Optimierte Steuerung in VoIP-Netzen für eine effiziente Ressourcennutzung

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

vorgelegt von

Thomas Steinert

geb. in Villingen

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Mitberichter:	Prof. Dr.-Ing. Ralf Steinmetz, TU Darmstadt
Tag der Einreichung:	11. Dezember 2003
Tag der mündlichen Prüfung:	16. Juni 2005

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2005

Inhaltsverzeichnis

Abkürzungen

A/D	Analog/Digital
ACF	Admission Confirm
ACG	Automatic Call Gapping
AD	Administrative Domain
ARJ	Admission Reject
ARPA	Advanced Research Projects Agency
ARQ	Admission Request
ASPA	Aggregate Server Access Protocol
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband ISDN
BE	Border Element
CAC	Connection Admission Control
CH	Clearing House
CODEC	Coder/Decoder
CTI	Computer Telephony Integration
DARPA	Defense Research Projects Agency
DCF	Disengage Confirm
DiffServ	Differentiated Services
DoS	Denial of Service
DRQ	Disengage Request
EFSM	Extended Finite State Machine
ENRP	Endpoint Name Resolution Protocol
Erl	Erlang
FTP	File Transfer Protocol
GK	Gatekeeper
GoS	Grade of Service
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IN	Intelligent Network
IntServ	Integrated Services
IP	Internet Protocol
IPv6	Internet Protocol, Version 6
ISDN	Integrated Services Digital Network
ITU	International Telecommunication Union
ITU-T	International Telecommunication Union - Telecommunication Sector
IVR	Interactive Voice Response
LAN	Local Area Network

LB	Leaky Bucket
LDAP	Lightweight Directory Access Protocol
LIV	Load Indicator Value
MAC	Media Access Control
MC	Multipoint Controller
MCU	Multipoint Control Unit
MG	Media Gateway
MGC	Media Gateway Controller
MIPS	Millions of Instructions per Second
MP	Multipoint Processor
MPLS	Multi-Protocol Label Switching
NOR	Number of Open Requests
OSI	Open Systems Interconnection
OvP	Overload Protection
PBX	Private Branch Exchange
PC	Personal Computer
PDU	Protocol Data Unit
PHB	Per-Hop Behaviour
PPP	Point to Point Protocol
PT	Percentage Throttling
QL	Queue Length
QoS	Quality of Service
RAC	Resources Available Confirm
RAI	Resources Available Indicate
RAS	Registration, Admission and Status
RFC	Request for Comments
RSVP	Resource Reservation Protocol
RTCP	RTP Control Protocol
RTP	Real Time Transport Protocol
RTT	Round Trip Time
SCM	Selected Communication Mode
SCP	Service Control Point
SCTP	Stream Control Transmission Protocol
SDH	Synchronous Digital Hierarchy
SDL	Specification and Description Language
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SS	Supplementary Service
SS7	Signalling System No. 7
SSP	Service Switching Point
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoD	Video-on-Demand
VoIP	Voice over IP

WAN	Wide Area Network
WCS	Weighted Connection States
WDM	Wavelength Division Multiplex
WIN	Window-Method
WWW	World Wide Web

Formelzeichen

a_j	Gewicht des Mittelwerts des Intervalls $k - N + j$, $j \in \{1, \ldots, N\}$
B	Blockierwahrscheinlichkeit
C	Kosten
$C_{intrazone}$	Kosten für die Weiterleitung einer Anforderung innerhalb des Gatekeeper-Clusters
$C_{intrazone,\ fail}$	Kosten für eine fehlgeschlagene Weiterleitung einer Anforderung innerhalb des Gatekeeper-Clusters
$C_{intrazone,\ success}$	Kosten für eine erfolgreiche Weiterleitung einer Anforderung innerhalb des Gatekeeper-Clusters
C_{local}	Kosten für die Bearbeitung einer Anforderung im lokalen Gatekeeper
$C_{local,\ fail}$	Kosten bei fehlgeschlagener Bearbeitung einer Anforderung im lokalen Gatekeeper
$C_{local,\ success}$	Kosten bei erfolgreicher Bearbeitung einer Anforderung im lokalen Gatekeeper
$\overline{D}$	Mittlere Antwortverzögerung
f_a	Anpassungsfaktor für sich ändernde Verkehrscharakteristika
g_ν	Gewichtungsfaktor für Klasse ν
h	Mittlere Bedienzeit
i	Intervallnummer
k	Intervallnummer
LIV_{WCS}	Lastindikatorwert für Lastindikator *Gewichtete Verbindungszustände*
m	Anzahl der Bedieneinheiten
N	Anzahl der Intervalle für die Mittelwertbildung
P	Power
P_B	Power mit Berücksichtigung der Blockierwahrscheinlichkeit
P_P	Power-Produkt
P_S	Power-Summe
$p_{intrazone,\ fail}$	Wahrscheinlichkeit für eine fehlschlagende Weiterleitung einer Anforderung innerhalb eines Gatekeeper-Clusters
$p_{local,\ fail}$	Wahrscheinlichkeit für eine fehlschlagende Bearbeitung einer Anforderung im lokalen Gatekeeper
R_{base}	Basis-Ressourcenverbrauch für Bearbeitung einer vollständigen Verbindung ohne zusätzliche Dienstanfragen
$R_{effective}$	tatsächlicher Ressourcenverbrauch für Bearbeitung einer vollständigen Verbindung
S_R	Anzahl zwischengespeicherter Verbindungsanforderungen
$S_{R,max}$	Maximale Anzahl zwischengespeicherter Verbindungsanforderungen

$\overline{SS_{num}}$	Mittelwert der Anzahl der zusätzlichen Dienstanfragen innerhalb einer Verbindung
S_T	Token-Anzahl
$S_{T,max}$	Maximale Token-Anzahl
T_G	Gap Time Intervall
T_{LB}	Leaky Bucket Zeitintervall
T_{TK}	Token-Erzeugungs-Zeitintervall
W	Fenstergröße
W_{max}	Maximale Fenstergröße
$W(S)$	Gewicht eines Verbindungszustands
$\hat{x}(k)$	Indikatorwert für das Intervall k
$\tilde{x}(i)$	Mittelwert für das Intervall i
α	Kosten pro Anforderung
β	Gewinn je erfolgreicher Anforderung
λ_x	Angebotsrate - Mittlere Anzahl eintreffender Anforderungen pro Zeiteinheit
λ_y	Durchsatzrate - Mittlere Anzahl erfolgreicher Anforderungen pro Zeiteinheit
ξ	Glättungsfaktor
ρ	Mittlere Auslastung

Optimierte Steuerung in VoIP-Netzen für eine effiziente Ressourcennutzung

Kurzfassung

Seit Jahren nimmt die Bedeutung der Internet-basierten Kommunikation zu. Durch die kontinuierliche Verbesserung der entsprechenden Netze in den Bereichen Zuverlässigkeit und Qualität der Datenübertragung ist es nun möglich, Telekommunikationsdienste erfolgreich in diesen Netzen einzuführen. Im Bereich der Internet-basierten Netze werden diese Dienste als VoIP-Dienste (Voice over Internet Protocol) bezeichnet. Somit erlauben Internet-basierte Netze die Integration der Daten- und der Telekommunikationsdienste und sind daher ein wichtiger Faktor für die Konvergenz der Netze.

Für eine weitreichende Verbreitung der VoIP-Dienste ist es notwendig, dass die Diensterbringung ebenso stabil wie bei den Telekommunikationsdiensten erfolgt. Daher muss die Diensterbringung auch in Hoch- und Überlastsituationen gewährleistet sein. Dies kann nur erreicht werden, indem die verfügbaren Ressourcen in optimierter Form genutzt werden.

Diese Arbeit behandelt Steuerungsverfahren für die effiziente Ressourcenverwendung in einer VoIP-Umgebung. Der Schwerpunkt der Verfahren liegt auf den Ressourcen der Steuerung, so dass viele Komponenten von der Anwendung dieser Verfahren profitieren. Die vorgestellten Verfahren entstammen sowohl aus der Daten- als auch aus der Telekommunikation, wobei sie für die Verwendung in einer VoIP-Umgebung entsprechend angepasst wurden. Des Weiteren wurden in dieser Arbeit neue Verfahren abgeleitet.

Kapitel 2 enthält eine Einführung zu VoIP. Es beinhaltet eine detaillierte Beschreibung der VoIP-Architektur der ITU-T-Empfehlung H.323 (International Telecommunication Union - Telecommunication Sector) und ihren Steuerungskomponenten, sowie der relevanten Signalisierprotokolle. In dieser Architektur stellt der *Gatekeeper* eine zentrale Komponente dar, da er für die Verwaltung und Steuerung einer Zone zuständig ist. Eine *Zone* besteht aus allen Komponenten, die bei einem Gatekeeper angemeldet sind. Der prinzipielle Ablauf der Operationen für die Steuerung der VoIP-Dienste ist zwar dem für klassische Telefoniedienste sehr ähnlich,

jedoch ist die Menge der bearbeiteten Daten höher und durch die Unterstützung unterschiedlicher Dienste auch inhomogener. Daher müssen die angewandten Verfahren für die optimierte Steuerung flexibel bezüglich dieser Eigenschaften von VoIP-Diensten sein. Darüber hinaus kann eine VoIP-Umgebung seine Struktur mittels entsprechender Steuerungstransaktionen verändern, da nur eine logische Zuordnung zwischen einer Komponente und ihrem steuernden Gatekeeper besteht. Im Gegensatz dazu besteht zwischen einem Telefonendgerät und seiner steuernden Vermittlungsstelle in der Regel eine physikalische Zuordnung, da das Telefon direkt mit seiner Vermittlungsstelle verbunden ist.

In Kapitel 3 werden Verfahren für eine optimierte Steuerung für eine H.323-basierte VoIP-Umgebung vorgestellt. Zunächst werden mögliche Leistungsdefinitionen beschrieben, die das Ziel einer optimierten Steuerung darstellen können. Es wird der prinzipielle Ablauf der optimierten Steuerung bestehend aus Lastzustandsbestimmung, Lastverteilung und Überlastabwehr definiert und für unterschiedliche Ressourcen einer VoIP-Umgebung wie z.B. die Übertragungskapazität oder ein Gateway angewandt.

Bei der Einordnung der Arbeit konnte festgestellt werden, dass die in dieser Arbeit behandelten Methoden und Verfahren einen neuartigen Ansatz für die optimierte Steuerung in einer VoIP-Umgebung darstellen.

Wie bereits erwähnt, stellt der Gatekeeper einen zentralen Punkt einer H.323-basierten VoIP-Umgebung dar, weshalb seine Funktionalität sicher gestellt werden muss. Der Großteil dieser Arbeit behandelt die optimierte Steuerung der Gatekeeper-Ressourcen. Für die Bestimmung seines Lastzustands werden verschiedene bekannte Lastindikatoren untersucht. Darüber hinaus wird ein neuer Lastindikator “Gewichtete Verbindungszustände” abgeleitet. Dieser erlaubt es, den zukünftigen Ressourcenbedarf innerhalb einer Signalisierbeziehung abzuschätzen.

Aus Zuverlässigkeits- und Skalierungsgründen wurde das Konzept des *Gatekeeper-Clusters* eingeführt. Ein Gatekeeper-Cluster besteht aus mehreren Gatekeepern, die gemeinsam eine Zone steuern. Dies erlaubt es, die anfallende Last mittels geeigneter *Intrazonen*-Lastverteilungsverfahren auf mehrere Gatekeeper zu verteilen. Da diese Gatekeeper sowohl Konfigurations- als auch Verbindungszustandsdaten der verwalteten Endpunkte gemeinsam verwenden, ist es notwendig, dass der Zugriff auf diese Daten durch die Cluster-Mitglieder entsprechend verwaltet wird. Insbesondere muss dabei die Konsistenz der Daten gewährleistet werden. Des Weiteren wird auf die Granularität der Lastverteilung eingegangen, d. h. ob einzelne Verbindungen, Verbindungsphasen oder Signalisiernachrichten jeweils einem anderen Gatekeeper zugeteilt werden. Schließlich müssen die Lastverteilungsverfahren für die Anwendung in einer VoIP-Umgebung adaptiert werden.

