichkeiten der Verteilungsfunktion. Entscheidender für die Erzeugung von Zufallsvariablen ist allerdings die Güte des QQ-Plots, da hier sichtbare Abweichungen zwischen empirischen und geschätzten Quantilen direkt eingehen. Der Verlauf zeigt gute Übereinstimmung im "normalen" Pausen-Bereich bis 2 Stunden aber Abweichungen für große Pausen.

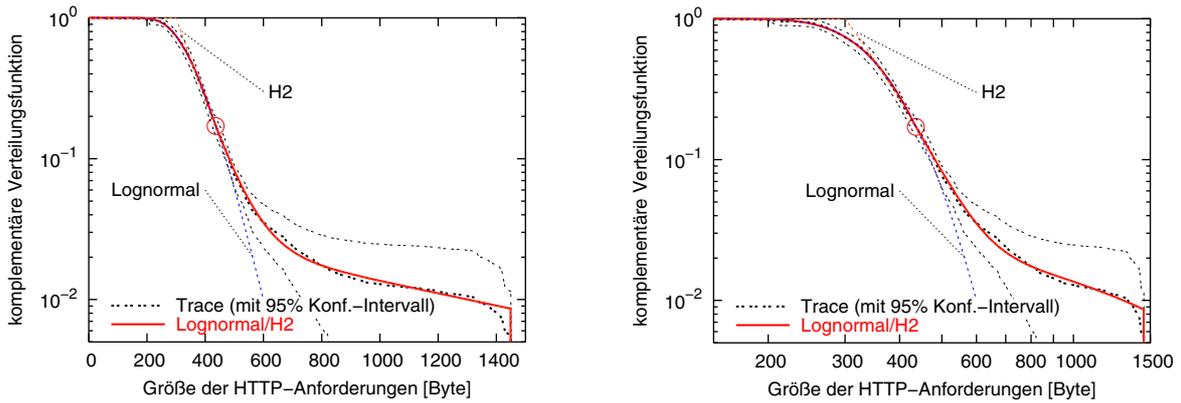


Bild 3.10: Approximation der Anforderungsgröße.

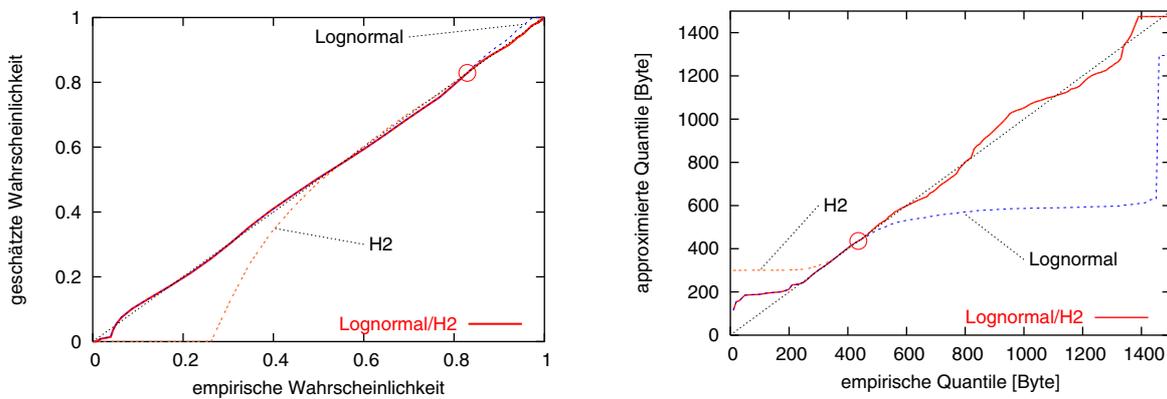


Bild 3.11: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

Wenn Sitzungspausen meßbar oder ausreichend genau abschätzbar sind, so kann das Modell um Prozesse zur Beschreibung von Sitzungsdauer und Sitzungsankünften erweitert werden. Diesbezüglich sind die Ergebnisse in [69,70] (Charakterisierung von Internet-Sitzungsverhalten bei Telefon- und ISDN-Einwahl) sowie [114,118] (Charakterisierung von Sitzungen aus Sicht des WWW-Servers) verwendbar.

3.2.2.5 Parameter der Dienstsicht

Die Modell-Parameter der Dienstsicht beschreiben die Beschaffenheit der HTTP-Kommunikation, welche durch die WWW-Seiten sowie die HTTP-Anforderungen charakterisiert ist.

Die Größe der HTTP-Anforderungsnachrichten ist in der Regel klein. Die Auswertung beschränkt sich auf Größen bis 1460 Bytes, d.h. der Größe eines IP-Paketes und faßt primäre und sekundäre Anforderungen zusammen. Die Größen der Kopffelder von IP, TCP und HTTP sind in der vorgestellten Beschreibung bereits enthalten.

Der in Bild 3.10 gezeigte Verlauf der Verteilung ähnelt bis auf den Kopfbereich einer hyperexponentiellen Verteilung zweiter Ordnung (H2). Im Kopfbereich kennzeichnet eine Spitze um 400 Bytes die Verteilungsdichtefunktion (nicht abgebildet). Durch eine Ergänzung mit einer

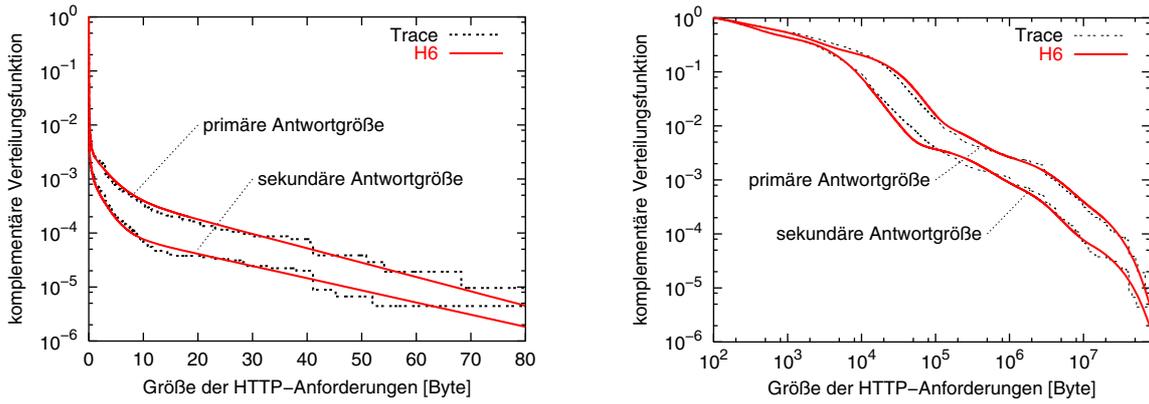


Bild 3.12: Approximation der HTTP-Antwortgrößen.

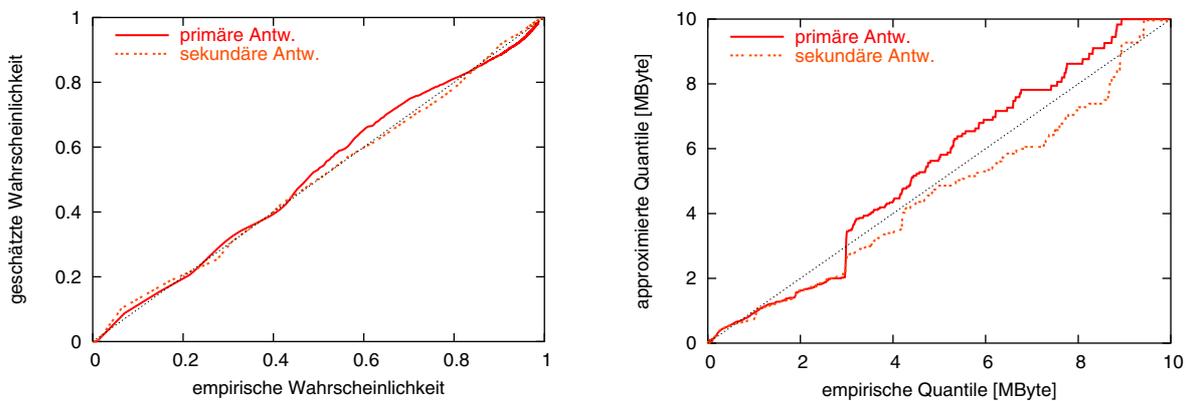


Bild 3.13: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

Lognormalverteilung kann auch dieser Bereich sehr genau beschrieben werden, so daß eine Kombination von Lognormal- und hyperexponentieller Verteilung die Kurve gut beschreibt, wie PP-Plot und QQ-Plot in Bild 3.11 zeigen.

WWW-Seiten sind durch die Größe des Basisdokuments, die Anzahl der darin eingebetteten Objekte und deren Größen bestimmt. Obwohl sich die Größe von Basisdokumenten und eingebetteten Objekte ähneln, werden sie getrennt modelliert. Die Länge der HTTP-Kopffelder wird bei der folgenden Beschreibung bereits berücksichtigt, wodurch eine Mindestantwortgröße von etwa 100 Bytes (inklusive aller TCP- und IP-Kopffelder) entsteht. Dieser Offset muß später wieder zu den hier beschriebenen Zufallsgrößen addiert werden.

Die Größen der HTTP-Antworten weisen Schwankungen über mehrere Größenordnungen auf. Die halb-logarithmische Darstellung in Bild 3.12 zeigt, daß die Verteilung für primäre und sekundäre Antwortgrößen annähernd exponentiell ausläuft. Damit kann eine hyperexponentielle Verteilungsfunktion höherer Ordnung den charakteristischen Verlauf ausreichend gut nachbilden. Die hier vorgestellte Approximation basiert auf der FW-Methode [76]. Die doppelt-logarithmische Darstellung zeigt, daß der unruhige Verlauf in beiden Fällen gut durch eine hyperexponentielle Verteilung 6. Ordnung (H6) nachgebildet werden kann. Die PP- und QQ-Plots in Bild 3.13 zeigen die Güte der Schätzungen.

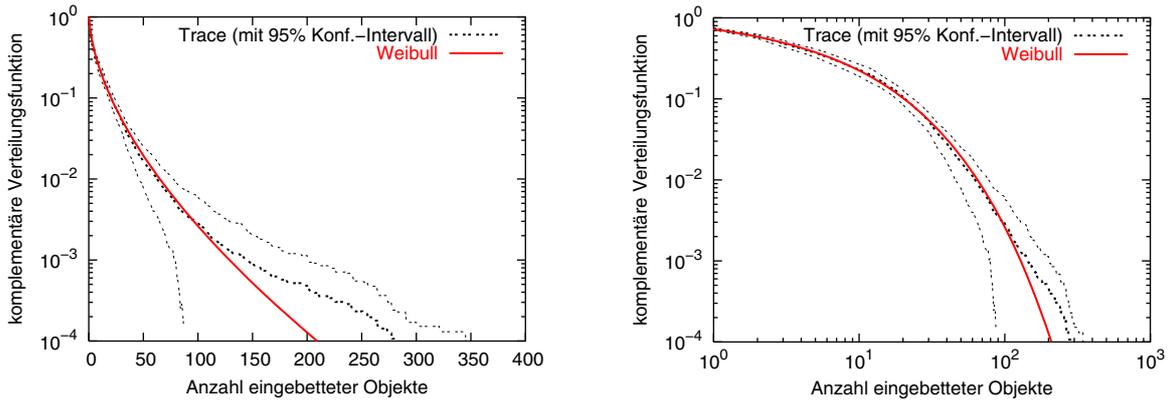


Bild 3.14: Approximation der Anzahl der eingebetteten Objekte.

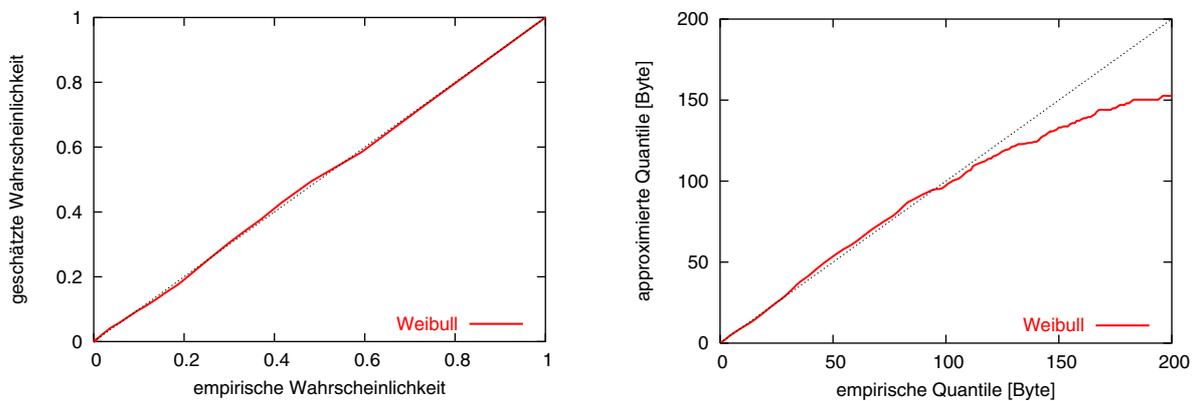


Bild 3.15: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

Ein Großteil der beobachteten Seiten enthält keine eingebetteten Objekte (etwa 45%). Dagegen finden sich in manchen WWW-Seiten mehrere hundert Objekte. Bild 3.14 zeigt die Verteilung der Anzahl eingebetteter Objekte für WWW-Seiten mit mindestens einem referenzierten Objekt. Ihr Verlauf wird bis etwa 200 Objekte gut durch eine Weibull-Verteilung beschrieben. Obwohl die Abweichung für höhere Werte signifikant ist, wird auf eine genauere Approximation durch Kombination mit anderen Verteilungsfunktionen verzichtet, da die Weibull-Verteilung noch im Bereich der 95%-Konfidenzintervalle liegt und offensichtlich kein *Heavy Tail* auftritt (Bild 3.15).

WWW-Portale werden mit zunehmender Komplexität immer weniger durch einfache WWW-Server mit statischem Inhalt realisiert, sondern durch hochdynamische Lösungen. Dabei werden Zugriffe transparent auf Server-Cluster verteilt und Inhalte werden erst auf Abruf zusammengestellt. Während solche Architekturen eine deutlich bessere Skalierung und flexiblere Einsatzmöglichkeiten gewährleisten, führen sie aber auch merkliche Verzögerungen ein. Diese Verzögerung kann in einer Größenordnung liegen, die in der Modellierung nicht vernachlässigt werden darf.

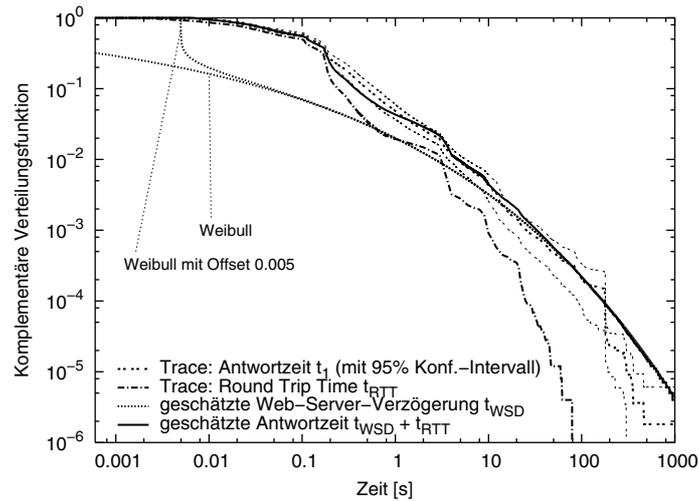


Bild 3.16: Schätzung der Verzögerung durch den WWW-Server.

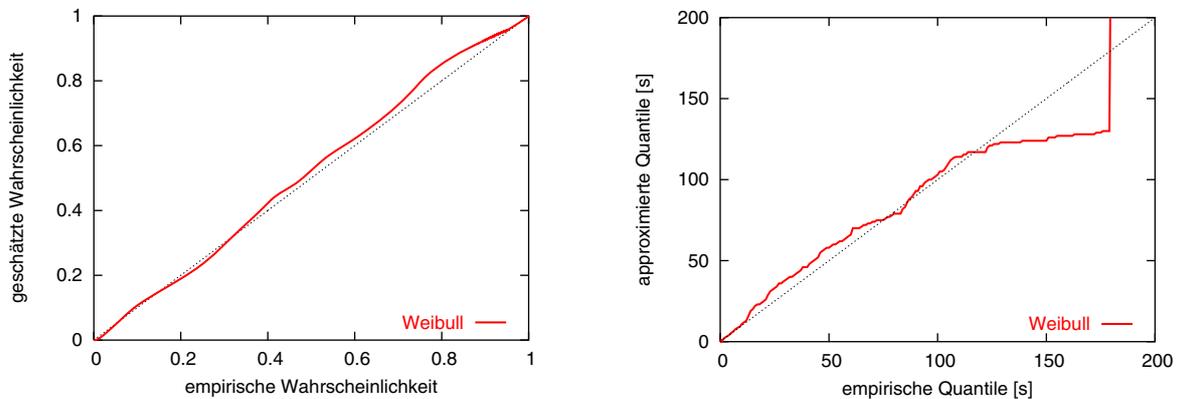


Bild 3.17: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

Die Wartezeit auf der Seite des Servers vom Eintreffen der Anforderung bis zum Absenden der Antwort hängt vom betrachteten Server-System ab. Für spezifische Einzelsysteme können durch direkte Messung sehr genaue statistische Beschreibungen gefunden werden, aber zur Charakterisierung von WWW-Verkehr zwischen Browsern und unterschiedlichen Servern unbestimmten Typs sind Beschreibungen von einzelnen Servern nur in großer Zahl hilfreich. Alternativ erlauben die Messungen, die im Rahmen dieser Arbeit durchgeführt wurden eine Abschätzung der Größenordnung und des qualitativen allgemeinen Verhaltens. Da der Meßpunkt in der Nähe der Clients liegt, beinhaltet die beobachtete Reaktionszeit der angesprochenen WWW-Server auch die Umlaufzeit (*Round Trip Time*) des Internet, so daß mit Kenntnis der typischen Umlaufzeit auch die Verteilung der Server-Verzögerung geschätzt werden kann.

Bild 3.16 zeigt die beobachteten Verteilungen der Umlaufzeit und der Antwortzeit, welche aus der Summe von Umlaufzeit und Server-Verzögerung besteht. Der ähnliche Verlauf der Antwortzeit- und Umlaufzeitverteilung verrät, daß die Antwortzeit bis etwa 3 Sekunden von der Umlaufzeit dominiert wird. Für größere Werte weicht sie deutlich ab, ist dann also im Wesentlichen durch den zweiten Term, die Web-Server-Verzögerung bestimmt. Die gesuchte Vertei-

lung für die Web-Server-Verzögerung soll daher den Schwanzverlauf der Antwortzeit möglichst gut nachbilden, aber im Kopfbereich unter die Umlaufzeit fallen. Wie Bild 3.16 zeigt, erfüllt eine auf den logarithmischen Schwanzbereich approximierte Weibull-Verteilung diese Bedingungen zufriedenstellend. Durch einen geringen Offset kann der Verlauf im Bereich sehr kleiner Zeiten – der nicht mehr allein durch die Umlaufzeit bestimmt ist – genauer angenähert werden. Aus der Summe der Zufallsvariablen der so geschätzten Web-Server-Verzögerung und der Umlaufzeit ergibt sich die abgebildete Verteilung der geschätzten Antwortzeit, die recht gut mit der beobachteten Antwortzeit übereinstimmt (Bild 3.17).

Für die Näherung dieser Größe kann kein effektives systematisches Vorgehen angegeben werden, da die zu approximierende Verteilung nur als ein Term der Summe zweier Zufallsvariablen definiert ist, und sich nicht einfach, beispielsweise durch eine Differenzbildung, direkt darstellen läßt. Somit kann auch die Güte einer Testfunktion nur nach Erzeugung der Summenverteilung und Vergleich mit der Ziel-Summenverteilung festgestellt und bewertet werden. Damit ist einerseits die Wahl eines geeigneten Verteilungstyps und dessen Startparameter schwierig und andererseits ist für jede Iteration ein relativ langer Prozeß der Summenbildung (Faltung bzw. Simulation) notwendig. Aufgrund des allgemeinen Charakters der hier beschriebenen Server-Verzögerung, genügt an dieser Stelle die gezeigte Schätzung für die nachfolgenden Betrachtungen. Es empfiehlt sich noch, eine Obergrenze für die Server-Verzögerung einzuführen, da WWW-Server typischerweise eine Zeitüberwachung einsetzen, um Blockierungen zu erkennen und dem Nutzer gegebenenfalls eine Fehlermeldung zukommen zu lassen. Der QQ-Plot zeigt, daß auch die Näherung ab etwa 120 Sekunden nicht mehr sehr genau ist.

3.2.2.6 Parameter der Kommunikationssteuerungsschicht

Für die Parameter der Kommunikationssteuerungsschicht wird keine stochastische Beschreibung angegeben, da sich diese Werte im Rahmen des Anwendungsbereichs des Modells nicht stochastisch verhalten. Die Kombination von WWW-Browser und WWW-Server legt zunächst die eingesetzte HTTP-Protokollversion fest. Sowohl für HTTP/1.0 als auch für HTTP/1.1 können persistente Verbindungen angenommen werden, da ältere Anwendungen, die nur HTTP/1.0 unterstützen, in der Regel auch den äquivalenten "keep alive"-Mechanismus einsetzen. Pipelining wird selbst bei HTTP/1.1 nur wenig eingesetzt. Weitere Anwendungsparameter, wie Zeitüberwachungsmechanismen oder die maximal zulässige Anzahl paralleler Verbindungen, unterscheiden sich von Produkt zu Produkt oder werden sogar vom Nutzer festgelegt, so daß sich eine so große Vielfalt ergibt, die sich höchstens in ständigen und verteilten Langzeitmessungen statistisch beschreiben läßt. Für den Einsatz des Verkehrsmodells müssen hierfür szenariospezifisch vereinfachende Annahmen getroffen werden.

3.2.3 Modellparameter

Die ermittelten Beschreibungen der Parameter des WWW-Verkehrsmodells sind in Tabelle 3.10 zusammengefaßt. Parameterbeschreibungen für eine weitergehende Modellierung des Sitzungsverhaltens können aus [69,70,114,118] entnommen werden.

Tabelle 3.10: Parametrisierung des WWW-Modells

Parameter		Beschreibung
Nutzer	Geduldsspanne [s]	<ul style="list-style-type: none"> NegExp(100)
	Betrachtungsdauer (sec)	<ul style="list-style-type: none"> Weibull/Pareto <ul style="list-style-type: none"> Weibull($\alpha=0.45259792$, $\beta=5.83644690$) (mit Offset von 0.05470528) Pareto($\alpha=0.24819309$; $k=0.52074007$) Schnittpunkt bei $x=59.075524607$
Dienst	Anforderungsgröße (Bytes)	<ul style="list-style-type: none"> LogNormal/HyperExp 2. Ordnung (H2) <ul style="list-style-type: none"> LogNormal($\sigma=0.23225805$, $\mu=5.85608752$) H2($m_1=996$, $p_1=0.02732672$, $m_2=71.8$, $p_2=1-p_1$) (mit Offset von 300) Schnittpunkt bei $x=435.711717$
	Basis-Dokumentgröße (kbytes)	<ul style="list-style-type: none"> HyperExp 6.Ordnung <ul style="list-style-type: none"> $m_1=16342864$ $p_1=0.0006028$ $m_2=2673791$ $p_2=0.002872$ $m_3=204021$ $p_3=0.01194$ $m_4=25000$ $p_4=0.2800$ $m_5=1500$ $p_5=0.450$ $m_6=150$ $p_6=0.2545852$ (mit Offset von 100Bytes)
	Größe v. eingeb. Objekten (Bytes)	<ul style="list-style-type: none"> HyperExp 6.Ordnung <ul style="list-style-type: none"> $m_1=19303269$ $p_1=0.0001157$ $m_2=2017768$ $p_2=0.001135$ $m_3=261214$ $p_3=0.00364$ $m_4=11839$ $p_4=0.0910$ $m_5=3896$ $p_5=0.425$ $m_6=220$ $p_6=0.4791093$ (mit Offset von 100Bytes)
	Anzahl eingeb. Objekte	<ul style="list-style-type: none"> 45%: keine Inlines, sonst Weibull ($\alpha=0.59187227$; $\beta=4.90819582$) (mit Offset von 0.28)
	Web-Server-Verzögerung	<ul style="list-style-type: none"> Weibull ($\alpha=0.16631565$; $\beta=0.00027277$)

Zur korrekten Bewertung dieser Parametrisierung muß angemerkt werden, daß von einer vereinfachten Sicht des WWW-Verkehrs und Nutzerverhaltens ausgegangen wurde. Viele Teilnehmer nutzen ausgiebig mehrere Browser-Fenster, um durch parallele Abfrage unterschiedlicher Inhalte geringere Wartezeit vor leeren Browser-Fenstern zu verbringen. Dieses Verhalten

läßt sich wiederum von Netzseite kaum erkennen und führt zu untypischem Einzelnutzerverhalten, das eigentlich durch Überlappung mehrerer verwobener Sitzungen desselben Nutzers entsteht. Bei schnellen Anbindungen wird dieses Nutzerverhalten tendenziell allerdings immer weniger sinnvoll so daß zu erwarten ist, daß dieser Effekt aufgrund der sehr guten Anbindung im beobachteten Netz nicht dominiert.

Weiterhin sind die eingesetzten Heuristiken zur Erkennung des Sitzungsverhaltens nicht auf die Erkennung von verschachtelten HTTP-Abrufen bei Pipelining ausgelegt. Bei auftretendem (unerkanntem) Pipelining werden daher zu große Antworten und zu wenige eingebettete Objekte registriert. Das übertragene Gesamtvolumen ist allerdings gleich, so daß sich ein Fehler nur in der Abhängigkeit der eingebetteten Objekte und der daraus resultierenden Verzögerungen auswirkt.

Bezüglich der Approximation bestehen sehr viele Freiheiten in der Wahl und Anwendung der Approximationsverfahren. Je nach dem auf welche Aspekte einer Verteilung Wert gelegt wird, kann eine Optimierung vorgenommen werden, wie z.B. durch die Betrachtung der doppelt logarithmischen Darstellung, die das *Tail*-Verhalten betont und den eigentlich dominanten Kopfteil einer komplementären Verteilung weniger stark gewichtet.

3.3 Bewertung des WWW-Verkehrsmodells

Das in Kapitel 3.1 vorgeschlagene Verkehrsmodell für den Dienst WWW ist durch die Einführung von Rückkopplungen auf Nutzer- und Dienstebene motiviert. Die in Kapitel 3.2 erfolgte Parametrisierung beruht auf der Auswertung bzw. auf der Schätzung von Verkehrswerten auf Grundlage der Verkehrsmessung. Das Verkehrsmodell, im folgenden als reaktiver WWW-Lastgenerator (RWLG) bezeichnet, wird nun in simulativen Untersuchungen eingesetzt, um die Stabilität seiner geschätzten Parameter, den Einfluß seiner charakteristischen Rückkopplungen sowie seine Einsatzmöglichkeiten zu bewerten (Abschnitt 3.3.3). Zuvor werden aber in Abschnitt 3.3.1 die Randbedingungen der simulativen Untersuchungen und das zugrundegelegte Simulationsszenario beschrieben. In Abschnitt 3.3.2 werden die zugrundegelegten Bewertungskriterien dargelegt.

3.3.1 Randbedingungen der Simulation

3.3.1.1 Implementierung

Die Einbindung der Transportebene in das Kommunikationsmodell des WWW-Lastgenerators erfordert den Einsatz eines Modells der TCP-Mechanismen insbesondere für Quittierung und Datenflußsteuerung. In Ergänzung zu der hier eingesetzten Simulationsbibliothek Simlib [172] wurde eine solche TCP-Erweiterung in [116] erarbeitet. In einer weiterentwickelten Form wird

sie im folgenden zur Steuerung der Ende-zu-Ende-Übertragung auf der Transportschicht eingesetzt.

Die Simulationsbibliothek Simlib stellt Objektklassen für die Modellierung von Kommunikationssystemen, die Simulationssteuerung sowie das Einlesen statistischer Parametergrößen und die Erfassung und statistische Verarbeitung von Meßwerten zur Verfügung. Die TCP-Erweiterung beinhaltet Klassen zur Modellierung von TCP-Verbindungen, welche nach Initiierung einer Übertragung für den vollständigen Empfang der Nachricht sorgen – unter Berücksichtigung von Datenflußsteuerung und verlustbedingten Paketwiederholungen. Einschränkend kann eine Übertragung dabei nur von einem Sender zum Empfänger erfolgen, d.h. ausgenommen von Quittierungen unterstützen die Verbindungen keinen Datenfluß in der Rückrichtung.

Zur Bewertung und Demonstration des reaktiven WWW-Lastgenerators wurde er mit Hilfe der erweiterten Simulationsbibliothek implementiert. Die unidirektionalen TCP-Verbindungen werden nur zur Simulation der Übertragung der HTTP-Antworten eingesetzt. Die Übertragungszeiten der zugehörigen HTTP-Anforderungen werden zuvor unter Annahme der Verlust-

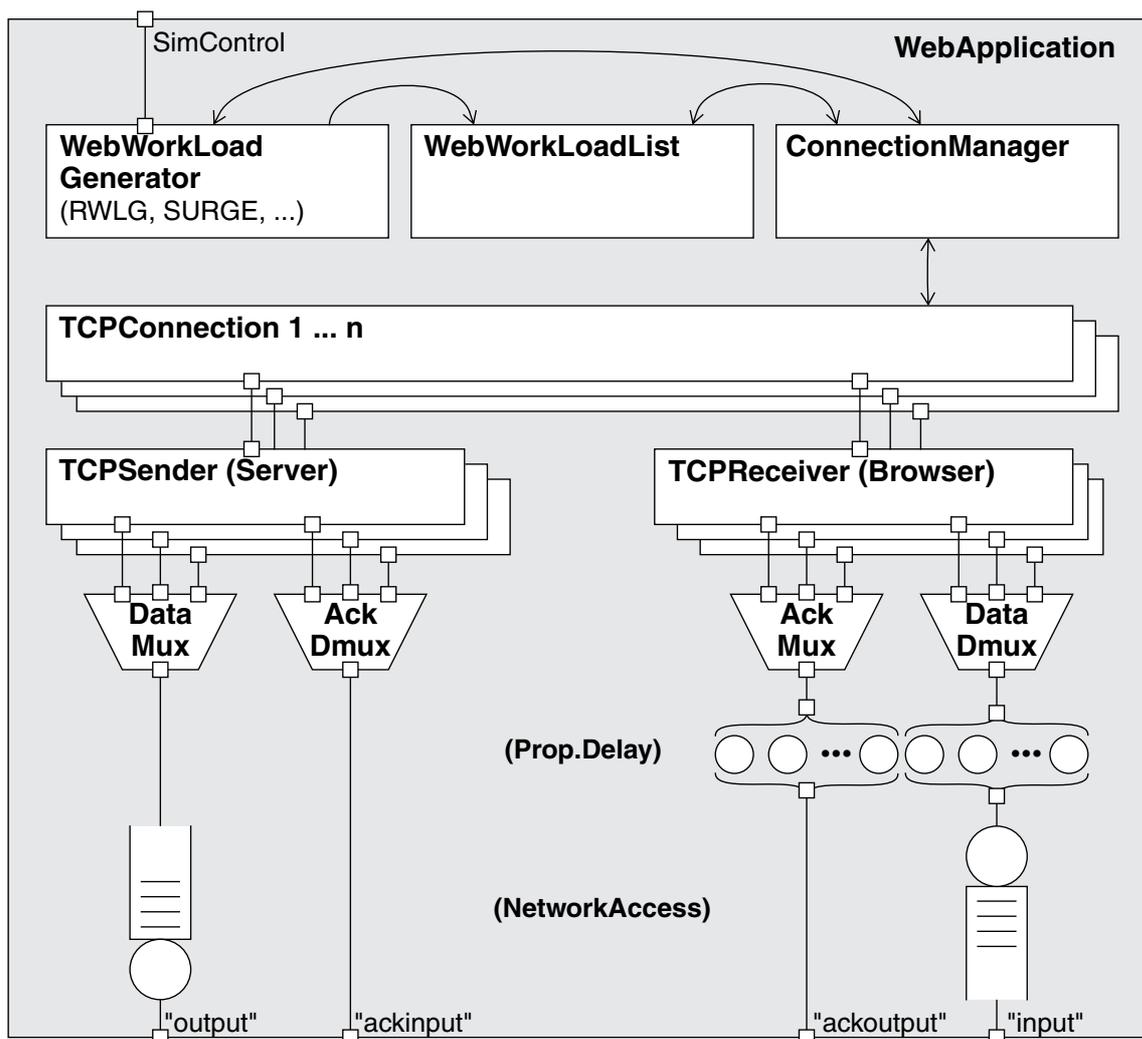


Bild 3.18: Komponenten des WWW-Verkehrsmodells

und Verzögerungsschwankungsfreiheit für den *Upstream* berechnet und bestimmen die Verzögerung des Übertragungsbeginns der HTTP-Antworten.

Bild 3.18 zeigt die Komponenten des WWW-Verkehrsmodells. Wesentlicher Bestandteil des Modells ist der eigentliche Lastgenerator (*WebWorkLoadGenerator*), der für die Erzeugung von Nutzeranforderungen sowie Anzahl und Größe der HTTP-Abrufe der entsprechenden WWW-Seite zuständig ist. Erfolgte Nutzeranforderungen werden zur Steuerung der Simulationsdauer eingesetzt. Der Lastgenerator legt die Daten der zu übertragenen Seite zentral ab (*WorkLoadList*) und informiert den Verbindungsmanager (*ConnectionManger*), der noch aktive Übertragungen abbricht und mögliche neue HTTP-Abrufe einleitet. Im Verbindungsmanager werden Abhängigkeiten der HTTP-Abrufe geprüft und mögliche Übertragungen freien Verbindungen zugeteilt. TCP-Sender und TCP-Empfänger werden unter einer Verbindung zusammengefaßt und sind über Multiplexer und optionale Elemente für Laufzeitverzögerungen und Zugangsrate an die Kommunikationsschnittstellen des Modells angeschlossen.

Neben dem reaktiven Lastgenerator wurde auch ein Lastgenerator entsprechend dem SURGE-Modell implementiert, der unabhängig von Rückmeldungen operiert und die Übertragung aller HTTP-Abrufe explizit auslösen muß. Simulationen von SURGE-Quellen benötigen aufgrund der Bestimmung der Übertragungszeitpunkte aller eingebetteten Objekte deutlich mehr Rechenzeit als für das rückgekoppelte Modell benötigt wird.

3.3.1.2 Simulations-Szenarien

Für die simulative Untersuchung eines rückgekoppelten Verkehrsmodells ist auch der Einsatz eines Netzmodells notwendig, das Verzögerungen und Verluste einführt. Um die Ergebnisse der Studien nicht unnötig komplex zu machen, wird ein Minimalmodell gewählt, das in Bild 3.19 dargestellt ist. Es wird durch eine begrenzte Warteschlange mit Bedieneinheit konstanter Bediendauer für die Übertragungsrate im Kernnetz und ein Verzögerungselement für die Signallaufzeit beschrieben. Wahlweise können auch zufällige Paketverluste auftreten. Im WWW-Verkehrsmodell ist zudem die Möglichkeit einer Begrenzung der Netzzugangsrate gegeben. Da Paketverluste auch bei geringer Link-Auslastung stattfinden sollen, wird die Verlustwahrscheinlichkeit in der Regel auf 0.01 gesetzt, so daß sporadisch Verzögerungen durch erzwungene TCP-Paketwiederholungen erzeugt werden. Für die Signallaufzeit werden 200 ms angenommen. Die Wahl einer konstanten Laufzeit führt allerdings eine signifikante charakteristische Eigenschaft der Übertragungsdauer ein, die beim Vergleich von Simulationsergebnissen mit dem original gemessenen Verkehr berücksichtigt werden muß. Für die Link-Rate wird zunächst 1Mbit/s gesetzt, das für wenige Quellen ausreichend ist – für die Betrachtung von sehr vielen überlagerten Quellen wird sie auf 1Gbit/s gesetzt, um die Charakteristik des aggregierten Verkehrs ungestört zu erfassen. Die Begrenzung der Zugangsraten der Quellen ist zunächst für die Betrachtung des Einzelverkehrs (Abschnitt 3.3.3.1) unbedeutend und wird erst bei der Betrachtung von überlagertem Verkehr (Abschnitt 3.3.3.2) und im Anwendungsszenario (Abschnitt 3.3.3.3) gesetzt.