Des Weiteren wird ein *Interzonen*-Lastverteilungsverfahren abgeleitet, welches es erlaubt, die Struktur einer VoIP-Umgebung zu verändern. Dies erfolgt, indem z. B. ein Gatekeeper eines

Clusters einem Gatekeeper-Cluster einer anderen Zone zugeordnet wird, so dass die verfügbaren Ressourcen beider Zonen optimal genutzt werden.

Wenn die Last an einem Gatekeeper zu groß ist, müssen Überlastabwehrmaßnahmen angewandt werden. In dieser Arbeit werden verschiedene Maßnahmen aus dem Bereich der Telekommunikation für die Verwendung in einer VoIP-Umgebung adaptiert und untersucht.

Nach einigen Implementierungsaspekten, die die vorgestellten Verfahren zur optimierten Steuerung betreffen, wird ein Ansatz für die Steuerung eines integrierenden Netzes beschrieben. In diesem Ansatz führt der Gatekeeper sowohl die Steuerung für die Daten- als auch für die Telekommunikationsdienste durch.

Kapitel 4 beschreibt die angewandten Untersuchungsmethoden. Zur Untersuchung existierender Gatekeeper-Implementierungen wurde der *PreServer* entwickelt. Dieser stellt eine prototypische Implementierung verschiedener Überlastabwehrmaßnahmen dar. Darüber hinaus wurde ein ereignisgesteuertes Simulationsprogramm erstellt, das die Untersuchung verschiedener Lastindikatoren, Intrazonen- und Interzonen-Lastverteilungsverfahren sowie Überlastabwehrmaßnahmen erlaubt. Dieses Programm ermöglicht die Bestimmung sowohl des stationären als auch des instationären Verhaltens der untersuchten Verfahren.

Die Ergebnisse der durchgeführten Untersuchungen und die Bewertung dieser Ergebnisse werden in Kapitel 5 vorgestellt. Zunächst wird ein einzelner Gatekeeper betrachtet. Die Ergebnisse der Untersuchungen mit dem PreServer zeigen die prinzipielle Wirksamkeit der Überlastabwehrmaßnahmen zur Maximierung des Durchsatzes bei gleichzeitiger Begrenzung der Antwortverzögerung und Verhinderung des Fehlschlagens von Verbindungen. Die Simulationen bestätigen diese Beobachtungen. Darüber hinaus zeigen sie die Reaktionsfähigkeit der Verfahren bei einem Lastsprung.

Das Verhalten der untersuchten Lastindikatoren ist nahezu gleich, so dass alle es erlauben, den aktuellen Lastzustand eines Gatekeepers in geeigneter Form zu bestimmen. Betrachtet man darüber hinaus den Implementierungsaufwand, besitzt der Lastindikator “Warteschlangenlänge” gegenüber “Gewichtete Verbindungszustände” durch seine einfache Funktionalität und seine Unabhängigkeit vom Inhalt einer Signalisiernachricht einige Vorteile.

Die Ergebnisse für die untersuchten Überlastabwehrmaßnahmen sind ebenfalls sehr ähnlich, wobei jeweils der Durchsatz bei gleichzeitiger Begrenzung der Antwortverzögerung und der Verhinderung fehlschlagender Verbindungen maximiert wurde.

Wenn der Ressourcenbedarf für die Bearbeitung einer Verbindung variiert, z. B. aufgrund der Durchführung zusätzlicher Dienste, muss die Konfiguration der optimierten Steuerungsverfahren adaptiert werden. Für die kontinuierliche Adaption ohne erneute Konfiguration der Verfahren wird ein *Anpassungsfaktor* vorgeschlagen, der mittels Messung des mittleren Ressourcenbedarfs einer Verbindung aktualisiert wird.

Die Untersuchung der Granularität der Lastverteilung in einem Gatekeeper-Cluster zeigt, dass die Verteilung auf Verbindungsebene vorteilhaft ist, da dadurch weniger Ressourcen als bei der Verteilung auf Verbindungsphasen- oder Nachrichten-Ebene benötigt werden und die Cluster-Mitglieder immer noch nahezu gleichmäßig belastet sind.

Wenn die Lastverteilungsverfahren ohne Überlastabwehr angewandt werden, zeigt das “Sender-Receiver”-Verfahren eine interessante Eigenschaft: Dieses Lastverteilungsverfahren leitet eine Anforderung nur dann an ein anderes Cluster-Mitglied weiter, wenn eines der Cluster-Mitglieder über genügend Kapazitäten verfügt, um die zusätzliche Anforderung erfolgreich zu bearbeiten. Ansonsten wird die Anforderung durch den ursprünglichen Gatekeeper bearbeitet. Dies verhindert die nutzlose Verteilung von Anforderungen, so dass ein höherer Durchsatz in bestimmten Überlastbereichen erreicht wird.

Falls die Lastverteilungsverfahren zusammen mit einer Überlastabwehrmaßnahme angewandt werden, erzielen alle Lastverteilungsverfahren außer der statischen Lastverteilung einen hohen Durchsatz, so dass der zusätzliche Aufwand für die Verwaltung der gemeinsamen Daten gerechtfertigt ist. Darüber hinaus werden die Antwortzeiten begrenzt und das Fehlschlagen von Verbindungen wird verhindert. Wegen der geringen Unterschiede zwischen dem Verhalten der verschiedenen dynamischen Lastverteilungsverfahren muss ihre Implementierung betrachtet werden. Der zentral gesteuerte “Round Robin”-Algorithmus benötigt neben den Gatekeepern eine zusätzliche Komponente, den *Dispatcher*, der die ankommenden Nachrichten auf die Gatekeeper des Clusters verteilt. Dagegen benötigt ein Lastverteilungsverfahren mit verteilter Steuerung, wie z. B. das “Sender-Receiver”-Verfahren, keine zusätzliche Komponente. Jedoch müssen alle Cluster-Mitglieder über die Funktionalität der Lastverteilung verfügen. Diese Funktionen werden auch in Hoch- und Überlastsituationen durchgeführt, in denen die Ressourcen knapp sind. In einem Cluster mit zentraler Steuerung sind die Gatekeeper nahezu unabhängig von der Lastverteilung, so dass Standard-Gatekeeper verwendet werden können.

Zur Bestimmung des zeitlichen Verhaltens des Interzonen-Lastverteilungsverfahrens wurden instationäre Simulationen durchgeführt. Diese zeigen die Verbindung zwischen den verfügbaren Ressourcen für die Durchführung der Lastverteilung und der Dauer der entsprechenden Aktionen. Darüber hinaus werden die Auswirkungen auf die Dienstgüte aufgezeigt. Weitere Untersuchungen zeigen, dass die Weiterleitung eines Gatekeepers von einem wenig zu einem überlasteten Cluster zu einer effizienteren Ressourcennutzung führt und daher auch zu einem höheren Durchsatz. Dagegen zeigt die Weiterleitung einzelner Endpunkte von einer überlasteten zu einer wenig belasteten Zone keinen Effekt, da der Lastanteil dieser Endpunkte sehr gering gegenüber der gesamten Last in der Zone ist.

Insgesamt betrachtet, erlauben die vorgestellten Verfahren für eine optimierte Steuerung die effiziente Nutzung der verfügbaren Ressourcen. Dies führt zu einem maximierten Durchsatz, während gleichzeitig die Antwortverzögerungen begrenzt werden und das Fehlschlagen von

Verbindungen verhindert wird. Darüber hinaus können die untersuchten Verfahren für eine erweiterte Form des Gatekeepers verwendet werden. Dieser ist neben der Verwaltung der VoIP-Dienste einer Zone auch für die Datendienste zuständig, so dass eine optimierte Steuerung eines Netzes möglich ist, das Daten- und Telekommunikationsdienste integriert.

Control optimization in VoIP networks for efficient resource utilization

Summary

In the last years Internet-based communication networks have become increasingly more important. Because of their continuous improvement concerning reliability and quality of data transmission, even telecommunication services can be deployed successfully in these networks. In the context of Internet-based networks, these services are referred to as VoIP (Voice over Internet Protocol) services. Internet-based networks therefore permit to integrate both data and telecommunication services and are an important factor for the convergence of networks.

For a widespread deployment of VoIP services it is necessary that the operation of these services is as stable as it is known for telecommunication services. Therefore, the service provision has to be guaranteed also in high and overload situations. This can only be achieved if the available resources are used in an optimized way.

This thesis addresses control methods for efficient resource usage in a VoIP environment. The focus of the methods is on the control resources where multiple components benefit from the application of these methods. The presented methods are based on methods from both the data and the telecommunication sector. For the use in a VoIP environment they are adapted correspondingly. Furthermore, new methods are derived in this thesis.

Chapter 2 contains an introduction to VoIP. It includes a detailed description of the VoIP architecture of ITU-T recommendation H.323 (International Telecommunication Union - Telecommunication Sector) and its control components as well as the relevant signalling protocols. In this architecture the *gatekeeper* is a central component, because it is responsible for the administration and the control of a zone. A *zone* consists of all components registered at a gatekeeper. The principal sequence of operations for the control of VoIP services is very similar to the one for the control of classical telephone services, but the amount of processed data is higher and more variable for VoIP services. Therefore, the applied methods for optimized control have to

be flexible concerning this complexity. Furthermore, a VoIP environment permits to change its structure by means of control actions, as there is only a logical relation between a component and its controlling gatekeeper. In contrast, the relation between a telephone terminal and its controlling telecommunication switch is usually a physical one, because the telephone is directly connected to the switch.

In chapter 3 methods for optimized control of an H.323-based VoIP environment are presented. At first possible performance definitions, which can be the aim for an optimized control, are described. The general sequence of operations consisting of the tasks load state determination, load distribution and overload protection is defined and applied to different resources in an VoIP environment, as e. g. the transmission path or a gateway.

The discussion of related publications and product descriptions shows that the methods and procedures addressed in this thesis represent a novel approach to control optimization in a VoIP environment.

As already mentioned, the gatekeeper represents the central point in an H.323-based VoIP environment. Therefore, its proper functionality has to be assured. The main part of this thesis addresses the optimized control for gatekeeper resources. For the determination of its load states existing load indicators are investigated. Furthermore, a new load indicator, "weighted connection states", is derived. It permits to estimate the future resource requirements within a signalling relation.

For scalability and reliability reasons the concept of a *gatekeeper-cluster* is introduced. A gatekeeper-cluster consists of several gatekeepers controlling a zone, which permits to distribute the load to several gatekeepers by means of *intrazone* load distribution methods. Because these gatekeepers share both configuration and connection state data of the controlled endpoints, it is necessary that the members of the cluster are able to access this data in an appropriate way. This includes methods for ensuring the consistence of the data. Related to this subject is the granularity of the load distribution. This means whether each connection, each connection phase or each signalling message can be assigned to a different gatekeeper. Finally the load distribution methods have to be adapted for the use in a VoIP environment.

Furthermore, an *interzone* load distribution method is derived. This method allows to change the structure of a VoIP environment, e. g. by moving a gatekeeper from one cluster to the gatekeeper-cluster of another zone, so that the available resources of both zones can be used in an optimized way.

If the load at a gatekeeper is too high, overload protection procedures have to be applied by this gatekeeper. Several methods known from the telecommunication sector are adapted in this thesis for the use in a VoIP environment.

After some implementation aspects concerning the presented control optimizing methods, an approach for a converged communication network with integrated control is described. In this approach the gatekeeper performs the control for both data and telecommunication services.

Chapter 4 addresses the applied investigation methods. For investigating existing gatekeeper implementations the *PreServer* was developed. This system represents a prototypical implementation of overload protection procedures. Furthermore, an event-driven simulation tool was generated, which allows the investigation of load indicators, intrazone and interzone load distribution methods as well as overload protection procedures. This simulation tool permits the determination of the stationary and the transient behaviour of the investigated methods.

The results of the conducted investigation and the evaluation of these results are presented in chapter 5. At first a single gatekeeper is considered. The results of the tests with the PreServer indicate the principle validity of overload protection procedures to maximise throughput while limiting the answer delay and preventing connection failures. The simulations confirm these observations. Furthermore, they demonstrate the reactivity of the procedures in case of a load step.

The investigated load indicators behave in almost the same way, so that all of them permit to derive the current load state of a gatekeeper appropriately. Concerning the implementation, the load indicator "queue length" possesses some advantages compared to the load indicator "weighted connection state" because of its simple functionality and its independence of the signalling message content.

The investigated overload protection procedures are also very similar. All of them achieve to maximise throughput while limiting answer delay and preventing connection failures.

If the amount of resources required for the processing of a connection varies, because e. g. of the application of supplementary services, the configuration of the control optimization methods has to be adapted. For a continuous adaptation without re-configuration of the methods an *adaptation factor* is suggested, which is updated by means of measurements of the mean resource usage for a connection.

The investigation of the load distribution granularity in a gatekeeper-cluster demonstrates that the distribution on the connection level is favourable because it consumes less resources than the distribution on the connection phase and on the message level, while the cluster members are still nearly uniformly loaded.

When applying load distribution methods without overload protection the "sender-receiver"-method shows an interesting property: this load distribution method forwards a request to another cluster member only if one of the other cluster members has sufficient resources to process this additional request. Otherwise the request will be handled at the original gate-

keeper. This prevents useless distribution of requests, which results in a higher throughput in a certain overload load range.

If the load distribution methods are applied with an overload protection procedure, all methods except the static load distribution achieve a high throughput so that the supplementary effort for the shared data administration is justified. Furthermore, the answer delays are limited and connection failures are prevented. Because of the small difference between the behaviour of the dynamic load distribution methods, their implementation has to be taken into account. The centrally controlled "round-robin" algorithm needs besides the gatekeepers a supplementary component, the *dispatcher*, to distribute incoming messages to the gatekeepers of the cluster. In contrast, a load distribution method with distributed control, as e. g. the "sender-receiver" method, needs no supplementary component, but all cluster members have to be able to perform the load distribution functions. These functions are applied even in high and overload situations, where resources are sparse. In a cluster with central control the gatekeepers are almost independent from the load distribution, so that standard gatekeeper implementations can be used within this cluster type.

To determine the temporal behaviour of the interzone load distribution method, transient simulations are applied. They show the relation between the available resources for the execution of the method and the duration of the corresponding actions. Furthermore, the impact on the quality of service is illustrated. Further investigations demonstrate that the moving of a gatekeeper from a less loaded to an overloaded cluster leads to a more efficient resource utilisation and, therefore, to a higher throughput. In contrast, the moving of endpoints from an overloaded to a less loaded zone has no effect, because of the small portion of affected load compared with the overall load at the zone.

In conclusion, the presented methods and procedures for an optimized control allow the efficient utilization of the available resources. This results in a maximised throughput while answer delays are limited and connection failures are prevented. In addition, the investigated methods could be applied to an extended version of a gatekeeper, which is not only responsible for the administration of the VoIP services in a zone, but also for the data services, so that an integrated control of a converged network could be achieved.

Kapitel 1

Einleitung

Seit Jahren nimmt die Bedeutung der Internet-basierten Kommunikationsnetze sehr stark zu, wobei diese Netze, von wenigen Ausnahmen abgesehen, für die Datenkommunikation verwendet wurden. Diese hat im Vergleich zur Telekommunikation geringere Anforderungen an das zeitliche Verhalten sowie an die Verfügbarkeit des Netzes. Durch die weite Verbreitung des Internet und die Einführung neuer Dienste stiegen diese Anforderungen jedoch immer weiter an, so dass in diesem Bereich vielfältiger Forschungsbedarf bestand und noch immer besteht. Der dadurch entstehende technologische Fortschritt führt zu einer immer größeren Zuverlässigkeit dieser Netze, so dass die Einführung von Telekommunikationsdiensten, wie z. B. dem Telefoniedienst, möglich wird. Damit entsteht mittels Internet-basierter Kommunikationsnetze ein Dienste-integrierendes Kommunikationsnetz, das sowohl Daten- als auch Telekommunikationsdienste unterstützt.

Im Bereich des Internet werden die Telekommunikationsdienste unter dem Begriff *Voice over IP* – VoIP (IP – *Internet Protocol*) zusammengefasst. Ihnen werden u. a. laut [118] große Zuwachsraten und somit eine glänzende Zukunft vorausgesagt, was auch durch einen Bericht der ITU (*International Telecommunication Union*) aus dem Jahre 2001 bestätigt wird [37].

1.1 Motivation

Damit die VoIP-Dienste mit den Diensten der klassischen Telekommunikation konkurrieren können, muss neben der Qualität der Datenübertragung auch die Zuverlässigkeit des Systems in Hoch- und Überlastphasen sichergestellt sein. Dies kann nur erreicht werden, indem die vorhandenen Ressourcen in optimierter Form genutzt werden und auf Änderungen der Belastung entsprechend reagiert wird. Des Weiteren sollen die Vorteile, die sich aus der Internet-Architektur ergeben, wie z. B. Flexibilität und Erweiterbarkeit, erhalten bleiben.

In dieser Arbeit werden Verfahren für eine optimierte Steuerung einer VoIP-Umgebung vorgestellt, die es erlauben, die verfügbaren Ressourcen effizient zu nutzen. Dabei werden vor allem die Ressourcen der Steuerung betrachtet, da ihre Wirkbreite besonders groß ist und somit viele Komponenten einer VoIP-Umgebung davon profitieren.

Die beschriebenen Verfahren entstammen teilweise der Datenkommunikation, wie z. B. die Lastverteilung innerhalb einer Gruppe von Rechnern, und teilweise der Telekommunikation, wie z. B. die Überlastabwehr in einer Vermittlungsstelle, wobei sie für die Verwendung in einer VoIP-Umgebung entsprechend angepasst werden. Des Weiteren werden auch neue Verfahren abgeleitet. Durch das Zusammenwirken verschiedener Verfahren wird somit eine optimierte Nutzung der zur Verfügung stehenden Ressourcen erreicht.

1.2 Gliederung der Arbeit

Im Anschluss an diese Einleitung erfolgt in Kapitel 2 eine Einführung zu VoIP. Dabei werden zunächst die notwendigen Grundlagen der Kommunikationstechnik gegeben. Dies beinhaltet auch eine kurze Beschreibung der Architektur und der relevanten Protokolle des Internets. Darüber hinaus wird auf die Konvergenz der Daten- und Telekommunikation eingegangen, die eine treibende Kraft bei der Entwicklung und der Einführung von VoIP-Diensten darstellt. Anschließend wird der Ablauf der Nutzdatenübertragung, die u. a. auch die Übertragung der Sprachdaten beim Telefoniedienst umfasst, beschrieben. Dabei wird auch allgemein auf dienstgüteunterstützende Verfahren für Internet-basierte Netze eingegangen. Schließlich wird die VoIP-Architektur nach der ITU-T-Empfehlung H.323 (ITU-T – *International Telecommunication Union - Telecommunication Sector*) vorgestellt. Dies beinhaltet eine Beschreibung der wichtigsten Steuerelemente und der relevanten Signalisierprotokolle. Da die weiteren Betrachtungen auf dieser Architektur basieren, erfolgt diese Beschreibung entsprechend detailliert. Des Weiteren werden die wesentlichen Unterschiede zwischen der Steuerung für VoIP-Dienste im Vergleich zur klassischen Telefonie aufgezeigt.

In Kapitel 3 werden Verfahren für eine optimierte Steuerung einer VoIP-Umgebung, die auf der Empfehlung H.323 basiert, beschrieben. Dazu werden zunächst mögliche Leistungsdefinitionen gegeben, die das Ziel einer optimierten Steuerung darstellen können. Anschließend wird der allgemeine Ablauf der Steuerungsoptimierung, der die Bestimmung des aktuellen Lastzustands mittels Lastindikatoren, die Lastverteilung und die Überlastabwehr umfasst, beschrieben.

Zur Einordnung der Arbeit wird eine Übersicht über relevante Verfahren für eine optimierte Steuerung aus Literatur und bestehenden Produkten, die sowohl der Daten- als auch der Telekommunikation entstammen, gegeben. Des Weiteren werden bestehende Ansätze und Untersuchungen für VoIP-Umgebungen beschrieben. Anschließend erfolgt eine Abgrenzung zu den in dieser Arbeit vorgestellten Verfahren.

Für die Steuerungsoptimierung für verschiedene Ressourcen, wie z. B. den Transportpfad oder *Gateways*, wird auf mögliche Verfahren eingegangen, die ihre effiziente Nutzung erlauben.

Der Schwerpunkt dieser Arbeit liegt auf den Verfahren zur optimierten Verwendung der Ressourcen der Steuerung. In einer H.323-basierten VoIP-Umgebung ist der sog. *Gatekeeper* für die Steuerung einer *Zone*, die alle beim Gatekeeper angemeldeten Komponenten umfasst, zuständig. Zunächst werden verschiedene Verfahren zur Bestimmung des aktuellen Lastzustands eines Gatekeepers mittels entsprechender Lastindikatoren präsentiert. Dabei wird auch ein neuer Lastindikator abgeleitet, der den sich ändernden Ressourcenverbrauch während einer VoIP-Kommunikation widerspiegelt. Anschließend werden Verfahren für die Lastverteilung vorgestellt. Dazu wird zunächst der sog. *Gatekeeper-Cluster* eingeführt, bei dem mehrere Gatekeeper gemeinsam die Steuerung übernehmen. Dieses aus dem Bereich der *Web-Server* bekannte Prinzip muss jedoch für VoIP-Dienste in adaptierter Form angewandt werden. Dabei spielt insbesondere die Datenverwaltung und der Zugriff auf gemeinsam verwendete Daten eine Rolle. Des Weiteren erfolgt eine Beschreibung der adaptierten Lastverteilungsverfahren. Neben dieser Form der Lastverteilung innerhalb einer Zone ist auch eine Lastverteilung über Zonengrenzen hinweg möglich. Dazu wird ein entsprechendes Verfahren abgeleitet. Anschließend werden Überlastabwehrmaßnahmen für Gatekeeper beschrieben, die Anwendung finden, wenn die bei einem Gatekeeper ankommende Last zu groß ist, so dass keine erfolgreiche Bearbeitung der Anforderungen gewährleistet werden kann. Dabei werden verschiedene aus der Telekommunikation bekannte Verfahren, die für VoIP-Dienste entsprechend adaptiert werden, vorgestellt. Schließlich werden einige Realisierungsaspekte für die beschriebenen Verfahren betrachtet.

Wenn die Ressourcen eines Kommunikationsnetzes, das Daten- und Telekommunikationsdienste integriert, effizient genutzt werden sollen, müssen die entsprechenden Ressourcen gemeinsam verwaltet werden. Auf diese Problematik wird ebenfalls eingegangen, wobei ein Vorschlag für die Durchführung einer derartigen gemeinsamen Verwaltung gegeben wird.

Zur Untersuchung der vorgestellten Verfahren zur Steuerungsoptimierung werden die in Kapitel 4 beschriebenen Methoden angewandt. Für die Untersuchungen mittels einer prototypischen Implementierung einzelner Überlastabwehrmaßnahmen wurde ein Testsystem entwickelt, das Untersuchungen zusammen mit existierenden Gatekeeper-Realisierungen erlaubt. Für die Simulationsstudien wurde ein Simulationswerkzeug erstellt, das die detaillierte Untersuchung der verschiedenen Verfahren sowohl im stationären als auch im instationären Fall ermöglicht.

In Kapitel 5 werden die Ergebnisse der durchgeführten Untersuchungen präsentiert und eine Bewertung dieser Ergebnisse vorgenommen. Dabei wird zunächst ein einzelner Gatekeeper betrachtet. Dazu werden Studien sowohl mittels prototypischer Implementierung als auch mittels Simulationen durchgeführt. Ziele dieser Studien sind die Ermittlung des Verhaltens verschiedener Lastindikatoren und Überlastabwehrmaßnahmen im stationären und instationären

Fall. Des Weiteren werden die Auswirkungen durch zusätzliche Dienste, die eine Veränderung der Verkehrscharakteristika bewirken, betrachtet.