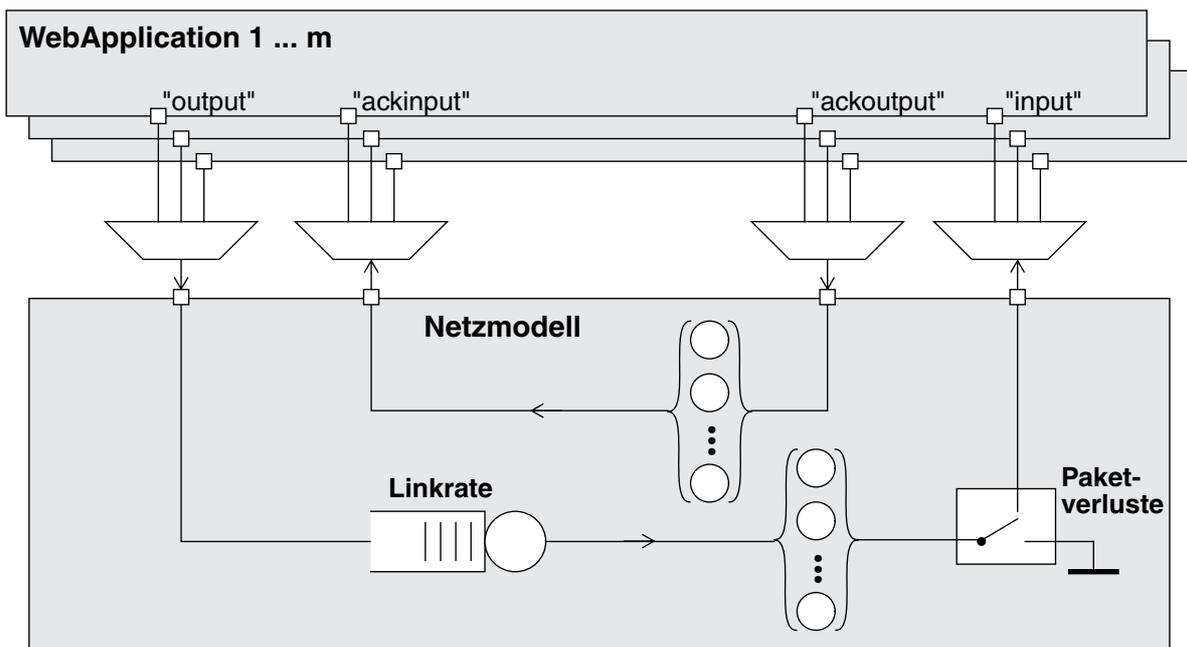


Bild 3.19: Komponenten des rückgekoppelten WWW-Verkehrsmodells mit angeschlossenem Netzmodell der simulierten Szenarien

Die WWW-Verkehrsquelle verwendet in der Regel maximal 4 parallele Verbindungen und der Server antwortet mit einer mittleren Verzögerung von 250 ms. Alle Grundvorgaben für die Simulations Parameter sind auch in Anhang B.1 aufgeführt.

Die Simulationsdauer wird über die Anzahl erfolgter Seitenanforderungen gesteuert. Da insbesondere Statistiken bzgl. der Seiten-Eigenschaften erhoben werden, muß also eine ausreichend große Anzahl von Nutzeranforderungen erfolgen. In den nachfolgenden Simulation wurden in der Regel 10 x 20000 WWW-Seiten abgerufen.

3.3.2 Bewertungskriterien

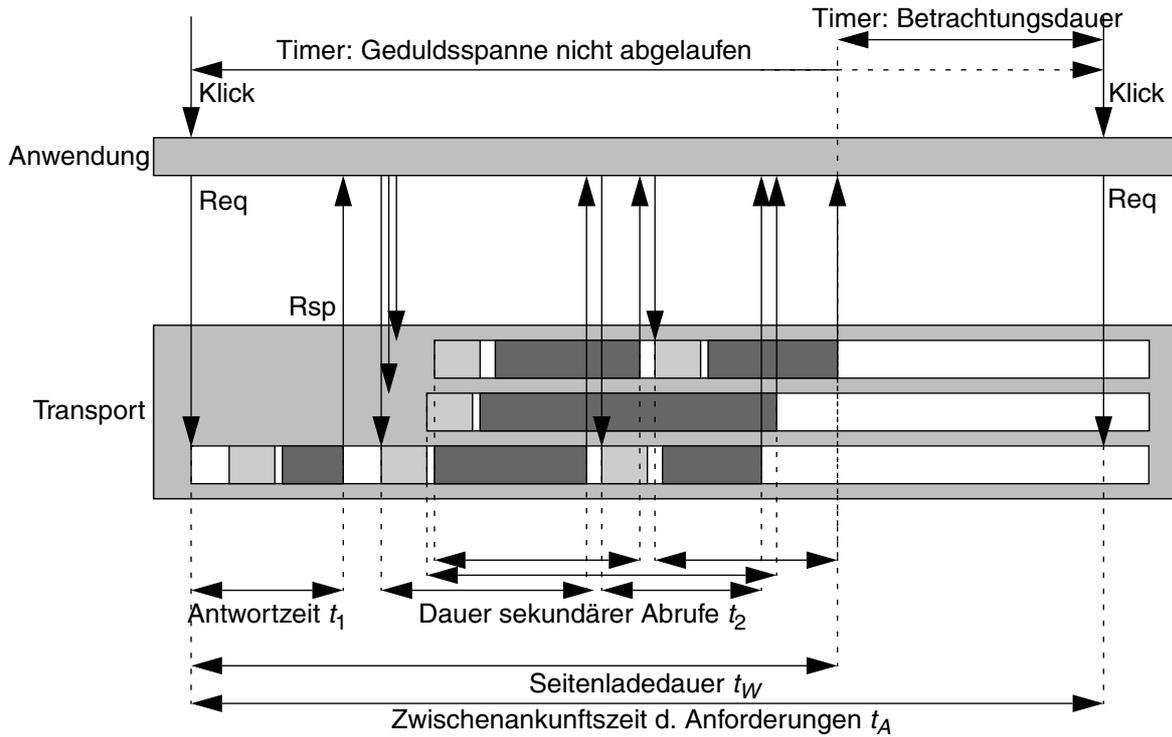
Die Bewertung der Simulationen kann auf Dienstebene und auf Paketebene erfolgen. Die Nutzersicht des Verkehrsmodells erlaubt eine quantitative Bewertung der erreichten Dienstgüte und bezieht sich auf eine einzelne WWW-Quelle. Wie bereits in Kapitel 3.1.2 beschrieben wird, soll die Bewertung anhand der folgenden Verkehrsgrößen erfolgen:

- Ladedauer einer WWW-Seite t_W
- Antwortzeit t_1 (Dauer des primären Abrufs)
- Dauer einzelner sekundärer Abrufe t_2
- Abbruchwahrscheinlichkeit des Ladevorgangs p_U (Ungeduld)

In Bild 3.20 sind diese Größen für den normalen Ladevorgang sowie für den Abbruch eines Ladevorgangs veranschaulicht. Seiten- und Objektgrößen sind in der Regel von geringem Interesse. Sie sind stochastisch von der Verkehrsquelle vorgegeben und weichen nur bei Abbruch des Ladevorgangs davon ab. Für eine Bewertung sind sie eventuell bei Berücksichtigung von Nutzererwartungen interessant, die beispielsweise für große Dokumente größere Geduldsspannen und Bedeutung zuordnet. Eine so detaillierte Betrachtung soll an dieser Stelle aber nicht erfolgen. Die Zwischenankunftszeit der Seiten-Anforderungen ist ein Maß für das Verkehrsaufkommen und wird nur unterstützend herangezogen.

Direkt auf Paketebene sichtbare Verkehrscharakteristiken beschreiben weniger die vom Nutzer erfahrene Dienstgüte als vielmehr die erzeugte Netzbelastung. Für Linkauslastung oder Warteschlangen-Verhalten kann bei einigermaßen realitätsnaher Parametrisierung kein Effekt durch den Verkehr einzelner WWW-Quellen identifiziert werden. Erst bei Überlagerung mehrerer Quellen zu einem immer stärker büschelhaften Verkehr, zeigen sich Effekte der kurzzeitigen Überlast. Daher ist insbesondere die Büschelhaftigkeit des erzeugten Verkehrs interessant und als deren Folge der Grad der asymptotischen Selbstähnlichkeit, die bei Überlagerung mehrerer Quellen erzeugt wird. Die Betrachtung auf Paketebene fokussiert sich folglich auf Paket-Zwischenankunftszeiten, welche die hohe Dynamik des Verkehrs beschreiben.

vollständiger Seitenabruf



abgebrochener Seitenabruf

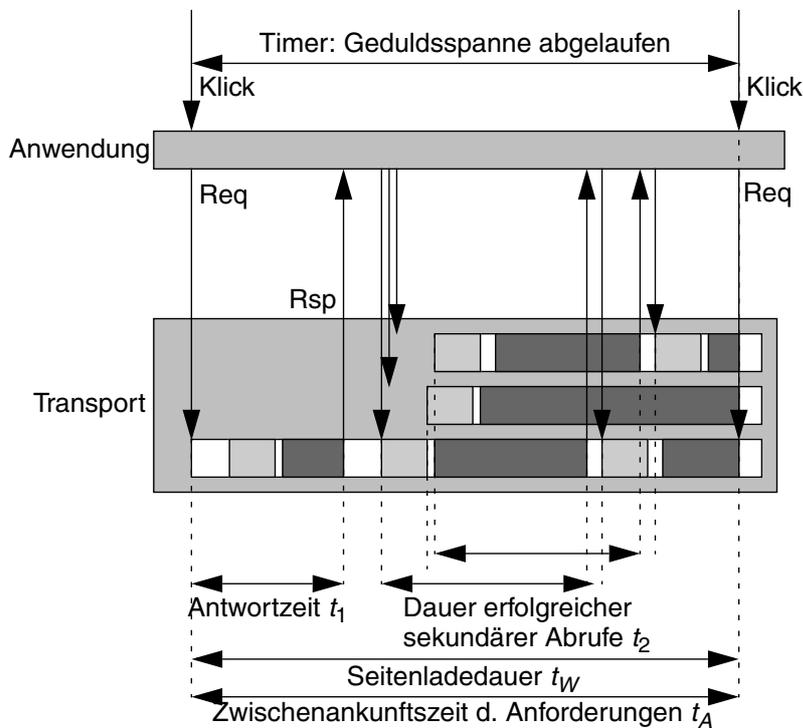


Bild 3.20: Verkehrsgrößen zur Bewertung der Dienstgüte von WWW-Diensten im Normalfall (oben) und bei Abbruch aufgrund von Nutzerungeduld (unten)

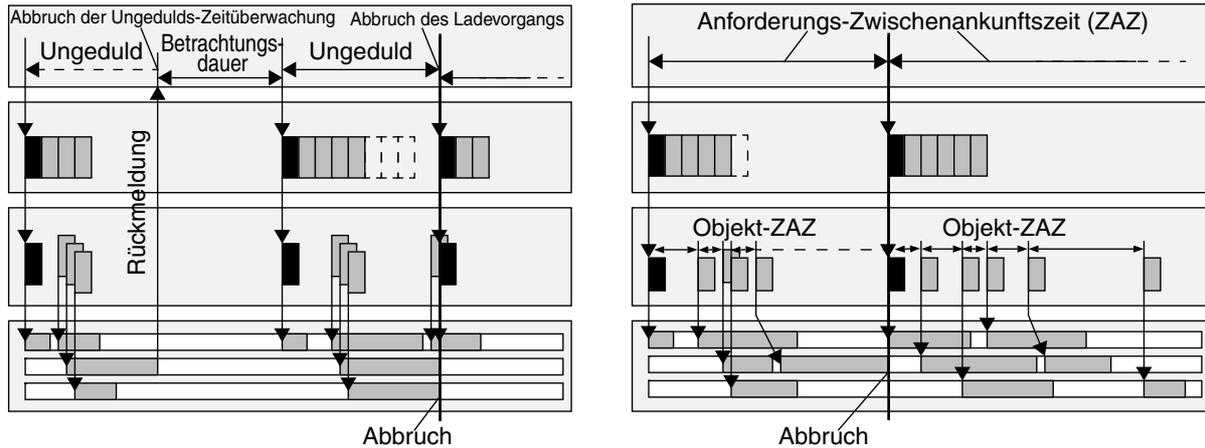


Bild 3.21: Gegenüberstellung des vorgeschlagenen reaktiven Verkehrsmodells (links) und SURGE als Referenz-Verkehrsmodell (rechts)

Um eine vergleichende Bewertung der betrachteten Größen zu ermöglichen, werden einige Simulationen auch mit einem Referenzverkehrsmodell durchgeführt. Hierfür wird ein angepasstes SURGE-Modell eingesetzt. Bei den Anpassungen handelt es sich im wesentlichen um das Aufsetzen des SURGE-Modells auf das selbe TCP-Verbindungsmanagement-Modul, welches die Anzahl parallel nutzbarer Verbindungen und die TCP-Eigenschaften festlegt. Insbesondere wird impliziert, daß bei Eintreffen einer neuen Nutzeranforderung alle aktiven oder wartenden Übertragungen abgebrochen werden und daß eintreffende sekundäre Anforderungen auf freie Verbindungen warten müssen. SURGE sieht zu Simulationsbeginn eigentlich die Festlegung eines Server-Verbunds mit einem festen Seitenangebot vor, aus welchem dann zur Simulationszeit die zu übertragenden Seiten ausgewählt werden. Für den simulativen Vergleich an dieser Stelle werden alle Seiten allerdings bei jeder Anfrage neu erzeugt. In Bild 3.21 werden die beiden Verkehrsgeneratoren gegenübergestellt. Während der reaktive Lastgenerator alle sekundären Abrufe nach abgeschlossenem primären Abruf zur Übertragung freistellt, werden die sekundären Abrufe beim SURGE-Modell zu zufälligen Zeitpunkten getätigt. Insbesondere sind die Nutzeranforderungen bei SURGE unabhängig von allen Vorgängen auf untergeordneten Modellebenen. Auch die Parameter des SURGE-Modells wurden entsprechend den gemessenen Verkehrsgrößen angepaßt, d.h. die empirischen Verteilungen für die Anforderungs- sowie die Objektzwischenankunftszeit wurden als Modellparameter eingesetzt und für Umfang der Seiten sowie Größen der Objekte wurden die selben Verteilungen eingesetzt wie für das reaktive Modell.

3.3.3 Bewertung des reaktiven WWW-Modells

3.3.3.1 Auswirkungen der Parameter-Schätzung auf die Modellstabilität

Die Bewertung erfolgt zunächst durch eine Betrachtung der mit der Parameter-Schätzung eingeführten Unsicherheit. Die Heuristiken zur Identifikation der durch den Nutzer ausgelösten primären HTTP-Anforderungen wurden bereits in Kapitel 3.2.2 bewertet. Die weiteren durch Schätzung erhaltenen Größen, deren Einfluß auf das Modellverhalten an dieser Stelle untersucht werden soll, sind die Geduldsspanne und die Verzögerung auf Seite des WWW-Servers.

Einfluß der Nutzergeduld

In Abschnitt 3.2.2.2 wurde die Nutzergeduld anhand der meßbaren Betrachtungsdauer und der Zwischenankunftszeit der Seitenanforderungen geschätzt. Es wird eine negativ exponentielle Verteilung mit Mittelwert von 100 s vorgeschlagen. Simulationen mit verschiedenen Mittelwerten für die Geduldsspanne und bei unterschiedlich starker Verzögerung durch den WWW-Server (t_{WSD} , *Web Server Delay*) machen den Einfluß der Nutzergeduld auf das Modellverhalten sichtbar. In erster Linie wirkt sich die Geduld natürlich auf die Häufigkeit der Ladeabbrüche aus, wie Bild 3.22 zeigt. Bei einem Mittelwert von 100 Sekunden für die Geduldsspanne bewegt sich diese Abbruchwahrscheinlichkeit noch in einem signifikanten Bereich.

Ladeabbrüche beeinflussen den erzeugten Verkehr durch Unterdrückung längerer Seiten und anschließender Anforderung neuer Seiten. Bild 3.23 und Bild 3.24 zeigen, wie sich Mittelwerte und Verteilungen der tatsächlich übertragenen Seitengröße bei unterschiedlicher mittlerer Geduldsspanne verhalten. Ab einer mittleren Geduldsspanne von etwa 100 Sekunden werden im Mittel nur noch wenig mehr eingebettete Objekte übertragen. Das mittlere Volumen der übertragenen Daten je Seite steigt allerdings mit zunehmender Geduld weiterhin an. Das unterschiedliche Verhalten beider Maße für die Seitengröße wird durch die höhere Varianz für die Objektgrößen erklärt.

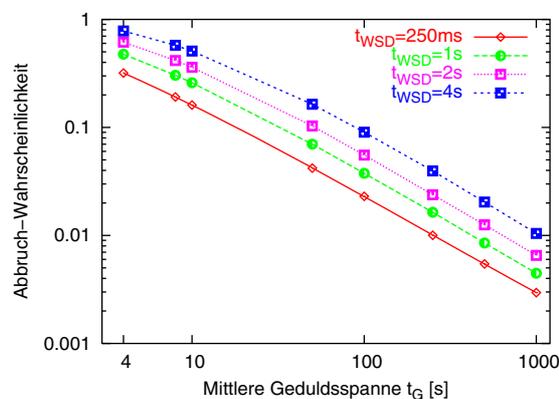


Bild 3.22: Wahrscheinlichkeit für einen Ladeabbruch in Abhängigkeit der mittleren Geduldsspanne (für verschiedene konstante Verzögerungen durch den WWW-Server)

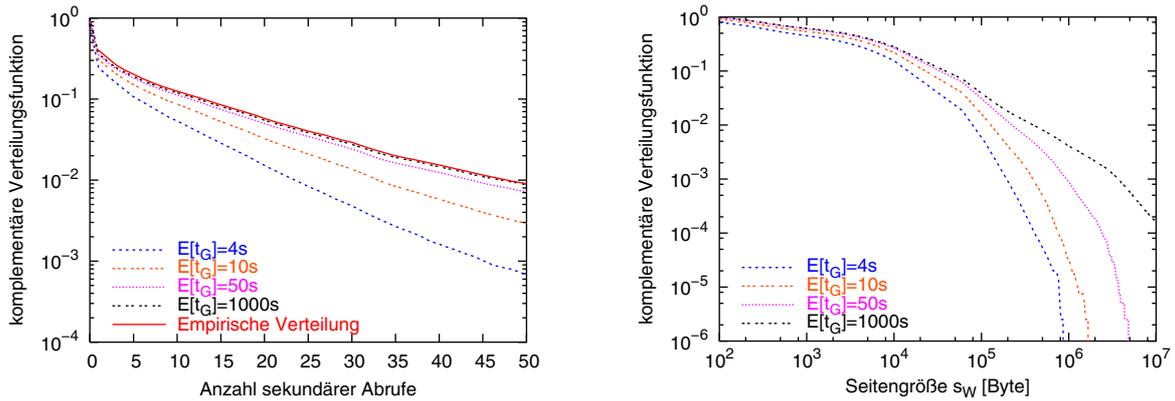


Bild 3.23: Größenverteilungen der tatsächlich geladenen WWW-Seiten bei unterschiedlichen mittleren Geduldsspannen (bei konstanter Server-Verzögerung von 250ms)

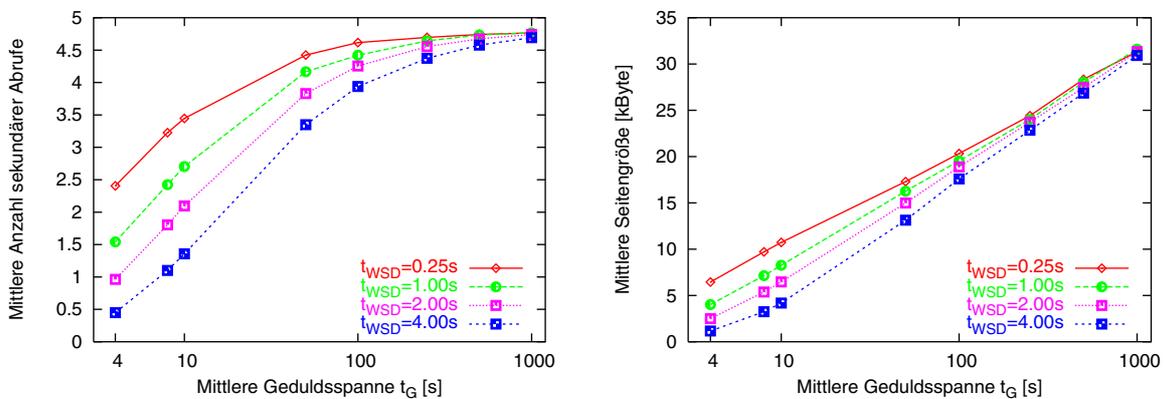


Bild 3.24: Mittlere Größe der tatsächlich geladenen WWW-Seiten bei unterschiedlichen mittleren Geduldsspannen (bei konstanter Server-Verzögerung)

Bei geduldigen Nutzern erhöht sich entsprechend die mittlere Seitenladedauer. Bild 3.25 und Bild 3.26 zeigen, daß sich auch bei sehr hohen Geduldsspannen bis etwa 10 min (500 s) noch erhebliche Änderungen der Seitenladedauer ergeben. Die in Bild 3.25 ebenfalls eingezeichnete empirische Verteilung der Seitenladedauer aus dem gemessenen Verkehr dient hier nur als Vergleich. Sie wird durch die Simulation nicht genauer abgebildet da im gewählten Szenario deutlich vereinfachte Beschreibungen für Lauf- und Antwortzeiten gewählt wurden.

Die Betrachtung zeigt, daß die Wahl des Mittelwerts von 100s für die Nutzergeduld ihr einerseits genügend Gewichtung läßt, so daß Seitengröße und Ladedauer noch signifikant begrenzt werden, andererseits aber Ladeabbrüche nicht zu häufig vorkommen, so daß die anderen Modell-Parameter den erzeugten Verkehr nach wie vor charakterisieren, wie z.B. die Anzahl eingebetteter Objekte, die im Mittel 4.5 gemessen wurde und auch in der Simulation für $E[t_G]=100s$ und $t_{WSD}=250ms$ erreicht wird.

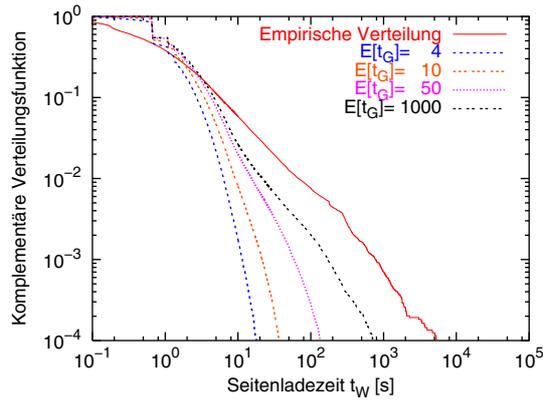


Bild 3.25: Verteilungen der Seitenladedauer bei unterschiedlichen mittleren Geduldsspannen.

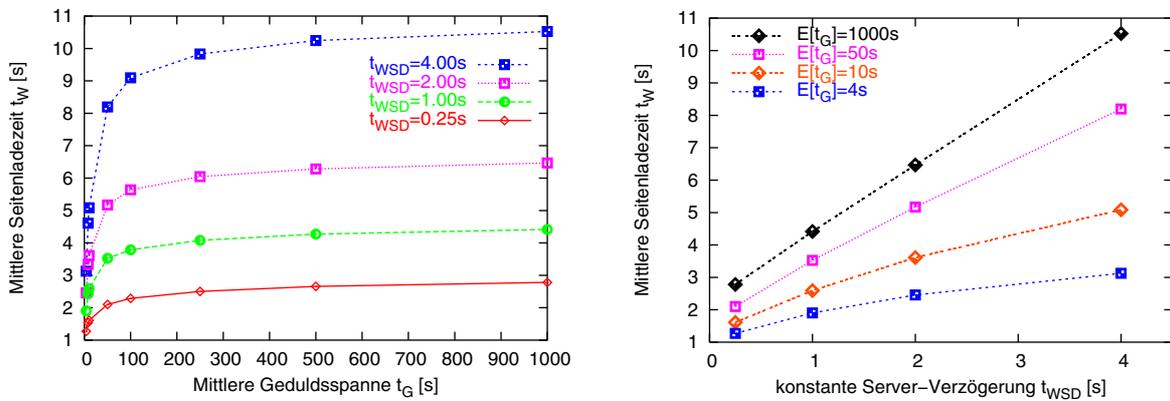


Bild 3.26: Mittlere Seitenladezeit in Abhängigkeit der (konstanten) Server-Verzögerung bei verschiedenen mittleren Geduldsspannen.

Einfluß der Server-Verzögerung

In Kapitel 3.2.2 wurde ebenfalls eine Schätzung der Verzögerung auf Seiten des Servers vorgeschlagen (WSD, *Web Server Delay*). Die Schätzung beruht auf dem Querschnitt der unterschiedlichsten Server bei unterschiedlichsten Lastverhältnissen und darf nur als grober Richtwert betrachtet werden. Um so interessanter ist der Einfluß, den dieser unsichere Modellparameter auf den erzeugten Verkehr hat. Die Verzögerung liegt mit einem Mittelwert von 0.5 ms in der Größenordnung der Umlaufzeit. Durch die vorgeschlagene Pareto-Verteilung mit unendlicher Varianz werden allerdings auch deutlich größere Verzögerungen erzeugt.

Um beobachtete Effekte eindeutig identifizieren zu können, werden zunächst Simulationen mit konstanten Server-Verzögerung betrachtet, also einer einfachen Verlängerung der Ladezeiten jedes Objekts. In Bild 3.26 ist gut zu erkennen, daß die mittlere Seitenladedauer mit höherer Server-Verzögerung überproportional ansteigt, da diese in Abhängigkeit der Anzahl eingebetteter Objekte und der Anzahl der verwendeten parallelen Verbindungen mehrmals auftritt. Bild 3.27 zeigt, wie sich die Verteilung der Antwortzeit sowie der Seitenladezeit bei steigender Server-Verzögerung entwickeln. Da viele Seiten keine eingebetteten Objekte enthalten, sind

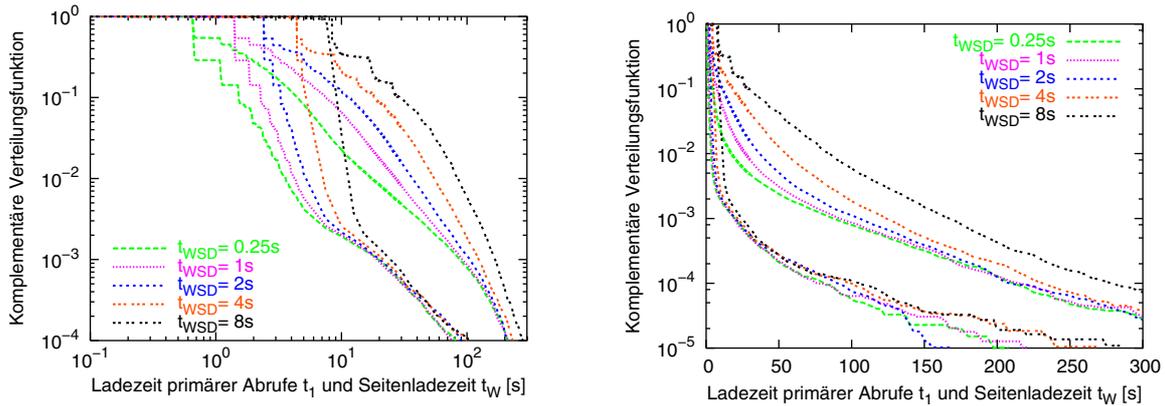


Bild 3.27: Verteilung der Antwortzeit und der Seitenladezeit in Abhängigkeit der (konstanten) Server-Verzögerung (links doppel logarithmisch, rechts einfach logarithmisch)

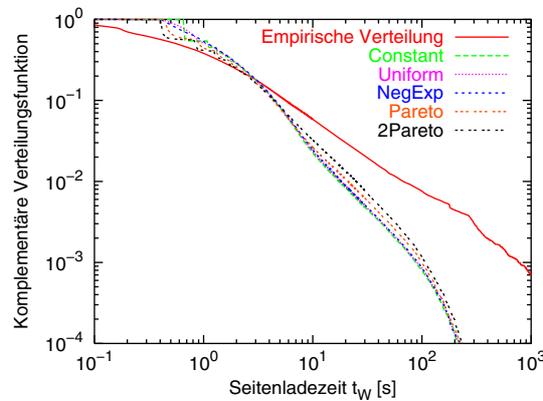


Bild 3.28: Verteilung der Seitenladezeit bei Server-Verzögerung mit unterschiedlicher Varianz

Antwortzeit und Seitenladezeit identisch und beide Kurven demnach anfangs deckungsgleich. Die Seitenladezeit erreicht bei höherer Server-Verzögerung allerdings deutlich größere Werte, was sich im flacheren Verlauf der Kurve zeigt. Die Seitenladezeit hat wiederum direkte Auswirkungen auf die Abbruchwahrscheinlichkeit aufgrund der Ungeduld des Nutzers. In Bild 3.26 ist deutlich zu sehen, daß sich die Wahl der Geduldsspanne insbesondere bei langen Verzögerungen auswirkt und Bild 3.22 zeigt, wie mit der Server-Verzögerung auch die Abbruchwahrscheinlichkeit steigt.

Simulationen mit Gleichverteilung, negativ-exponentieller Verteilung und der vorgeschlagenen Pareto-Verteilung für die Server-Verzögerung zeigen prinzipiell dasselbe Verhalten. Mit steigender Varianz der Server-Verzögerung ändert sich das Verhalten der Seitenladedauer oder der Abbruchwahrscheinlichkeit nur geringfügig, wie Bild 3.28 zeigt. Lediglich die signifikante Stufe am Kopf der Verteilung, die durch die konstante Verzögerung der Antwortzeit hervorgerufen wird, verschwindet.

Aus Sicht des Netzes wirken sich hohe Server-Verzögerungen in längeren Pausen zwischen den Übertragungsphasen aus, so daß insgesamt ein etwas büschelförmigeres Verhalten ent-

steht. Daher ist eine Betrachtung der bei Überlagerung vieler Quellen erzeugten asymptotischen Selbstähnlichkeit interessant. In Bild 3.29 ist der Varianz-Zeit-Plot der Paketzwischenankünfte von 64 überlagerten Quellen abgebildet. Die eingezeichneten Hilfslinien für die Schätzung des Hurst-Parameters zeigen zwar, daß der Summenverkehr eine signifikante asymptotische Selbstähnlichkeit aufweist, aber die Ergebnisse für die verschiedenen Server-Verzögerungen unterscheiden sich kaum, so daß kein besonders starker Einfluß der Server-Verzögerung auf das Paketverhalten festgestellt werden kann.

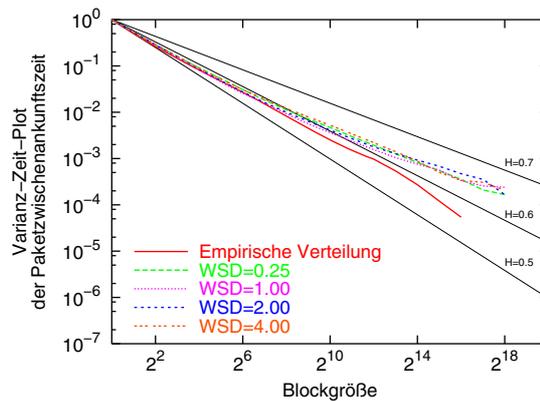


Bild 3.29: Hurst-Schätzung bei aggregiertem Verkehr mit Verzögerung auf Server-Seite.

3.3.3.2 Auswirkungen der eingeführten Rückkopplung

Die charakteristische Eigenschaft des vorgeschlagenen Modells sind die Rückkopplungen auf Dienst- und Nutzerebene. Zum einen werden dabei Strukturabhängigkeiten von HTML-Dokumenten abgebildet und zum anderen wird mit der Nutzergeduld neben der Betrachtungsdauer eine zusätzliche Zeitüberwachung eingeführt, welche einen Abbruch des Ladevorgangs insbesondere bei schlechtem Netzzustand unterstützt. Mit diesen Mechanismen soll das charakteristische Verkehrsverhalten des WWW-Dienstes besser nachgebildet werden. Die Auswirkungen der Modellierung können am besten im Vergleich mit einem nicht-rückgekoppelten Verkehrsmodell dargelegt werden, wenn beide Modelle bei Netzverzögerungen (z.B. durch hohe Web-Server-Verzögerungen) und Netzstörungen (z.B. Paketverluste) untersucht werden. Als Referenzmodell dient hier eine angepaßte Realisierung des SURGE-Modells, das bereits in Abschnitt 3.1.3 vorgestellt und in Abschnitt 3.3.2 dem reaktiven Modell gegenübergestellt wurde.

Da Seitenabrufe bei SURGE unabhängig ausgelöst werden, ändert sich, im Gegensatz zum reaktiven Modell, auch nicht die Abruftrate. Der Abbruch langer Ladevorgänge kommt bei diesem Modellansatz relativ häufig vor, wie Bild 3.30 zeigt. Das liegt aber auch daran, daß die Übertragungen eingebetteter Objekte nicht sofort bei Eintreffen der primären Antwort freigestellt werden, sondern zufällig ausgelöst und damit auch über einen größeren Zeitraum verteilt werden.

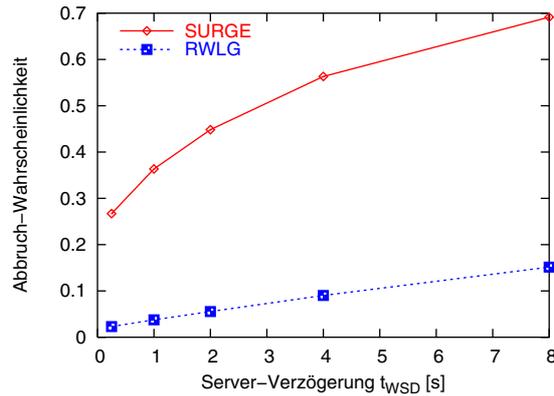


Bild 3.30: Seitenlade-Abbruchwahrscheinlichkeiten bei RWLG und bei SURGE.

Die tatsächlich übertragenen Seitengrößen weichen beim SURGE-Modell folglich auch deutlich stärker von den Vorgaben der Modellparameter ab. Bild 3.31 zeigt, daß die Seitengröße (Anzahl übertragener Objekte sowie Seitenumfang) für das SURGE-Modell bei steigender Verzögerung im Netz, welche hier nur durch die Server-Verzögerung realisiert wird, deutlich abnimmt. Für die Anzahl der übertragenen Objekte (links) ist zum Vergleich auch die empirische Verteilung eingezeichnet, die direkt als Modellparameter eingesetzt wird, also als Sollvorgabe des Modells betrachtet werden kann. Das reaktive Modell zeigt natürlich eine ähnliche Tendenz, aber die Abweichungen von der Vorgabe sind deutlich geringer ausgeprägt.

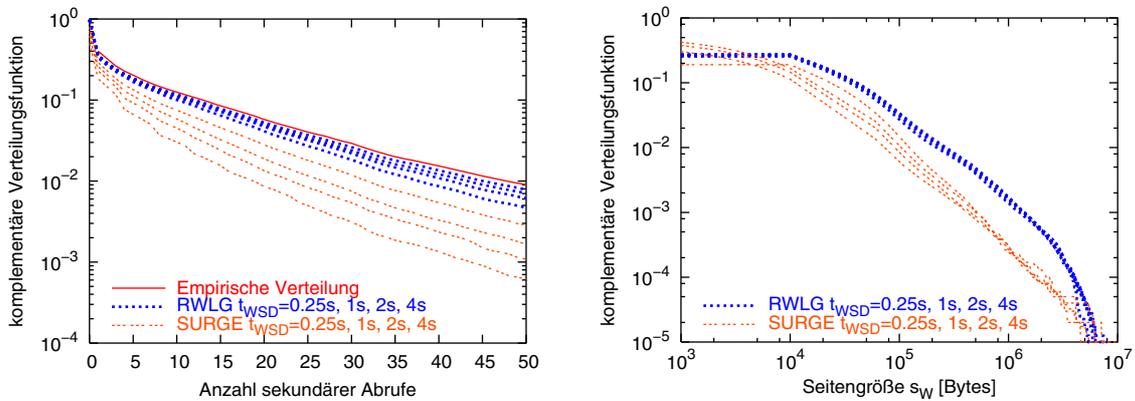


Bild 3.31: Anzahl übertragener Objekte je Seite (links) und Seitengröße (rechts) beim reaktiven Modell und bei SURGE bei unterschiedlicher Server-Verzögerung.

Neben der direkten Verzögerung durch Laufzeit, Netzzugang und vor allem Server-bedingten Wartezeiten werden Übertragungen auch durch Paketverluste und der dadurch ausgelösten Wiederholungen bei Verwendung von TCP verzögert. In Bild 3.32 sind die Seitengrößen bei verschiedenen Paketverlustwahrscheinlichkeiten für SURGE und das reaktive Modell dargestellt. Es zeigt sich, daß SURGE-Verkehr deutlich empfindlicher auf Netzstörungen reagiert. Die beobachtete Ladedauer (Bild 3.33) verhält sich allerdings umgekehrt, d.h. beim reaktiven Modell steigt die Ladedauer mit der Paketverlustwahrscheinlichkeit deutlich empfindlicher als

beim SURGE-Modell, da die Geduldsspanne hoch genug eingestellt ist, um auch längere Ladedauern zu tolerieren.

Die Seitenladedauer als Dienstgüte-Maß muß für beide Modelle daher unterschiedlich interpretiert werden, da sie durch die Nutzergeduld und Ladeabbrüche entscheidend geprägt ist. Insbesondere im Falle von SURGE handelt es sich bei den Seitenladezeiten zum größeren Teil um abgebrochene Ladevorgänge.