Anschließend erfolgen die Untersuchungen für einen Gatekeeper-Cluster, die simulativ durchgeführt werden. Zunächst werden die Auswirkungen der Granularität der Lastverteilung aufgezeigt. Dabei beschreibt die Granularität, auf welcher Ebene die Lastverteilung erfolgt, d. h. ob einzelne Verbindungen, Verbindungsphasen oder Signalisiernachrichten jeweils einer verarbeitenden Komponente zugeordnet werden. Die weiteren Untersuchungen umfassen die Lastverteilungsverfahren innerhalb einer Zone, wobei diese zunächst ohne zusätzliche Durchführung einer Überlastabwehr betrachtet werden, bevor ihre Wirksamkeit gemeinsam mit einer Überlastabwehrmaßnahme ermittelt wird.

Schließlich wird das beschriebene Verfahren für die Lastverteilung über Zonengrenzen hinweg simulativ untersucht, wobei insbesondere der Verlauf der Lastverteilung betrachtet wird.

Kapitel 6 schließt diese Arbeit ab und fasst die wichtigsten Ergebnisse zusammen. Des Weiteren wird auf mögliche Erweiterungen und zusätzlichen Forschungsbedarf im Umfeld dieser Arbeit eingegangen.

Kapitel 2

Voice over IP – VoIP

Der Begriff *Voice over IP* – VoIP bezeichnet nicht nur die Sprachkommunikation über Internet-basierte Kommunikationsnetze, sondern er wird wesentlich allgemeiner aufgefasst. Er umfasst die gesamte Multimediakommunikation mittels Internet-basierter Netze und schließt somit u. a. Videoübertragung, gemeinsame Anwendungen, wie z. B. die gleichzeitige Bearbeitung eines Dokuments durch mehrere Benutzer, und Videokonferenzen mit ein. Daher können die Begriffe VoIP, Internet-Telefonie, IP-Telefonie und Multimediakommunikation über Internet-basierte Netze austauschbar verwendet werden.

Dieses Kapitel stellt die Prinzipien der Multimediakommunikation mittels Internet-basierter Kommunikationsnetze vor. Dazu wird im folgenden Abschnitt 2.1 zunächst eine Einführung in dieses Themengebiet gegeben, bei dem u. a. die relevanten Grundlagen vermittelt werden. Im anschließenden Abschnitt 2.2 wird der Austausch der Nutzdaten während der Kommunikation beschrieben. Dabei werden mögliche Probleme und bestehende Lösungsmöglichkeiten aufgezeigt. Schließlich wird in Abschnitt 2.3 die Signalisierung zur Steuerung der Kommunikation vorgestellt, wobei neben der detaillierten Vorstellung des von der ITU-T definierten Rahmenwerks H.323 auch Unterschiede zur Steuerung der kanalvermittelnden Telefonie aufgezeigt werden.

2.1 Einführung

Nach einer Einführung in die Grundlagen der Kommunikationstechnik in Abschnitt 2.1.1 wird im folgenden Abschnitt 2.1.2 das Internet und seine wichtigsten Eigenschaften vorgestellt, bevor in Abschnitt 2.1.3 auf die Konvergenz der Kommunikationsnetze eingegangen wird.

2.1.1 Grundlagen der Kommunikationstechnik

In diesem Abschnitt werden die für den weiteren Verlauf der Arbeit notwendigen Grundlagen vorgestellt. Dabei erfolgt in Abschnitt 2.1.1.1 zunächst eine Definition der Begriffe *Kommunikationsdienst* und *Dienstgüte*. Anschließend werden in den Abschnitten 2.1.1.2 und 2.1.1.3 die prinzipiellen Verfahren der Tele- und der Datenkommunikation vorgestellt. Abschließend werden in Abschnitt 2.1.1.4 die wichtigsten Eigenschaften dieser Kommunikationskonzepte aufgezeigt und darauf basierend werden die Konzepte miteinander verglichen.

Es sei darauf hingewiesen, dass dieser Abschnitt nur eine kurze Übersicht über die Grundlagen der Kommunikationstechnik gibt. Eine detailliertere Einführung der vorgestellten Begriffe und Konzepte kann z. B. in [67, 115] gefunden werden.

2.1.1.1 Kommunikationsdienst und Dienstgüte

Unter einem Kommunikationsdienst (im weiteren Verlauf als Dienst bezeichnet) versteht man nach [67] „alle funktionellen Eigenschaften eines Kommunikationsnetzes, welche eine bestimmte Kommunikationsform zwischen Endgeräten unterstützen, einschließlich aller funktionellen, qualitativen und rechtlichen Aspekte“. Diese Definition beinhaltet nicht nur die technischen Eigenschaften, wie Netzschnittstellen und Prozeduren, sondern bezieht sich auch auf die Qualität eines Dienstes, der durch die sog. *Dienstgüte* beschrieben wird, sowie auf rechtliche Aspekte, z. B. zur Wahrung des Fernmeldegeheimnisses.

Beispiele für Kommunikationsdienste sind Telefonie, Dateitransfer, E-Mail, SMS (*Short Message Service*) aber auch Rundfunk und Fernsehen.

Für die Telekommunikation werden darüber hinaus erweiterte Dienstmerkmale und sog. zusätzliche Dienste definiert, die den Basisdienst, in der Regel die Sprachkommunikation zwischen zwei Teilnehmern, um verschiedene Optionen erweitern und damit dem Benutzer mehr Möglichkeiten für die Kommunikation zur Verfügung stellen. Beispiele für erweiterte Dienstmerkmale und zusätzliche Dienste sind die Anzeige der Rufnummer des Rufenden beim Zielteilnehmer, automatischer Rückruf, Dreier-Konferenz, Halten einer Verbindung, Televotum und gebührenfreie Rufe.

Die Dienstgüte (QoS – *Quality of Service*) legt nach [17, 67] die Qualitätsmerkmale eines Kommunikationsdienstes aus der Sicht eines Benutzers fest. Beispiele für derartige Qualitätsmerkmale sind Verfügbarkeit, Zuverlässigkeit oder Übertragungsqualität. Die ITU untergliedert diese Sicht noch in die beiden Aspekte Netzgüte (*Network Performance*) und Verkehrsgüte (GoS – *Grade of Service*). Dabei definiert die Netzgüte die Fähigkeit eines Kommunikationsnetzes, einen geforderten oder vereinbarten Dienst zu erbringen und die Verkehrsgüte legt den Bereich der Dienstgüte fest, der von der Dimensionierung der Netzressourcen und der Netzorganisation abhängt. Für die Verkehrsgüte können objektive Metriken angewandt wer-

den, wie z. B. Informationsverlustwahrscheinlichkeit, Informationsverzögerung, Schwankung der Informationsverzögerung (*Jitter*) oder Verbindungsaufbauverzögerung.

2.1.1.2 Telekommunikation

Unter dem Begriff Telekommunikation wird in der Regel die Sprachkommunikation über kanalvermittelnde Netze verstanden. Bei den kanalvermittelnden Netzen werden jeder Kommunikationsbeziehung ein oder mehrere Kanäle konstanter Bandbreite exklusiv zur Verfügung gestellt. Die Realisierung eines Kanals hängt vom Multiplexverfahren ab, das die gemeinsame Übertragung verschiedener Kanäle über ein physikalisches Übertragungsmedium bezeichnet. Die in Tabelle 2.1 dargestellten Multiplexverfahren und Kanalrealisierungen sowie Kombinationen davon werden für die kanalvermittelte Kommunikation angewandt.

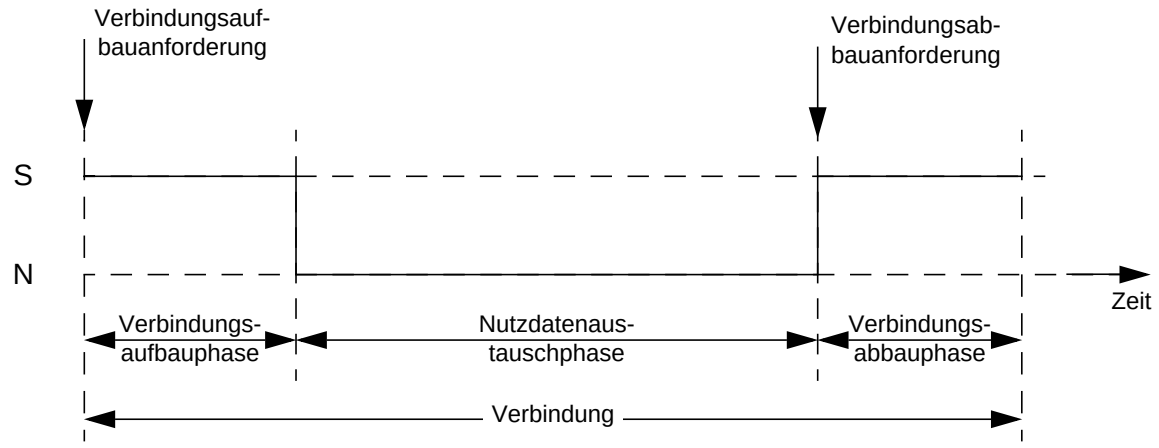
Multiplexverfahren	**Kanalrealisierung**
Raummultiplex	Leitung z.B. innerhalb eines Leitungsbündels
Frequenzmultiplex	Frequenzband
Synchrones Zeitmultiplex	Zeitlage
Wellenlängenmultiplex	Wellenlänge
Codemultiplex	Code

Tabelle 2.1: Multiplexverfahren und Kanalrealisierungen der kanalvermittelten Kommunikation

Die kanalvermittelte Kommunikation ist stets verbindungsbezogen, d. h. dass vor der eigentlichen Übertragung der zu kommunizierenden Daten zunächst eine Verbindung aufgebaut wird, um z. B. die benötigten Ressourcen zu reservieren, und dass nach dem Ende der Kommunikation die Verbindung abgebaut wird, um z. B. die verwendeten Ressourcen wieder frei zu geben, damit sie von anderen Verbindungen genutzt werden können.

Der Verbindungsauf- und -abbau ist Bestandteil der Verbindungssteuerung. Der für diese Steuerung des Dienstes notwendige Informationsaustausch zwischen den beteiligten Komponenten wird als Signalisierung bezeichnet. Bild 2.1 verdeutlicht die drei Phasen einer Verbindung.

Allgemein betrachtet erfolgt die Kommunikation zwischen Komponenten in einem Netz nach einem Protokoll. Dieses legt alle formalen und prozeduralen Eigenschaften der Kommunikation fest. So wird z. B. der Ablauf der Signalisierung für die Verbindungssteuerung mittels der entsprechenden Signalisierprotokolle definiert.



S Austausch von Steuerdaten – Signalisierung
N Austausch von Nutzdaten

Bild 2.1: Phasen einer Verbindung (aus [67])

2.1.1.3 Datenkommunikation

Für die Datenkommunikation wird meist ein paketvermittelndes Verfahren verwendet, bei dem die zu übertragenden Daten in Paketen fester oder variabler Länge übertragen werden. Ein Paket besteht aus einem sog. Kopf (*Header*), der die für dieses Paket relevanten Steuerinformationen enthält, und dem Nutzdatenanteil (*Payload*), der aus den zu übertragenden Daten besteht. Zur Übertragung der Pakete werden die Multiplexverfahren asynchrones Zeitmultiplex mit konstanter oder mit variabler Blocklänge angewandt. Bei diesen Multiplexverfahren existiert kein exklusiv einer Kommunikationsbeziehung zugeordneter physikalischer Kanal, sondern die jeweiligen Ressourcen werden gemeinsam von den einzelnen Kommunikationsbeziehungen benutzt. Die dadurch entstehenden Konflikte beim Zugriff auf Ressourcen werden durch Pufferung der Pakete, bis die jeweilige Ressource verfügbar ist, aufgelöst.

In der Datenkommunikation wird sowohl die verbindungslose als auch die verbindungsorientierte Kommunikation verwendet. Dabei muss unterschieden werden, auf welche Schicht des entsprechenden Schichtenmodells sich dieses Prinzip bezieht. In Bild 2.2 ist beispielsweise für das OSI-Referenzmodell (OSI – *Open Systems Interconnection*) nach [36] dargestellt, wie in der Vermittlungsschicht die Kommunikation zwischen den vermittelnden Netzknoten sowie zwischen Endgerät und vermittelndem Netzknoten verbindungslos erfolgt, während in der Transportschicht eine verbindungsorientierte Kommunikation durchgeführt wird. Dies wird auch als Ende-zu-Ende Verbindung bezeichnet.