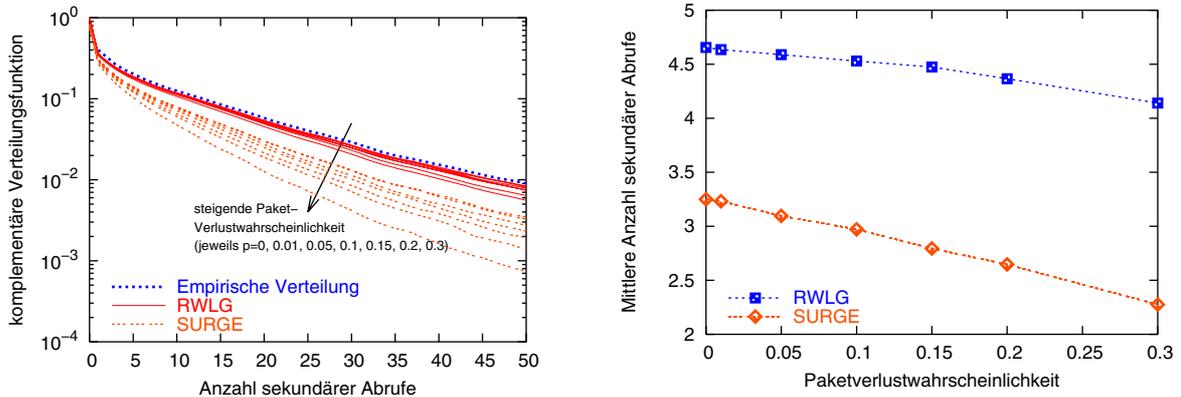


Bild 3.32: Anzahl übertragener Objekte je Seite bei unterschiedlichen Paketverlustwahrscheinlichkeiten.

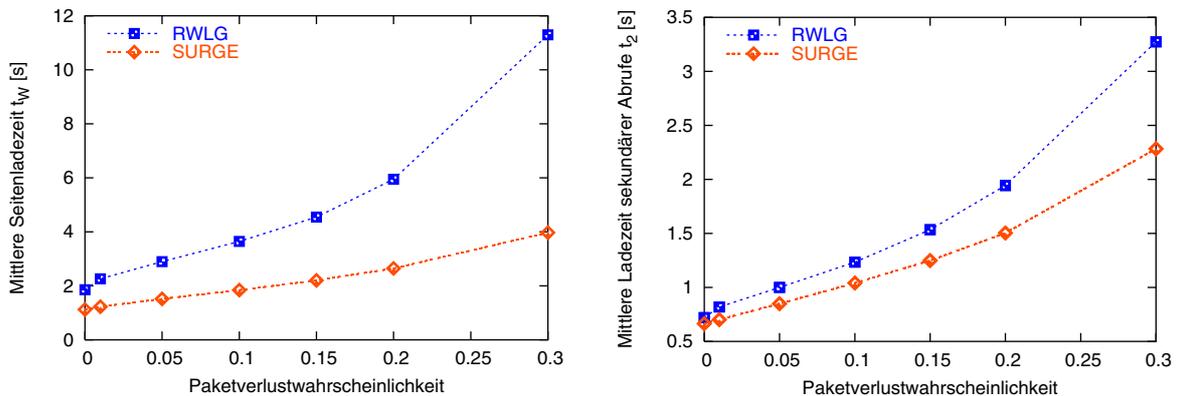


Bild 3.33: Ladezeit bei unterschiedlichen Paketverlustwahrscheinlichkeiten.

Neben dem zeitlichen Verhalten aus Nutzersicht ist auch interessant, wie der erzeugte Verkehr aus Netzsicht charakterisiert ist. Besonders das Verhalten bei Überlagerung mehrerer Quellen soll betrachtet werden, da bei büschelförmigem Verkehr selbstähnliche Effekte auftreten, welche das Netz überproportional belasten. Bild 3.34 zeigt den Varianz-Zeit-Plot für die Paketzwischenankunftszeit des Summenverkehrs von mehreren reaktiven Lastgeneratoren (links) und mehreren SURGE-Quellen (rechts). Die Steigung der Kurven ist ein direktes Maß für den Hurst-Parameter, welcher den Grad der Selbstähnlichkeit beschreibt – dessen Schätzwert ist in Bild 3.35 abgebildet. Prinzipiell zeigen beide Quellen das gleiche Verhalten: bei wenigen Quellen liegen die Kurven auf der Linie für $H=0.5$, also keiner innewohnenden Selbstähnlichkeit. Mit steigender Quellenzahl tendiert auch der Varianz-Zeit-Plot zu einem höheren Grad

der Selbstähnlichkeit, der bei Erreichen der maximalen Link-Bandbreite wieder abklingt (hier nicht gezeigt).

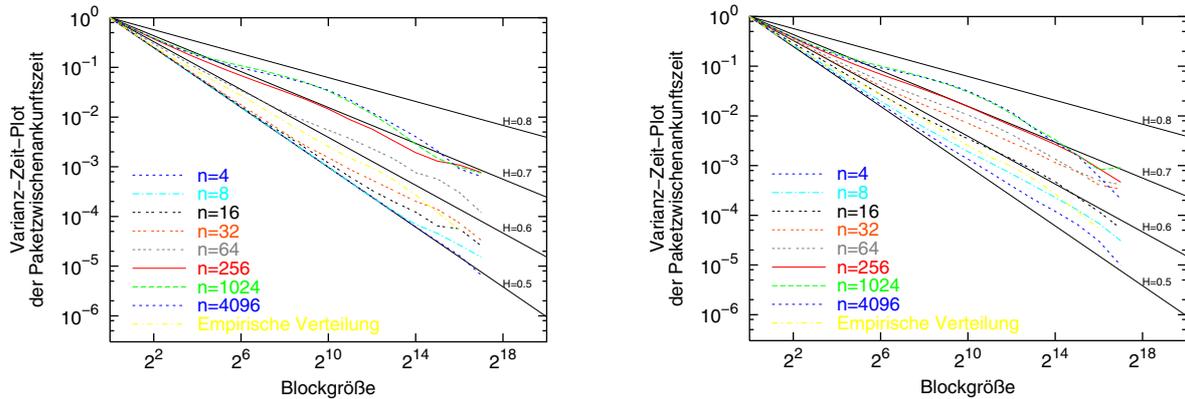


Bild 3.34: Varianz-Zeit-Plot der Paketzwiseankunftszeit des Summenverkehrs bei Überlagerung von RWLG-Quellen (links) und SURGE-Quellen (rechts).

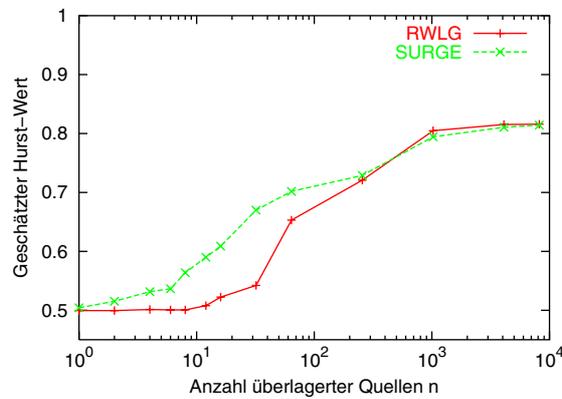


Bild 3.35: Geschätzte Hurst-Parameterwerte der Paketzwiseankunftszeit.

Der Summenverkehr von überlagerten SURGE-Quellen zeigt bereits bei wenigen Quellen stärkeres selbstähnliches Verhalten als beim reaktiven Modell. Ab etwa 200 Quellen verhalten sich die Summenverkehre beider Modelle wieder ähnlich. Die Ursachen hierfür dürften zunächst in der Tatsache begründet sein, daß SURGE-Quellen die Objekte einer WWW-Seite sporadisch nach zufälliger Zwischenankunftszeit übertragen, also einen stärker büschelförmigen Verkehr erzeugen als das reaktive Modell, das alle Objekte einer Seite als Block überträgt, also längere Wartezeiten und intensivere aber gleichmäßigere Aktivitätsphasen erzeugt. Zudem weisen SURGE-Quellen im betrachteten Szenario etwa 5 mal höhere Paketraten auf, die durch längere Anforderungs-Zwischenankunftszeiten des reaktiven Modells im untersuchten Szenario begründet sind. Damit ist der direkte Vergleich der Modelle zwar erschwert, aber das Verhalten der Selbstähnlichkeit wird durch die unterschiedlichen Paketraten nur geringfügig beeinflusst.

Auch die Autokorrelationsfunktionen beider Quellen verhalten sich ähnlich (Bild 3.36). Während bei wenigen überlagerten Quellen keine Korrelation festgestellt werden kann, nimmt sie mit stärkerem Verkehrsaufkommen deutlich zu, bis, bei hoher Link-Auslastung, die TCP-

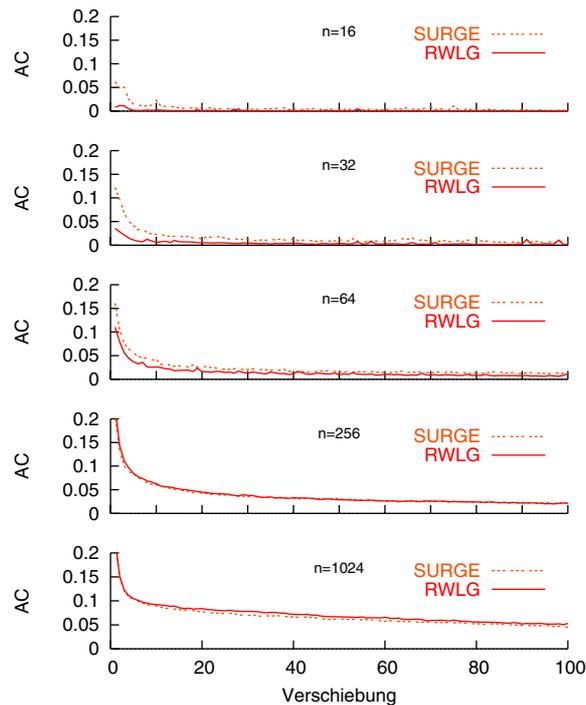


Bild 3.36: Autokorrelationsfunktion der Paketzwiseankunftszeit des Summenverkehrs bei Überlagerung von reaktiven Quellen (RWLG) und SURGE-Quellen.

Mechanismen und die Signallaufzeit die Verkehrscharakteristik dominieren und eine starke Korrelation verursachen. Für das SURGE-Modell wird bei der Überlagerung mehrere Quellen eine schnellere Zunahme der Autokorrelation beobachtet. Dies kann durch die höheren Paketräte und die weniger stark gebündelte Übertragung von eingebetteten Objekten erklärt werden kann, was bei Überlagerung schneller zu einem regelmäßigen Paketfluß führt.

Die eingeführten Rückkopplungen schlagen sich bei Netzstörungen und -verzögerungen also hauptsächlich in erhöhten Seitenladezeiten nieder, die aufgrund der Nutzergeduld auch in gewissem Maß toleriert werden. Dagegen generiert das SURGE-Modell HTTP-Anforderungen unabhängig weiter – die längeren Objektladezeiten führen bei gleichbleibenden Anforderungszeitpunkten zu häufigeren Abbrüchen, so daß insgesamt auch weniger Daten übertragen werden. Der Summenverkehr der beiden Modelle unterscheidet sich lediglich im Bereich weniger Quellen. Durch das stärker blockorientierte Übertragungsverhalten des reaktiven Modells kommt es erst bei höherem Aggregationsgrad zu Selbstähnlichkeit.

3.3.3.3 Exemplarische Anwendung des Verkehrsmodells

Eine typische Fragestellung bei der Parametrisierung von WWW-Browsern ist die Anzahl von parallelen Verbindungen, die maximal zugelassen werden sollen. Die parallele Übertragung von Objekten einer Seite beschleunigt einerseits den Ladevorgang, führt andererseits aber zu erhöhter Netzlast, die zu Paketverlusten und Wartezeiten führen kann. Durch Einsatz des reakt-

tiven WWW-Lastgenerators können die Auswirkungen dieser Browser-Einstellungen untersucht und quantitativ bewertet werden.

Die Simulationsergebnisse in Bild 3.37 zeigen, wie sich die Seitenladezeit (links) bei mehreren parallelen TCP-Verbindungen zunächst stark verringert. Bei mehr als vier parallelen Verbindungen ist kaum noch ein Gewinn zu erzielen - im Falle geringer Zugangsraten verlängert sich sogar der Ladevorgang. Die rechts abgebildete Ladezeit der einzelnen Objekte vergrößert sich bei parallelen Übertragungen, da die Zugangsbandbreite geteilt werden muß.

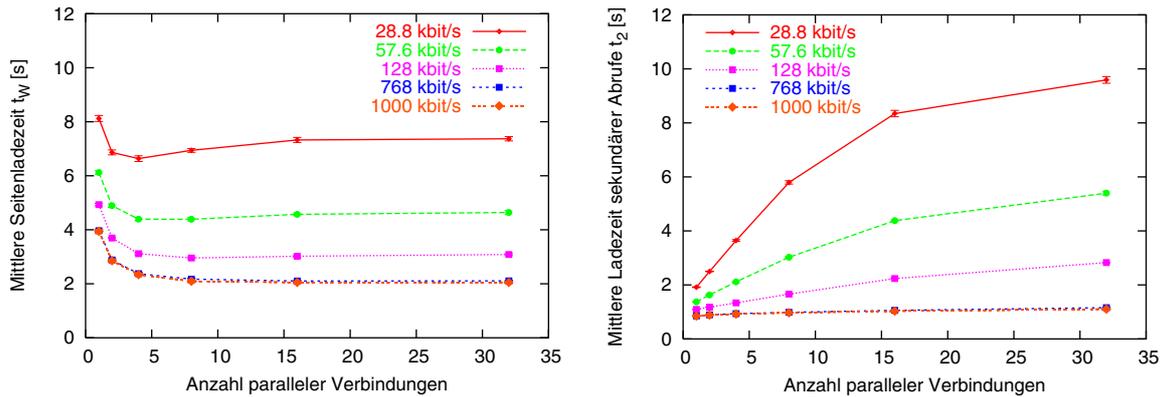


Bild 3.37: Mittlere Seitenladezeit und Ladezeit eingebetteter Objekte bei Verwendung von parallelen Verbindungen und unterschiedlichen Zugangsraten.

Betrachtet man nun, wieviele der maximal erlaubten TCP-Verbindungen tatsächlich verwendet werden, stellt sich eine starke Abhängigkeit von der Zugangsbandbreite heraus. In Bild 3.38 ist die mittlere Anzahl tatsächlich gleichzeitig aktiver TCP-Verbindungen abgebildet. Nutzer mit geringer Zugangsrate machen von parallelen Verbindungen deutlich stärkeren Gebrauch. Die Nutzung so weniger paralleler Verbindungen bei hochratiger Netzanbindung belegt, daß die meisten WWW-Seiten nur wenige parallel übertragbare Objekte enthalten. Daß bei geringer Rate trotzdem im Mittel so viele Verbindungen genutzt werden, ist in den erheblich längeren Ladezeiten begründet, da die Anzahl tatsächlich aktiver Verbindungen über ein zeitliches Integral während der aktiven Phasen bestimmt werden.

Für Nutzer mit langsamen Netzzugang empfiehlt es sich also, nicht mehr als vier parallele Verbindungen zu erlauben. Allerdings kann bereits mit zwei Verbindungen, wie dies in der HTTP-Spezifikation empfohlen wird, ein fast gleichwertiger Gewinn erzielt werden. Für Nutzer mit höheren Zugangsraten bringen mehr als 4 parallele Verbindungen nur noch eine minimale Beschleunigung des Ladevorgangs.

In Bild 3.39 bis Bild 3.41 sind nochmals die wesentlichen Unterschiede der beiden verwendeten Modelle sichtbar. Der Einsatz des SURGE-Modells führt zu erheblich mehr Ladeabbrüchen und mißt dadurch geringere Seitenladezeiten. Das SURGE-Modell kann im Mittel deutlich mehr TCP-Verbindungen nutzen, da zwischen primären und sekundären Abrufen nicht

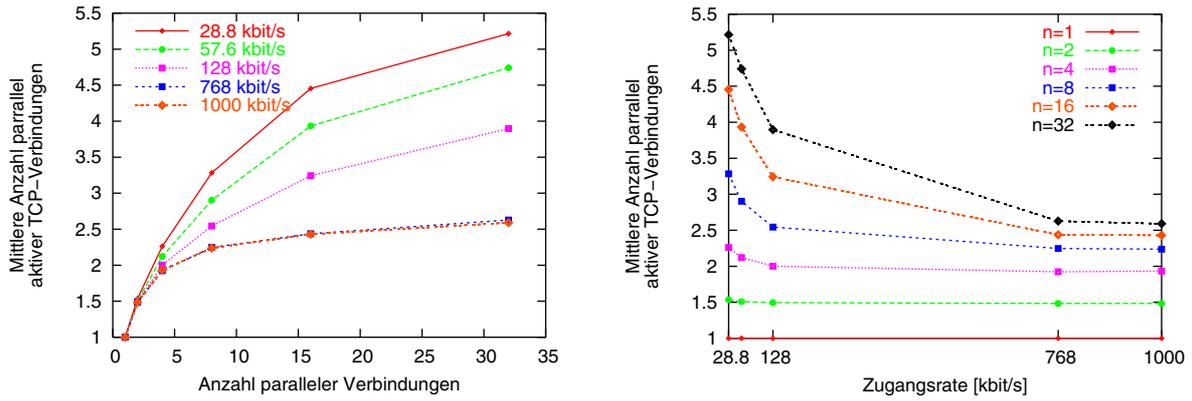


Bild 3.38: Mittlere Anzahl tatsächlich aktiver TCP-Verbindungen.

unterschieden wird, der primäre Abruf also nicht exklusiv übertragen wird. Bei hohen Zugangsraten nähern sich die Ergebnisse beider Modelle an, da Gewichtung des primären Abrufs und die Abbruchwahrscheinlichkeit sinken. Bei geringen Zugangsraten weichen die Modelle aber auch im Verhalten ab. Eine simulative Bestimmung der optimalen Anzahl parallel einzusetzender Verbindungen würde mit dem SURGE-Modell zu zwei Verbindungen führen.

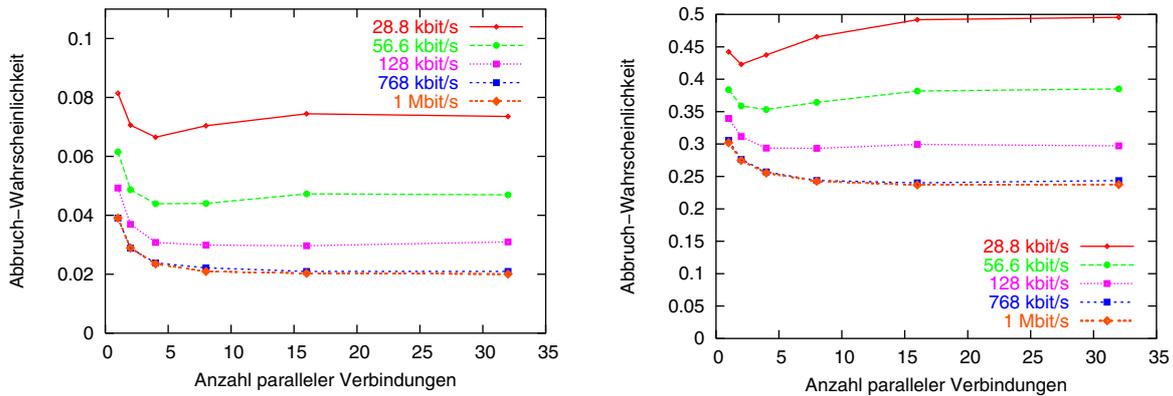


Bild 3.39: Lade-Abbruchwahrscheinlichkeit bei Verwendung paralleler Verbindungen für das rückgekoppelte Modell (links) und für das SURGE-Modell (rechts)

Der Modellansatz erlaubt durch die direkte Beobachtung der identifizierten Dienstgüteparameter eine einfache Bewertung der Leistungsfähigkeit eines Kommunikationssystems bezüglich des Dienstes WWW. Es können sowohl Einflüsse von Dienstparametern (parallele verbindungen) als auch von Netzeigenschaften untersucht werden. Die Rückkopplungseigenschaften des reaktiven Modells zeichnen es vor allem bei der Betrachtung von Einzelverkehren aus.

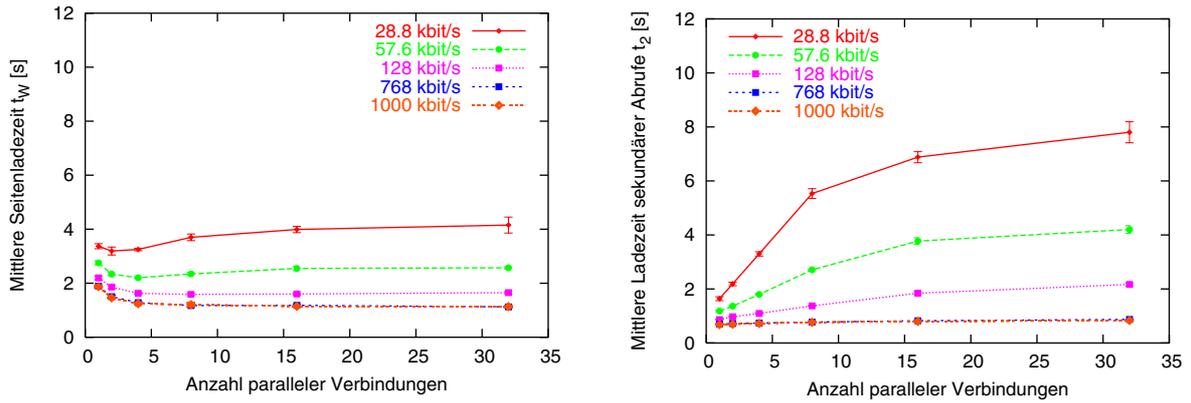


Bild 3.40: Mittlere Seitenladezeit und Ladezeit eingebetteter Objekte (SURGE-Modell).

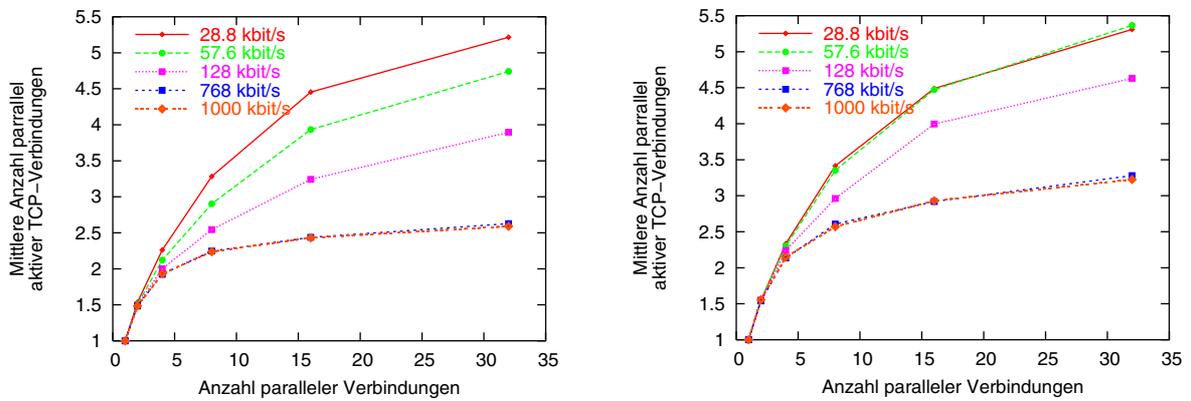


Bild 3.41: Mittlere Anzahl tatsächlich aktiver TCP-Verbindungen für das rückgekoppelte Modell (links) und das SURGE-Modell (rechts).

Kapitel 4

Verkehrsmodellierung des verteilten Echtzeitdienstes Netzspiele

4.1 Modellierung

4.1.1 Einordnung

Computerspiele werden immer populärer und entwickeln sich zur Grundlage einer sehr schnell wachsenden milliarden schweren Unterhaltungsindustrie, die bereits mit dem Volumen der Filmindustrie vergleichbar ist [168a]. Obwohl hierfür zum Großteil nur lokal laufende Spiele verantwortlich sind, steigt auch der Anteil der Netzspiele (Online-Spiele) enorm an. Einkünfte aus Online-Spielen haben bereits die Milliarden-Dollar-Marke überstiegen [182,186a]. Für Netzanbieter wird die Erfüllung von Spiele-Anforderungen zu einem immer wichtigeren Verkaufsargument [162,184].

Netzspiele umfassen dabei ein sehr weites Feld, das von kurzweiligen Pausenspielen bis hin zu hochauflösenden 3D-Abenteuerspielen reicht. Sie sind für Alleinspieler bis hin zu tausenden von vernetzten Mitspielern konzipiert. Die Spieldauer reicht von wenigen Minuten bis hin zu etliche Stunden und manche Netzspiele finden in einer persistenten Welt statt, die von den Spielern über Jahre hinweg regelmäßig besucht wird. Als Spieleplattformen fungieren in erster Linie der flexible Heimcomputer sowie dedizierte, an den Fernseher angeschlossene Spielekonsolen aber zunehmend auch Web-Browser und mobile Endgeräte. Vor dem Hintergrund der komplexen Programmentwicklung, der notwendigen Unterstützung unterschiedlichster Plattformen sowie dem Trend zur Vernetzung von immer mehr Spielern, werden zunehmend Programmmodule für Grafik [187,171], Physik [170], Sound oder künstliche Intelligenz [162] sowie Kommunikations-Infrastrukturen von Drittanbietern [164] eingesetzt.

Die weitere Entwicklung der Computerspieleindustrie wird mit Sicherheit sehr dynamisch sein. Sie zeigt ein hohes jährliches Wachstum von 30-50% und erfährt in kurzen Zyklen grundlegende technische Neuerungen. In Zukunft kann beispielsweise bei Teamspielen eine Unterstützung durch Sprachkommunikation erwartet werden. Auch soziale Strukturen, die durch Spielergemeinschaften aufgebaut werden, werden technische Auswirkungen haben, beispielsweise durch die Einführung eines Beobachtermodus, bei dem hunderte von Spielern passiv eine Partie zwischen zwei Spielergruppen verfolgen können.

Diese Entwicklung hat ebenfalls Einfluß auf die Kommunikationsnetze. Netzspiele tragen bereits heute signifikant zum Internet-Verkehrsaufkommen bei. In [55] wurde gezeigt, daß 3-4% aller IP-Pakete im Kernnetz nur 6 populären Spielen zugeordnet werden konnten. Das Verkehrsvolumen ist allerdings gering, da die Pakete von Spielen in der Regel sehr klein sind. Neben dem steigenden Verkehrsanteil sind Netzspiele für Netzbetreiber auch aufgrund ihrer besonderen Anforderungen von großem Interesse. Insbesondere für solche Spiele, die rasche Reaktionen der Nutzer erfordern, müssen Transportnetze strenge Echtzeitbedingungen einhalten, um für Kunden und Dienstanbieter attraktiv zu sein. Andererseits basieren viele zeitkritische Spiele auf UDP anstatt TCP, wodurch sich der Anteil des aggressiven Verkehrs (ohne Selbstregulierung wie bei TCP) erhöht.

Hinsichtlich der Kommunikationsanforderungen müssen Netzspiele grundsätzlich in solche für Alleinspieler und solche für mehrere Spieler unterschieden werden. Für Alleinspieler lassen sie sich durch eine einfache Punkt-zu-Punkt-Beziehung beschreiben. Das auftretende Verkehrsaufkommen ist dabei eher gering, da Alleinspieler im Internet typischerweise strategische Aufgaben lösen und relativ wenige Spieleingaben übertragen werden. Aus Sicht des Transportnetzes heben sich solche Spiele nicht sonderlich von Diensten wie dem WWW oder Telnet ab.

Dagegen sind Netzspiele mit mehreren Teilnehmern im Prinzip interaktive verteilte Anwendungen, deren Zustand von allen Teilnehmern beeinflußt wird. Die Kommunikationsbeziehungen und das Kommunikationsaufkommen sind für diese Anwendungen weitaus komplexer. Aus der oben beschriebenen Vielfalt der Netzspiele resultieren entsprechend unterschiedliche Ausprägungen der Netzanforderungen. Für eine methodische Beschreibung der Netzanforderungen von Netzspielen wird daher eine Klassifizierung der Netzspiele vorgenommen. In [130] wurden Kriterien für eine Klassifizierung anhand technischer Gesichtspunkte vorgestellt. Tabelle 4.1 zeigt daran angelehnte Klassifizierungskriterien, die um die Echtzeitanforderungen erweitert wurden.

- Der Mehrspielergrad eines Spiels hängt eng mit dem Kommunikationsmodell zusammen. Spiele ersten Grades haben so wenige Spieler, daß deren Vernetzung fast keinen Beschränkungen unterliegt. Hier kann eine vollvermaschte *peer-to-peer* Kommunikation eingesetzt werden, in welcher jeder Client vollständig über die Aktionen der anderen Clients informiert wird. Bei größeren Spielen mit über 16 Spielern werden die Clients auf Grund der in

Tabelle 4.1: Kriterien zur Klassifizierung von Netzspielen

Kriterium	Klassifizierung
Mehrspielergrad	<ol style="list-style-type: none"> 1. gering (1-16 Spieler) 2. groß (16-200 Spieler) 3. massiv (über 200 Spieler)
Kommunikationsmodell	<ul style="list-style-type: none"> • Stern (Client/Server) • Vollvermaschung (Peer-to-Peer) • Hybrid
Behandlung neuer Spieler	<ul style="list-style-type: none"> • spontan (Mitmachen zu beliebigem Zeitpunkt) • fest (nur zum Sitzungsbeginn)
Spieldauer	<ul style="list-style-type: none"> • persistent • begrenzt (Dauer einer Partie)
Echtzeitanforderungen	<ul style="list-style-type: none"> • gering • hoch • sehr hoch

der Regel geringen Zugangsbandbreite der Endgeräte nur noch sternförmig vernetzt. Clients werden von einem zentralen Server nur noch mit den für sie relevanten Informationen versorgt. Bei sehr großen Spielen mit tausenden von Teilnehmern müssen vermaschte Server-Farmen eingesetzt werden. Die für Clients relevanten Informationen werden vorgefiltert und beispielweise über ein Publisher/Subscriber-Modell verteilt [20,78].

- Das gewählte Kommunikationsmodell bestimmt direkt Laufzeiten und Bandbreitenbedarf. Durch Vollvermaschung können Nachrichten schneller ausgetauscht werden, aber jeder Client benötigt eine ausreichende Zugangsbandbreite, um Daten an und von allen Mitspielern zu übertragen. Ein Client/Server-Modell verringert den Bandbreitenbedarf der Clients, da sie in jedem Zyklus nur eine Nachricht absenden und eine zusammenfassende Nachricht des Servers empfangen müssen. Dies führt aber zu einer erhöhten Laufzeit, da diese Nachrichten nacheinander übertragen werden müssen.
- Ob neue Spieler spontan in ein laufendes Match einsteigen können oder nicht, beeinflusst in erster Linie die Anwendungsarchitektur und das Spielprinzip, aber auch die Dynamik des Verkehrs.
- Die Spieldauer betrifft die Lebensdauer des Servers, bzw. eines Matches. Bei Spielen mit kurzen Partien tragen die Teilnehmer in der Regel mehrere Partien aus, so daß sie zwischen etwa 30 Minuten bis 6 Stunden aktiv sind.
- Die Echtzeitanforderungen hängen vom Spielprinzip ab. Schnelle Reaktionsspiele haben höhere Anforderungen (50-150ms) als taktische (500ms) oder strategische (Sekunden) Spiele mit eher ruhigerem Spielfluß.

Tabelle 4.2: Klassifizierung von Computerspielen

Klassifizierung	Beschreibung
<p>Reaktionsspiele</p> <p><i>Mehrspielergrad:</i> 2 (2-64 Spieler) <i>Komm.-Modell:</i> Client/Server <i>neue Spieler:</i> spontan <i>Dauer:</i> 5-15min <i>Echtzeitanforderungen:</i> sehr hoch</p>	<p>Spiele, die schnelle Reaktionen verlangen: schnelle 3D-Spiele (Quake, Unreal, Half-Life, Counter-Strike), Flugsimulationen und Sportspiele.</p>
<p>Taktikspiele</p> <p><i>Mehrspielergrad:</i> 1 (2-8 Spieler) <i>Komm.-Modell:</i> C/S, Vollverm. <i>neue Spieler:</i> fest <i>Dauer:</i> 30min-4h <i>Echtzeitanforderungen:</i> hoch</p>	<p>Spiele, die taktische Planung erfordern, aber in Echtzeit ablaufen und relativ schnell auf Spielerentscheidungen reagieren müssen: Echtzeitstrategiespiele (Age of Empires), Rollenspiele (Diablo).</p>
<p>Denkspiele</p> <p><i>Mehrspielergrad:</i> 1 (2-8 Spieler) <i>Komm.-Modell:</i> C/S, Vollverm. <i>neue Spieler:</i> fest <i>Dauer:</i> 30min-4h <i>Echtzeitanforderungen:</i> gering</p>	<p>Spiele, die überlegtes Vorgehen erfordern und ohne Zeitdruck ablaufen: rundenbasierte Strategiespiele (Civilization), Adventures.</p>
<p>Massive Multiplayer-Spiele</p> <p><i>Mehrspielergrad:</i> 3 (>200 Spieler) <i>Komm.-Modell:</i> Hybrid <i>neue Spieler:</i> spontan <i>Dauer:</i> persistent <i>Echtzeitanforderungen:</i> hoch</p>	<p>Meist Rollenspiele in einer persistenten Welt, die sich durch ständige Präsenz und eine extrem große Spielerzahl auszeichnen (EverQuest, Ultima Online). Oft als MMOG (<i>Massively Multiplayer Online Games</i>) bezeichnet.</p>

Netzspiele laufen in Zyklen ab, in denen jeder Client den anderen Clients eigene Aktionen mitteilt und über die Aktionen der anderen Clients informiert wird. Nach der Auswertung, Darstellung und Verarbeitung der Nutzereingaben beginnt der nächste Zyklus. Das Erreichen eines gemeinsamen Spielzustands kann dabei synchron oder asynchron erfolgen. Synchron ablaufende Spiele leiden allerdings unter jedem einzelnen Paketverlust, da das Spiel angehalten wird, bis alle Teilnehmer synchronisiert sind [21,165b]. Insbesondere bei strategischen Spielen mit geringeren zeitlichen Anforderungen [144] kommt dies zum Einsatz, da ohnehin jede Spielinformation entscheidend sein kann und nicht verloren gehen darf, weshalb dann in der Regel auch TCP eingesetzt wird. Asynchrone Spiele führen den lokalen Spielzustand ständig dem abweichenden globalen Spielzustand nach und setzen Kompensationsmethoden ein, um Verzögerungen und Verluste zu überspielen [23,27,131]. Diese Technik benötigt keine zeitaufwändigen Sicherungsmechanismen und setzt daher auf UDP auf. Sie wird sehr erfolgreich bei den meisten modernen Spielen eingesetzt, die rasche Reaktionen erfordern.

Tabelle 4.2 zeigt eine grobe Klassifizierung für typische Vertreter der Spiele-Genres.

4.1.2 Eingrenzung

Aufgrund der großen Vielfalt der Netzspiele, der ständigen Weiterentwicklung sowie der kurzen Produkt-Lebensdauer kann eine sinnvolle Verkehrscharakterisierung dieses Dienstes nicht anhand einzelner Spiele durchgeführt werden. Durch Abstraktion können aber die in Tabelle 4.2 aufgeführten Dienst-Klassen definiert werden, die eine verallgemeinerte Charakterisierung erlauben. Im folgenden beschränken wir uns auf die Untersuchung der Klasse der Reaktionsspiele, da diese die weitaus größten Anforderungen an die Netzbetreiber stellen und weil diese Spiele zu den populärsten Online-Spielen gehören [168b]. Auf die Betrachtung der Taktikspiele wird verzichtet, da sie zum einen nicht so stark im Internet vertreten sind und zum anderen nur etwas geringere Anforderungen an das Netz stellen. Denkspiele haben keinen nennenswerten Anteil am Internet-Verkehr. Allein die massiven Netzspiele (MMOG, *massivley multiplayer online games*) sind eine Klasse mit grundsätzlich unterschiedlichen Anforderungen, die eine genauere Betrachtung wünschenswert macht. Diese komplexen Spiele werden allerdings sorgfältig geplant und auf eigens optimierte Netze aufgesetzt. Die Kommunikation innerhalb der Server-Cluster, die bei diesen Spielen von besonderem Interesse sind, betreffen daher nicht den allgemeinen Internet-Verkehr, der in dieser Arbeit untersucht wird. Die Charakterisierung der Kommunikation zu den Clients und auch direkt zwischen den Clients (Chat-Funktion) wird an dieser Stelle nicht erfolgen.

Die weitere Charakterisierung der Reaktionsspiele basiert im wesentlichen auf dem beliebtesten Vertreter dieser Klasse, "Half Life: Counter-Strike". Anfang 2002 waren jederzeit etwa 6000 aktive Half Life Server registriert, etwa zehn mal so viele wie bei vergleichbaren Spielen [92]. Counter-Strike kann zudem insofern stellvertretend für seine Spiele-Klasse betrachtet werden, als es wie viele andere Reaktionsspiele auf dem Programmcode von Quake [171] basiert. Das Kommunikationsaufkommen ist für alle Quake-basierten Spiele sehr ähnlich. Neben Quake muß auch Unreal [187] genannt werden, dessen Programmcode ebenfalls von vielen weiteren Spielen in Lizenz verwendet wurde. Es gibt weitere eigenständige Entwicklungen, die sich aber im Prinzip alle durch ein ähnliches Kommunikationsmodell und damit ähnlichen Verkehr auszeichnen.

Da die Verkehrsmodellierung mit dem Ziel der Leistungsuntersuchung und insbesondere der Bewertung der erreichten Dienstgüte erfolgen soll, kann eine weitere Einschränkung der Dienst-Betrachtung erfolgen. Reaktionsspiele werden in Phasen ausgetragen. Nachdem ein Spiel auf einem Server aufgesetzt wurde, meldet sich der Server bei einem Master-Server, der eine weltweite Liste aller aktiven Server führt. Ein Spieler sucht entweder auf dieser Liste nach einem geeigneten Server (die Antwortzeit wird für jeden Server ermittelt und angezeigt) oder er wählt ihn direkt an, wenn er die Server-IP-Adresse bereits kennt. Die Kommunikation während dieser Vorbereitungsphase, sowie der Nachbereitungsphase (Chat, Statistiken betrachten)

ist nicht echtzeitkritisch. Für eine Bewertung des Dienstes ist es daher sinnvoll, nur die aktiven Phasen des Spiels zu betrachten.

4.1.3 Dienstgüte-Metriken für Reaktionsspiele

Zur Bewertung eines Kommunikationsnetzes hinsichtlich der Eignung für Netzspieleverkehr müssen objektive Kriterien gefunden werden, welche die vom Nutzer subjektiv empfundene Zufriedenheit des Dienstes beschreiben. Die Dienstgüte kann aus Sicht der Anwendung z.B. über die Reaktionsschnelligkeit auf Nutzereingaben beschrieben werden. Aus Sicht eines paketbasierten Transportnetzes läßt sich die Güte einer Kommunikation im Wesentlichen durch Verlustwahrscheinlichkeiten, Verzögerungen und Verzögerungsschwankungen von Paketen ausdrücken. Bild 4.1 verdeutlicht diese unterschiedlichen Anforderungen und die Sichtbarkeit von Dienstgütekriterien auf den Kommunikationsschichten Nutzer, Dienst und Transport. Die Dienstgüte wird jeweils durch äußere Einflüsse beeinflusst. So hat die Erwartungshaltung des Spielers oder sein Verhalten (aggressiv oder abwartend) Einfluß auf seine Zufriedenheit. Eine entscheidende Bedeutung fällt auch der Implementierung eines Spiels zu, von der die Empfindlichkeit auf Verzögerungen direkt abhängt. Beispielsweise berichten Spieler, daß das Spiel "Quake III Arena" sehr empfindlich auf Verzögerungen reagiert, so daß schon die Qualität der Netzanbindung über Sieg oder Niederlage entscheidet. Durch Berücksichtigung der Verzögerung im Spielkonzept (z.B. durch Trägheit) oder den Einsatz von Bewegungsvorhersage von Figuren und Objekten und der ständigen Korrektur der Abweichungen im Client kann eine sehr viel geringere Empfindlichkeit gegenüber Verzögerungen erreicht werden. Mit solchen Techniken können moderne Spiele selbst bei sehr schlechten Verbindungen Paketverluste und -verzögerungen ausgleichen und gute Spielbarkeit erreichen [23,27,131]. Aus Sicht eines Netzbetreibers sind diese Kriterien aber unabänderbar. Er hat nur Einfluß auf die Güte des Pakettransports indem er eine optimale Architektur, Dimensionierung und Konfiguration seines Netzes vornimmt.

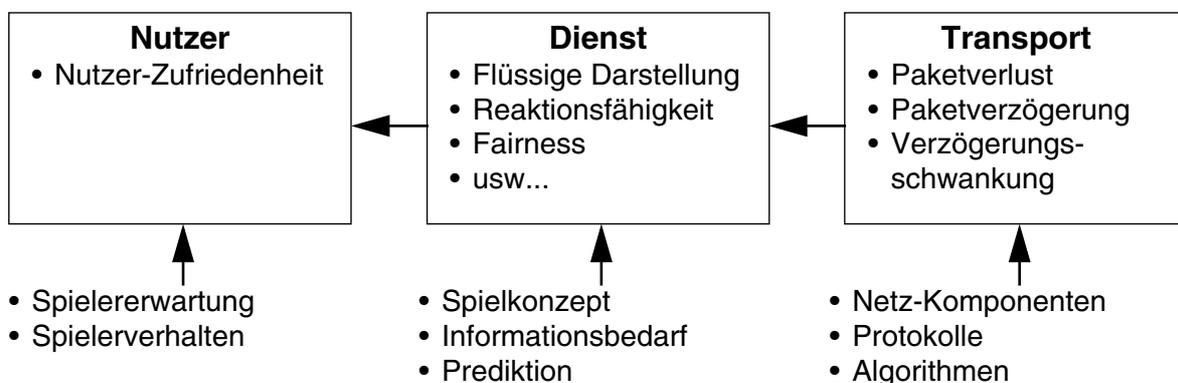


Bild 4.1: Einflüsse auf die Dienstgüte von Reaktionsspielen

Reaktionsspiele zeichnen sich insbesondere durch sehr hohe Echtzeitanforderungen aus, die sich im Prinzip direkt auf die Paketverzögerung übertragen läßt. In [11] wurde tatsächlich ein direkter Zusammenhang zwischen der Verzögerung der Anbindung der Spieler und ihrer Erfolgsquote gemessen. In [94] wird berichtet, daß Spieler sich bei der Wahl der Server nach der Schnelligkeit der Anbindung orientieren, auch wenn sie bei deutlicher (absichtlich herbeigeführter) Verschlechterung der Anbindung vermutlich aus sozialen Motiven nicht mehr den Server wechseln. Bei schnellen interaktiven Spielen wird allerdings anstelle der einfachen Verzögerung die Antwortzeit (*Round Trip Time*) gemessen, die als "Lag" bezeichnet wird. In Tabelle 4.3 sind quantitative Angaben zur von Spielern empfundenen Dienstgüte bei bestimmten Antwortzeiten angegeben. Diese Werte beruhen auf einer nicht repräsentativen Zusammenfassung von Spielermeinungen [183a,149] und können daher nur als grobe Richtwerte dienen.

Tabelle 4.3: Dienstgüte-Merkmal "Lag"

Round Trip Time (Lag)	Spieler-Empfinden
< 50 ms	exzellent
50 - 100 ms	gut
100 - 150 ms	beeinträchtigt
150 - 200 ms	merklich beeinträchtigt
> 200 ms	schlecht

Die Antwortzeit ist für Spieler das entscheidende Dienstgütemerkmal. Sie wird aber sowohl von der Umlaufzeit im fehlerfreien Fall als auch von der Paketverlustwahrscheinlichkeit bestimmt. Außerdem senden Server und Client ihre Pakete asynchron – im Rahmen dieser Arbeit wurde beispielsweise eine feste Senderate von 16.6 Paketen/s für den Server ($r_S = 1/60\text{ms}$) und eine feste Senderate von 24.4 Paketen/s bei Clients ($r_C = 1/41\text{ms}$) ermittelt¹. Eine Antwort auf eine Aktion des Spielers kann aber nur erfolgen, wenn die Nachricht des Clients rechtzeitig vor Absenden eines Pakets am Server eintrifft. Bild 4.2 zeigt, wie sich bei den beobachteten Paketraten, konstanter Signallaufzeit τ , konstanter Verarbeitungszeit am Server T_S und vernachlässigten Übertragungsdauern des Client-Pakets h_C und des Server-Pakets h_S die Antwortzeit (Lag) durch die Verschiebung der Server- und Client-Nachrichten ständig ändert, also auch ohne netzbedingte Schwankungen nicht regelmäßig ist. Sie folgt in etwa einer Gleichverteilung zwischen $2\tau + h_S + h_C + T_S$ und $2\tau + h_S + h_C + T_S + 1/r_S$, d.h. sie schwankt gleichmäßig mit der Server-Zwischenankunftszeit $1/r_S$, die mit 60 ms im Größenbereich einer typischen Ende-zu-Ende-Übertragungsdauer liegt. Bild 4.3 veranschaulicht dies indem eine volle Periode der asynchronen Ankunfts- bzw. Sendezeitpunkte am Server dargestellt wird. Im unteren Teil wird für jedes Client-Paket der zusätzlichen Lag-Anteil (ΔLag)

¹ Anmerkung: Zur besseren Unterscheidung wird im Weiteren mit r die Paketräte am Sender bezeichnet, während mit T die Zwischenankunftszeit am Empfänger gemeint ist, so daß $r \neq 1/T$.

bezogen auf $1/r_S$ gezeigt, welcher durch die unterschiedlichen konstanten Senderaten zustandekommt.

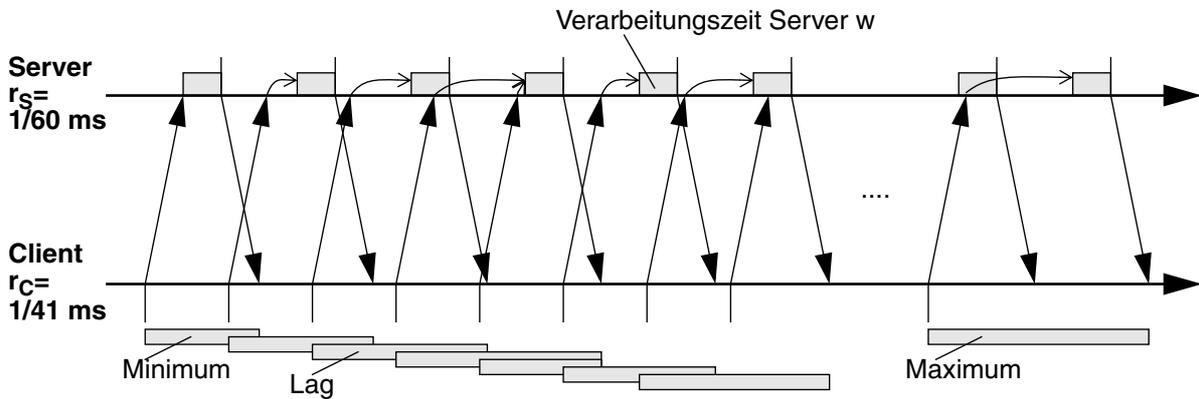


Bild 4.2: Betrachtung der theoretischen Antwortzeit (Lag).

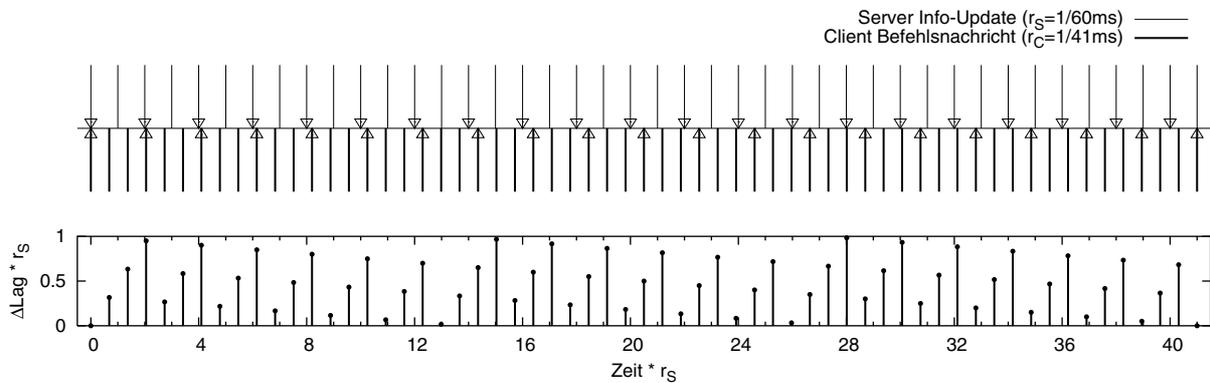


Bild 4.3: Veranschaulichung der inhärenten Antwortzeit-Schwankung

Paketgrößenänderungen aufgrund der Teilnehmerzahl wirken sich nur bei langsamen analogen Modemzugängen auf die Antwortzeit aus. Die zusätzliche Verzögerung bewegt sich hier höchstens im Bereich einiger Millisekunden und ist im Vergleich zu den Auswirkungen von Netzstörungen gering. Deutliche Schwankungen sind neben der oben erläuterten inhärenten Ungleichmäßigkeit der Antwortzeit vor allem durch Paketverluste zu erwarten. Jedes verlorene Paket verlängert die Antwortzeit in der Regel um $1/r_S$. Insbesondere mehrere Paketverluste in Folge führen also zu merklichen Störungen.

Paketverluste und Verzögerungsschwankung werden in Spielerkreisen kaum diskutiert. Während eine kleine Paketverlustrate die Dienstgüte kaum stört, werden Spiele mit sehr schlechten Verbindungen gar nicht erst begonnen oder aber abgebrochen. Hohe Verzögerungsschwankungen führen selbst bei sehr geringer Antwortzeit zu erheblichen Störungen – dagegen können viele Spieler bei geringen Verzögerungsschwankungen ihre Spielweise mit Erfolg einer hohen Antwortzeit anpassen. Bild 4.4 zeigt, wie eine entsprechende Bewertung der Dienstgüte aufgrund gemessener Leistungsparameter erfolgen kann. Die vorgeschlagenen Entscheidungsgrenzen haben nur illustrativen Charakter. Die gewählten Bereiche beschreiben die oben erläu-

terten Zusammenhänge qualitativ: das erste Szenario schneidet trotz größerer mittlerer Antwortzeit besser ab, als das zweite, in welchem starke Paketschwankungen die Spielbarkeit trotz geringer Antwortzeiten beeinträchtigen. Paketverluste gehen hierbei indirekt in Form von erhöhten Antwortzeiten und Schwankungen ein. Andere Entscheidungsgrenzen sind natürlich denkbar, müssen aber letztendlich in Feldversuchen bewertet werden. Im Rahmen dieser Arbeit wird auf eine solche empirische Bewertung verzichtet.

Bild 4.4 veranschaulicht die Dienstgüte ausgehend aus der Sicht des Spielers. Aus Netzsicht ist die Bestimmung der Antwortzeit eines speziellen Dienstes allerdings aufwändig. An Stelle der bidirektionalen Betrachtung kann auch aus einfacheren unidirektionalen Betrachtungen die Dienstgüte von Netzspielen abgeschätzt werden. Der Erwartungswert der Latenzzeit $E[d]$ (*delay*) und die Varianz der Paketzwischenankunftszeit $VAR[T]$ können durch einfache dienstunspezifische Messungen erfaßt werden (im verlustfreien Fall entspricht die Varianz der Paketzwischenankunftszeit der Varianz der Latenzzeit $VAR[T] = VAR[d]$). Paketverluste treten auch hier indirekt in Erscheinung, indem sie eine höhere Schwankung der Paketzwischenankunftszeiten verursachen.

Eine Annäherung an den bidirektionalen Betrachtungsfall kann erreicht werden, wenn die Paketverlustwahrscheinlichkeit zur Abschätzung einer angepaßten mittleren Latenzzeit verwendet wird. Eine höhere Antwortzeit aufgrund von Paketverlusten kann zum Zwecke der Dienstgütebewertung durch eine entsprechend höhere Latenzzeit beschrieben werden. Da jeder Paketverlust eine zusätzliche Verzögerung von $1/r$ verursacht, kann die entsprechend schlechtere Dienstgüte durch eine angepaßte Latenzzeit d' wie folgt ausgedrückt werden

$$d' = d + p\frac{1}{r} + p^2\frac{1}{r} + p^3\frac{1}{r} + \dots = d + \left(\frac{1}{1-p} - 1\right)\frac{1}{r} \quad (\text{geometrische Reihe}) \quad (4.1)$$

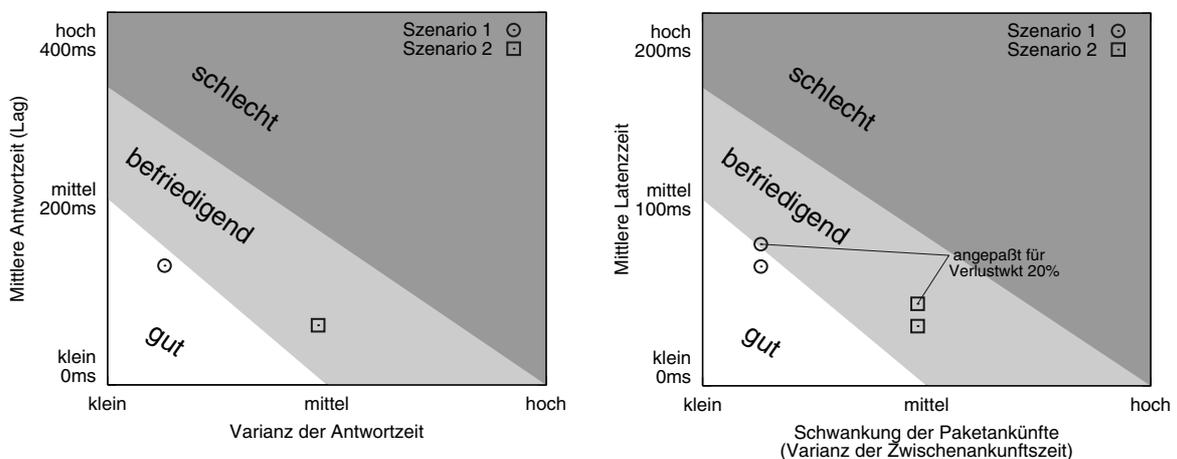


Bild 4.4: Exemplarische Bewertungsmethodik für Netzspiele anhand von dienstspezifischen Dienstgütekriterien (links) bzw. allgemeinen Dienstgütekriterien (rechts)

Aufgrund der konstanten Paketrate r des Senders kann die Paketverlustwahrscheinlichkeit p anhand der mittleren Paketzwehenankunftszeit T am Empfänger abgeschätzt werden, welche die tatsächliche Paketrate beschreibt:

$$p = 1 - \frac{E[r_{\text{Empfänger}}]}{r_{\text{Sender}}} = 1 - \frac{1}{r E[T]} \quad (4.2)$$

Damit kann eine Dienstgütebewertung allein aufgrund von dienstunspezifischen unidirektionalen Messungen der Latenzzeit d und der Paketzwehenankunftszeit T erfolgen.

4.1.4 Bestehende Verkehrscharakterisierungen

Da Netzspiele erst seit wenigen Jahren zu einem signifikanten Anteil am Netzverkehr beitragen, wurden sie bislang auch kaum als relevante Dienstklasse betrachtet und untersucht. Zudem geht die Vielfalt der Spiele mit einer großer Vielfalt der netztechnischer Umsetzungen einher, die sich erst seit den Spielen *Quake* (1995) und *Unreal* (1998) als einigermaßen netztauglich erweist.

In [28] stellte Borella im Jahr 1999 erstmals ein Paketverkehrsmodell für das Spiel *Quake* vor. Die Modellierung beruht auf der Messung von zwei kurzen Spielen mit drei bzw. vier Clients in einem lokalen Netz. Obwohl die Ergebnisse demnach nur eine grobe Abschätzung liefern, werden grundlegende Eigenschaften und Anforderungen des Spieleverkehrs beschrieben. *Quake* verwendet eine Client/Server-Architektur und den Datagramm-Dienst UDP. Der von Clients erzeugte Verkehr unterscheidet sich deutlich vom Server-Verkehr. Clients senden Pakete deterministischer Länge alle 10- 25 ms, während Server Pakete mit variierender Länge mit etwas geringerer Rate an jeden Client senden. Da Pakete an alle Clients sofort aufeinander folgen, zeigt der Server eine Burst-Charakteristik. Es werden deutliche Unterschiede zwischen Charakteristiken einzelner Clients bzw. zwischen den beiden Servern beobachtet, die auf den Einfluß unterschiedlicher Rechnerkonfigurationen zurückgeführt werden. Tabelle 4.4 zeigt die Parameter der vorgeschlagenen Approximationen für jeden Client. Die Definitionen der verwendeten Approximationsverteilungen sind in Anhang A zu finden.

In [105] werden Paketverkehrsmuster der Reaktionsspiele *Quake World* (ein Nachfolger von *Quake*) und *Unreal Tournament* charakterisiert und simulativ untersucht, allerdings nur unter Verwendung von wiedereingespielten Meßreihen bzw. durch die Verwendung der empirischen Verteilungsfunktionen der Paketgrößen und -zwischenankunftszeiten von Clients und Server. Diese Arbeit zeigt, daß unterschiedliche Reaktionsspiele vergleichbare Verkehrscharakteristiken erzeugen.

Tabelle 4.4: Quake-Verkehrsmodellierung nach Borella [28].

Verkehrsparameter	Approximation	Modellparameter	
		Messung 1 (3 Clients)	Messung 2 (4 Clients)
Client Paketgröße [Bytes]	Deterministisch(a)	a = 24	a = 24
Client Paket-Zwischenankunftszeit [ms]	Deterministisch(a) <50%	a = 13	a = 14
	Exponentiell(a) >50%	a = 19.61	a = 18.38
	Extremwert(a,b)	a = 20.63, 23.06 b = 4.39, 5.23	a = 36.38, 15.89, 30.26 b = 8.01, 2.98, 6.30
Server Paketgröße [Bytes]	Extremwert(a,b)	a = 89.92 b = 34.84	a = 65.58 b = 19.12
Server Paket-Zwischenankunftszeit [ms]	Extremwert(a,b)	a = 44.2 b = 9.52	a = 22.84 b = 5.29

Das mit Abstand beliebtesten Reaktionsspiel ist *Counter Strike*, das Gegenstand der meisten Untersuchungen ist. Es werden neben Paketverkehrscharakteristiken [51,77] auch Sitzungsdauern der Spieler [44,90] oder die empfundene Dienstgüte [94,91] untersucht.

In [90] ermitteln Henderson und Bhatti die aktiven Teilnehmer durch Abfrage von aktuellen Informationen von unterschiedlichen Spiele-Servern (Anzahl der Teilnehmer und deren bisherige Spieldauer). Sie berichten, daß sich Nutzer stark an der Tageszeit orientieren und an Wochenenden deutlich aktiver sind, wie dies auch bei Telefonnutzung oder der Internet-Einwahl beobachtet wird [69,70]. Der Gesamtprozeß darf also nicht als stationär betrachtet werden. Unter Berücksichtigung der saisonalen Effekte schlagen die Autoren einen autoregressiven Prozeß ARIMA nach Box-Jenkins [30] (*autoregressive-integrated-moving-average*) zur Modellierung der Teilnehmerzahl vor. Ein ARIMA-Prozeß zeichnet sich durch die Abhängigkeit der Elemente einer Zeitreihe von ihren vorhergehenden Elementen (AR-Anteil) und von einem Fehleranteil der vorhergehenden Elemente aus (MA-Anteil). ARIMA-Prozesse werden in der Form $(p, d, q) \times (P, D, Q)_s$ klassifiziert. Das erste Parameter-Tripel beschreibt die von saisonalen Effekten bereinigte Zeitreihe mit den Parametern p für die Ordnung des autoregressiven Prozesses (AR), d für die Ordnung des Integrationsprozesses (I), welcher die d -malige Differenzierung der Reihe zur Erreichung von Stationarität beschreibt und q für die Ordnung des moving-average Prozesses (MA). Das zweite Parameter-Tripel beschreibt die ARIMA-Parameter des saisonalen Anteils mit der Periode s . Der vorgeschlagene Prozeß $(2, 1, 1) \times (0, 1, 1)_{48}$ impliziert ein Zeitintervall von 30 Minuten zwischen den einzelnen Werten, berücksichtigt also saisonale Änderungen im Tagesverlauf.

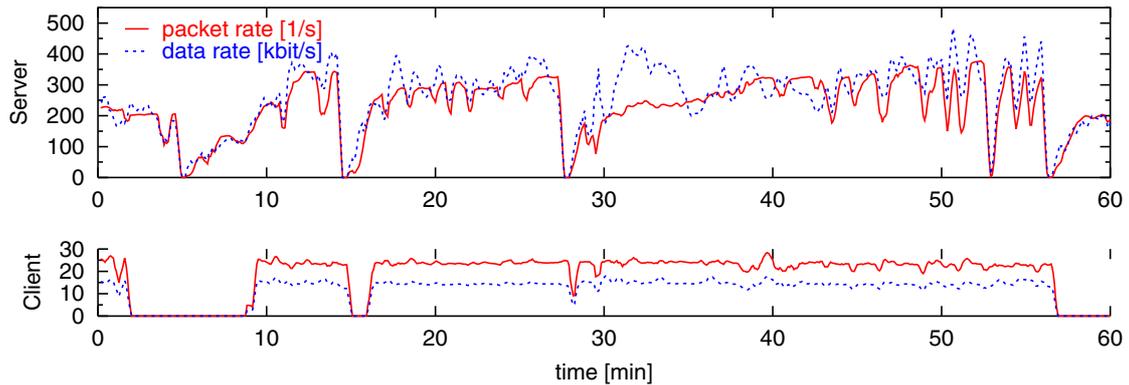


Bild 4.5: Beispielverlauf für von Server und Client erzeugten Verkehr (LAN)

4.1.5 Client/Server-Modellierung

Aufgrund der relativ einfachen Verkehrscharakteristik der Spiele, die exemplarisch aus der in Bild 4.5 gezeigten Messung ersichtlich ist, erscheint eine detaillierte hierarchische Modellierung zur Erzeugung von Paketverkehr nicht notwendig. Wie in Abschnitt 4.2.2 gezeigt wird, gibt es allerdings signifikante Abhängigkeiten der Paketgrößen des Servers von der Anzahl der beteiligten Spieler, die berücksichtigt werden sollten.

Das in dieser Arbeit vorgeschlagene Verkehrsmodell (siehe auch [69,72]) baut auf dem von Borella vorgeschlagenen Modell auf. Pakete werden unabhängig für Client und Server anhand von Erneuerungsprozessen für Paketgröße und Paket- bzw. Burst-Zwischenankunftszeiten erzeugt. Zusätzlich wird eine Nutzerschicht eingeführt, die über Zugangsrate und Sitzungsdauer die Anzahl der aktiven Spieler bestimmt und entsprechend die Anzahl der Client-Quellen und die Paketgrößenverteilung des Server-Verkehrsgenerators anpaßt. Bild 4.6 zeigt die beiden Schichten des Modells. Die Sitzungs-Zwischenankunftszeiten einzelner Teilnehmer können mit Messungen am Server-Zugangsnetz nicht ausreichend erfaßt werden und wurden im Rahmen dieser Arbeit nicht weiter untersucht. Diese Größe ist allerdings nur für die Beschreibung des Verkehrsaufkommens für bestimmte Teilnehmer in einem längeren Zeitraum sinnvoll. Die Sitzungs-Zwischenankunftszeiten am Server sind dagegen gut erfaßbar und für die Beschreibung des Teilnehmerverhaltens meist völlig ausreichend.

Aufgrund des sehr dynamischen Teilnehmerverhaltens und der genannten Verkehrsabhängigkeiten sowie der Randbedingungen (Client-spezifische Verkehrsmuster aufgrund heterogener Hardware und Netz-Zugangsrate, notwendige Identifizierung der aktiven Mitspieler) ist die Parametrisierung dieses Modells aufwändig.

Eine tiefere Modellierung durch Beschreibung der Spieleraktivität oder des Spielverlaufs in der Anwendung könnte genauere Verkehrsmuster mit z.B. erhöhtem Kommunikationsaufkommen bei Zusammenkunft mehrerer Spieler-Figuren erzeugen, würde aber gleichzeitig eine zu enge Festlegung auf ein bestimmtes Spiel bedeuten. Das hier vorgeschlagene Ver-

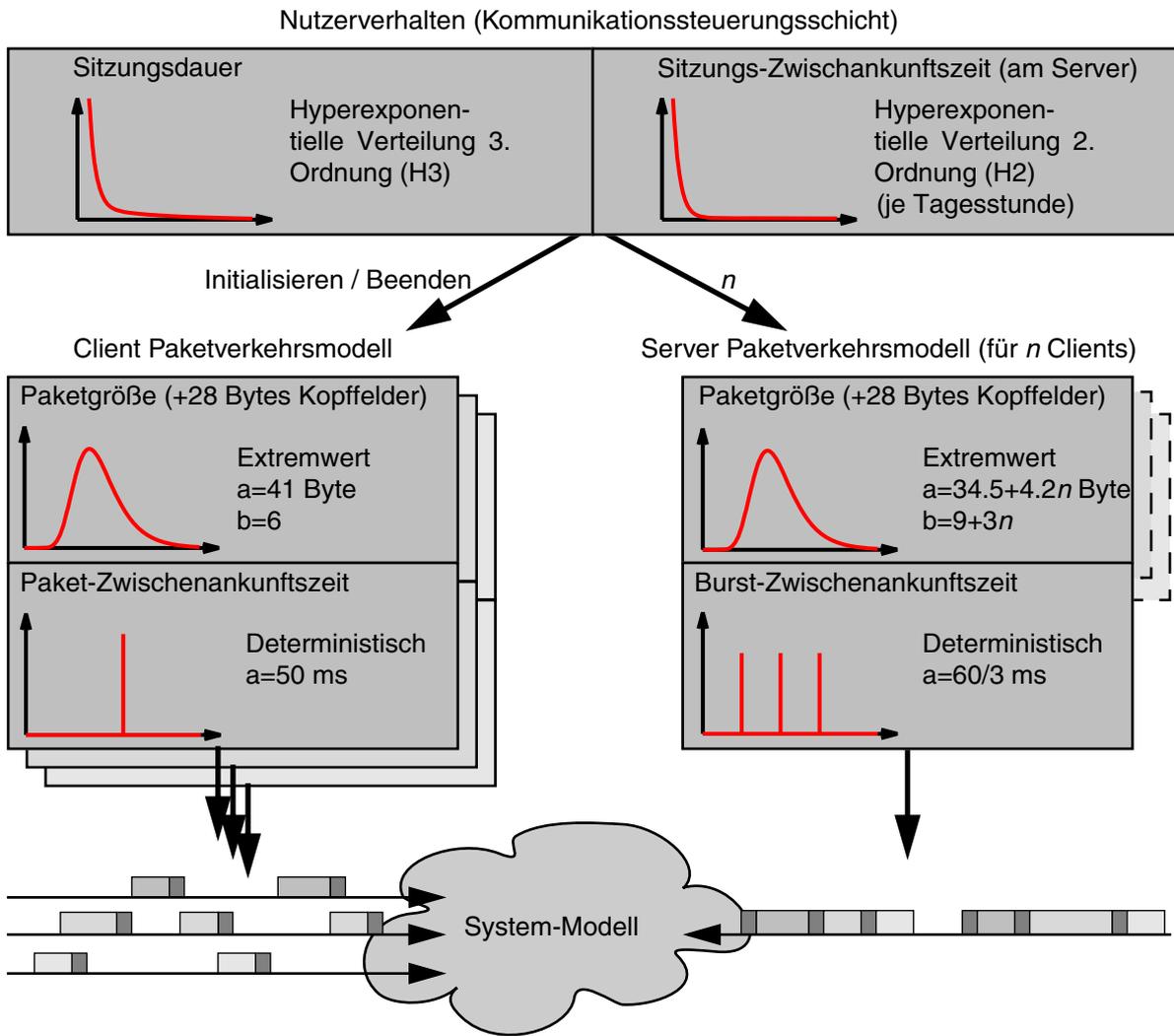


Bild 4.6: Verkehrsmodell für schnelle Netzspiele mit Client/Server-Architektur

kehrmodell ist ein Kompromiß zwischen detaillierter Verkehrsbeschreibung und Allgemeingültigkeit für die Klasse der Reaktionsspiele.

4.2 Empirische Bestimmung der Modellparameter

Das in Abschnitt 4.1 vorgeschlagene Verkehrsmodell bildet Spielerverkehr direkt auf der Paketschicht nach. Abgesehen von der Beschreibung der Anzahl aktiver Teilnehmer findet keine Lastmodellierung auf höheren Kommunikationsschichten mehr statt. Die Bestimmung der Modell-Parameter basiert daher direkt auf der Charakteristik des Paketverkehrs und hängt damit prinzipiell vom Zustand des Netzes und dem Ort der Messung ab.

Bild 4.7 zeigt ein typisches Szenario eines Reaktionsspiels. Ein neuer Teilnehmer ruft zunächst eine Liste aktiver Spiele-Server von einem zentralen Master-Server ab und kann sich dann für die Teilnahme an einem Spiel auf einem dieser Server entscheiden. Dabei helfen ihm Status-

Informationen, wie beispielsweise die Anzahl anderer aktiver Spieler, die gemessene Antwortzeit des Servers oder die eingestellten Spielregeln, die von den aufgelisteten Servern abgefragt werden. Teilnehmer, die sich für eine Partie entschieden haben, treten dem laufenden Spiel dynamisch bei und können es ebenso jederzeit wieder verlassen. Während einer Partie kommunizieren die Endgeräte nur noch über ihren Spiele-Server, fungieren also als Clients. Clients und Spiele-Server zeigen unterschiedliche Verkehrsmuster auf und werden entsprechend getrennt charakterisiert.

Im folgenden wird nur der Verkehr eines laufenden Spiels betrachtet. Messungen an den Punkten (1) bzw. (2) im Bild unterscheiden sich nicht nur bezüglich der Bündelung sondern insbesondere bezüglich der Verteilung der Paketankünfte, da diese in ähnlicher Größenordnung wie die im Internet typischerweise beobachtete Verzögerungsschwankung liegen. Es ist daher sinnvoll, sowohl auf Seite des Clients (2) also auch auf Seite des Servers (1) zu messen, um die Charakteristiken jeweils nahe der Verkehrsquelle ohne Netzeinflüsse zu erfassen.

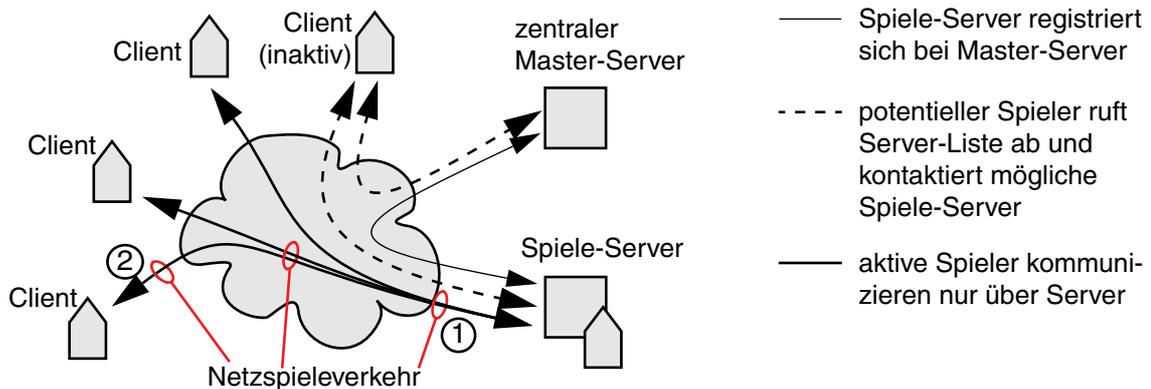


Bild 4.7: Client/Server-Architektur und unterschiedliche Meßpunkte

4.2.1 Messung

Da die Messung an einzelnen Clients ineffizient ist, wird stattdessen die Messung einer Spielpartie in einem lokalen Netz (LAN) herangezogen, in welchem störende Netzeinflüsse als vernachlässigbar angenommen werden können. Für die Erfassung auf Seite des Servers ist eine längere Beobachtungsdauer mit vielen Teilnehmern wünschenswert. Eine solche Messung erfolgte im Rahmen der Studien von Henderson [91] und seine Meßdaten konnten auch für die vorliegende Arbeit verwendet werden.

Die Messungen erfolgten über Monitor-Ausgänge am serverseitigen Switch bzw. Router über welchen der bidirektionale Verkehr des Spiele-servers an einen PC weitergeleitet wurde, der ihn mit Hilfe der Meß-Anwendung *tcpdump* aufzeichnete. Tabelle 4.5 zeigt dabei exemplarisch die erfaßten Kopffelder des IP-Protokolls. Da die Kommunikation einer Partie immer über einen zentralen Server läuft, taucht dessen IP-Adresse in jedem Paket auf. Im Falle der LAN-Mes-

sung können so auch mehrere parallel betriebene Spiele anhand der unterschiedlichen Server-Adressen unterschieden werden. Die wesentlichen Verkehrsinformationen sind der Zeitstempel, die Nutzlast (Paketlänge ohne 28 Bytes UDP/IP-Kopffelder), die Client-Adresse sowie die Übertragungsrichtung.