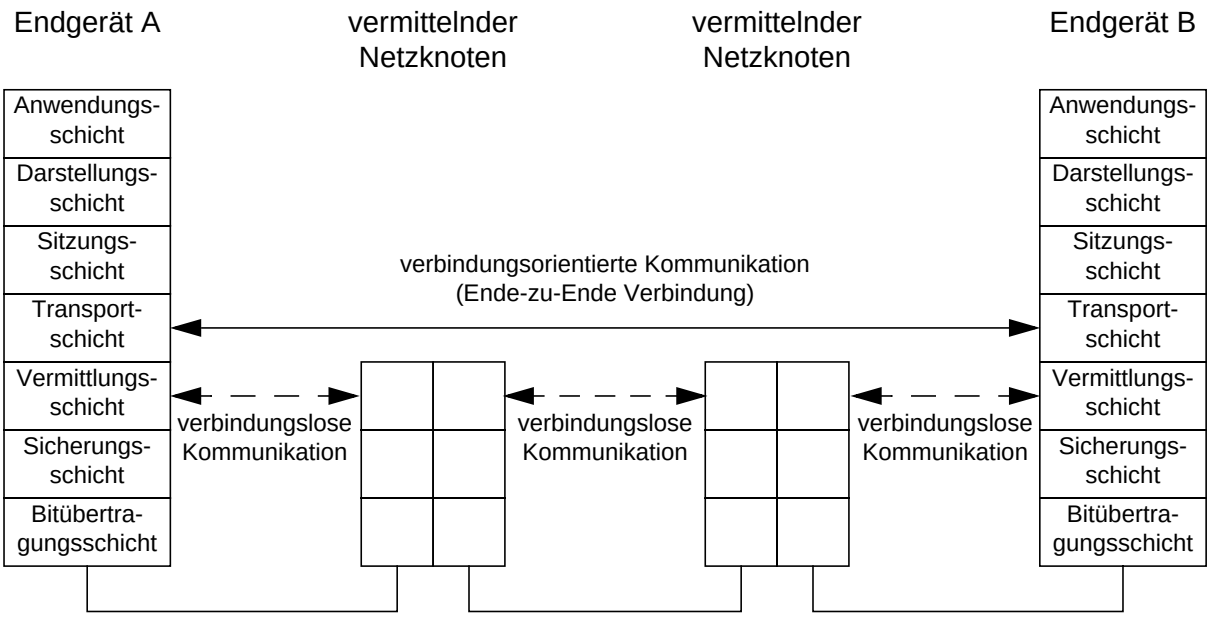


Bild 2.2: Beispiel für verbindungslose und verbindungsorientierte Kommunikation im OSI-Referenzmodell

Verbindungslose Kommunikation

Wenn die Kommunikation in der Vermittlungsschicht verbindungslos erfolgt, werden die Pakete auch als Datagramme bezeichnet. Dabei enthält der Paketkopf die vollständige Information, die benötigt wird, um das Paket vom Sender zum Empfänger zu übertragen. In den jeweiligen Netzknoten wird diese Information ausgewertet, um das Paket in Richtung des Empfängers weiterzuleiten. Da jedes Paket in den Netzknoten als einzelnes Paket betrachtet wird, kann es vorkommen, dass Pakete einer verbindungslosen Kommunikationsbeziehung unterschiedliche Wege durch das Netz zum Empfänger nehmen, so dass die Reihenfolge der empfangenen Pakete nicht unbedingt der Reihenfolge der gesendeten Pakete entspricht. Des Weiteren wird der Verlust von Paketen weder vom Empfänger noch vom Sender erkannt.

Um eine größere Zuverlässigkeit bei der verbindungslosen Kommunikation zu erreichen, kann ein quittierter Datagrammdienst angewendet werden. Dabei wird für jedes empfangene Datagramm eine Quittierung an den Sender als Bestätigung für den korrekten Empfang gesendet.

Verbindungsorientierte Kommunikation

Erfolgt die Kommunikation in der Vermittlungsschicht verbindungsorientiert, wird bei der Datenkommunikation über paketvermittelnde Netze das Prinzip der virtuellen Verbindung zu Grunde gelegt. Dabei wird in der Verbindungsaufbauphase durch den Austausch entsprechender Signalisierpakete ein Weg vom Sender zum Empfänger bestimmt und der virtuellen Verbindung für jeden Übertragungsabschnitt eine eindeutige Kennzeichnung zugeordnet. Es werden jedoch, im Gegensatz zu einer kanalvermittelten Verbindung, keine physikalischen Res-

sourcen der Verbindung exklusiv zugeordnet. Während der Nutzdatenaustauschphase enthalten die Paketköpfe nur die notwendigen Informationen, um die Pakete den entsprechenden virtuellen Verbindungen zuzuordnen. In der Verbindungsabbauphase werden durch den Austausch der entsprechenden Signalisierpakete die die Verbindung betreffenden Daten in den beteiligten Komponenten entfernt. Diese Art der Kommunikation erlaubt die Erkennung von Paketverlusten und ermöglicht die Sicherung der korrekten Reihenfolge der Pakete.

2.1.1.4 Vergleich der Konzepte

Um die Konzepte der Tele- und der Datenkommunikation zu vergleichen, werden zunächst die wichtigsten Merkmale dieser Kommunikationstechniken vorgestellt.

Wie bereits in Abschnitt 2.1.1.2 beschrieben, wird in der Telekommunikation die Kanal- bzw. Durchschaltevermittlung angewandt. Daher sind jeder Kommunikationsbeziehung ein oder mehrere physikalische Kanäle exklusiv zugeordnet, so dass während der gesamten Nutzdatenaustauschphase eine konstante Übertragungskapazität zur Verfügung steht. Dies bewirkt, dass in der Nutzdatenaustauschphase keine Konflikte beim Zugriff auf die übertragungstechnischen Ressourcen auftreten und die Verzögerung vom Senden der Daten bis zu ihrem Empfang beim Zielteilnehmer minimal ist. Des Weiteren ist diese Verzögerung konstant und hängt nicht von der Gesamtbelastung des Netzes ab. Auch die Wahrscheinlichkeit, dass Daten bei der Übertragung verfälscht werden oder verloren gehen, ist sehr gering. Nur bei der Übertragung über eine Luftschnittstelle, wie es z.B. beim Mobilfunk notwendig ist, ergeben sich höhere Verluste, die jedoch durch die physikalischen Randbedingungen hervorgerufen werden und daher unabhängig davon sind, ob es sich um Kanal- oder Paketvermittlung handelt.

Die exklusive Nutzung der Kanäle bringt auch Nachteile mit sich: Während der Kommunikation können Pausen, bei denen keine Daten übertragen werden, nicht für den Datenaustausch anderer Kommunikationsbeziehungen genutzt werden. Pausen können dabei Gesprächspausen oder bereits die Abstände zwischen einzelnen Wörtern bei der Sprachkommunikation darstellen. Dies führt somit zu einer Verschwendung übertragungstechnischer Ressourcen. Des Weiteren kann die Übertragungskapazität der zugeordneten Kanäle auch nicht kurzzeitig überschritten werden.

Durch die Verbindungsorientierung bei der Telekommunikation ist der Aufwand für die Steuerung nicht zu vernachlässigen. Hauptbestandteil der Steuerung ist dabei die Verwaltung und Zuteilung der zur Verfügung stehenden Ressourcen. Für den Verbindungsaufbau beinhaltet dies beispielsweise die Suche nach einem Weg und nach entsprechenden freien Kanälen für die Verbindung. Des Weiteren ist der Steuerungsaufwand für die Realisierung zusätzlicher Dienste zu beachten.

Im Gegensatz zur exklusiven Nutzung der Nutzkanäle werden für die Steuerung Ressourcen gemeinsam verwendet, z. B. für die Auswertung der gewählten Telefonnummer. Des Weiteren

führen verschiedene Komponenten gemeinsam die Verwaltung bestimmter Ressourcen durch, z. B. die Verwaltung der einzelnen Kanäle auf einem Übertragungsabschnitt. Durch die dabei entstehenden Zugriffskonflikte kann der aktuelle Zustand dieser Steuerkomponenten die Steuerung einzelner Verbindungen erheblich beeinflussen. So kann es bei einer bereits sehr stark belasteten Vermittlungsstelle vorkommen, dass die Bearbeitung einer neuen Verbindungsanfrage so stark verzögert wird, dass dies durch den Benutzer bemerkbar ist und als Störung des Dienstes interpretiert werden könnte.

Bei der Datenkommunikation werden durch die Anwendung der Paketvermittlung i. A. weniger übertragungstechnische Ressourcen als bei der kanalvermittelnden Telekommunikation verschwendet, da diese Ressourcen von anderen Kommunikationsbeziehungen verwendet werden können. Dabei ist aber zu beachten, dass durch die Steuerinformation, die jedes Paket in Form des Paketkopfs enthält, zusätzliche zu übertragende Daten entstehen, die die effektive Ausnutzung der Ressourcen verringern.

Da es beim Zugriff auf die gemeinsam genutzten Ressourcen bei paketvermittelnden Netzen zu Konflikten kommen kann, die durch Pufferung der Pakete aufgelöst werden, sind erhebliche Verzögerungen der Pakete möglich, die für den einzelnen Teilnehmer nicht vorhersehbar sind. Darüber hinaus können diese Verzögerungen zwischen den einzelnen Paketen einer Kommunikationsbeziehung stark variieren, da sich die Belegung der einzelnen Ressourcen schnell verändert. Des Weiteren kann es durch die endliche Größe der Puffer zu Paketverlusten kommen, da für zu puffernde Pakete, die auf die Zuteilung einer Ressource warten müssen, kein Platz mehr vorhanden ist.

Die gemeinsame Verwendung der Ressourcen führt somit dazu, dass sowohl die Verzögerung, die Schwankung der Verzögerung und die Verlustwahrscheinlichkeit der einzelnen Pakete als auch die für eine Kommunikationsbeziehung zur Verfügung stehende Übertragungskapazität von der Gesamtbelastung des Netzes abhängt. Dabei verringert i. A. eine neu hinzukommende Kommunikationsbeziehung die Dienstgüte aller anderen, bereits bestehenden Kommunikationsbeziehungen. Diese Effekte können jedoch minimiert werden, indem eine entsprechende Steuerung für das Netz angewandt wird, die jedoch deutlich komplexer als beim kanalvermittelnden Telekommunikationsnetz ist.

Wenn die paketvermittelte Datenkommunikation in der Vermittlungsschicht verbindungsorientiert durchgeführt wird, ist der Steuerungsaufwand für den Verbindungsauf- und -abbau ähnlich groß wie bei der kanalvermittelnden Telekommunikation. Jedoch muss der für eine entsprechende Dienstgüte notwendige Mehraufwand, z. B. für die Verbindungsannahmesteuerung (CAC – *Connection Admission Control*), die entscheidet, ob eine Verbindung zugelassen oder abgelehnt wird, beachtet werden. Darüber hinaus ist im Falle einer Dienstgüteunterstützung auch während der Nutzdatenaustauschphase eine Steuerung des Netzes notwendig, damit der gemeinsame Zugriff auf die Netzressourcen entsprechend geregelt wird.

Ein Vertreter der paketvermittelten, verbindungsorientierten Kommunikation, bei dem die Gewährleistung einer Dienstgüte unterstützt wird, ist der *Asynchronous Transfer Mode* (ATM) [32, 68]. Zur Dienstgüteunterstützung wurden von der ITU-T verschiedene Dienstgüteklassen definiert [49], deren virtuelle Verbindungen entsprechend in den beteiligten Netzknoten behandelt werden.