Tabelle 4.5: Erfasste Daten des Netzspielevverkehrs im Internet

Zeitstempel, Quelladresse, Zieladresse, Protokoll, Nutzlast				
988858926.726338	128.16.10.120.27015	> 24.1.157.49.27005:	udp	172 (DF)
988858926.729279	62.163.38.218.62165	> 128.16.10.120.27015:	udp	58
988858926.750525	62.179.162.68.27005	> 128.16.10.120.27015:	udp	40
988858926.750595	65.4.62.146.27005	> 128.16.10.120.27015:	udp	36
988858926.751198	172.148.7.253.27005	> 128.16.10.120.27015:	udp	46
988858926.758097	62.163.38.218.62165	> 128.16.10.120.27015:	udp	52
988858926.765623	24.1.157.49.27005	> 128.16.10.120.27015:	udp	44
988858926.766844	128.16.10.120.27015	> 24.249.99.111.6050:	udp	105 (DF)
988858926.767494	128.16.10.120.27015	> 172.148.7.253.27005:	udp	184 (DF)
988858926.780580	24.249.99.111.6050	> 128.16.10.120.27015:	udp	44
988858926.786115	128.16.10.120.27015	> 24.1.157.49.27005:	udp	141 (DF)
988858926.793359	62.163.38.218.62165	> 128.16.10.120.27015:	udp	45

Die LAN-Messung fand im Rahmen einer zweitägigen Veranstaltung der Studenten der Universität statt und erfasste Daten von mehreren Spiele-Servern. Sie diente auch der Bewertung der Meßgenauigkeit der PC-basierten Meßsysteme in [12]. Insgesamt wurde dabei Spielevkehr von etwa 50 Teilnehmern beobachtet.

Die Internet-Messung war deutlich umfangreicher. Im gesamten Mai 2001 wurden Paketinformationen aller Clients eines eigens aufgesetzten Spieleservers aufgezeichnet. Täglich tauschten etwa 35000 Clients 20 Millionen IP-Pakete mit dem Server aus. Der Großteil der Teilnehmer nimmt aber nicht an einer Partie teil, sondern informiert sich nur über die Netzanbindung des Servers. Betrachtet man nur solche Clients, die längere Zeit kommunizieren, reduziert sich ihre Zahl auf etwa 300 (>2400 Pakete am Tag) bzw. etwa 230 (>12000 Pakete am Tag). Während der Messung sind die sporadischen Besucher nicht zu erkennen und müssen daher bei der Datenauswertung erkannt und herausgefiltert werden.

4.2.2 Parameterbestimmung

Das Ziel der betrachteten Verkehrsmodellierung ist die Beschreibung der aktiven Phasen eines Online-Spiels. Die Messung kann aber nicht auf diese Phasen begrenzt werden und erfaßt daher den gesamten Verlauf mehrerer Spiel-Partien bzw. den gesamten Verkehr über mehrere Tage. Somit müssen bei der Parametrisierung die für die Modell-Beschreibung sinnvollen Größen aus der Gesamtmessung bestimmt und beschrieben werden.

Zunächst muß aus den gesamten Meßdaten der Verkehr jedes einzelnen Clients gefiltert werden, der durch die Client-IP-Adresse eindeutig identifiziert werden kann. Dadurch können die

Zwischenankunftszeiten des Client-Verkehrs bestimmt werden. Die zur Modellbeschreibung gewünschten Wahrscheinlichkeitsverteilungen der Zwischenankunftszeiten und der Paketgrößen können aus den Meßreihen der unterschiedlichen Clients entweder aus der Hintereinanderreihung aller Client-Meßreihen oder aber für jeden Client einzeln erzeugt werden. Die zuerstgenannte Vorgehensweise ist dann zulässig, wenn die Meßreihen der Clients gleichwertig angenommen werden können. Allerdings hat bereits Borella in [28] gezeigt, daß sich unterschiedliche Rechnerkonfigurationen auf die Verkehrscharakteristik auswirken. Auch die unterschiedlichen Netzanbindungen beeinflussen die Verkehrscharakteristiken der Clients – beispielsweise durch netzbedingte Verzögerungsschwankungen und damit unterschiedliche Varianzen der Paket-Zwischenankunftszeit. Aufgrund dieser Gegebenheiten führt eine gemeinsame statistische Auswertung aller Client-Verkehre zu einer "Verschmierung" der eigentlich sehr regelmäßigen Paket-Zwischenankunftszeiten, deren Mittelwert sich aber von Client zu Client unterscheidet.

Im Fall der Einzelauswertung für jeden Client ergibt sich eine große Schar von Wahrscheinlichkeitsverteilungen, die zur Modellbeschreibung auf eine einzige Verteilung zurückgeführt werden muß. Dabei ist zu entscheiden, ob Ergebnisse von Clients mit nur geringer Beteiligung am Spiel ebenso stark gewichtet werden, wie Ergebnisse von sehr aktiven Spielern, die den Großteil des tatsächlichen Verkehrs erzeugen. Außerdem wurde eine Abhängigkeit der Verkehrscharakteristik von der Anzahl der aktiven Teilnehmer des Spiels beobachtet, d.h. der Verkehr eines Clients ändert sich, wenn andere Teilnehmer dem Spiel beitreten oder es verlassen. Um dieses Verhalten zu beschreiben, müssen die entsprechenden Verkehrsgrößen für Phasen mit gleichbleibender Spielerzahl also wiederum unterschieden, d.h. getrennt erfaßt und ausgewertet werden.

Eine getrennte Auswertung der Pakete nach Client-Adresse (hunderte), Richtung (*upstream/downstream*) und Aktivitätsphasen mit konstanter Spielerzahl (2-32) würde allerdings zu tausenden von Verteilungsfunktionen führen, die nicht mehr sinnvoll auf eine einfach Parameter-Beschreibung zurückzuführen wären. Die hier vorgenommene Parameterauswertung basiert deshalb im Wesentlichen auf der blockweisen Auswertung der gesamten Meßdaten sowie der exemplarischen Auswertung ausgewählter aktiver Clients ohne Berücksichtigung der Aktivitätsphasen.

Bei der blockweisen Auswertung werden die Pakete jedes Clients innerhalb des Betrachtungsfensters (*sliding window*) für jede Übertragungsrichtung gezählt. Clients, die eine gewisse Paketrate überschreiten, werden als aktiv gewertet und ihre Paket-Zwischenankunftszeiten und Paketgrößen gehen in die statistische Auswertung für den für dieses Fenster bestimmten Aktivitätsgrad (Anzahl aktiver Clients) ein. Diese Methode weist dabei den großen Vorteil auf, daß außerhalb des aktuellen Betrachtungsfensters keine Daten einzelner Clients gehalten werden müssen (etwa zur Bestimmung der Paket-Zwischenankunftszeit) und ihre Komplexität somit

nicht vom Umfang der Meßreihe abhängt. Die Schwierigkeit liegt dabei zum einen in der Wahl der Fenstergröße, die in einem Bereich liegen sollte, in dem der Prozeß der Teilnehmeranzahl nicht stark schwankt und zum anderen in der Wahl der minimalen mittleren Paketrate, welche innerhalb des Fensters überschritten werden soll, um einen Client als aktiv einzustufen. Nach Auswertungen mit unterschiedlichen Fenstergrößen und Stabilitätsprüfungen der Ergebnisse wurde ein Wert von 30 Sekunden für die Fenstergröße als ausreichend ermittelt. Als minimale mittlere Paketrate wurden 5 Pakete pro Sekunde angenommen, um auch Clients mit nur geringer Aktivität einzubeziehen. Im Vergleich dazu sendet ein aktiver Client etwa 24 Pakete pro Sekunde, d.h. der Schwellwert wurde sehr vorsichtig gewählt. Bild 4.8 zeigt das Prinzip der Bestimmung der Teilnehmerzahl und die davon abhängige statistische Auswertung von Paketabständen und -größen.

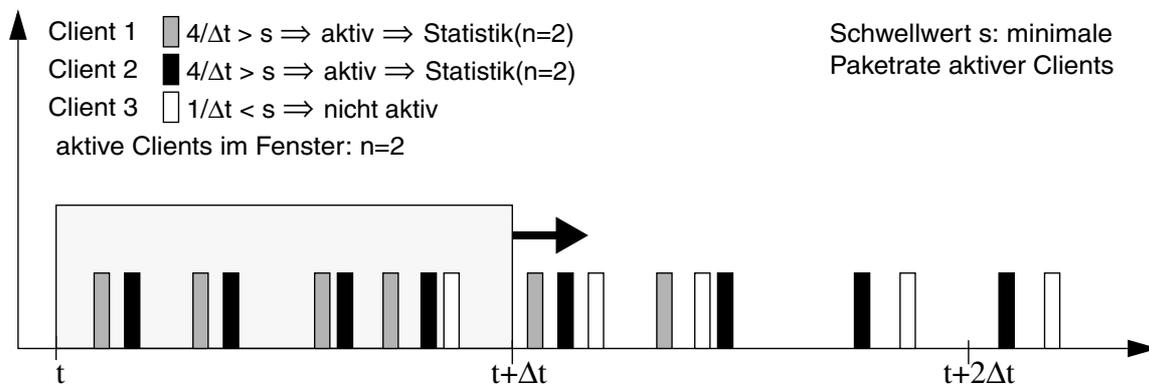


Bild 4.8: Blockweise Auswertung des Spielverkehrs mit dynamischer Teilnehmerzahl

Der Verlauf der so erhaltenen Mittelwerte von Paketgrößen und -abständen in Abhängigkeit von der Anzahl der aktiven Teilnehmer ist in Bild 4.9 dargestellt. Die mittleren Größen der vom Server erzeugten Pakete nehmen eindeutig linear mit der Anzahl der Spieler zu. Auch die mittlere Paketrate des Servers nimmt bei steigender Last leicht ab. Die Größen der Client-Pakete bleiben dagegen ebenso wie deren Rate in etwa konstant. Aufgrund dieser Ergebnisse wurde bei der Beschreibung der Modell-Parameter lediglich für die Paketgrößen des Servers eine Abhängigkeit von der Teilnehmerzahl berücksichtigt. Die Auswertung der Client-Paketgrößen erfolgt aus der Gesamtheit aller aktiven Clients ohne Berücksichtigung von Reihenfolge oder Client-Zuordnung. Für die Auswertung der Paket-Zwischenankunftszeiten muß

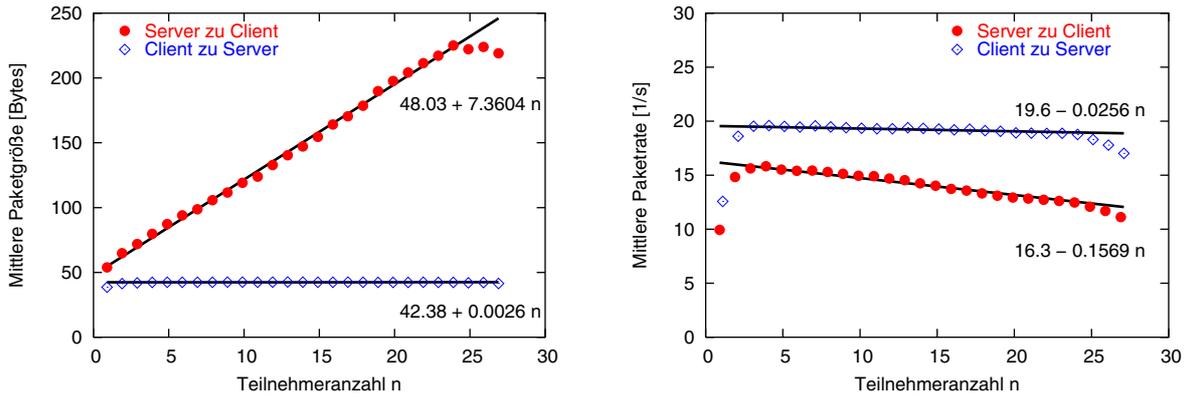


Bild 4.9: Abhängigkeit der Paketgröße und Paketrage von der Anzahl aktiver Teilnehmer

allerdings eine getrennte Auswertung je Client erfolgen, um deren unterschiedliche Verkehrscharakteristiken erfassen zu können.

4.2.2.1 Paketgrößen des Servers

Die oben beschriebene Abhängigkeit der Paketgrößen von der Anzahl der aktiven Teilnehmer äußert sich in unterschiedlichen Wahrscheinlichkeitsverteilungen für jede Aktivitätsphase. Bild 4.10 veranschaulicht, wie unterschiedlichen die Größenverteilungen bei 4, 8, 12 und 16 aktiven Teilnehmern ausfallen. Es ist ein Trend zu im Mittel größeren aber auch stärker gestreuten Paketen bei zunehmender Teilnehmerzahl erkennbar. Die Verteilungen wurden für jede Aktivitätsphase und für jeden der 30 Tage gesondert erstellt und durch Minimierung der Fehlerquadrate mit einer Extremwertfunktion angenähert, deren zwei Parameter das Maximum und die Form der Verteilung festlegen. Die Wahl der Extremwertfunktion beruht auf dem Wunsch nach Vergleichbarkeit zu der Modellierung durch Borella. Alternativ kann in diesem Zusammenhang auch die Lognormalfunktion eingesetzt werden. Für jede Aktivitätsphase wurden die Parameter der abgebildeten Gesamt-Approximation aus den Parametern der jeweils 30 approximierten Extremwertfunktionen durch Mittelung gewonnen.

Bild 4.11 zeigt auf der linken Seite diese Approximationen für alle Aktivitätsphasen und auf der rechten Seite den linearen Verlauf der Parameter der Extremwertfunktion. Damit ist eine einfache Beschreibung der Paketgrößen des Servers erreicht, denn die notwendige Paketgrö-

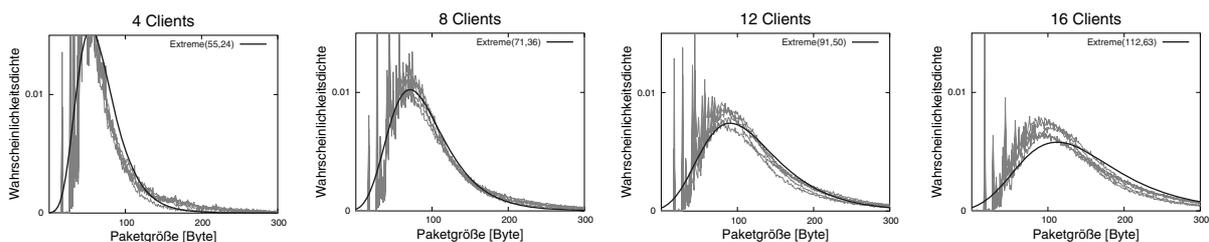


Bild 4.10: Verteilungsdichtefunktionen der Paketgrößen des Servers ausgewählter Meßreihen bei jeweils fester Anzahl aktiver Teilnehmer und ihre Approximation.

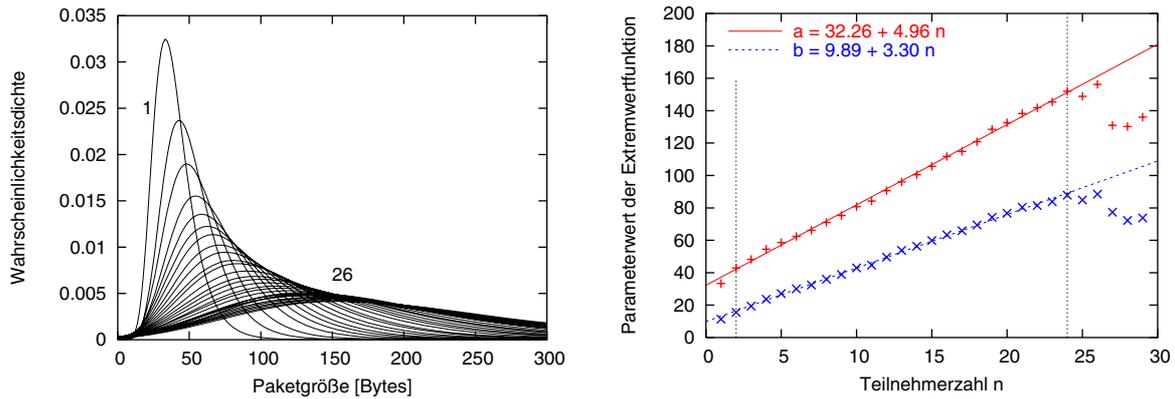


Bild 4.11: Die Approximationen der Wahrscheinlichkeitsdichtefunktion der Paketgrößen des Servers in Abhängigkeit von der Teilnehmerzahl (links) und die Beschreibung der entsprechenden Parameter der gewählten Extremwertfunktion (rechts).

ßen-Verteilung für die vorgegebene Anzahl aktiver Teilnehmer läßt sich einfach aus den linearen Funktionen der beiden Parameter bestimmen.

$$a = 34.5 + 4.2n \tag{4.3}$$

$$b = 9 + 3 n \tag{4.4}$$

Die so beschriebenen Größen stellen die Nutzlast der UDP-Pakete dar, d.h. es müssen noch 28 Bytes für die IP- und UDP-Kopffelder addiert werden. Die kleinste beobachtete Nutzlast beträgt 7 Bytes, die bei der Erzeugung der Zufallsgröße nicht unterschritten werden sollte. Ebenso müssen Paketgrößen auf insgesamt 1500 Bytes begrenzt werden.

4.2.2.2 Paket-Zwischenankunftszeiten des Servers

Für die Bestimmung der Paket-Zwischenankunftszeiten müssen längere Meßreihen für einzelne Clients ausgewertet werden, so daß der oben eingesetzte blockorientierte Ansatz nicht ausreichend genaue Ergebnisse liefern kann. Alternativ wurden 44 aktive Clients ausgewählt und ihre Paketabstände untersucht. Exemplarisch zeigt Bild 4.12 auf der linken Seite die entsprechende Wahrscheinlichkeitsdichtefunktion für einige einzelne Clients sowie für alle 44 Clients gemeinsam. Der charakteristische Verlauf zeigt eindeutig, daß Pakete an einzelne Clients streng periodisch etwa alle 60 ms ausgesendet werden und deutlich seltener um Vielfache von 10 ms verzögert werden. Die Schwankungsbreite der Verteilungen ist durch variable Serverlast und Wechsel der Teilnehmerzahl und damit der Sendereihenfolge einzelner Clients zu erklären.

Auf der rechten Seite zeigt Bild 4.12, daß der Summenverkehr nicht der naheliegende Büschelverkehr mit einem kompakten Büschel von Paketen an alle aktiven Clients alle 60 ms ist. Vielmehr verteilt der Server die Pakete in jedem 60 ms-Zyklus lastabhängig auf einen, zwei oder drei Büschel, bei t , $t + 20$ ms oder $t + 40$ ms. Dennoch soll für die Beschreibung des

Sendevorgangs im Verkehrsmodell des Servers vereinfachend eine konstante Zykluszeit mit gleichbleibender Aufteilung der Client-Pakete auf nur ein Bündel angenommen werden.

4.2.2.3 Paket-Zwischenankunftszeiten der Clients

Die Paket-Zwischenankunftszeiten der Clients müssen für jeden Client einzeln betrachtet werden, da es von Client zu Client deutliche Unterschiede geben kann. Die Verteilungen der Paketabstände bei Beobachtung auf der Seite des Servers (also des Empfängers), die in Bild 4.13 abgebildet sind, zeigen eine signifikante Varianz, die aber im Bereich der typischen Paketverzögerungsschwankungen liegt, so daß kein Rückschluß auf die Verkehrscharakteristik an der Quelle gezogen werden kann.

Eine Beobachtung des Client-Verkehrs auf Seite der Clients wurde durch die Messung im LAN vorgenommen, in welchem Paketverzögerungsschwankungen so gering sind, daß sie vernachlässigt werden können. Die Ergebnisse in Bild 4.14 zeigen für die Paket-Zwischenankünfte der Clients jetzt deutliche Spitzen um 40 ms bzw. 42 ms bei vernachlässigbarer Schwankung. Das Bild zeigt auch, daß teilweise Doppelspitzen auftreten, die vermutlich auf

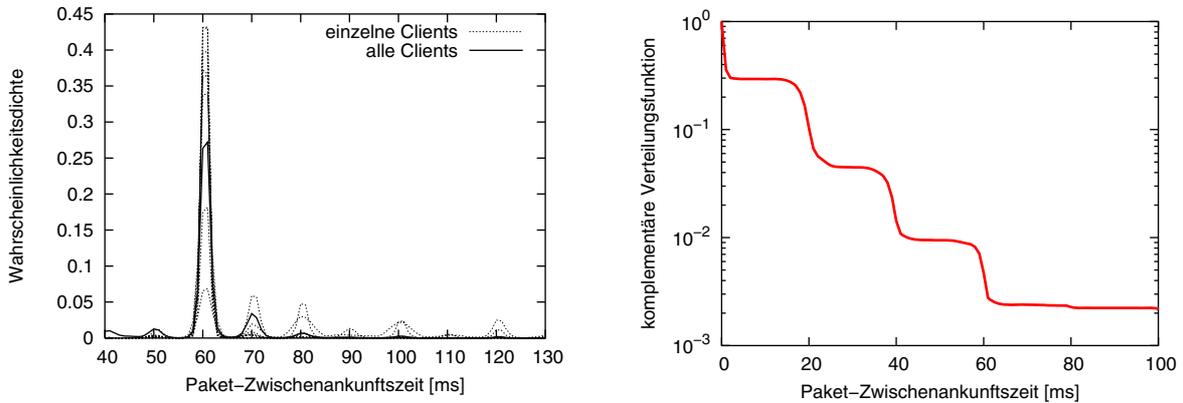


Bild 4.12: Paket-Zwischenankunftszeiten des Servers je Client (links) und für den Summenverkehr (rechts)

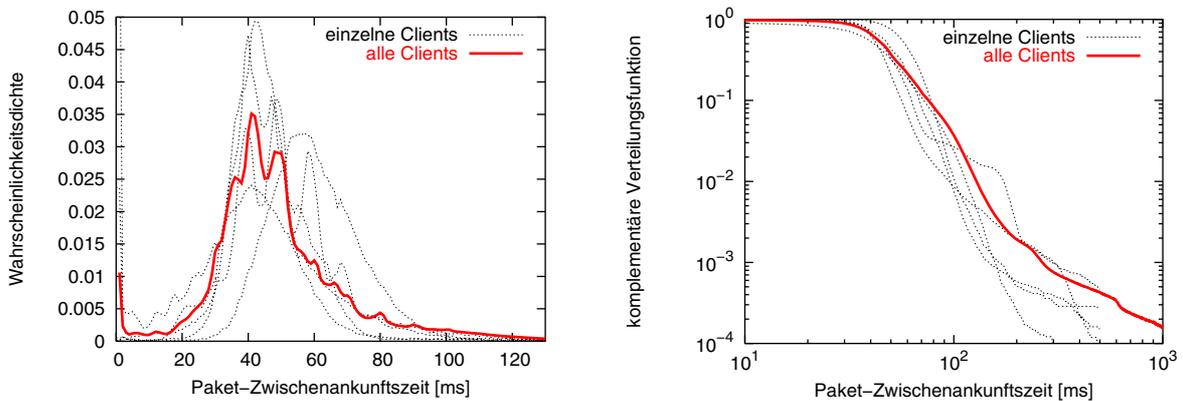


Bild 4.13: Paket-Zwischenankunftszeiten der Clients (Server-seitig)

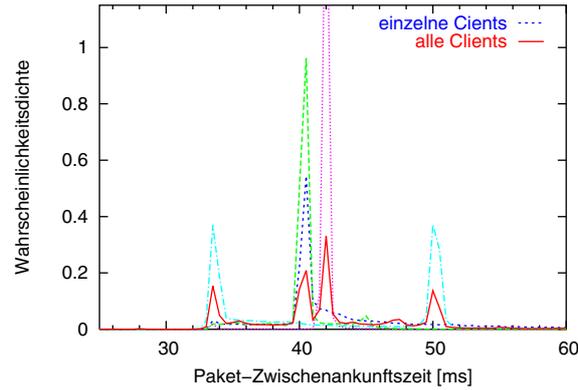


Bild 4.14: Paket-Zwischenankunftszeiten der Clients (Client-seitig)

Verzögerungen beim Aussenden zurückzuführen sind. Vereinfachend soll für die Modell-Beschreibung eine konstante Paketrage von etwa 24 Paketen/s (41 ms) angenommen werden.

4.2.2.4 Paketgrößen der Clients

Die Paketgrößen der Clients zeigen keine Abhängigkeiten von der Anzahl der aktiven Teilnehmer – da ein Client auch bei mehreren Teilnehmern keine zusätzlichen Informationen übertragen muß ist diese Unabhängigkeit plausibel. Die Verteilungen für die einzelnen Clients sind sich ähnlich, weisen allerdings eine etwas schmalere Charakteristik auf als die gemeinsame Verteilung aller Paketgrößen (Bild 4.15). Für die Approximation dieser Verteilung wird eine Extremwertfunktion vorgeschlagen, die durch Minimierung der Fehlerquadrate auf jeden der untersuchten 44 Clients optimiert wird. Durch Mittelung der jeweiligen Orts- und Formparameters ergibt sich die abgebildete Approximation für die Paketgrößenverteilung der Clients.

4.2.2.5 Teilnehmerverhalten

Die Anzahl der aktiven Teilnehmer eines typischen Reaktionsspiels ist aufgrund des kurzweiligen Spielprinzips und der Möglichkeit, am Spiel jederzeit teilzunehmen oder es zu verlassen sehr dynamisch. Wie oben gezeigt wird, hängen von der Teilnehmerzahl die Größen der Pakete

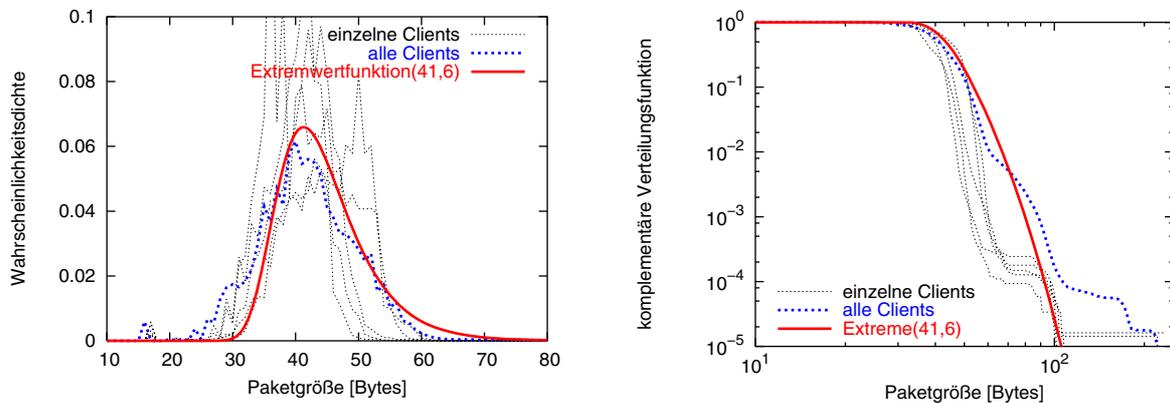


Bild 4.15: Paketgrößen der Clients

des Servers ab. Für eine vollständige Modellierung eines Netzspieles ist daher auch eine Beschreibung auf Kommunikationssteuerungsschicht (Sitzungen) wünschenswert.

Bei Untersuchungen von Teilbereichen der Spielekommunikation z.B. einem exemplarischen Teilnehmerzugang oder dem Serverzugang, genügt zur Erzeugung des typischen Verkehrs ein abstrakter Prozeß zur Beschreibung der Teilnehmerzahl, wie er beispielsweise in [90] durch einen $ARIMA(2, 1, 1) \times (0, 1, 1)_{48}$ Prozeß beschrieben wird (siehe Abschnitt 4.1.4).

Alternativ dazu wird die dynamische Teilnehmeranzahl im folgenden durch die Sitzungs-Zwischenankunftszeiten am Server und die Sitzungsdauern bestimmt, welche wiederum durch Erneuerungsprozesse beschrieben werden. Client-Quellen können bei Sitzungsbeginn dynamisch instanziiert und bei Sitzungsende gelöscht werden, womit eine Gesamtansicht einer Netzspiele-Kommunikation möglich wird.

Die Bestimmung dieser Sitzungsgrößen aus der Paketmessung erfordert zunächst die Erkennung der Sitzungsdaten jedes Clients. Wie weiter oben bereits beschrieben wurde, kann kein eindeutiger Rückschluß auf das Sitzungsverhalten gezogen werden, da auch Verkehr auftritt, der keiner Sitzung zugeordnet werden darf. Analog zum oben beschriebenen Vorgehen werden daher Betrachtungsfenster eingesetzt, in welchen ein Client als aktiv gewertet wird, wenn er eine mittlere Mindest-Paketrate erreicht. Da weiter oben in Bild 4.13 die mittlere Paketrate eines aktiven Clients kaum unter 10/s fällt (Zwischenankunftszeit 100 ms) und sich der Verkehr inaktiver Spieler in der Regel auf wenige Pakete beschränkt, wird hier eine Mindest-Paketrate von 10/s gewählt. Durch Zusammenfügen der aktiven Phasen eines Clients können für ihn zusammenhängende Sitzungen bestimmt werden. Allerdings treten häufig kurze Unterbrechungen auf, die nicht vom Ende und Neubeginn einer Sitzung unterschieden werden können. Daher werden Pausen bis zu 90 Sekunden nur als Unterbrechung innerhalb einer Sitzung gewertet. Bei der Wahl eines kleineren Schwellwertes erhöht sich die Anzahl der resultierenden Sitzungen signifikant und bei größeren Pausen bleibt sie in etwa konstant.

Bild 4.16 zeigt die komplementäre Verteilungsfunktion der Dauer der so ermittelten Sitzungen sowie die Annäherung durch eine hyperexponentielle Verteilung dritter Ordnung (H3) und eine Weibull-Verteilung. Die PP- und QQ-Plots in Bild 4.17 verdeutlichen, daß die Sitzungsdauer am besten durch die hyperexponentielle Verteilung beschrieben werden kann. Die Sitzungsdauer zeigt keine Abhängigkeit von der Tageszeit. In [90] wird zudem die Unabhängigkeit der Sitzungsdauer von der Anzahl der Teilnehmer gezeigt und für ihre Beschreibung wird eine negativ exponentielle Verteilung vorgeschlagen (Mittelwert vermutlich um 30 min). In [44] wird eine Weibull-Verteilung mit $\alpha=0.5$ und $\beta=20$ für die Sitzungsdauer von Online-Spielen vorgeschlagen. Die Weibull-Verteilung kann auch zur Beschreibung der in dieser Arbeit bestimmten Verteilung eingesetzt werden, allerdings mit anderen Parametern ($\alpha=0.67$ und

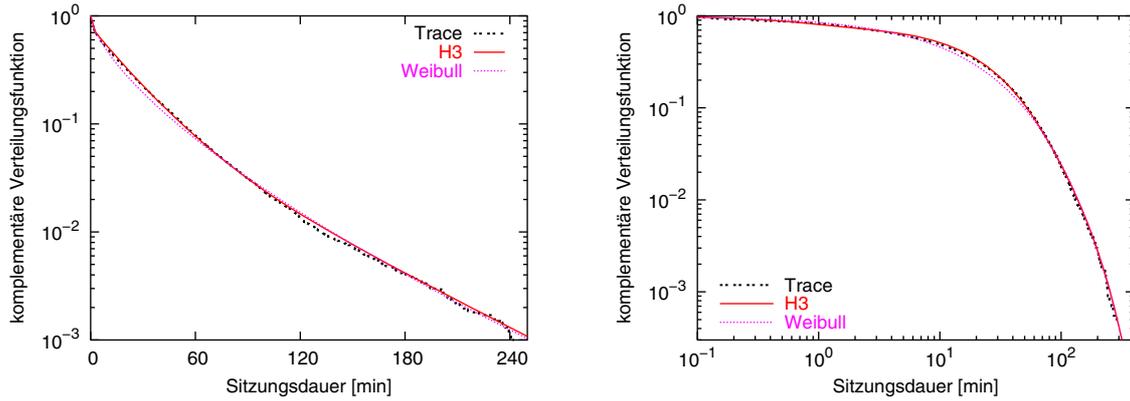


Bild 4.16: Verteilung der Sitzungsdauern im Spielverkehr

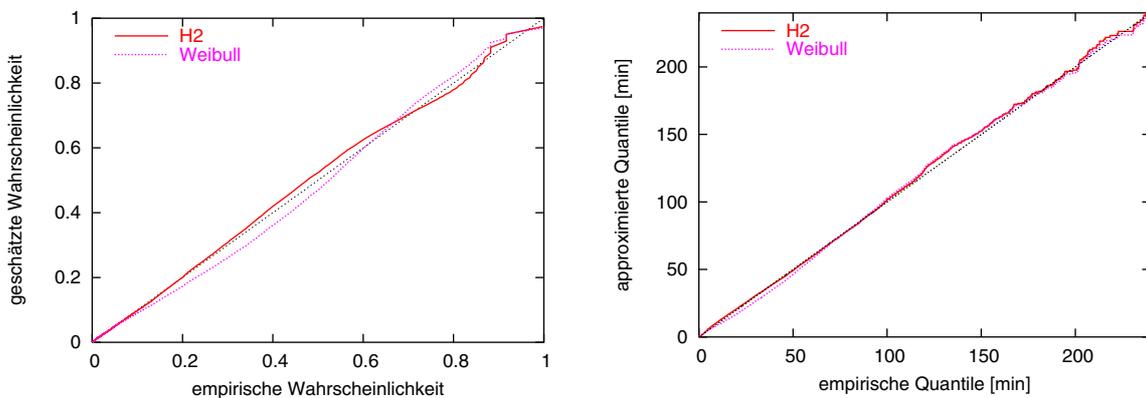


Bild 4.17: Bewertung der Approximation durch PP-Plot (links) und QQ-Plot (rechts).

$\beta=14.43$). Alle genannten Parameter beziehen sich auf eine in Minuten angegebene Sitzungsdauer.

Für die Beschreibung der Zwischenankunftszeiten der Sitzungen am Server muß die Tageszeit berücksichtigt werden. Bild 4.18 zeigt, daß die entsprechenden Verteilungsfunktionen für jede Stunde des Tages einem klaren Trend folgen, der eine erhöhte Intensität am Nachmittag und Abend zeigt. Am frühen Morgen beginnen neue Sitzungen mit deutlich höherem Abstand. Daß dennoch eine relativ hohe Rate erreicht wird, liegt vermutlich an der Teilnahme von Spielern in anderen Regionen der Welt, so daß die tageszeitlichen Abhängigkeiten nur durch die Mehrheit der geographisch nahe gelegenen Teilnehmer entstehen. Die in Bild 4.18 gezeigten Verteilungen lassen sich näherungsweise durch hyperexponentielle Verteilungen beschreiben, deren durch Minimierung der Fehlerquadrate bestimmte Parameter im rechten Teil der Bildes gezeigt sind. Die genauen Werte der Parameter können aus Tabelle B.4 im Anhang entnommen werden und beziehen sich auf Sitzungs-Zwischenankunftszeiten in Sekunden.

Nach [90] verringert sich die Zwischenankunftszeit für neu eintretende Spieler am Server leicht wenn bereits viele Teilnehmer aktiv sind, d.h. die Wahrscheinlichkeit, daß noch mehr

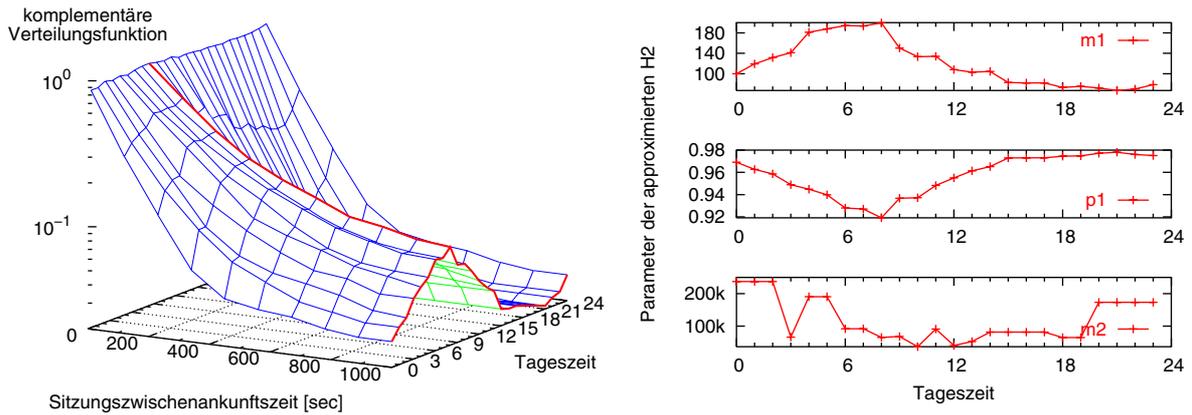


Bild 4.18: Verteilungen der Sitzungs-Zwischenankunftszeiten für jede Tagesstunde und die Parameter ihrer Approximationen mit der hyperexponentiellen Verteilung 2. Ordnung.

Spieler eintreten steigt mit der Anzahl bereits aktiver Spieler. Dieser Effekt tritt allerdings nur sehr schwach in Erscheinung und wird an dieser Stelle deshalb vernachlässigt.

Generell muß bei der Modellierung des Teilnehmerverhaltens selbstverständlich die Beschränkung auf die maximal unterstützte Spielerzahl für jeden Server (hier 32) beachtet werden.