Wenn die Datenkommunikation in der Vermittlungsschicht verbindungslos durchgeführt wird, fällt kein Steuerungsaufwand für Verbindungsauf- und -abbau in dieser Schicht an. Darüber hinaus müssen keine Zustandsinformationen über einzelne Verbindungen innerhalb des Netzes gehalten werden. Damit skaliert ein derartiges Netz sehr gut mit der Anzahl der Kommunikationsbeziehungen hinsichtlich der Steuerung. Der Nachteil ist jedoch der höhere Steuerungsaufwand pro Paket im Vergleich zu Datenpaketen einer virtuellen Verbindung, da bei der verbindungslosen Kommunikation z. B. für jedes Paket die Suche nach einem Weg in Richtung des Empfängers neu durchgeführt werden muss, während dies bei der verbindungsorientierten Kommunikation nur in der Verbindungsaufbauphase notwendig ist. Beim Ausfall eines Übertragungsabschnitts kann jedoch mit der verbindungslosen Kommunikation schneller reagiert werden, da nahezu kein Steuerungsaufwand benötigt wird, um den ausgefallenen Abschnitt zu umgehen.

Um Dienstgüte in verbindungslosen, paketvermittelnden Netzen zu unterstützen, muss das Netz entsprechend gesteuert werden. Auf diese Fragestellung wird in Abschnitt 2.2 etwas genauer eingegangen.

Der wichtigste Vertreter der verbindungslosen, paketvermittelten Datenkommunikation ist das Internet, das in Abschnitt 2.1.2 vorgestellt wird.

Im Folgenden werden schließlich die Konzepte der Tele- und der Datenkommunikation bezüglich einiger relevanten Kriterien verglichen.

- Dienstgüte
  Die Dienstgüte ist bei der Telekommunikation durch die exklusive Nutzung der übertragungstechnischen Ressourcen im Vergleich zur heutigen paketvermittelten Datenkommunikation sehr hoch. Um eine ähnliche Dienstgüte zumindest annähernd zu erreichen, ist daher bei der paketvermittelnden Datenkommunikation ein entsprechend hoher Steuerungsaufwand notwendig.

- Ressourcenausnutzung
  Durch die gemeinsame Verwendung der übertragungstechnischen Ressourcen kann bei der paketvermittelten Datenkommunikation eine höhere Ausnutzung der zur Verfügung stehenden Ressourcen erzielt werden. Dabei müssen die zusätzlich zur Nutzlast zu übertragenden Paketköpfe beachtet werden, die bei kurzen Paketen, wie sie bei der paketorientierten

Sprachübertragung vorkommen, relevant werden können. Darüber hinaus ist der Ressourcenaufwand für die notwendigen Paketpuffer nicht zu vernachlässigen.

- Steuerungsaufwand
  Der Steuerungsaufwand bei der paketvermittelten, in der Vermittlungsschicht verbindungsorientierten Datenkommunikation hängt stark von der, falls vorhandenen, Dienstgüteunterstützung ab. Ansonsten ist der Aufwand für die Verbindungssteuerung vergleichbar mit dem der Telekommunikation. Bei der verbindungslosen Datenkommunikation ist der Steuerungsaufwand sehr gering, da keine Verbindungen verwaltet werden müssen, sondern nur einzelne Pakete weitergeleitet werden. Insbesondere wenn nur wenige Pakete in einer Kommunikationsbeziehung zu übertragen sind, ist die Steuerung einzelner Pakete effektiver als wenn zunächst eine Verbindung aufgebaut und dann wieder abgebaut werden muss.

- Skalierbarkeit
  Am besten skaliert das Konzept der paketvermittelten, in der Vermittlungsschicht verbindungslosen Datenkommunikation, da keine Kommunikationsbeziehungs-spezifischen Informationen innerhalb des Netzes gehalten werden müssen, wobei die durch den Einzelnen erfahrene Dienstgüte von der aktuellen Belastung des Netzes abhängt. Damit kann die Anzahl der Teilnehmer des Netzes einfach vergrößert werden, jedoch kann die Dienstgüte dementsprechend schlechter werden. Das virtuelle Verbindungsprinzip der paketvermittelten, verbindungsorientierten Datenkommunikation skaliert etwas besser, als die kanalvermittelnde Telekommunikation, da die zur Verfügung stehenden Ressourcen besser ausgenutzt werden, jedoch hängt dies von dem angewandten Verfahren zur Dienstgüteunterstützung ab.

- Erweiterbarkeit und Flexibilität
  Dieser Punkt bezieht sich auf die Integration neuer Dienste, die Erweiterung bestehender Funktionalitäten sowie auf die Flexibilität bezüglich der unterstützten Dienste innerhalb eines Kommunikationsnetzes.

  - Da bei der paketvermittelten, in der Vermittlungsschicht verbindungslosen Datenkommunikation keine Teilnehmer- oder verbindungsspezifischen Daten innerhalb des Netzes gehalten werden, können neue Dienste einfach integriert werden. Dabei müssen die Endgeräte, die diese Dienste verwenden sollen, sie auch entsprechend unterstützen. Jedoch müssen nicht alle am Netz angeschlossenen Endgeräte diese Erweiterungen anwenden können. Dies ist eine der Hauptursachen des Erfolges des Internets: Neue Dienste können einfach integriert werden, da nur die beteiligten Endgeräte in der Lage sein müssen, diese zu unterstützen – das Netz selbst ist davon nicht betroffen.

  - Um das Dienstespektrum bei der verbindungsorientierten Kommunikation zu erweitern, muss die Steuerung des Netzes diese Erweiterungen entsprechend unterstützen. Bei der Telekommunikation wird dabei das Prinzip angewandt, dass die Endpunkte sehr einfach

gehalten werden und die Dienstbearbeitung im Netz durchgeführt wird. Damit müssen Erweiterungen im gesamten Netz durchgeführt werden, wobei nahezu alle Endgeräte diese dann anwenden können.

- Da bei paketvermittelnden Datenkommunikationsnetzen keine Kanäle konstanter Übertragungskapazität angewendet werden, können Dienste mit unterschiedlichem, sogar variablem Bedarf an Übertragungskapazität integriert werden.

2.1.2 Internet

Dieser Abschnitt stellt den wichtigsten Vertreter der Datenkommunikationsnetze, die in der Vermittlungsschicht verbindungslos arbeiten, das Internet, vor. Dazu wird in Abschnitt 2.1.2.1 seine Entstehungsgeschichte und seine Entwicklung zu einem Dienste-integrierenden Netz kurz beschrieben, anschließend werden in Abschnitt 2.1.2.2 seine Architektur sowie die wichtigsten angewendeten Protokolle erläutert.

2.1.2.1 Entwicklungsgeschichte

Die ARPA[1] (*Advanced Research Projects Agency*), eine Organisation des Verteidigungsministeriums der USA, die die Forschung der Universitäten und des Militärs fördert und koordiniert, nahm 1969 das sog. ARPANET mit vier Knoten in Betrieb. Dieses paketvermittelnde Netz erlaubte einerseits den Zugriff auf entfernte Rechnerressourcen, war andererseits auch selbst Forschungsgegenstand vieler Untersuchungen.

1982 wurde der Begriff Internet geprägt und 1983 wurde das komplette Netz auf das *Internet Protocol* (IP) [83] umgestellt. Wie der Name bereits Nahe legt, erlaubt IP die Verbindung verschiedener Netztechnologien, solange sie das *Internet Protocol* unterstützen. Eine genauere Beschreibung der Internet-Architektur mit seinen wichtigsten Protokollen erfolgt in Abschnitt 2.1.2.2.

Mit der Einführung des *World Wide Web* (WWW) 1991 und der Verfügbarkeit von entsprechenden Anwendungen für den Zugriff auf die Inhalte des WWW (sog. *Browser*), z. B. *Mosaic* 1993, wurde das Internet der breiten Öffentlichkeit bekannt. Neben dem Zugriff auf Texte und Bilder spielten zunehmende multimediale Daten eine Rolle. Dabei wurden aber auch die Nachteile des Internets immer deutlicher: Durch das Prinzip der verbindungslosen Paketvermittlung und der nicht oder kaum vorhandenen Dienstgüteunterstützung konnte auf bestimmte, sehr beliebte Inhalte manchmal gar nicht oder nur sehr langsam zugegriffen werden.

In den Firmen entstanden eigene, IP-basierte Kommunikationsnetze, die von außen in der Regel nur beschränkt oder gar nicht erreichbar waren. Ein derartiges privates Netz wird als

[1] Mittlerweile wurde diese Organisation in DARPA (*Defense Advanced Research Projects Agency*) umbenannt.

Intranet bezeichnet, wobei es sich über viele Standorte eines Unternehmens erstrecken kann. Wie in den privaten Telefonnetzen werden in diesen Intranets Innovationen schneller eingeführt als in den öffentlichen Netzen, da diese Netze eine kleinere Ausbreitung als das globale Internet besitzen und in der Regel zentral gewartet werden. Dies führt zum schnelleren Einsatz neuer Technologien, wie z. B. zur Dienstgüteunterstützung.

Die Übertragung von Telefongesprächen über das Internet wurde nach [72] bereits in den frühen Phasen angedacht. Jedoch kam es erst in der Mitte der 90er Jahre zu den ersten Produkten [118], die von Internet-Nutzern angewendet wurden, um das teurere Telefonnetz bei Ferngesprächen zu umgehen. Dabei spielte die Dienstgüte kaum eine Rolle, da man sehr wenig für diesen Dienst bezahlte. Diese ersten Internet-Telefonie-Produkte waren proprietär, d. h. dass Kommunikation nur zwischen gleichen Produkten möglich war. Dies änderte sich durch die 1996 einsetzende Standardisierung in diesem Bereich.

2.1.2.2 Architektur und Protokolle

Im Folgenden wird zunächst die dem Internet zu Grunde liegende Architektur vorgestellt, bevor das Schichtenmodell sowie die wichtigsten Protokolle des Internet präsentiert werden. Eine ausführliche Beschreibung ist z. B. in [22] enthalten.

Das Internet verbindet paketvermittelnde Netze mittels sog. *Router*, wobei die einzelnen Netze unterschiedliche Übertragungstechnologien verwenden können. Damit entsteht, wie in Bild 2.3 dargestellt, aus verschiedenen Einzelnetzen ein grosses, zusammenhängendes Netz.

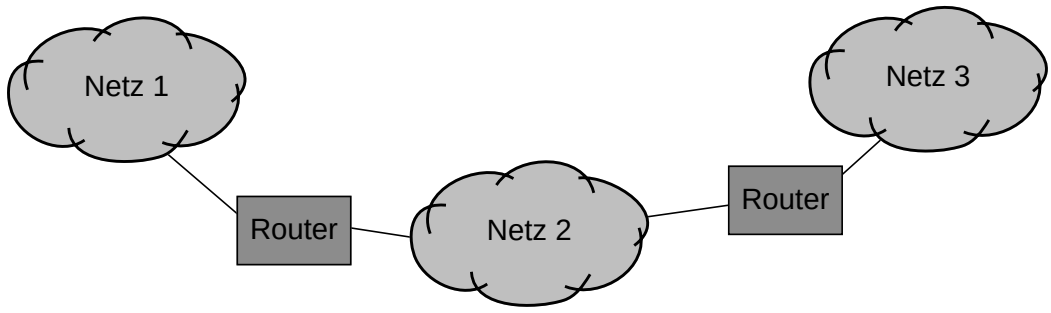


Bild 2.3: Verbindung unterschiedlicher Netze mittels Router

Das Schichtenmodell der Internet-Architektur unterscheidet sich etwas vom OSI-Referenzmodell. In Bild 2.4 sind die prinzipiellen Unterschiede dargestellt:

- Die Netz- bzw. Medienzugriffsschicht (*Network Access*) regelt den Zugriff auf das physikalische Übertragungsmedium und wird daher auch als MAC-Schicht (*Media Access Control*) bezeichnet. Zusammen mit der Bitübertragungsschicht stellt sie den technologieabhängigen Teil des Schichtenmodells dar. In lokalen Netzen (LAN – *Local Area Network*) sind sie meist durch das *Ethernet* realisiert, in Weitverkehrsnetzen (WAN – *Wide Area Network*) könnte beispielsweise ATM, SDH (*Synchronous Digital Hierarchy*) oder auch WDM

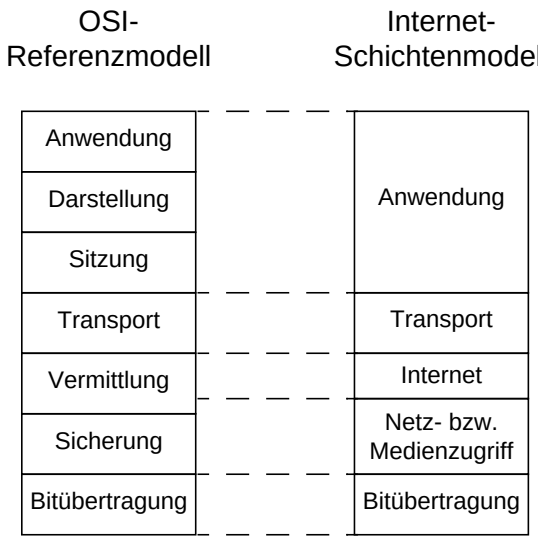


Bild 2.4: Vergleich zwischen OSI-Referenzmodell und Internet-Schichtenmodell

(*Wavelength Division Multiplex*) angewendet werden. Für Anwender, die über das klassische Telefonnetz mittels eines Modems auf das Internet zugreifen, wird häufig das PPP (*Point to Point Protocol*) verwendet.

- In der Internet-Schicht wird als Protokoll das IP (*Internet Protocol*) verwendet, das als gemeinsame Basis für alle darüberliegenden Schichten dient und somit die darunterliegenden Übertragungs- und Netztechnologien abstrahiert. Die Internet-Schicht wird auch als IP-Schicht bezeichnet. Im weiteren Verlauf der Arbeit werden diese beiden Begriffe synonym verwendet.

- Die relevanten Protokolle der Transportschicht des Internet sind TCP (*Transmission Control Protocol*, [84]), das eine Ende-zu-Ende Verbindung zwischen zwei am Internet angeschlossenen Kommunikationsendpunkten ermöglicht, und UDP (*User Datagram Protocol*, [85]), das verbindungslos arbeitet und im Prinzip nur eine Schnittstelle zur Internet-Schicht realisiert.

- Die Anwendungen des Internet setzen in der Regel direkt auf der Transportschicht auf. Beispiele für Anwendungsprotokolle sind HTTP (*Hypertext Transfer Protocol*, [26]), FTP (*File Transfer Protocol*, [86]) und RTP (*Real Time Transport Protocol*, [95]), das für den Austausch von Echtzeitdaten, wie sie bei der Sprachkommunikation anfallen, vorgesehen ist.

Die zentrale Schicht des Internet-Schichtenmodells ist die IP-Schicht. Sie vereinheitlicht die Verwendung unterschiedlicher Netztechnologien der darunterliegenden Schichten. Die darüberliegenden Anwendungen können dann unabhängig von der zur Verfügung stehenden Netztechnologie realisiert werden, indem sie eine entsprechende Schnittstelle zur IP-Schicht bzw. zu TCP/UDP besitzen. Dieses Prinzip führte zu einer breiten Unterstützung von verschiedenen Netztechnologien und zu einer Vielzahl IP-basierter Anwendungen, was durch das sog. Sand-

uhrmodell in Bild 2.5 verdeutlicht wird. Eine ausführlichere Beschreibung der wichtigsten dieser Anwendungen kann z. B. in [17, 22] gefunden werden.

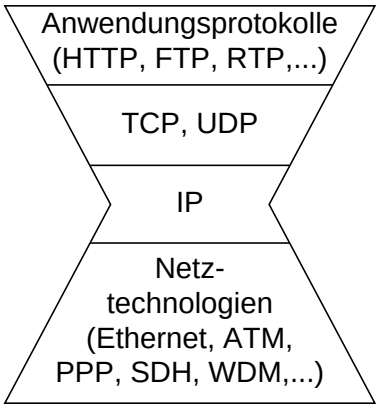


Bild 2.5: Sanduhrmodell der Internet-Schichten

Die Router des Internet arbeiten in der IP-Schicht, d. h. dass sie nur Informationen bis zu dieser Schicht, in der Regel die IP-Adresse, auswerten, um die Pakete in Richtung des Empfängers weiterzuleiten. Daher sind bei neuen Anwendungen, die auf der IP-Schicht oder den darüberliegenden Transportschichten TCP und UDP basieren, keine Erweiterungen in den weiterleitenden Knoten notwendig, was zu der großen Innovationskraft des Internet führte.

Derzeit wird im Internet die Version 4 von IP verwendet, wobei die nächste relevante Version, Version 6 (IPv6, [24]) standardisiert ist und in einigen Testnetzen angewendet wird. Anders als bei der Einführung von IP im Internet 1983 wird die Umstellung auf IPv6 sukzessive erfolgen, da mittlerweile Millionen von Knoten davon betroffen sind, und außerdem eine große Anzahl von Anwendungen angepasst werden muss.

TCP erweitert die Funktionalität der IP-Schicht um eine Ende-zu-Ende Verbindung. Damit steht den Anwendungen ein zuverlässiger Kommunikationsdienst zur Verfügung. Dazu führt TCP Maßnahmen zur Fehlererkennung und -behebung, zur Reihenfolgesicherung, zur Flusssteuerung sowie zur Verbindungssteuerung durch.

Der Verbindungsaufbau bei TCP erfolgt in drei Schritten (*Three-Way-Handshake*) wie in Bild 2.6 dargestellt ist. Erst nach dem Senden der ACK-PDU (*Protocol Data Unit*) beginnt der Vollduplex-Datenaustausch, d. h. dass beide Seiten gleichzeitig Daten empfangen und senden können.

Die Flusssteuerung von TCP passt die Senderate der Pakete einerseits an die Rate, mit der der Empfänger die Pakete verarbeiten kann (Empfangsrate), und andererseits an die aktuelle Lastsituation im Netz an. Dies wird dadurch erreicht, dass zu Beginn der Verbindung mit einer kleinen Senderate, realisiert durch eine entsprechend beschränkte Sende-Fenstergröße von nicht bestätigten Paketen, gesendet wird. Diese Rate wird im weiteren Verlauf der Verbindung

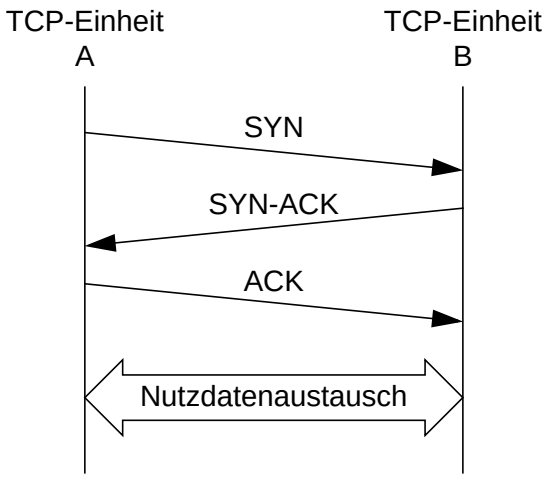


Bild 2.6: Verbindungsaufbau bei TCP von Einheit A nach B

immer weiter erhöht, bis es zu vermeintlichen Paketverlusten kommt. Diese werden durch Zeitüberwachungen (im weiteren Verlauf als *Timer* bezeichnet) der Bestätigungspakete erkannt. Daraufhin wird die maximale Sende-Fenstergröße und damit auch die Senderate halbiert. Anschließend wird die Senderate wieder langsam erhöht.

Dieses Verfahren hat den Vorteil, dass sich TCP sehr gut an die Lastsituation des Netzes anpasst. Dabei beeinflussen jedoch neu hinzukommende Verbindungen bereits bestehende und verringern deren Dienstgüte. Dies hat zur Folge, dass die Übertragungsrate im Laufe einer Verbindung erheblich schwanken kann. Die Fehlerbehebung bei TCP erfolgt durch wiederholtes Senden der verloren gegangenen Pakete. Da dies Ende-zu-Ende durchgeführt wird, ist die Zeitdauer, bis ein wiederholtes Paket schließlich beim Empfänger ankommt, für Echtzeitanwendungen, bei denen bestimmte Antwortzeiten eingehalten werden müssen, meist zu lang. Aus diesen Gründen wird in der Regel TCP für derartige Anwendungen nicht verwendet.

UDP stellt den höheren Schichten einen verbindungslosen, ungesicherten Kommunikationsdienst zur Verfügung. Dies bedeutet, dass diese höheren Schichten mit Paketverlusten, Reihenfolgeänderungen, Paketduplizierungen und dem Verlust der Kommunikationsfähigkeit umgehen können müssen. Da die Kommunikation jedoch verbindungslos durchgeführt wird, ist kein Aufwand für die Verbindungssteuerung notwendig, insbesondere fällt keine Verbindungsaufbauverzögerung an, so dass die Daten ohne Verzug übertragen werden können. Des Weiteren ist für bestimmte Anwendungen die rechtzeitige Ankunft eines Paketes notwendig. So ist beispielsweise bei der Sprachkommunikation ein verspätetes Paket, wie es bei Wiederholungen von verlorenen Paketen vorkommen kann, ebenso wertlos, wie ein Paket, das nie ankommt. Daher verwendet das für die Echtzeitkommunikation vorgesehene RTP, das in Abschnitt 2.2.1 vorgestellt wird, UDP als Transportprotokoll.

Sowohl TCP als auch UDP verwenden für die Adressierung neben den IP-Adressen der Internet-Schicht sog. *Ports*, die über Nummern identifiziert werden. Dabei existieren Ports, die

explizit bestimmten Diensten zugeordnet sind [87]. Somit ist eine Kommunikationsbeziehung in der Transportschicht eindeutig durch Sender-IP-Adresse, Sender-Portnummer, Empfänger-IP-Adresse und Empfänger-Portnummer gekennzeichnet.

2.1.3 Konvergenz

Unter dem Begriff der Konvergenz wird in der Kommunikationstechnik das Zusammenwachsen unterschiedlicher Kommunikationsprinzipien verstanden. Im Rahmen dieser Arbeit ist insbesondere die Konvergenz der Tele- und Datenkommunikation von Interesse. Eng mit dem Begriff der Konvergenz ist dabei die Diensteintegration verbunden, bei der es um das Einbinden unterschiedlicher Kommunikationsdienste in einem Netz geht.

Mit der Konvergenz der Netze sind mehrere Ziele verbunden. Der zentrale Punkt ist sicherlich die Vereinheitlichung der Netzinfrastruktur für die verschiedenen Kommunikationsdienste. Dies würde enorme Kosteneinsparungen mit sich bringen, da nur eine Netztechnologie gewartet und weiterentwickelt werden müsste. Des Weiteren wären für den Endteilnehmer weniger verschiedene Zugangstechnologien notwendig, was für eine größere Transparenz der Kommunikationsdienste für den Teilnehmer sorgen würde. Schließlich erhofft man sich durch das Zusammenwachsen der Tele- und der Datenkommunikation eine Vielzahl neuer Dienste, die bisher nicht oder nur mit relativ hohem Aufwand realisiert werden konnten. Als Beispiel seien dazu die unter dem Stichwort CTI (*Computer Telephony Integration*) zusammengefassten Dienste zu nennen. Diese werden in sog. *Call Center* angewandt, die die Schnittstelle zum Kunden für viele Unternehmen, z. B. im Banken- und Versicherungssektor, erweitern. Sie erlauben beispielsweise, dass beim Anruf eines Kunden alle wichtigen Kundendaten dem entsprechenden Kundenberater sofort zur Verfügung stehen, so dass dieser den Kunden schnell und kompetent beraten kann. Ein weiteres Beispiel sind Videokonferenzen, bei denen gemeinsam eine Anwendung von mehreren Konferenzteilnehmern verwendet wird (*Application Sharing*), um z. B. gemeinsam Skizzen zu entwerfen.

Bis zur Einführung des ISDN (*Integrated Services Digital Network*) ab 1988 waren die Kommunikationsnetze Dienste-spezifisch, d. h. für nahezu jeden Dienst gab es ein spezielles Netz, z. B. das Telefonnetz, das Kabelfernsehnetz oder das DATEX-P Paket-Datenübertragungsnetz. Mit dem ISDN ist es möglich, unterschiedliche Dienste mit einem gemeinsamen Netz zu unterstützen und mit einem gemeinsamen Verfahren, der ISDN-Signalisierung, zu steuern. Die Möglichkeiten des ISDN bezüglich der Integration von Kommunikationsdiensten werden wenig ausgenutzt: Der dominierende Dienst im ISDN ist die Telefonie. Jedoch wurden mit dem ISDN und dem 1998 in das Telefonnetz integrierten IN (*Intelligent Network*) zusätzliche Dienste eingeführt, die den Basisdienst Telefonie wesentlich erweiterten. Das B-ISDN (*Broadband-ISDN*) auf Basis der ATM-Technologie sollte diese Form der Diensteintegration

weiter fortführen [107], jedoch wurde es durch den Erfolg und die rasante Entwicklung des Internet etwas in den Hintergrund gedrängt.