4.2.3 Modellparameter

Die Parameter des Netzspiele-Modells sind in Tabelle 4.6 zusammengefaßt. Bei der Verwendung der Modells soll beachtet werden, daß die hier bestimmten Parameter auf der Auswertung nur eines Spiels beruhen, das zwar derzeit weit verbreitet ist und als repräsentatives Reaktionsspiel betrachtet werden darf, sich aber von Vertretern anderer Spieleklassen grundlegend unterscheidet. Da zukünftige Reaktionsspiele Gruppen-Sprachkommunikation integrieren, muß die Modellierung entsprechend erweitert werden.

Tabelle 4.6: Parameter des Netzspieleverkehrs-Modells

Parameter		Beschreibung
Teilnehmerzahl	Sitzungs-Zwischenankunftszeit am Server [sec]	hyperexponentielle Verteilungen 2. Ordnung je Tagesstunde z.B. Hauptverkehrsstunde 21:00-22:00 (s.a. Tabelle B.4): $p_1=0.978044635969065$, $p_2=1.0-p_1$ $m_1=67.0581631383333$, $m_2=173495.471425679$
	Sitzungsdauer [min]	hyperexponentielle Verteilung 3. Ordnung mit $p_1=0.123$, $p_2=0.666$, $p_3=1-p_1-p_2$ $m_1=52.74$, $m_2=20.78$, $m_3=0.72$
Server	Büschel-Zwischenankunftszeit [ms]	Deterministisch (60)
	Paketgröße [Byte]	$\max\{\text{Extremwertfunktion}(a(n),b(n)) + 28; 1500\}$ mit $a = 34.5 + 4.2n$ und $b = 9 + 3n$
Client	Paket-Zwischenankunftszeit [ms]	Deterministisch (41)
	Paketgröße [Byte]	$\max\{\text{Extremwertfunktion}(41,6) + 28; 1500\}$

4.3 Bewertung des Spieleverkehrsmodells

4.3.1 Bewertungsziel und Randbedingungen

Das vorgeschlagene Verkehrsmodell erzeugt für jeden Client einen Paketstrom mit relativ einfacher Verkehrscharakteristik. Für einen einzelnen Client betrachtet kann Spielerverkehr annähernd durch eine konstante Paktrate bei fester Paketgröße beschrieben werden. Die Besonderheit des Verkehrs liegt zum einen in der Dynamik der Teilnehmerzahl und der davon abhängigen Charakteristik des Summenverkehrs, die insbesondere für vom Server erzeugten Verkehr signifikant variiert. Zum anderen müssen besondere Echtzeitbedingungen eingehalten werden, um eine akzeptable Dienstgüte zu erreichen. Die Bewertung des Modells soll entsprechend dieser beiden Besonderheiten durch Simulation erfolgen. Das Ziel der Bewertung ist, mangels vergleichbarer Modelle und aufgrund der einfachen Verkehrscharakteristik, nicht die Validierung des Modells, sondern die Demonstration seiner sinnvollen Anwendbarkeit in einem exemplarischen Szenario. Zunächst sollen das Simulationsszenario skizziert sowie einige Randbedingungen der Implementierung erläutert werden.

Im gewählten Szenario wird nur der vom Server erzeugte Summenverkehr betrachtet. Der Spielerverkehr wird auf Empfängerseite gemessen, so daß Störungen erfaßt und ausgewertet werden können. Die Störungen in Form von Paketverlusten und Paketverzögerungen werden durch Überlagerung mit einem abstrakten büschelförmigen Hintergrundverkehr und durch eine begrenzte Warteschlange erzeugt (Bild 4.19). Der Hintergrundverkehr setzt sich aus mehreren unabhängigen M /Pareto-Quellen zusammen, die jeweils mit negativ exponentiell verteilter Burst-Zwischenankunftszeit von 100 ms Nachrichten erzeugen, deren Länge einer Pareto-Verteilung folgt und die in Pakete zu maximal 1500 Bytes aufgeteilt werden. Innerhalb eines Bursts werden die Pakete mit etwa 100 Mbit/s an den Multiplexer gesendet. Die Nachrichtenlänge variiert mit einem Variationskoeffizienten $c = 1.5$, was bei der Pareto-Verteilung einem Parameterwert $\alpha \approx 2.2$ entspricht. Alternativ können Pakete auch mit voreingestellter Verlustwahrscheinlichkeit verworfen werden. Pakete des Spielerverkehrs werden gesondert markiert und können so im Demultiplexer unterschieden und ihre Verkehrscharakteristik gemessen wer-

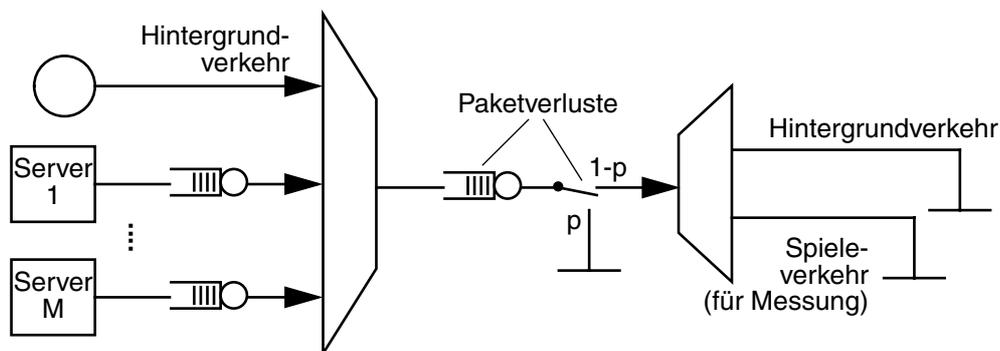


Bild 4.19: Simulationsmodell des Server-erzeugten Spielerverkehrs

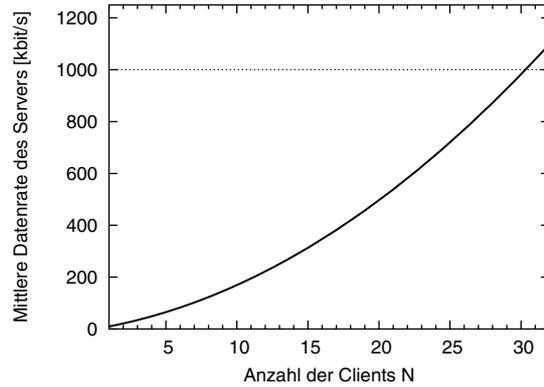


Bild 4.20: Vom Spiele-Server erzeugtes Angebot

den. Die Bedieneinheit verarbeitet die ankommende Last byteweise, so daß eine Übertragungskapazität von 10 Mbit/s beschrieben wird. Der Spiele-Server erzeugt maximal etwa 1.2 Mbit/s (s. Bild 4.20), so daß die Verbindung ohne Hintergrundverkehr ausreichend dimensioniert ist.

Das Szenario wird mit unterschiedlichen Parametern betrachtet. Im einen Fall (Szenario 1) wird die mittlere Hintergrundlast A_H nur bis etwa 60% erhöht (mittlere Nachrichtenlänge 1250 Bytes, bis zu 60 Quellen), so daß kaum Paketverluste und geringe Verzögerungsschwankungen zu erwarten sind. Im zweiten Fall (Szenario 2) wird die mittlere Hintergrundlast A_H auf das bis zu 6-fache der Leitungskapazität erhöht (mittlere Nachrichtenlänge 12500 Bytes, bis zu 60 Quellen), so daß erhebliche Paketverluste auftreten. Die so zusammen mit dem Spielerverkehr erzielte tatsächliche Auslastung ρ ist in Bild 4.21 abgebildet und die hohen Paketverluste, die in Szenario 2 erzeugt werden, in Bild 4.22.

Die Verlustwahrscheinlichkeit an der Warteschlange sinkt bei steigender Teilnehmerzahl, da hierdurch der Anteil des regelmäßigen Spielerverkehrs zunimmt und die Verluste, die hauptsächlich bei Bursts des Hintergrundverkehrs auftreten, relativ zurückgehen. Die absoluten Paketverluste steigen natürlich.

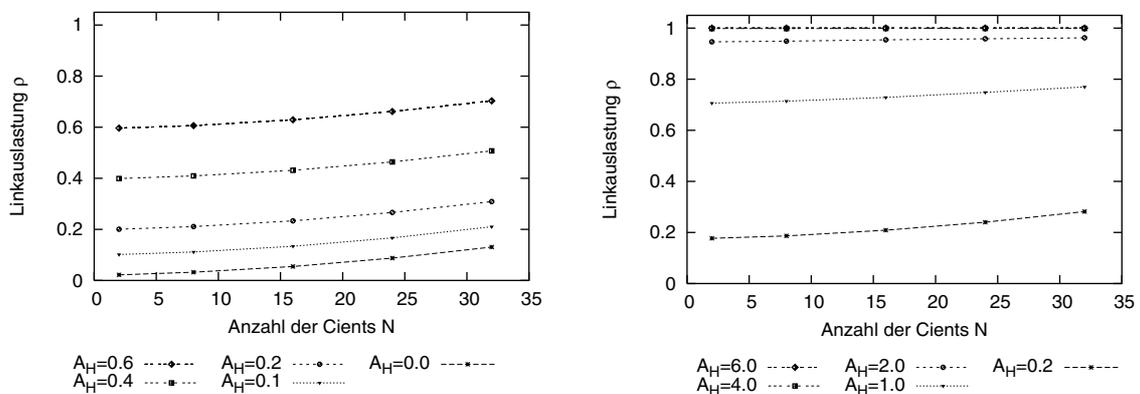


Bild 4.21: Auslastung durch Hintergrundverkehr und Spielerverkehr in Szenario 1 (links) und Szenario 2 (rechts)

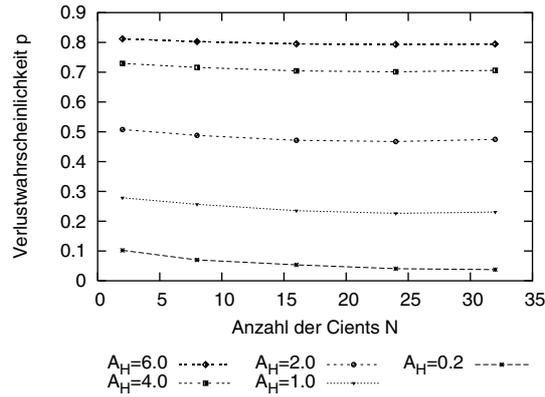


Bild 4.22: Paketverlusten an der Warteschlange in Szenario 2

Für beide Szenarien werden bei steigender fester Spieleranzahl verschiedene Dimensionierungen für die Warteschlangengröße Q und das Verhalten bei unterschiedlicher Hintergrundlast A_H bzw. Paketverlustwahrscheinlichkeit p (ohne Hintergrundlast) untersucht und abschließend eine Bewertung der jeweils erreichten Dienstgüte vorgenommen.

Für die simulative Untersuchung des vom Server erzeugten Verkehrs wurde ein Generator auf der Grundlage der Simulationsbibliothek Simlib realisiert [172]. Da Spieleverkehr auf UDP aufsetzt, ist keine Nachbildung der komplexen TCP-Mechanismen notwendig. Allerdings erfordert die gewählte Modellierung eine Ergänzung der Simulationsbibliothek, um die Abhängigkeit der Paketgrößenverteilung von der Anzahl aktiver Teilnehmer zur Laufzeit der Simulation zu beschreiben.

4.3.2 Untersuchung der Verkehrscharakteristik

Die Charakteristik des Spieleverkehrs wird durch normale Hintergrundlast (Szenario 1) kaum beeinflusst. Es wird lediglich eine Paketverzögerungsschwankung aufgrund der Warteschlange beobachtet, die weiter unten kurz betrachtet wird. Im Szenario 2 werden dagegen starke Paketverluste erzeugt, die zu einem deutlichen Einbruch der Übertragungsrate führen (Bild 4.23).

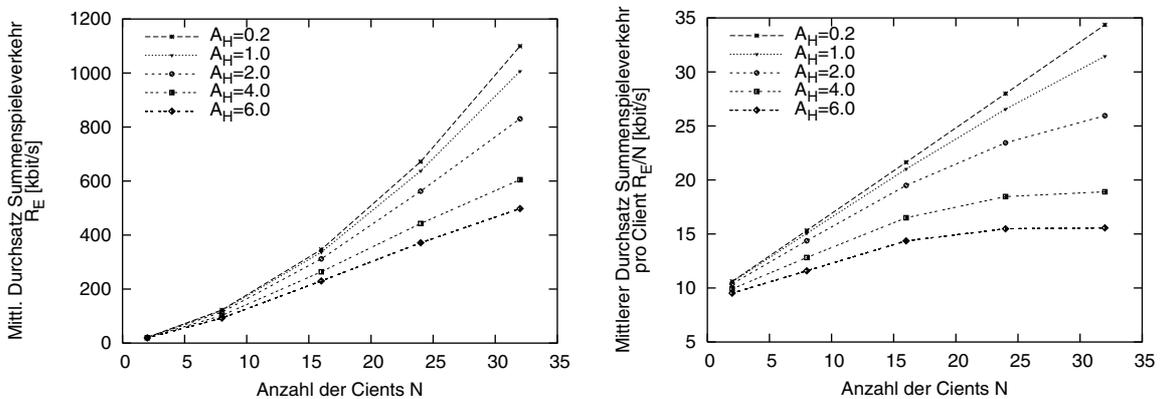


Bild 4.23: Gesamtdurchsatz des Spieleverkehrs (links) und Durchsatz pro Client (rechts)

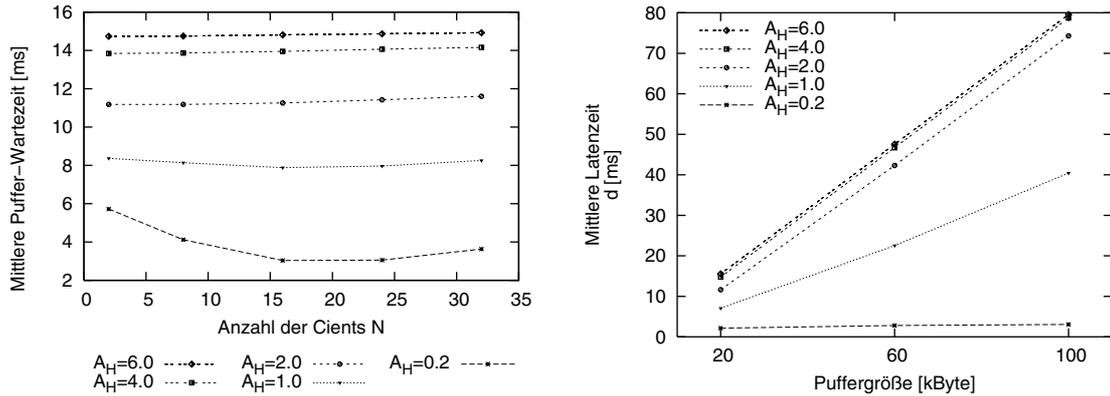


Bild 4.24: Wartezeit am Puffer (links für 20k Puffergröße) und Latenzzeiten (rechts) bei unterschiedlicher Netzauslastung

Durch hohe Paketverluste sinkt sowohl der Gesamtdurchsatz des Spielerverkehrs (links) als auch der Durchsatz pro Client (rechts) deutlich.

Die Übertragungsrate allein erlaubt nur eine grobe Bewertung der Dienstgüte für einen Client. Das gilt insbesondere dann, wenn dem Meßsystem die Anzahl aktiver Spieler unbekannt ist. Wichtigstes Dienstgütekriterium von Spielerverkehr ist die Antwortzeit, die durch Paketverzögerung und Paketverluste bestimmt wird.

Im betrachteten Szenario wird keine absolute Paketverzögerung modelliert. Die zusätzliche Modellierung einer typischen Laufzeit würde sich lediglich in einem entsprechenden Offset niederschlagen und wurde für diese Betrachtung unterlassen. Damit ergibt sich die Latenzzeit allein aus der Wartedauer und der Bearbeitungsdauer der Bedieneinheit.

In Bild 4.24 ist links gezeigt, wie sich die Wartedauer und auch die Latenzzeit bei Überlast erhöht. Die Wartedauer ist nahezu unabhängig vom Teilnehmer-abhängigen Spielerverkehr, da sie hauptsächlich durch den dominanten Hintergrundverkehr entsteht. Lediglich bei geringer Hintergrundlast ($A_H = 0.2$) sind Spiele- und Hintergrundverkehr etwa gleichwertig und die mittlere Wartezeit der Spielepakete sinkt wenn sich der Anteil des gleichmäßigeren Spielerverkehrs mit der Anzahl der Teilnehmer erhöht, der zwischen Burst-Phasen sehr schnell abgearbeitet werden kann. Bei Überlast füllt sich der Puffer und die Wartedauer hängt direkt von der Puffergröße ab.

Aus der Regelmäßigkeit der eintreffenden Pakete können Rückschlüsse über Verzögerungsschwankungen und Paketverluste gezogen werden. Die Auswirkungen von Paketverlusten werden im folgenden gesondert betrachtet. Aufgrund der periodisch gesendeten Pakete ergibt sich bei Paketverlusten eine charakteristische stufenförmige Verteilung der Paketzweikanakunftszeit (Bild 4.25). Bei steigender Verlustwahrscheinlichkeit flacht die Verteilung ab. Der Abstand der Stufen entspricht dabei dem Sendeintervall und ihre Höhe der Wahrscheinlichkeit für keinen, einen und mehrere Paketverluste in Folge. Die Verlustwahrscheinlichkeit wurde hierbei auf feste Werte gesetzt und die Pakete wurden durch keinerlei Hintergrundverkehr

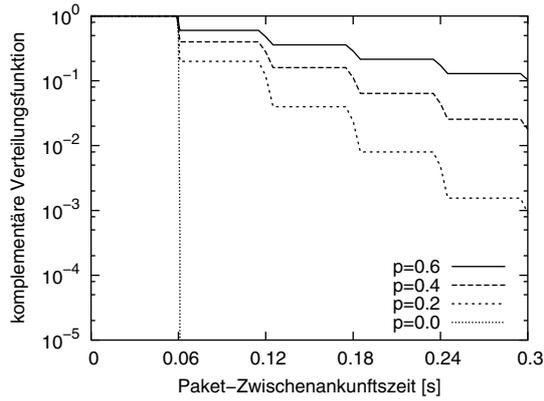


Bild 4.25: Paketverluste verursachen stufenförmigen Verlauf

gestört, so daß andere Ursachen für Verzögerungsschwankungen ausgeschlossen werden können. Damit kann der Verlauf als charakteristische Auswirkung von Paketverlusten betrachtet werden.

In Verbindung mit Störverkehr zeigt sich ein etwas anderes Verhalten, da sowohl Verzögerungsschwankungen als auch Paketverluste entstehen. Bild 4.27 zeigt den entsprechenden Ver-

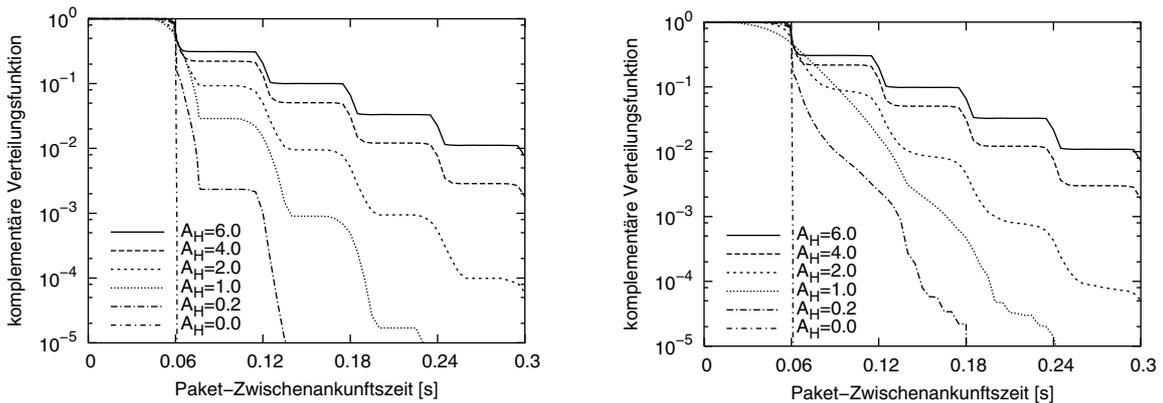


Bild 4.26: Paketverluste verursacht durch Hintergrundverkehr mit Überlast (A_H)
(links: Warteschlangenlänge 20kBytes, rechts: 100kBytes)

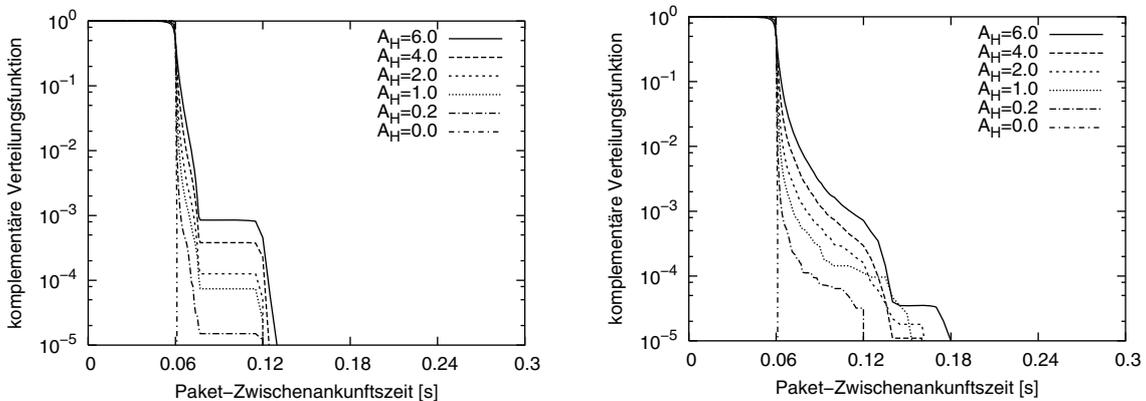


Bild 4.27: Paketverluste verursacht durch Hintergrundverkehr mit normaler Last (A_H)
(links: Warteschlangenlänge 20kBytes, rechts: 100kBytes)

lauf der Paketzwehenankunftsverteilung für das Szenario 1 (normale Last), in welchem es kaum zu Paketverlusten kommt. Besonders bei Verwendung eines größeren Puffers (rechte Seite) zeigt die Verteilung keine klaren Stufen mehr auf, sondern verschmiert. Bei Überlast (Szenario 2) zeigt sich eine deutlich abweichende Verteilung (Bild 4.26). Hier treten vermehrt Paketverluste auf, welche zur Abflachung des Verlaufs führen und insbesondere bei extremer Überlast treten wieder charakteristische Stufen in Erscheinung. Bei großem Puffer (rechte Seite) überschreitet die maximale Wartedauer das Sendeintervall, so daß sich nur mehrere Verluste in Folge in stufigem Verlauf zeigen. Bei extremer Überlast ist die Warteschlange wiederum dauerhaft gefüllt, wodurch die zusätzliche Verzögerung weniger stark schwankt und wieder ein Treppenverlauf zu sehen ist.

4.3.3 Dienstgütebewertung

Mit der in Abschnitt 4.1.3 vorgestellten Bewertungsmethodik kann der Zusammenhang von Paketverzögerung und Schwankung der Paketzwehenankünfte so dargestellt werden, daß sich die am Meßpunkt beobachtete Dienstgüte direkt ablesen läßt. Da das Verfahren auf dienstspezifische Messungen verzichtet, wird die Dienstgüte allerdings nur für eine Richtung ermittelt und muß gegebenenfalls durch eine entsprechende Messung für die Gegenseite ergänzt werden. Bei der unidirektionalen Betrachtung wird kein Zusammenhang zwischen Nachrichten und Antworten hergestellt, so daß die durch Paketverluste verursachte Verzögerung der Antwortzeit nicht direkt erfaßt wird. Um die für die Dienstgüte relevante Verzögerung abzubilden, muß daher eine von der Verlustwahrscheinlichkeit abhängige Anpassung der beobachteten Paketverzögerung erfolgen. Bild 4.28 zeigt die in der Simulation erreichte Dienstgüte für ein Spiel mit 24 Teilnehmern und einer festen Laufzeit von 50 ms bei vier unterschiedlichen fest eingestellten Paketverlustwahrscheinlichkeiten. Der Einfluß der Warteschlange kann in diesem Szenario vernachlässigt werden. Der untere Verlauf ergibt sich ohne Anpassung der

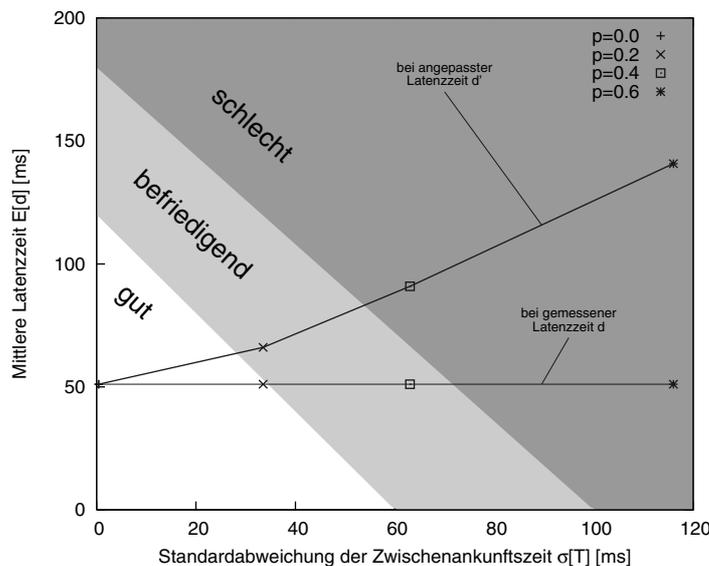


Bild 4.28: Dienstgütebewertung bei festen Paketverlustwahrscheinlichkeiten

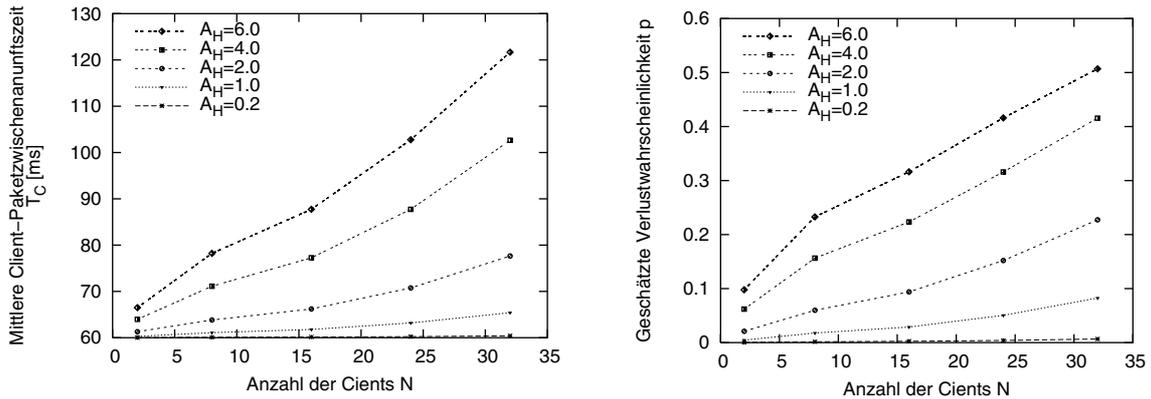


Bild 4.29: Mittlere Paketzwischenankunftszeit je Client (links) und die daraus abgeschätzte Paketverlustwahrscheinlichkeit (rechts)

Verzögerung: Paketzwischenankünfte schwanken allein aufgrund von Paketverlusten. Wird berücksichtigt, daß sich bei Paketverlusten auch die Antwortzeit (Lag) erhöht, ergibt sich der obere Verlauf mit einer deutlich pessimistischeren Bewertung.

Da es im betrachteten Szenario bei normaler Last (Szenario 1) weder zu signifikanten Paketverlusten noch zu Paketverzögerungen kommt, wird es bei allen gewählten Parametersätzen mit guter Dienstgüte bewertet (ohne Abbildung). Im Überlastfall von Szenario 2 stellt sich aber die Frage, inwieweit trotz Störungen noch eine akzeptable Dienstgüte erreicht wird.

Zunächst muß die durch Pufferüberlauf verursachte Paketverlustwahrscheinlichkeit bestimmt werden. Bei bekanntem Sendeintervall kann die Paketverlustwahrscheinlichkeit aufgrund der mittleren Paketzwischenankunftszeit nach Gleichung (4.2) berechnet werden. Bild 4.29 zeigt die mittlere Paketzwischenankunftszeit am einzelnen Client für unterschiedliche Verkehrswerte und die entsprechende Abschätzung der Paketverlustwahrscheinlichkeit.

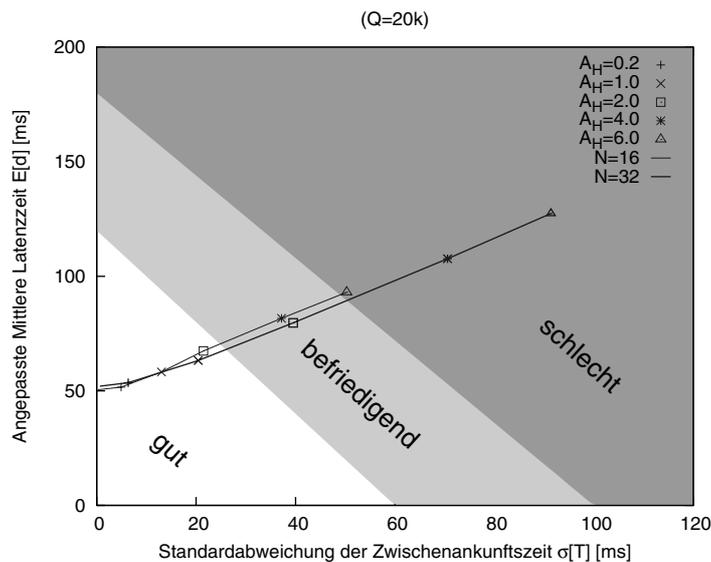


Bild 4.30: Dienstgüteebewertung für zunehmende Last (Hintergrundlast und Teilnehmerzahl)

Damit kann die Dienstgüte im Überlast-Szenario eingeschätzt werden. Bild 4.30 zeigt wie sich die Dienstgütbewertung bei steigendem Hintergrundverkehrsangebot erwartungsgemäß verschlechtert. Bei hoher Spielerzahl ist der störende Einfluß des Hintergrundverkehrs deutlich stärker. Bild 4.31 zeigt die Situation bei konstantem Hintergrundverkehr mit Verkehrsangebot $A_H = 2$. Bei steigender Spielerzahl verschlechtert sich die Dienstgüte, da das Gesamtverkehrsangebot und damit auch die Verluste steigen, so daß sich sowohl die angepaßte Latenzzeit als auch die Schwankung der Paketankünfte erhöht. Deutlich ist zu erkennen, daß der Einfluß großer Warteschlangen besonders bei Überlast negativ in Erscheinung tritt. Bild 4.32 zeigt nochmal das Verhalten bei steigender Hintergrundlast für 24 Teilnehmer und für alle drei Warteschlangenlängen. Deutlich ist zu sehen, daß die Warteschlangen erst dann zu einer signifikant

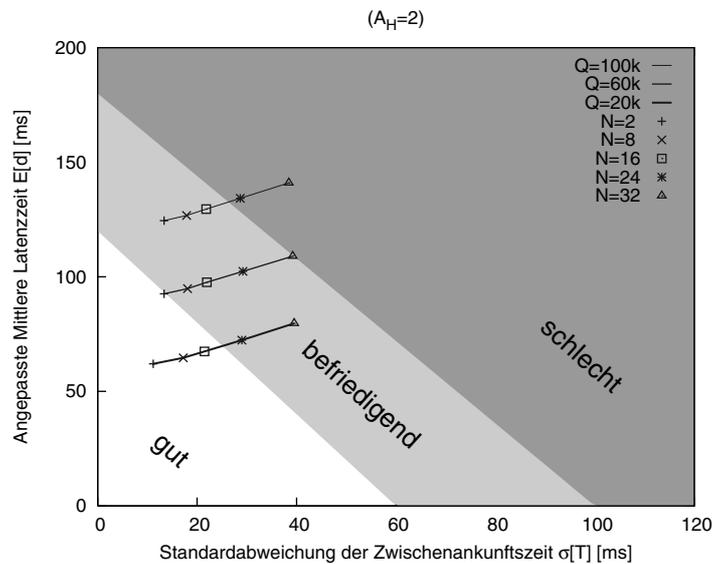


Bild 4.31: Dienstgütbewertung bei steigender Teilnehmerzahl und für unterschiedliche Puffergrößen

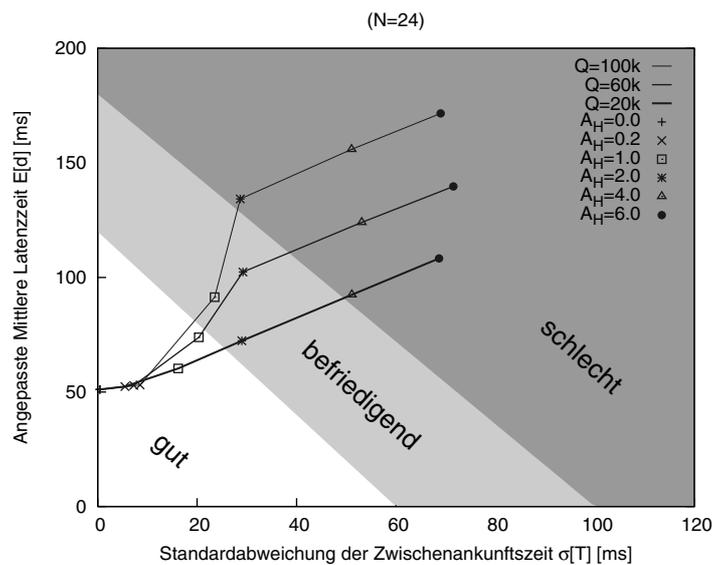


Bild 4.32: Dienstgütbewertung für zunehmende Hintergrundlast und für unterschiedliche Puffergrößen

erhöhten Latenzzeit führen, wenn der angebotene Hintergrundverkehr die Leitungskapazität überschreitet.

Das vorgeschlagene Verkehrsmodell und die zugehörige Bewertungsmethodik ermöglichen dienstspezifische Untersuchungen von Kommunikationsnetzen und ihre Bewertung hinsichtlich der erreichten Dienstgüte. Es erlaubt Studien, welche die Auswirkungen und Abhängigkeiten der Dienst- und Netz-Parameter sichtbar machen, so daß Netze entsprechend dimensioniert oder Dienste entsprechend eingegrenzt werden können.

Kapitel 5

Zusammenfassung und Ausblick

In der vorliegenden Arbeit werden Verkehrsmodelle zur Erzeugung von Verkehren der spezifischen Internet-Dienste WWW und Netzspiele auf Paketebene vorgestellt. Diese Modellierung ermöglicht simulative Untersuchungen des Verhaltens von Kommunikationseinrichtungen bei Belastung durch den spezifischen Dienst und kann zur Bewertung der Eignung eines Netzes für diesen Dienst oder zur Untersuchung der Einflüsse des Dienstes auf ein Netz genutzt werden.

Eine Besonderheit der Modellierung einzelner Dienste liegt in der Möglichkeit, auch die Endteilnehmer der Kommunikation einzubeziehen, d.h. die Kommunikationsbeziehungen auf höheren Schichten nachzubilden und den Verkehr der Paketebene indirekt zu erzeugen. Ein solcher hierarchischer Ansatz führt zu einer noch überschaubaren Zahl von Modellparametern mit konkreter und verständlicher Bedeutung. Zudem erlaubt er, Rückkopplungen auf mehreren Ebenen nachzubilden und er ermöglicht die Erfassung von Dienstgüte-Metriken aus Nutzer- oder Anwendungssicht. Ein solches Modell kann auf diese Weise die Verkehrslast durch invariante Parameter beschreiben und echten Ende-zu-Ende-Verkehr für ein beliebig gewähltes Netzmodell erzeugen, ohne daß eine Anpassung an den Netzzustand oder den Betrachtungsort erfolgen muß.

Die Modellierung erfolgt auf Basis eigener Messungen, welche zur Verkehrsanalyse für den Modellentwurf sowie zur Modellparametrisierung durchgeführt wurden. Die Messung im Netz erlaubt nur eine begrenzte Sicht auf die Kommunikation auf höheren Schichten, so daß die nicht direkt bestimmbar Parameter geschätzt werden müssen. Es werden entsprechende Schätzmethode untersucht und validiert.

WWW-Verkehrsmodell

In der simulativen Bewertung des vorgeschlagenen WWW-Verkehrsmodells zeigt sich, daß die Rückkopplungseffekte auf Nutzerebene insbesondere bei schlechten Netzzuständen (Paketver-

lust, Paketverzögerung) bzw. langsamen Netzzugängen und langen Server-Antwortzeiten sichtbar werden. Der direkte Vergleich zu SURGE, einem sehr beliebten Modell, das ohne Rückkopplung auf Nutzer- und Dienstebene arbeitet, zeigt aber, daß diese Rückkopplungen im normalen Betriebszustand einen vernachlässigbaren Einfluß auf die Paketverkehrscharakteristik haben, da beide Modelle sehr ähnliche Ergebnisse liefern. Bei der Bildung von stärker aggregiertem Verkehr durch die Überlagerung von Verkehren vieler WWW-Quellen verschwimmen die Charakteristiken des Paketverkehrs, so daß alternativ auch einfachere Modelle zur Erzeugung von Summenverkehren genutzt werden können.

Die besondere Stärke des Modells liegt in der Möglichkeit zur Untersuchung des Verkehrsverhaltens eines Dienstes. Damit können Einflüsse verschiedener Modellparameter (und damit auch Dienst-Parameter) mit physikalischer Bedeutung auf die Verkehrscharakteristik bzw. auf die Netzbelastung untersucht werden (z.B. maximale Anzahl paralleler Verbindungen). Es zeigt sich, daß bei der Überlagerung vieler Quellen auch das funktionell orientierte Modell erwartungsgemäß zu selbstähnlichem Paketverkehr führt. Damit können auch Einflüsse von Modellparametern auf den Summenverkehr des Dienstes untersucht werden. Eine weitere Stärke liegt in der Sichtbarkeit der Dienstgüte aus Nutzersicht (Seitenladezeit), also der möglichen Bewertung der Nutzerzufriedenheit.

Einschränkend muß festgestellt werden, daß bei der Modellierung des Dienstes eine Verallgemeinerung und Vereinfachung vorgenommen wurde, so daß sich das Modell im Grunde auf das einfache Abrufen von Webseiten beschränkt. Der Dienst WWW hat sich in den letzten Jahren aber deutlich darüber hinaus weiterentwickelt und seinen Charakter durch Streaming-Anwendungen, Web-Services und Web-Proxies deutlich geändert. Die Modellierung beschränkt sich daher nur noch auf einen Anwendungsteil des gesamten Dienstespektrums und sollte bei Bedarf entsprechend ergänzt werden.

Die Bewertung des hierarchischen WWW-Modells ist aufgrund der Rückkopplungen sehr schwierig, da damit nicht nur die Modell-Parameter betrachtet, sondern auch die Parameter des Netzes (Übertragungsrate im Netzzugang und -kern, Laufzeit, Puffergrößen, Verlustwahrscheinlichkeiten, ...) berücksichtigt werden müssen.

Spieleverkehrsmodell

Das vorgeschlagene Spieleverkehrsmodell ist schwieriger zu bewerten, da es einen gleichmäßigeren Verkehr mit geringer Bitrate ohne jegliche Rückkopplung erzeugt, der fast ebensogut durch eine konstante Übertragung beschrieben werden kann. Der Nutzen der Betrachtung liegt vielmehr in der Verkehrscharakterisierung und der Erkenntnis von typischen Verkehrsgrößen und Dienstgüte-Metriken. Das Spieleverkehrsmodell beschreibt den vom Client und den vom Server erzeugten Einzelverkehr sowie dessen Abhängigkeit von der Anzahl aktiver Spieler und kann netzunabhängig eingesetzt werden. Es kann eine beliebige Aggregation von Ein-

zelverkehren bei einer einstellbaren Anzahl aktiver Spieler betrachtet werden. Die simulative Bewertung des Modells zeigt, wie es zur Untersuchung typischer Spiele-Verkehre, beispielsweise zur Dimensionierung des Netzzugangs eines Spieleservers eingesetzt werden kann.

Das Modell beschreibt nur den Paketverkehr von typischen Vertretern der schnellen und populären Action-Spiele, welche die höchsten Anforderungen an das Kommunikationsnetz stellen. Eine Ergänzung mit typischen Verkehrsmodellen für andere wichtige Spielearten mit anderen Kommunikationsbeziehungen und Anforderungen steht noch aus. Zudem ist auch dieser Dienst sehr dynamisch und wird durch ständig neu erschienene Produkte und zusätzliche Eigenschaften wie z.B. Sprachkommunikation geprägt.

Dienstspezifische Verkehrsmodellierung

Verkehrsmodelle einzelner Dienste werden zur Leistungsuntersuchung bei geringer Aggregation eingesetzt. Zur Systembeschreibung bei stark aggregiertem Verkehr sind abstraktere und effizientere Modelle besser geeignet. Die typischen Untersuchungen dienen daher der grundsätzlichen Bewertung von Diensteeigenschaften und spezifischen Kommunikationsmechanismen und der Bewertung oder Optimierung von Netzen bezüglich eines Dienstes oder eines Dienste-Bouquets. Insbesondere für die drahtlose Übertragung von multimedialen Inhalten auf mobile Endgeräte ist aufgrund der knappen Bandbreite eine detaillierte Leistungsuntersuchung mit entsprechend genauer Modellierung geboten. In der heutigen Zeit bereiten Mobilfunkanbieter ständig neuartige Dienste vor, um sich abzuheben und Mehrwerte vermarkten zu können, so daß der Bedarf einer entsprechenden Analyse dieser neuen Dienste auch zukünftig gegeben ist.

Eine Hauptanforderung für entsprechende Verkehrsmodelle dieser Dienste ist eine schnelle Realisierung, so daß sie entsprechend flexibel erweiterbar sein oder systematischer entwickelt werden müssen. Die Komplexität der interaktiven Dienste führt allerdings zu einer entsprechenden Komplexität bei der Modellbildung, die sowohl für die Modellierung selbst als auch für die Parameterisierung und Validierung gilt. Für eine automatisierte vereinfachte Parametrisierung schlagen Lan und Heidemann in [111] die direkte Verwendung von empirischen Verteilungsfunktionen vor. Solche Ansätze müssen für den weitergehenden Einsatz von dienstspezifischen Modellen konsequent weiterentwickelt werden, um sowohl den schnellen Produkt- und Forschungszyklen als auch der schnellen Entstehung neuer Dienste gerecht werden zu können.

Literaturverzeichnis

- [1] Abdouni Khayari, R.: Fitting world-wide web request traces with the EM-algorithm, *Performance Evaluation*, No. 51, pp. 175-191, 2003
- [2] Abdulla, Ghaleb: Analysis and Modeling of World Wide Web Traffic, *Dissertation*, Virginia Polytechnic Institute and State University, Mai 1998
- [3] Abry, P.; Veitch, D.: Wavelet analysis of long-range-dependent traffic, *IEEE Transactions on Information Theory*, 44(1), pp.2-15, January 1998
- [4] Anagnostou, M.E.; Sanchez-P., J.-A.; Venieris, I.S.: A Multiservice User Descriptive Traffic Source model, *IEEE Transactions on Communications*, Vol.44,No.10, pp. 1243-1246, October 1996
- [5] Adams, Andrew; Bu, Tian; Friedman, Timur; Horowitz, Joseph; Towsley, Don; Cáceres, Ramón; Duffield, Nick; Lo Presti, Francesco; Moon, Sue B.; Paxson, Vern: The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior, *IEEE Communications Magazine*, vol. 38, no. 5, May 2000, <http://www.comsoc.org/ci1/private/2000/may/Towsley.html>
- [6] Allman M., Paxson V., Stevens W.: TCP Congestion Control, *IETF, RFC 2581* April 1999
- [7] Allman, Mark: A Web Server's View of the Transport Layer, *ACM Computer Communication Review*, vol. 30, no. 4, Oct. 2000 <http://www.acm.org/sigcomm/ccr/archive/2000/oct00/allman.pdf>
- [8] Arlitt, Martin; Williamson, Carey L.: Web Server Workload Characterization: The Search for Invariants. *In Proceedings of SIGMETRICS*, pp. 126-137, April 1996
- [9] Arlitt, Martin; Friedrich, Rich; Jin, Tai: Workload Characterization of a Web Proxy in a Cable Modem Environment. *HP Labs Technical Report HPL-1999-48*, 990422, 1999, <http://www.hpl.hp.com/techreports/1999/HPL-1999-48.html>, <http://www.comsoc.org/~ni/public/2000/may/index.html>
- [10] Arlitt, Martin; Jin, Tai: A Workload Characterization Study of the 1998 World Cup Web Site," *IEEE Network*, Vol.14, No.3, May 2000.
- [11] Armitage, Grenville: Sensitivity of Quake3 Players To Network Latency, *IMW2001 workshop poster session*, Nov 2001, http://www.geocities.com/gj_armitage/q3/quake-results.html

- [12] Arnold, Thomas: Untersuchung der Eignung von PCs als IP-Netzverkehrsmeßgerät. Studienarbeit, *Studienarbeit*, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 2001
- [13] Baran, Paul: On Distributed Communications, vols I-XI, *RAND Corporation Research Documents*, August 1964,
<http://www.rand.org/publications/RM/baran.list.html>
- [14] Barford P., Crovella, M.E.: An architecture for a WWW Workload generator, *In Proceedings of World Wide Web Consortium Workshop on Workload Characterization*, Oktober 1997
- [15] Barford P., Crovella, M.E.: Generating Representative Web Workloads for network and Server Performance Evaluation, *In Proceedings of Performance '98/ACM SIGMETRICS '98*, pages 151-160, Madison, 1998
- [16] Barford P., Bestavros A., Bradley A. and Crovella M.E.: Changes in Web Client Access Patterns: Characteristics and Caching Implications. *In World Wide Web, Special Issue on Characterization and Performance Evaluation*, Vol. 2, pages 15-28, 1999
- [17] Barford, P.; Crovella, M.E.: Measuring Web Performance in the Wide Area. *ACM Perf. Eval. Review*, vol. 27, no. 2, pp. 37–48, 1999
- [18] Barford, P.: Modeling, Measurement And Performance Of World Wide Web Transactions, *dissertation*, 2001
- [19] Barford, Paul; Bestavros, Azer; Byers, John; Crovella, Mark: On the Marginal Utility of Network Topology Measurements," *in ACM SIGCOMM Internet Measurement Workshop*, (San Francisco, USA), Nov. 2001,
<http://www.acm.org/sigcomm/imw2001/imw2001-papers/55.pdf>
- [20] Bauer, Daniel; Rooney, Sean; Scotton, Paolo: „Network Infrastructure for Massively Distributed Games“, *Proceedings for Netgames 2002*, pp. 36-43, Braunschweig, Apr. 2002
- [21] Bettner, P.; Terrano, M.: 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond, *Game Developers Conference*, 2001,
http://www.gdconf.com/archives/proceedings/2001/prog_papers.html
- [22] Beyer; Hackel; Pieper; Tiedge: „Mathematik für Ingenieure und Naturwissenschaftler“, 8. Auflage, Teubner Stuttgart Leipzig, Bad Langensalza, 1999
- [23] Bernier, Y.W.: Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization, *Game Developers Conference 2001*,
http://www.gdconf.com/archives/proceedings/2001/prog_papers.html
- [24] Berners-Lee, T. ; Connolly, D.: Hypertext Markup Language 2.0, *IETF, RFC 1866*, 1995
- [25] Berners-Lee, T.; Fielding, R.; Frystyk, H.: Hypertext transfer protocol - HTTP/1.0, *IETF RFC 1945*, Okt 1995

- [26] Bodamer S., Charzinski J.: Evaluation of effective bandwidth schemes for self-similar-traffic, *In ITC Specialist Seminar on IP Traffic Measurement, Modelign and Management*, Monterey, 2000
- [27] Bonham S., Grossman D., Portnoy W., Tam K.: Quake: An Example Multi-User Network Application - Problems and Solutions in Distributed Interactive Simulations, *CSE 561 Term Project Report*, University of Washington, July 2000, <http://www.cs.washington.edu/homes/grossman/projects/561projects/quake/>
- [28] Borella, M.S.: Source Models of Network Game Traffic, *Networld + Interop'99 Engineer's Conference*, Mai 1999
- [29] Bosch, K.: Elementare Einführung in die angewandte Statistik, 6. Auflage, Vieweg,, Braunschweig, 1997
- [30] Box, G.E.; Jenkins, G.M.: Time series analysis: forecasting and control, McGraw-Hill, London, UK, 1970
- [31] Brakmo L., O'Malley S. , Peterson L. : TCP Vegas: New techniques for congestion detection and avoidance. *In Proceedings of SIGCOMM '94*, pp. 24-35, Aug. 1994
- [32] Braden, R.: Extending TCP for Transactions – Concepts, *IETF, RFC 1379*, Nov. 1992
- [33] Braden, R.: T/TCP - TCP Extensions for Transactions Functional Specification, *IETF, RFC 1644*, Jul. 1994
- [34] Bradner, S.: The Internet Standards Process - Revision 3, *IETF, RFC 2026*, Okt. 1996
- [35] Bray, T.: Measuring the Web, *Computer Networks and ISDN Systems*, vol. 28, no. 13, pp. 994–1005, Paris, France, Mai 1996, http://www5conf.inria.fr/fich_html/papers/P9/Overview.html
- [36] Breitbart, Y.; Chan, C.-Y.; Garofalakis, M.; Rastogi, R.; Silberschatz, A.: Efficiently Monitoring Bandwidth and Latency in IP Networks," *in Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Anchorage, Apr. 2001
- [37] Breslau, L.; Cao, P.; Fan, L.; Philipps, G.; Shenker, S.: Web Caching and Zipf-like Distributions: Evidence and Implications. *In Proceedings of INFOCOM 1999*, pp. 126-134, 1999
- [38] Broder, A. ; Kumar, R.; Maghoul, F. et al.: Graph structure in the web. *In Proc. of the 9th International World Wide Web Conference*, Amsterdam, Netherlands, May 2000
- [39] Bronstein I.N., Semendjajew K.A., Musiol G., Mühlig H.: Taschenbuch der Mathematik, Verlag Harri Deutsch, Frankfurt am Main, 1993
- [40] Cáceres, R.; Duffield, N.G.; Feldmann A. et al.: Measurement and analysis of IP network usage and behavior. *IEEE Comm. Mag*, vol. 38, no. 5, pp. 144–151, May 2000, <http://www.comsoc.org/ci1/private/2000/may/Duffield.html>

- [41] Cáceres, Ramon; Duffield, Nick; Friedman, Timur: Impromptu Measurement Infrastructures using RTP," *in Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, New York (USA), Jun. 2002, <http://www.ieee-infocom.org/2002/papers/924.pdf>
- [42] Casilari, E.; Reyes-Lecuona, A.; González, F.J.; Díaz-Estrella, A.; Sandoval, F.: Characterisation of Web Traffic, *IEEE GlobeCom 2001*, San Antonio, Texas (USA), Nov 2001
- [43] Cerf, Vint: Cerf's Up: Internet History, *Online article*, http://www.worldcom.com/global/resources/cerfs_up/internet_history/
- [44] Chang, Francis; Feng Wu-chang: Modeling Player Session Times of On-line Games. *In Proceedings of ACM NetGames 2003*, Redwood, 2003
- [45] Charzinski, J.: Measured HTTP Performance and Fun Factors. *In Proc. ITC*, pp. 1063–1074, Salvador, BA, Brasil, 2001
- [46] Charzinski, J.; Färber, J.; Vicari, N.: Verkehrsmessung und Lastmodellierung im Internet. *Fachzeitschrift Praxis der Informationsverarbeitung und Kommunikation (PIK)*, Vol. 25.2, April-Jun 2002, S. 64-72.
- [47] Chen, Thomas M.: Network Traffic Measurements and Experiments," *IEEE Communications Magazine*, vol. 38, no. 5, May 2000.
- [48] Choi, H.-K.; Limb, J.O.: A behavioral Model of Web Traffic, *Proc. of ICNP'99*, Toronto, Oct. 1 - Nov. 03, 1999, <http://www.cc.gatech.edu/fac/John.Limb/>
- [49] claffy, k.: CAIDA: Visualizing the Internet. *Internet Computing Online*, <http://computer.org/internet/v5n1/caida.htm>, 2001
- [50] Clark, D.D.: Window and Acknowledgement Strategy in TCP, *RFC 813*, July 1982
- [51] Claypool, Mark; LaPoint, David; Winslow, Josh: Network Analysis of Counter-strike and Starcraft, *In Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Phoenix, Arizona, USA, April 2003. <http://www.cs.wpi.edu/~claypool/papers/net-game/>
- [52] Cohen, E.; Kaplan, H.: Prefetching the Means for Document Transfer: A New Approach for Reducing Web Latency. *In Proc. IEEE Infocom*, Tel Aviv, Israel, 2000
- [53] Comer, Douglas E.; Stevens, David L.: *Internetworking with TCP/IP*, Vol. 2, 2. Ausgabe, Prentice Hall, 1994
- [54] Courcoubetis, Costas; Siris, Vasilios A.: „Procedures and tools for analysis of network traffic measurements“, *Performance Evaluation*, No. 48, pp. 5-23, 2002
- [55] McCreary, S.; claffy, k.: Trends in wide area IP traffic patterns - A view from Ames Internet Exchange, *ITC Spec. Seminar*, 2000, <http://www.caida.org/outreach/papers/AIX0005>

- [56] McCreary, S.; claffy, k.: Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. *In ITC Specialist Seminar on IP Traffic Modeling, Measurement and Management*, Monterey, CA, Sept. 2000,
<http://www.caida.org/outreach/papers/AIX0005>
- [57] Cunha, C.; Bestavros, A.; Crovella M.: Characteristics of WWW client-based traces. *Technical Report TR-95-010*, Boston University Computer Science Dep., June 1995
- [58] Cox, D.R.; Miller, H.D.: *The Theory of Stochastic Processes*, Chapman and Hall, London, 1965
- [59] Crovella, M.E.; Carter, R.L.. Dynamic Server Selection in the Internet. *In Proc. IEEE Workshop Arch. and Impl. of High Perf. Comm. Subsystem. HPCS*, Mystic, CT, USA, pp. 158–162, Aug. 1995
- [60] Crovella, M.E.; Bestavros, A.: Explaining World Wide Web Traffic Self-Similarity, *Technical Report TR-95-015*, Computer Science Department, Boston University, 1995
- [61] Crovella, M.E.; Bestavros A.: Self-similarity in World Wide Web traffic: Evidence and possible causes. *Proc. of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pp.160-169, Philadelphia, May 1996
- [62] Deng, Shuang: Empirical Model of WWW Document Arrivals at Access Link, *In Proc. of International Computer Conference (ICC)*, pp. 1797-1802, September 1996
- [63] Douglis, F.;Feldmann, A.; Krishnamurthy, B.; Mogul, J.C.: Rate of Change and other Metrics: a Live Study of the World Wide Web. *In Proc. Usenix Symposium on Internet Technologies and Systems*, Monterey, CA, USA, pp. 147–158, 1997
- [64] Downey, A.B.: Evidence for long-tailed distributions in the Internet," *in ACM SIGCOMM Internet Measurement Workshop*, (San Francisco, USA), Nov. 2001,
<http://www.acm.org/sigcomm/imw2001/imw2001-papers/35.pdf>
- [65] Downey, A.B.: The structural cause of file size distributions, *In Proceedings of MAS-COTS 2001*, 2001
- [66] Elek, Viktoria; Karlsson, Gunnar; Ronngren, Robert: Admission Control Based on End-to-End Measurements, *In Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), Mar. 2000,
<http://www.ieee-infocom.org/2000/papers/239.ps>
- [67] Estan, Christian; Varghese, George: New Directions in Traffic Measurement and Accounting, *In SIGCOMM Symposium on Communications Architectures and Protocols*, (Pittsburgh,PA), pp. 14, Aug. 2002,
<http://www.acm.org/sigcomm/sigcomm2002/papers/traffmeas.pdf>
- [68] Eun, Do Young; Shroff, Ness: A Measurement-Analytic Framework for QoS Estimation Based on the Dominant Time Scale, *in Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Anchorage, Alaska), Apr. 2001

- [69] Färber, J.; Bodamer, S.; Charzinski, J.: Evaluation of dial-up behaviour of Internet users, *ITG Fachtagung*, Stuttgart, Oktober 1998
- [70] Färber, J.; Bodamer, S.; Charzinski, J.: Statistical evaluation and modelling of Internet dial-up traffic, *SPIE Photonics East'99*, Boston, September 1999
- [71] Färber, J.: Network Game Traffic Modelling. *In Proceedings of Netgames 2002*, pp.53-57, Braunschweig, April 2002
- [72] Färber, J.: Traffic Modelling for Fast Action Network Games, *to appear in Multimedia Tools and Applications*, 2003
- [73] Fall, Kevin, Floyd, Sally: Simulation-based comparisons of Tahoe, Reno and SACK TCP, *ACM SIGCOMM Computer Communication Review*, vol. 26 , no. 3, pp. 5-21, July 1996
- [74] Feldmann, A.; Gilbert, A.C.; Willinger, W.: Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. *In Proceedings of ACM SIGCOMM'98*, pp. 42–55, Vancouver, B.C., Canada, 1998
- [75] Feldmann, A.; Gilbert, A.; Huang, P.; Willinger, W.: Dynamics of IP traffic: A study of the role of variability and the impact of control. *In Proceedings of ACM SIGCOMM'99*, pp.251-262, Aug. 1999
- [76] Feldmann, A.; Whitt, W.: Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models, *Performance Evaluation*, No. 31(3), pp. 245-279, 1998
- [77] Feng, Wu-chang; Chang, Francis; Feng, Wu-chi; Walpole, Jonathan: Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server. *In Proceedings of the Internet Measurement Workshop 2002*, November 2002, <http://www.icir.org/vern/imw-2002/proceedings.html>
- [78] Fiedler, Stefan; Wallner, Michael; Weber, Michael: A Communication Architecture for Massive Multiplayer Games, *In Proceedings for Netgames 2002*, pp. 14-22, Braunschweig, April 2002
- [79] Fielding, R.; Irvine, U.C.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P.; Berners-Lee, T.: Hypertext Transfer Protocol -- HTTP/1.1, *IETF, RFC 2616 Draft (Work in Progress)*, June 1999
- [80] Floyd, S.; Henderson, Tom: The NewReno Modification to TCP's Fast Recovery Algorithm, *IETF, RFC 2582*, April 1999
- [81] Floyd, S.; Mahdavi, J.; Mathis, M.; Podolsky, M.: An Extension to the Selective Acknowledgement (SACK) Option for TCP, *IETF, RFC 2883*, July 2000
- [82] Fraleigh, C.; Moon, S.; Lyles, B.; Cotton, C.; Khan, M.; Moll, D.; Rockell, R.; Seely, T.; Diot, C.: Packet-Level Traffic Measurements from the Sprint IP Backbone, *IEEE Network*, vol. 17, no. 6, pp. 6-16, 2003

- [83] McGregor, A.J., Pearson, M.W.; Cleary, J.C.: Improving the performance of HTTP over high bandwidth-delay product circuits. *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) '01*, Phoenix, Arizona, pp. 97-102, Jan. 2001 <http://moat.nlanr.net/Papers/>
- [84] Guérin, R.: Equivalent Capacity and Its Application to Bandwidth Allocation in high-Speed Networks, *IEEE Journal of Selected Areas in Communications*, Vol. 9, No. 7, September 1991
- [85] Habib, A.; Abrams, M.: Analysis of Sources of Latency in Downloading Web Pages. *In Proc. Webnet*, San Antonio, TX, USA, 2000
- [86] Hafner, Katie; Lyon, Matthew: Where Wizards Stay Up Late: The Origins of the Internet. Simon and Schuster, New York, 1996, p.64
- [87] Hassan, Mahbub; Wu, Jim; Breen, James: "GPS-Based Infrastructure for Internet Performance Measurement", *technical report*, Monash University, Melbourne, Australia, Jun. 2000, <http://www.cse.unsw.edu.au/~jimw/publications/tr.pdf>
- [88] Hassan, Mahbub; Wu, Jim: APM: Asynchronous Performance Measurement for the Internet", *in the 6th Asia-Pacific Conference on Communications (APCC2000)*, (Seoul, South Korea), pp. 1335--1339, Nov. 2000, <http://www.cse.unsw.edu.au/~jimw/publications/apcc2000.pdf>
- [89] Heidemann, J.; Obraczka, K.; Touch, J.: Modeling the Performance of HTTP Over Several Transport Protocols, *ACM/IEEE Transactions on Networking*, Vol. 5, pp. 616--630, October 1997
- [90] Henderson, T.; Bhatti, S.: Modelling user behaviour in networked games, *Proceedings of the 9th ACM international conference on Multimedia*, pp.212 - 220, 2001, Ottawa, Canada
<http://doi.acm.org/10.1145/500141.500175>
- [91] Henderson, T.: Latency and User Behaviour on a Multiplayer Game Server. *Proc. of Third International COST264 Workshop Networked Group Communication*, 2001: pp. 1-13, London, UK, November 7-9, 2001
<http://link.springer.de/link/service/series/0558/papers/2233/22330001.pdf>
- [92] Henderson, T., Persönliche Kommunikation, Netgames 2002, Braunschweig, April 2002
- [93] Henderson, T.: The effects of relative delay in networked games, *Dissertationsschrift*, Department of Computer Science, University College London, Februar 2003
- [94] Henderson, T.: Networked games - a QoS-sensitive application for QoS-insensitive users?, *In Proceedings of ACM SIGCOMM 2003 Workshops*, pp. 141-147, Karlsruhe, 2003

- [95] Hessenmüller, H.; Achtmann, K.; Charzinski, J.; Frank, M.; Färber, J.; Lehmann, E.; Paul, S.; Quecke, U.; Tölle, J.; Zirwas, W.: Das europäische Projekt AMUSE - Der HFR Feldversuch in München, *Der Fernmelde Ingenieur*, Vol. 7/8, August 1999
- [96] Hlavacs, H.; Kotsis, G.: Modeling User Behavior: A Layered Approach, *In Proceedings of the Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS'99*, 1999
- [97] Hubbard, B.B.: Wavelets: Die Mathematik der kleinen Wellen, Aus dem Amerikan. von Michael Basler, Birkhäuser, Berlin, 1997
- [98] Huitema, C.; Weerahandi, S.: Internet Measurements: the Rising Tide and the DNS Snag. *In Proc. ITC Spec. Sem. on IP Traffic*, pp. 2.1–2.10, Monterey, CA, USA, 2000
- [99] ITG-Empfehlung 5.2-03: Begriffe der Nachrichtenverkehrstheorie, *Begriffswerk der ITG-Fachgruppe 5.2.1, Informationstechnische Gesellschaft*, Okt. 1997, http://www.comnets.rwth-aachen.de/~itg/FG521/home_ger.html
- [100] ITU-T Recommendation E.800: Terms and Definitions Related to Quality of Service and Network Performance including Dependability, (08/1994), International Telecommunication Union, Genf, März 1994
- [101] ITU-T Recommendation I.211: B-ISDN service aspects, (03/1993), International Telecommunication Union, Genf, März 1993
- [102] ITU-T Recommendation I.350: General Aspects of Quality of Service and Network Performance in Digital Networks, including ISDNs, (03/1993), International Telecommunication Union, Genf, März 1993
- [103] Jacobson V.: Congestion Avoidance and Control, *Computer Communication Review*, vol. 18, no. 4, pp. 314-329, Aug. 1988. <ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>
- [104] Jacobson V.: Modified TCP Congestion Avoidance Algorithm, *Technical Report (end2end-interest mailing list)*, April 30, 1990, <ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail> or <ftp://ftp.ee.lbl.gov/email/naj.90apr30.txt>.
- [105] Joyce, S.: Traffic on the Internet, *Report*, 2000, <http://citeseer.ist.psu.edu/joyce00traffic.html>
- [106] Khaunte, Sunil U.; Limb, John O.: Statistical Characterization of a World Wide Web Browsing Session, *Technical Report GA 30332-0280*, Georgia Institute of Technology, 1997
- [107] Kreyszig, Erwin: „Statistische Methoden und ihre Anwendungen“, Vandenhoeck & Ruprecht, Köthen 1965
- [108] Krishnamurthy, Balachander; Wills, Craig E. : Analyzing factors that influence end-to-end Web performance, *Computer Networks*, vol. 33, no. 1-6, pp. 17-32, Jun. 2000, <http://www.elsevier.com/locate/comnet>
- [109] Kühn, P.J.: Teletraffic Theory and Engineering, Vorlesungsskript Ausgabe 2003/2004

- [110] Labovitz, C.; Malan, G.R.; Jahanian, F.: Internet Routing Instability. *In Proc. ACM Sigcomm*, pp. 115–126, Cannes, France, 1997
- [111] Lan, Kun-chan; Heidemann, John: Rapid Model Parameterization from Traffic Measurements, *Technical Report*, USC Information Sciences Institute (ISI), 2002, <http://www.isi.edu/isi-technical-reports.html>
- [112] Le, Long; Aikat, Jay; Jeffay, Kevin; Smith, F. Donelson: The Effects of Active Queue Management on Web Performance, *In Proceedings SIGCOMM 2003*, pp.265-276, Karlsruhe, 2003
- [113] Leland W.; Taqqu M.; Willinger W.; Wilson D.: On the Self-Similar Nature of Ethernet Traffic (Extended Version), *IEEE/ACM Transactions on Networking*, No. 2, Vol. 1, pp. 1-15, 1994
- [114] Liu, Zhen; Niclausse, Nicolas; Jalpa-Villanueva, César: Traffic model and performance evaluation of Web servers, *Performance Evaluation*, vol. 46, no. 2-3, pp. 77-100, Oct. 2001
- [115] Liu, Zhen; Niclausse, Nicolas; Jalpa-Villanueva, César :WAGON: A Web Server Benchmarking Tool, Research Note, <http://www.sop.inria.fr/mistral/personnel/Zhen.Liu/wagon.html>
- [116] Lorang, Martin: Verbindungsorientierte und verbindungslose Mechanismen eines skalierbaren Verkehrsmanagements für IP-Netze, *Dissertation*, Institut für Kommunikation-snetze und Rechnersysteme, Uni Stuttgart, 2005
- [117] Luckie, Matthew J.; McGregor, Anthony J.; Braun, Hans-Werner: Towards Improving Packet Probing Techniques, *in ACM SIGCOMM Internet Measurement Workshop*, (San Francisco, USA), Nov. 2001, <http://www.acm.org/sigcomm/imw2001/imw2001-papers/85.pdf>
- [118] Mah, Bruce: An Empirical Model of HTTP Network Traffic, *Proc. of Infocom'97*, pp. 592-600, Kobe, 1997
- [119] Marsh, Ian; Hagsand, Olof; Andersson, Anders; Ahlgren, Bengt: Dimensioning Links for IP Telephony, *in Internet Telephony Workshop 2001*, (New York), Apr. 2001, <http://www.cs.columbia.edu/~hgs/papers/iptel2001/14.ps>
- [120] Mathis, M.; Mahdavi, J.; Floyd, S.; Romanow, A.: TCP Selective Acknowledgement Options, *IETF RFC 2018*, Oct 1996
- [121] Mathis, M.; Semke, J.; Mahdavi, J.; Ott, T.: The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *ACM Comp. Comm. Rev.*, vol. 27, no. 3, pp. 67–82, 1997
- [122] Molina, Maurizio; Castelli, Paolo; Foddis, Gianluca: Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE, *IEEE Network*, vol. 14, no. 3, May 2000, <http://www.comsoc.org/~ni/public/2000/may/index.html>

- [123] Mase, K.; Toyama, Y.: End-to-end measurement based admission control for VoIP networks, in *Conference Record of the International Conference on Communications (ICC)*, (New York, NY, USA), pp. 1194-1198, Apr. 2002
- [124] Micheel, Jörg; Donnelly, Stephen; Graham, Ian: Precision Timestamping of Network Packets, in *ACM SIGCOMM Internet Measurement Workshop*, (San Francisco, USA), Nov. 2001, <http://www.acm.org/sigcomm/imw2001/imw2001-papers/23.pdf>
- [125] Murata, M.: On a Network Dimensioning Approach for the Internet, *IEICE Transactions on Communications*, Vol. E85-B, No. 1, pp. 3-13, January 2002
- [126] Mogul, J.C. : Observing TCP Dynamics in Real Networks. *ACM Comp. Comm. Rev.*, vol. 22, no. 4, pp. 305–317, 1992
- [127] Moon, Sue: Internet Traffic Measurement, *KRNET 2001 Tutorial*, Seoul, Korea, June 2001
- [128] Morris, Robert; Lin, Dong: Variance of Aggregated Web Traffic, *In Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), Mar. 2000, <http://www.ieee-infocom.org/2000/papers/78.ps>
- [129] Nagle, J.: Congestion Control in TCP/IP Internetworks, *Comp. Comm. Review*, vol. 14, pp. 11-17, Oct 1984
- [130] Ng, Yu-Sheng: Internet Game Design, *Gamasutra Online Magazine*, 1997, <http://www.gamasutra.com/features/19970801/ng.htm>
- [131] Ng, Yu-Sheng: Designing Fast-Action Games for the Internet, *Gamasutra Online Magazine*, May 1997, http://www.gamasutra.com/features/19970905/ng_01.htm
- [132] Padmanabhan, V.N.; Mogul, J.C.: Using Predictive Prefetching to Improve World Wide Web Latency. *ACM Comp. Comm. Rev.*, vol. 26, no. 3, pp. 22–36, 1996
- [133] Paxson, Vern; Floyd, Sally: „Wide-Area Traffic: The Failure of Poisson Modeling“, *Proceedings of ACM SIGCOMM'94*, pp. 257-268, 1994
- [134] Paxson, Vern; Floyd, Sally: Why We Don't Know How To Simulate The Internet. *In Proceedings of 1997 Winter Simulation Conference*, pp. 1037 -1044, updated version, 1999, <http://www.icir.org/floyd/papers/wsc.ps>
- [135] Paxson, Vern; Floyd, Sally: Difficulties In Simulating The Internet. *IEEE/ACM Transactions on Networking*, Vol.9, No.4, pp.392-403, 2001, http://www.icir.org/floyd/papers/simulate_2001.ps
- [136] Reyes-Lecuona, A.; González-Parada, E.; Casilari, E.; Casasola, J.C.; Díaz-Estrella, A.: A page-oriented WWW traffic model for wireless system simulations, *In Proceedings of ITC 16*, pp. 1271-1280, 1999
- [137] Reynolds, J.; Poste, J.: Assigned Numbers, *IETF, RFC 1700*, Okt. 1994

- [138] Riska, A.: Aggregate Matrix Analytic Techniques and their Applicability in Computer Systems Performance Analysis, dissertation, Faculty of the Department of Computer Science The College of William & Mary in Virginia, März 2000, <http://www.cs.wm.edu/~riska/main/main.html>
- [139] Riska, Alma; Diev, Vesselin; Smirni, Evgenia: Efficient fitting of long-tailed data sets into Phase-type distributions, *Performance Evaluation Review*, Vol.30, No.3, pp. 6-8, Dezember 2002
- [140] Roberts, Lawrence G.: The Evolution of Packet Switching, *Proceedings of the IEEE*, Vol. 66, No. 1, November 1978, <http://www.packet.cc/files/ev-packet-sw.html>
- [141] Roughan, Matthew; Veitch, Darryl; Abry, Patrice: Real-time estimation of the parameters of long-range dependence, *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 467--478, August 2000
- [142] Rozalén, Federico Lozano: User Traffic Characterization of Internet services, *Technical Document*, COST 257 TD(98)44, 1998
- [143] Sass, D.; Junghans, S.: I2MP - A high precision measurement platform for IP traffic, 4. Würzburger ITG-Workshop IP Netzmanagement, Netzplanung und Optimierung, 27.-28. Juli 2004, <http://www3.informatik.uni-wuerzburg.de/ITG/2004/>
- [144] Sheldon, Nathan; Girard, Eric; Borg, Seth; Claypool, Mark; Agu, Emmanuel: The Effect of Latency on User Performance in Warcraft III, *In Proceedings of ACM NetGames*, Redwood City, California, USA, Mai 2003, <http://www.cs.wpi.edu/~claypool/papers/net-game/>
- [145] Shi, Weisong; Wright, Randy; Collins, Eli; Karamcheti, Vijay: Workload Characterization of a Personalized Web Site - And Its Implications for Dynamic Content Caching, *Technical Report*, TR2002-829, New York University, 2002
- [146] Smith, F. D.; Campos, F. H.; Jeffay, K.; Ott, D.: What TCP/IP Protocol Headers Can Tell Us About the Web. *Proc. of ACM SIGMETRICS 2001 / Performance 2001*, Cambridge, Juni 2001
- [147] Stevens, W. Richard: TCP/IP Illustrated, Vol. 1, Addison-Wesley, 1994
- [148] Tanenbaum, Andrew S.: Computer Networks, 3. Ausgabe, Prentice Hall, 1996
- [149] Trier, M.: Highspeed-Internet, *GameStar* (Spielezeitschrift), pp.164-165, März 2002
- [150] Vicari, Norbert: Measurement and modeling of WWW sessions: a comparison of Pareto and logarithmic histogram models. *In Proc. 5th Polish Teletraffic Symposium*, Warsaw, Poland, April 1998
- [151] Wallerich, Jörg: Design and Implementation of a WWW Workload Generator for the NS-2 Network Simulator, *Diplomarbeit*, Universität Saarbrücken, Fachbereich Informatik, August 2001, <http://www.net.informatik.tu-muenchen.de/~jw/nsweb/>

- [152] Wang, Zhe; Cao, Pei: Persistent Connection Behavior of Popular Browsers, Research Note, Dep. of Computer Science, University of Wisconsin, 1998, <http://www.cs.wisc.edu/~cao/papers/persistent-connection.html>
- [153] Weber, Hubert: “Einführung in die Wahrscheinlichkeitsrechnung und Statistik für Ingenieure“, 3. Auflage, Teubner Stuttgart, Karlsruhe, 1992
- [154] Wijata, Yulia Indrayani; Niehaus, Douglas; Frost, Victor S.: A Scalable Agent-Based Network Measurement Infrastructure, *IEEE Communications Magazine*, vol. 38, no. 9, Sep. 2000, <http://www.comsoc.org/livepubs/ci1/public/2000/sep/index.html>
- [155] Williamson, Carey: Internet Traffic Measurement, *IEEE Internet Computing*, Vol. 5, No. 6, pp. 70-74, Dez. 2001, <http://pages.cpsc.ucalgary.ca/%7Eccarey/papers/measurements.pdf>
- [156] Willinger, Walter; Taqqu, Murad S.; Leland, Will E.; Wilson, Daniel V.: Self-similarity through high variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, Vol.5, No.1, pp.71-86, Feb. 1997
- [157] Wills, Craig E.; Trott, Gregory; Mikhailov, Mikhail: Using bundles for Web content delivery, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol.42, No.6, pp. 797-817, August 2003
- [158] Wolman, Alec; Voelker, Geoffrey M.; Sharma, Nitin; Cardwell, Nean; Karlin, Anna; Levy, Henry M.: On the scale and performance of cooperative Web proxy caching. In *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles*, Kiawah Island, SC, Dez. 1999, <http://research.microsoft.com/users/alecw/sosp-1999.pdf>
- [159] Yajnik, M.; Moon, S.; Kurose, J.; Towsley, D.: Measurement and modelling of the temporal dependence in packet loss, *Proceedings of IEEE INFOCOM 99*, pp. 345-352, 1999, ftp://gaia.cs.umass.edu/pub/Yajn98_Meas-TR-98-78.ps
- [160] Zhang, Jinsong: Estimation of WWW User Behaviour From IP Traffic, *Master Thesis*, Universität Stuttgart, Februar 2003
- [161] Zhang, Li; Liu, Zhen; Xia, Cathy: Clock Synchronization Algorithms for Network Measurements, in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (New York, New York), Juni 2002, <http://www.ieee-infocom.org/2002/papers/329.pdf>

Quellen im Internet

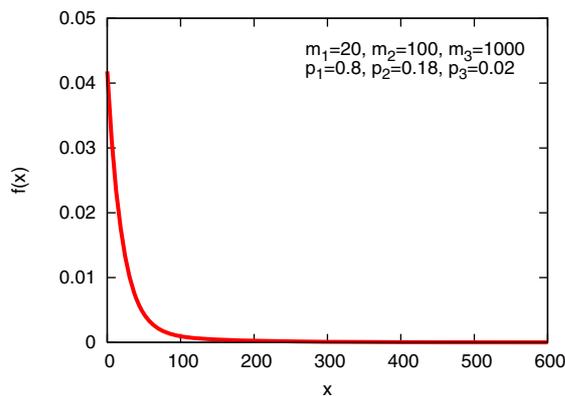
- [162] AI Interface Standards Committee, <http://www.igda.org/committees/ai.php>
- [163] Battle Foundry, Internet Service Provider (ISP) designed to support the online gaming, <http://www.battlefoundry.net/>
- [164] Butterfly Net, <http://www.butterfly.net>
see White Paper „Butterfly.net: Powering Next-Generation Gaming with Computing On-Demand“, IBM/IDC
- [165] Blizzard Entertainment, Spieleentwickler
 - a) Battlenet, <http://www.battle.net/>
 - a) Homepage, <http://www.blizzard.com>
 - b) WarCraft III, <http://www.blizzard.com/war3/>
- [166] CAIDA Homepage, Cooperative Association for Internet Data Analysis, Homepage: <http://www.caida.org/>
- [167] Ethereal Homepage, <http://www.ethereal.org>
- [168] Game Research, <http://www.game-research.com>
 - a) „The 6 myths of computer gaming“, Januar 2003, http://www.game-research.com/art_myths_of_gaming.asp
 - b) „Online gaming habits“, April 2002, http://www.game-research.com/art_online_gaming.asp
- [169] Williams, T.; Kelley C.: gnuplot: An Interactive Plotting Program, Version 3.7, Documentation, <http://www.gnuplot.info/>
- [170] Havok Homepage, Anbieter einer Physik-Engine für Spiele-Entwickler, <http://www.havok.com/>
- [171] ID Software (Quake, Doom), <http://www.idsoftware.com>
- [172] IKR Simlib, <http://www.ikr.uni-stuttgart.de/IKRSimLib/>
- [173] Internet Engineering Task Force (IETF), <http://www.ietf.org>
- [174] Internet Traffic Archive (ITA), <http://ita.ee.lbl.gov>
- [175] Internet Software Consortium (ISC): Internet Domain Survey, <http://www.isc.org/ds>
- [176] NLANR, National Laboratory for Applied Network Research Homepage, <http://www.nlanr.net/>
- [177] Netcraft: Web Server Survey, <http://www.netcraft.com/Survey/>
- [178] Network Simulator NS
 - a) Homepage <http://www.isi.edu/nsnam/ns/>

- b) Module für NS (u.a. WWW-Generatoren) <http://www.isi.edu/nsnam/ns/ns-contributed.html>
- [179] NetworkWorlFusion: The 30-second rule, online article, July 2002,
<http://www.nwfusion.com/columnists/2002/0722chen.html>
- [180] NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>, 2005
- [181] Novalogic, Spieleentwickler, <http://www.novalogic.com/>
- [182] Skotos Net Artikel zu Statistiken der Spieleindustrie
http://www.skotos.net/articles/BTH_04.html
- [183] Slashdot, Disussions-Forum, <http://www.slashdot.org>
a) „How Fast Too Slow? A Study Of Quake Pings“, Diskussion, May 2001,
<http://slashdot.org/article.pl?sid=01/05/24/2044233>
- [184] Speakeasy, ISP claiming to support gamers, <http://www.speakeasy.net/>
- [185] Tcpdump Homepage, <http://www.tcpdump.org>
- [186] Tom's Hardware Guide
a) "Breitband-Killer: Online-Spiele im Fokus", 8. November 2002,
http://business.tomshardware.de/facts/20021108/online_gaming-07.html
- [187] Epic Homepage (Unreal), <http://www.epicgames.com>
- [188] World Wide Web Consortium, W3ORG, <http://www.w3.org>
- [189] Zona research report: The Need for Speed II, *Zona Market Bulletin*, No. 5, April 2001,
http://www.avoka.com/resources/keynote/Zona_Need_For_Speed.pdf

Anhang A

Ausgewählte Verteilungen zur Approximation von Modellparametern

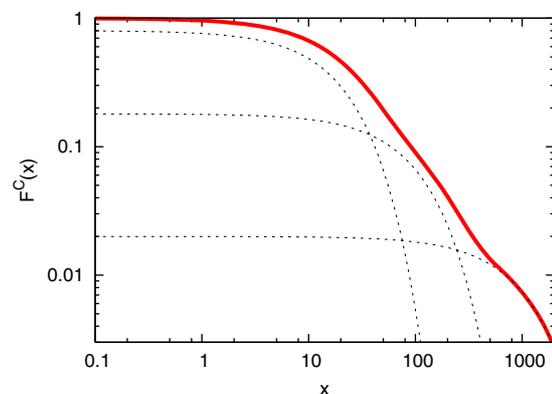
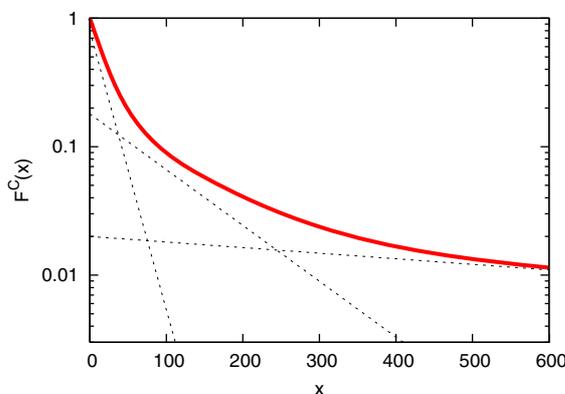
A.1 Hyperexponentielle Funktion n-ter Ordnung: $Hn(p_i, m_i)$



für $n=3$

$$f(x) = \frac{p_1}{m_1} e^{-\frac{x}{m_1}} + \frac{p_2}{m_2} e^{-\frac{x}{m_2}} + \frac{p_3}{m_3} e^{-\frac{x}{m_3}}$$
$$F^C(x) = p_1 e^{-\frac{x}{m_1}} + p_2 e^{-\frac{x}{m_2}} + p_3 e^{-\frac{x}{m_3}}$$

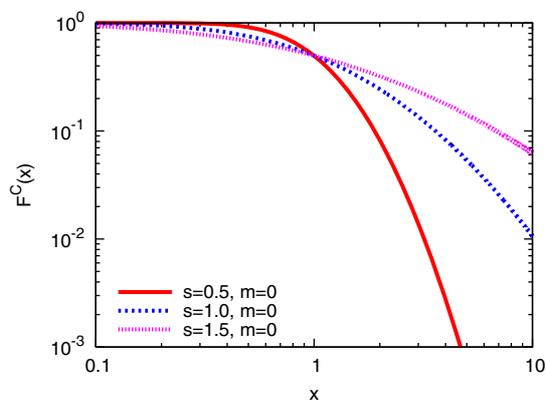
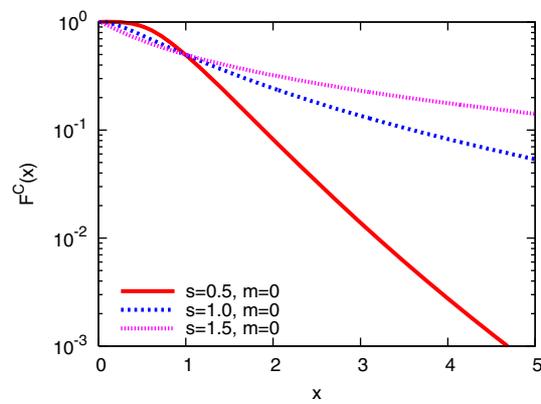
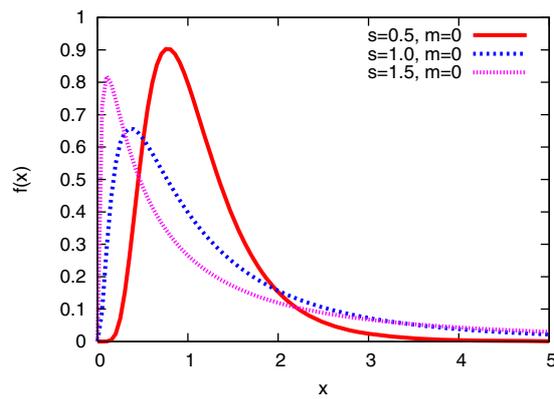
mit $\sum_i p_i = 1$



A.2 Lognormal-Verteilung: $L(\mu, \sigma)$

$$f(x) = \frac{1}{\sqrt{2\pi} \sigma x} e^{-\frac{(\ln x - \mu)^2}{2 \sigma^2}}, \alpha > 0$$

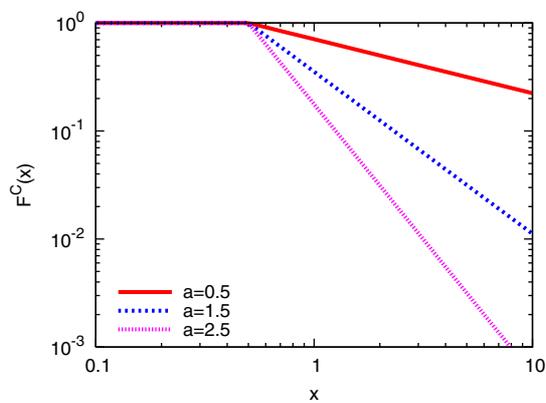
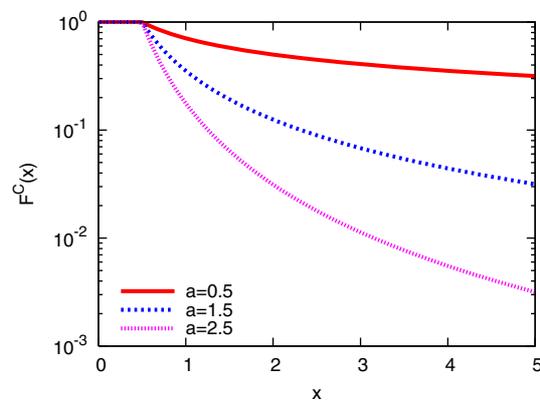
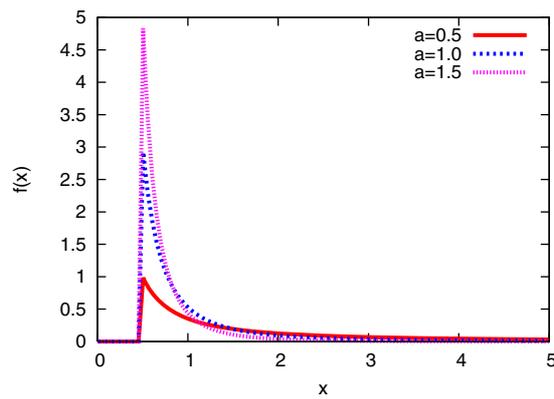
$F^C(x)$ (keine geschlossene Form)



A.3 Pareto-Verteilung: $P(\alpha, k)$

$$f(x) = \frac{\alpha k^\alpha}{x^{\alpha+1}}, \quad x > k$$

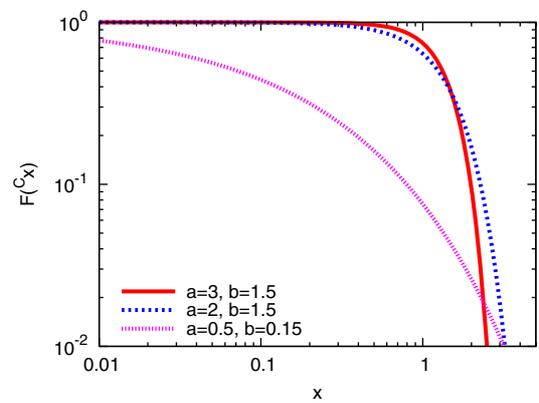
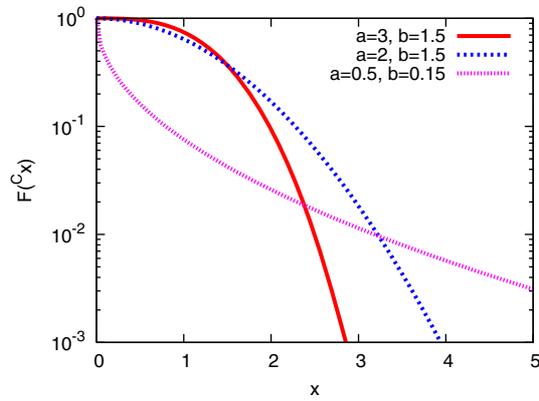
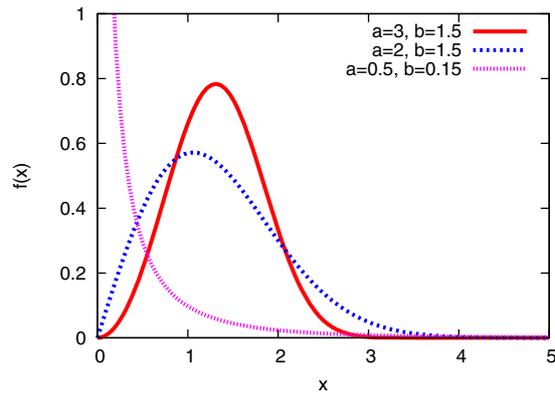
$$F^C(x) = \left(\frac{k}{x}\right)^\alpha, \quad x > k$$



A.4 Weibull-Verteilung $W(\alpha, \beta)$

$$f(x) = \alpha \beta^{-\alpha} x^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^\alpha}, \alpha, \beta > 0$$

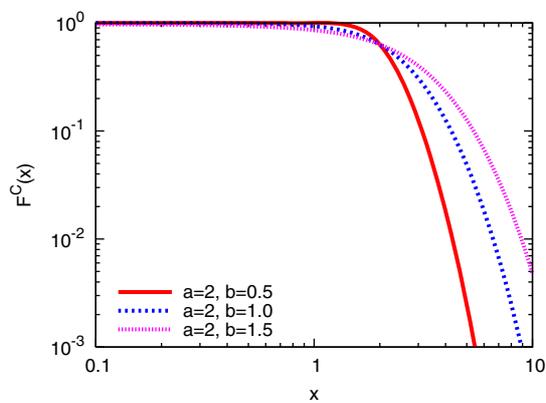
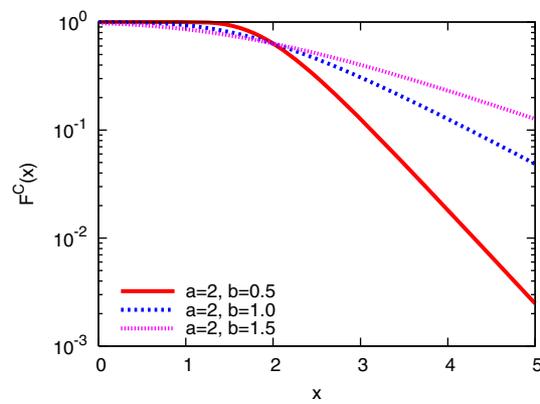
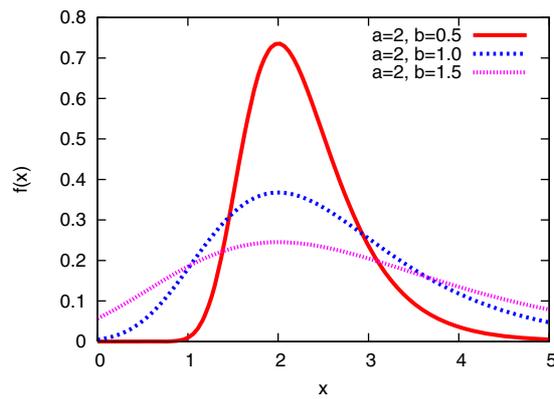
$$F^C(x) = e^{-\left(\frac{x}{\beta}\right)^\alpha}, \alpha, \beta > 0$$



A.5 Extremwert-Funktion (a,b)

$$f(x) = \frac{1}{b} e^{-\frac{x-a}{b}} e^{-e^{-\frac{x-a}{b}}}, \quad b > 0$$

$$F^C(x) = e^{-e^{-\frac{x-a}{b}}}$$



Anhang B

Parameter der Verkehrsmodelle

B.1 WWW-Verkehrsmodell

Tabelle B.1: Default-Parameter der WWW-Verkehrsquellen

```
WebApplication{
  WebServerDelayDist {
    //Constant { Mean=0.25; }
    //Uniform { LowerBound=0.0; UpperBound=0.5; }
    //NegExp { Mean=0.25; }
    Pareto { Alpha=1.14; MinValue=0.03; }
  }
  SenderNetworkAccess{
    BufferSize = 50000;      // Bytes
    //ServiceRate = 3600;   // in Bytes/s = 28.8 kbit/s
    //ServiceRate = 7200;   // in Bytes/s = 57.6 kbit/s
    //ServiceRate = 16000;  // in Bytes/s = 128 kbit/s
    //ServiceRate = 96000;  // in Bytes/s = 768 kbit/s
    ServiceRate = 125000;   // in Bytes/s = 1 Mbit/s
  }
  ReceiverNetworkAccess{
    BufferSize = 50000;     // Bytes
    ServiceRate = 125000;  // in Bytes/s = 1 Mbit/s
  }
  MaxNoOfParallelCons = 4;
  PropagationDelay = 0.400;
}

HFWebWorkLoadGenerator{
  NoOfInlinesDist{...}
  ReqSizeDist{...}
  RspSizeDist{...}
  PatienceTimeDist{...}
  UserThinkTimeDist{...}
}

SURGEWebWorkLoadGenerator{
  NoOfInlinesDist{...} //wie HF
  ReqSizeDist{...} //wie HF
  RspSizeDist{...} //wie HF
  RequestIATDist{...}
  InlineIATDist{...}
}
```

Tabelle B.2: Default-Parameter des Netzmodells

<pre>Link { BufferSize = 100000; ServiceRate = 125000; } Dropper { DropProb=0.01; } NoOfWebApplications=1;</pre>
--

Tabelle B.3: Verwendete Parameter der TCP-Module in den Simulationsszenarien

<pre>TCPSENDER{ Version = "Reno"; MSS = 1460; TCPIPHHeaderLength = 40; InitialCwnd = 2920; InitialSSThresh = 46720; InitialRTT = 1; MaxCwnd = 46720; DupACKThreshold = 3; TimerGranularity = 0.200; SWSATimerValue = 0.200; MaxRTO = 10000; MaxBackoff = 256; Nagle = false; GoBackN = true; Karn = true; FlightSizeRecovery = false; RenoConservation = true; ImmediateBackoffReset = false; }</pre>	<pre>TCPRECEIVER{ MSS = 1460; TCPIPHHeaderLength = 40; BufferSize = 65536; DelayedACK = true; ACKDelayTime = 0.200; SendPeriodicACKs = false; PeriodicACKInterval = 1.000; ACKSchedulingDelay = 0; MinUserBlockSize = 1; MaxUserBlockSize = 65536; UserBlockProcessingDelay = 0; } TCPConnection{ KeepAlive = true; }</pre>
---	--

B.2 Netzspiele-Verkehrsmodell

Tabelle B.4: Approximation der Sitzungs-Zwischenankunftszeit-Verteilungen am Server mit hyperexponentiellen Verteilungen 2. Ordnung für jede Stunde des Tages

Zeit	m_1	p_1	m_2
00	99.872147768163	0.969126648533833	237262.802188105
01	119.212157968451	0.962638069273394	237234.194564904
02	131.327965465055	0.958664022688866	237218.045379742
03	141.067237804657	0.948919276793285	66114.6023040209
04	181.234461058337	0.944815173417393	190427.837643722
05	187.770402947928	0.939865064200578	190394.44104428
06	194.36529430935	0.927971064488251	91875.761951157
07	193.512544374678	0.927053996975476	91874.7722545958
08	199.497480429933	0.919177993128325	65025.2880233841
09	150.299521534979	0.936750272414856	67888.6288436677
10	133.184888636878	0.937231402292692	36447.1491361314
11	134.023341528425	0.948238373138985	91071.8065975532
12	107.934472030865	0.955053754542815	40164.2251390333
13	103.050695152122	0.961307257186895	52395.0334678466
14	104.404439333802	0.965225775830976	81601.0820557306
15	82.7568862568868	0.972915435405993	81633.0023938495
16	81.8883233897253	0.973058972807051	81634.089862748
17	81.8880157231989	0.973060953004438	81634.089862759
18	73.1003911435356	0.974504629038285	64826.3169522715
19	75.0887418150479	0.974894965158346	64829.4142387466
20	72.0860417313939	0.977130986257368	173508.030480407
21	67.0581631383333	0.978044635969065	173495.471425679
22	70.0131073493206	0.976083928178297	173495.462147247
23	78.4535001565843	0.975049625432424	173495.461869204

Anhang C

Approximation von kombinierten Verteilungen

Für die Approximation der kombinierten Verteilungen, die in Kapitel 3.2.2 zur Beschreibung einiger Parameter des WWW-Modells verwendet werden, wurde in Kapitel 2.4.4.3 ein methodisches Vorgehen beschrieben. An dieser Stelle wird die Methode exemplarisch für alle eingesetzten Kombinationen veranschaulicht.

C.1 Weibull/Pareto

Die komplementäre Verteilungsfunktion der Weibull-Verteilung (für den Kopfbereich) legt nach Wahl von x_1 den Punkt (x_1, p_1) fest mit $F_{\text{Weibull}}(x_1) = p_1$. Die Pareto-Verteilung (für den *Tail*) ist wie folgt definiert:

$$F(x) = 1 - \left(\frac{k}{x}\right)^a, \quad x > k \quad (\text{C.1})$$

Im Übergangspunkt (x_1, p_1) muß $F_{\text{Pareto}}(x_1) = p_1$ gelten, also

$$\left(\frac{k}{x_1}\right)^a = 1 - p_1 \quad (\text{C.2})$$

$$a(k) = \frac{\ln(1 - p_1)}{\ln \frac{k}{x_1}} = \frac{\ln(1 - p_1)}{\ln k - \ln x_1} \quad (\text{C.3})$$

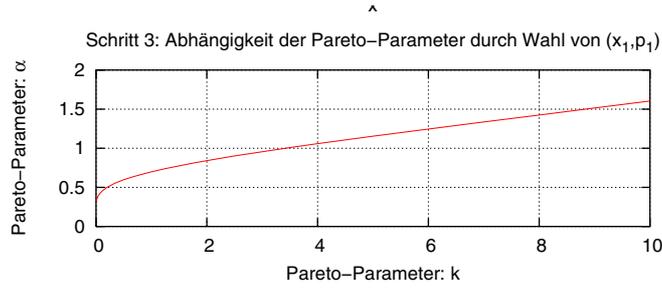


Bild C.1: Abhängigkeiten der Parameter der Pareto-Verteilung durch Punkt (x_1, p_1)

Durch Ersetzen von a mit $a(k)$ (Bild C.1) ist die Pareto-Funktion für die zweite Approximation so eingeschränkt, daß Sie immer durch den Punkt (x_1, p_1) läuft. Das Ergebnis der automatisierten Approximation ist in Bild C.2 dargestellt.

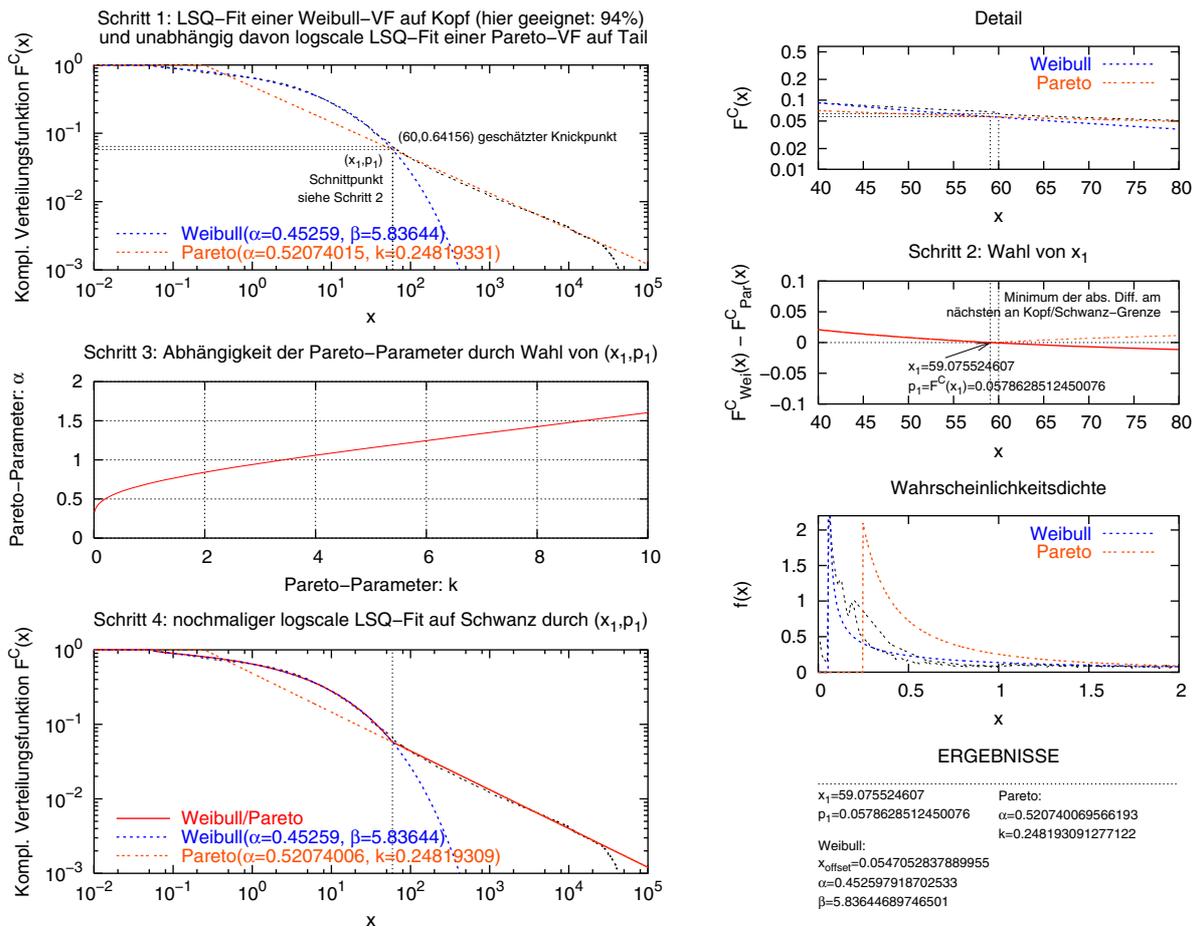


Bild C.2: Betrachtungsdauer

C.2 Lognormal/H2

Durch die Lognormal-Verteilung ist nach Wahl von x_1 analog zu Abschnitt C.1 der Übergangspunkt (x_1, p_1) festgelegt. Die Hyperexponentielle Verteilungsfunktion zweiter Ordnung (H2) ist definiert durch

$$F(x) = 1 - p e^{-\frac{x}{m_1}} - (1-p) e^{-\frac{x}{m_2}} \quad (\text{C.4})$$

und muß $F_{\text{H2}}(x_1) = p_1$ erfüllen, so daß

$$p e^{-\frac{x_1}{m_1}} + e^{-\frac{x_1}{m_2}} - p e^{-\frac{x_1}{m_2}} = 1 - p_1 \quad (\text{C.5})$$

$$p \left(e^{-\frac{x_1}{m_1}} - e^{-\frac{x_1}{m_2}} \right) = 1 - p_1 - e^{-\frac{x_1}{m_2}} \quad (\text{C.6})$$

$$p(m_1, m_2) = \frac{1 - p_1 - e^{-\frac{x_1}{m_2}}}{e^{-\frac{x_1}{m_1}} - e^{-\frac{x_1}{m_2}}}, \quad m_1 \neq m_2 \quad (\text{C.7})$$

Somit ist p durch m_1 und m_2 bestimmt, so daß die eingeschränkte Hyperexponentielle Verteilungsfunktion nur noch zwei Freiheitsgrade hat (siehe Bild C.3).

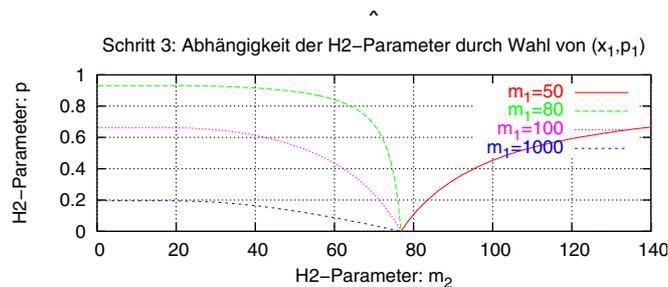


Bild C.3: Abhängigkeiten der Parameter der H2-Verteilung durch Punkt (x_1, p_1)

Wenn die hyperexponentielle Exponentialfunktion zweiter Ordnung (H2) wie hier mit einem Offset eingesetzt wird, muß beachtet werden, daß der Offset in Formel (C.7) einfließt, d.h. in diesem Fall bestimmt sich p zu

$$p = \frac{1 - p_1 - e^{-\frac{x_1 - x_{\text{offset}}}{m_2}}}{e^{-\frac{x_1 - x_{\text{offset}}}{m_1}} - e^{-\frac{x_1 - x_{\text{offset}}}{m_2}}}, \quad m_1 \neq m_2 \quad (\text{C.8})$$

Bild C.4 zeigt die Zwischenschritte und das Ergebnis der solchermaßen automatisierten Parametrisierung der Größen der HTTP Requests.

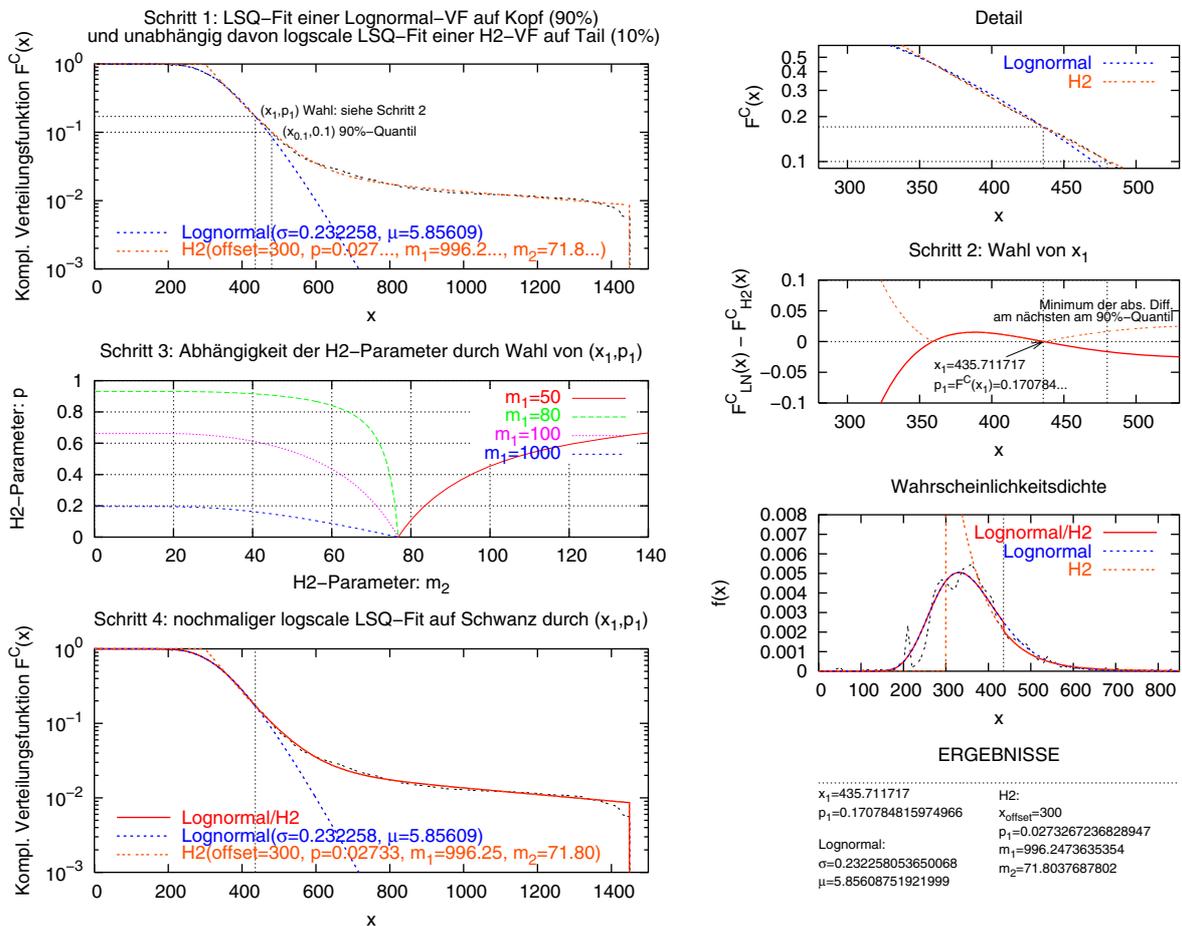


Bild C.4: GröÙe der HTTP-Anforderungen

C.3 Weibull / Lognormal

Für die Verteilungsfunktion der Lognormalfunktion existiert keine geschlossene Form. Nur die Verteilungsdichtefunktion kann folgendermaßen allgemein angegeben werden:

$$f(x) = \frac{e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma x}, \alpha > 0 \quad (C.9)$$

Da für die Lognormalfunktion keine Darstellung der Verteilungsfunktion existiert, kann auch die Randbedingung, daß Punkt (x_1, p_1) durchlaufen werden muß, nicht in die Formel eingearbeitet werden, so daß ein Fitting einer angepaßten Lognormalfunktion nicht möglich ist. Alternativ kann die Anpassung an den Übergangspunkt auch für die Näherungsfunktion für den

Kopfbereich durchgeführt werden. Dann wird (x_1, p_1) durch die approximierte Lognormalfunktion festgelegt und damit eine Randbedingung für die zweite Approximation der Weibull-Verteilung geschaffen. Die Weibull-Verteilungsfunktion ist wie folgt definiert:

$$F(x) = 1 - e^{-\left(\frac{x}{\beta}\right)^\alpha}, \alpha, \beta > 0 \quad (\text{C.10})$$

Wegen $F_{\text{Weibull}}(x_1) = p_1$ muß also gelten

$$e^{-\left(\frac{x_1}{\beta}\right)^\alpha} = 1 - p_1 \quad (\text{C.11})$$

$$\left(\frac{x_1}{\beta}\right)^\alpha = -\ln(1 - p_1) = \ln \frac{1}{1 - p_1} \quad (\text{C.12})$$

$$\beta = x_1 \left(\ln \frac{1}{1 - p_1} \right)^{-\frac{1}{\alpha}} \quad (\text{C.13})$$

Oder einfacherer mit $\lambda = \beta^{-\alpha}$, $n = \alpha$

$$F(x) = 1 - e^{-\lambda x^n}, \lambda, \alpha > 0 \quad (\text{C.14})$$

$$\lambda(n) = -\frac{\ln p_1}{x_1^n} \quad (\text{C.15})$$

Damit ist ein Parameter der Weibull-Funktion vom anderen abhängig und die eingeschränkte Weibull-Verteilung hat nur noch einen Freiheitsgrad.