Durch die einfache Technologie innerhalb des Netzes und die damit einhergehende Innovationskraft des Internet werden zunehmend mehr Dienste auf Basis IP-basierter Netze realisiert. Selbst die echtzeitkritische Sprachkommunikation soll in Zukunft mit gegenüber dem Telefonnetz vergleichbarer Qualität über diese Netze geführt werden. Gerade in diesem Bereich zeigen sich verstärkt Standardisierungsaktivitäten, wobei sowohl die ITU als Standardisierungsgremium für die Telekommunikation als auch die IETF (*Internet Engineering Task Force*) als Gremium für die Internet-basierte Datenkommunikation sehr aktiv sind. Des Weiteren verfügt nahezu jeder namhafte Hersteller von Telekommunikationssystemen mittlerweile über VoIP-Lösungen. Dies zeigt, dass die Konvergenz der Tele- und der Datenkommunikation über IP-basierte Netze führen wird.

2.2 Nutzdatenaustausch

In diesem Abschnitt wird auf den Nutzdatenaustausch, damit verbundene Probleme und einige dazugehörige Lösungsmöglichkeiten eingegangen. Dazu wird in 2.2.1 zunächst der Austausch von Nutzdaten mittels RTP vorgestellt. Bevor die Sprach- und Videodaten übertragen werden, müssen sie entsprechend codiert und beim Empfänger decodiert werden. Die dabei angewandten *Codec* (*Coder/Decoder*) und ihre Eigenschaften werden in Abschnitt 2.2.2 beschrieben. In Abschnitt 2.2.3 wird schließlich auf die Problematik der Dienstgüteunterstützung beim Nutzdatenaustausch eingegangen. Es werden die wichtigsten existierenden Verfahren kurz vorgestellt.

2.2.1 Transport

Beim Transport der Nutzdaten für VoIP-Dienste muss zunächst unterschieden werden, welchem Dienst diese Daten zuzuordnen sind. Für Datenapplikationen, wie z. B. *Application Sharing* oder Telefax, ist eine fehlerfreie Übertragung sehr wichtig, während die Verzögerung keine große Rolle spielt. Daher wird für diese Dienste in der Regel TCP als Transportprotokoll verwendet.

Im Gegensatz zu den Datenapplikationen ist die Verzögerung bei Sprach- und Videodaten ein sehr wichtiger Faktor, da bei zu großer Verzögerung die Kommunikation sehr gestört oder gar nicht möglich ist. Dagegen beeinflusst der Verlust einzelner Pakete die Dienstgüte nicht wesentlich. Da bei Ende-zu-Ende Paketwiederholungen die daraus resultierende Verzögerung zu groß wäre, wird für diese Dienste UDP verwendet. Um die für diese Echtzeitdienste notwendigen Protokollfunktionen allgemein zur Verfügung zu stellen, wurde von der IETF das

RTP definiert [95], das nahezu unabhängig vom darunterliegenden Transportprotokoll ist und in IP-Umgebungen UDP verwendet.

Zur Unterstützung der Echtzeitkommunikation verfügt RTP über folgende Funktionalitäten:

- Zeitstempel
  Jedes RTP-Paket wird beim Erzeugen mit einem Zeitstempel versehen, so dass der Empfänger die einzelnen Pakete synchronisiert an die Anwendung weitergeben kann. Damit ist sowohl *Intra-Media-Synchronisation*, zur Synchronisation der Daten innerhalb eines Mediums, als auch *Inter-Media-Synchronisation*, wie sie z. B. bei der Lippensynchronisation für Sprach- und Videodaten notwendig ist, möglich.

- Reihenfolgenummern
  Da die einzelnen Pakete unterschiedliche Wege durch das Netz nehmen können, erlauben die Reihenfolgenummern die Wiederherstellung der ursprünglichen Reihenfolge. Des Weiteren können Paketverluste erkannt werden, wobei aber RTP keine Paketwiederholungen initiiert.

- Profile
  RTP erlaubt die Definition von Profilen für die übertragenen Nutzdaten. Damit können für unterschiedliche Nutzdatentypen verschiedene Prozeduren festgelegt werden, die für die Interpretation der Daten angewendet werden sollen.

- Unterstützung der Umcodierung
  Wenn die Nutzdaten während der Übertragung umcodiert werden müssen, erfolgt dies durch einen sog. *Translator*. Dies kann z. B. in einer Konferenz notwendig sein, wenn nicht alle Teilnehmer die gleiche Codierung der Nutzdaten durchführen können.

- Unterstützung des Mischens
  RTP unterstützt das Mischen von Nutzdaten, indem angegeben wird, welche Komponente die Daten gemischt hat und welche Komponenten Daten dazu beigetragen haben. Dies ist bei Konferenzen notwendig, wenn z. B. die Sprachsignale mehrerer Teilnehmer zu einem gemeinsamen Sprachsignal zusammengeführt werden.

RTP garantiert weder eine bestimmte Dienstgüte, noch werden Ressourcen für die Übertragung der Daten reserviert. Dies ist außerhalb des Bereichs von RTP.

Dagegen unterstützt RTP mit dem ebenfalls in [95] definierten Steuerungsprotokoll RTCP (*RTP Control Protocol*) die Überwachung bestimmter Parameter des Datenaustauschs. Dazu senden die beteiligten Endpunkte periodisch jeweils Sender- und Empfängerberichte (*Sender Report* und *Receiver Report*), aus denen die Sende- und Empfangsrate, Paketverlustrate, Verzögerung und Variation der Verzögerung abgeleitet werden können. Es ist aber Aufgabe der darüberliegenden Anwendung, diese Angaben auszuwerten und entsprechend zu reagieren.

2.2.2 Codierung

Bevor die zu übertragenden Daten mittels RTP transportiert werden können, müssen sie zunächst entsprechend codiert und in Pakete verpackt werden. Beim Empfänger werden die Nutzdaten aus den Paketen entnommen und anschließend decodiert, so dass sie dem Empfänger im gewünschten Format zur Verfügung gestellt werden können. Die dafür angewandten Verfahren werden als *Codec* (*Coder/Decoder*) bezeichnet. Im Folgenden werden einige wichtige Codec für die Sprach- und die Videocodierung sowie ihre Eigenschaften vorgestellt.

In Bild 2.7 ist der prinzipielle Ablauf der Audioübertragung vom Mikrofon des Sprechenden bis zum Lautsprecher des Hörenden dargestellt. Zunächst wird das analoge Sprachsignal in entsprechenden Abständen abgetastet und mittels eines A/D-Wandlers (A/D – *Analog/Digital*) in ein digitales Signal umgewandelt. Anschließend wird dieses abgetastete Digitalsignal entsprechend dem verwendeten Codec codiert. Nach der Übertragung des resultierenden Paketes wird es zunächst in den sog. *Jitter-Buffer* geschrieben. Der Jitter-Buffer dient dem Ausgleich von Laufzeitschwankungen, so dass die Pakete in den gleichen Zeitabständen weitergegeben werden, wie sie gesendet wurden. Dabei ist aber zu beachten, dass die Ende-zu-Ende Verzögerung nicht zu groß wird. Darauf wird noch in Abschnitt 2.2.3 eingegangen. Anschließend werden die Sprachdaten wieder decodiert und schließlich in ein Analog-Sprachsignal umgewandelt, das z. B. über den Lautsprecher ausgegeben wird.

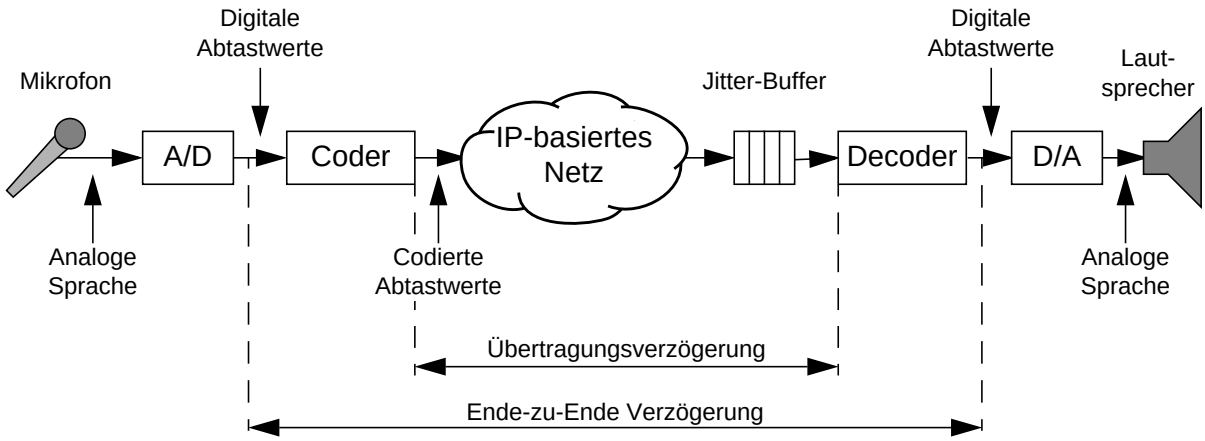


Bild 2.7: Prinzipieller Ablauf der Sprachübertragung über IP-basierte Netze

In Tabelle 2.2 sind die wichtigsten Sprach-Codec, die von der ITU-T spezifiziert wurden, und ihre relevanten Eigenschaften dargestellt:

- Bitrate
  Eine wichtige Aufgabe der Codec ist die Komprimierung der Daten, die sich in der benötigten Bitrate ausdrückt. Bei der dargestellten Bitrate handelt es sich um die Netto-Bitrate, d. h. die Steuerungsinformationen in den Köpfen der Pakete sind nicht berücksichtigt.

Codec	Bitrate [kbit/s]	Komplexität im Vergleich zu G.726	Algorithmische Verzögerung [ms]	Qualität [MOS]
G.711	64	sehr gering	0.125	4.0
G.723.1	5.3	8	37.5	3.6
G.723.1	6.3	8	37.5	3.9
G.726	32	1	0.125	3.85
G.728	16	15	0.625	3.61
G.729	8	10	15	3.9
G.729A	8	6	15	3.7

Tabelle 2.2: Vergleich einiger Sprach-Codec (aus [70])

- Komplexität
  Die Anzahl von MIPS (*Millions of Instructions per Second*) für Codier- und Decodieralgorithmen legt die Komplexität eines Codec fest. In Tabelle 2.2 werden die entsprechenden Werte der einzelnen Codec mit dem von G.726 ins Verhältnis gesetzt.

- Algorithmische Verzögerung
  Die durch Codierung und Decodierung hervorgerufene Verzögerung setzt sich aus der algorithmischen Verzögerung und der Verarbeitungsverzögerung (*Processing Delay*) zusammen. Die Verarbeitungsverzögerung entsteht durch den Vorgang des Codierens und Decodierens und hängt somit von der Komplexität des Codec und von der Leistungsfähigkeit der verwendeten Systemplattform ab. Die algorithmische Verzögerung dagegen ist konstant für einen Codec. Sie setzt sich aus der Rahmengröße, d. h. der Anzahl der enthaltenen Abtastwerte, und ggf. der Anzahl der Abtastwerte, die vorausschauend in den Rahmen für die Komprimierung einbezogen werden, zusammen. Beispielsweise enthält ein Rahmen bei G.723.1 Sprachdaten für 30 ms, d. h. es sind 240 Abtastwerte enthalten, und es wird für 7.5 ms vorausschauend auf weitere Abtastwerte zugegriffen, um einen G.723.1-Rahmen zu codieren. G.711 dagegen enthält nur einen Abtastwert pro Rahmen und benötigt keinen Vorgriff auf weitere Abtastwerte zur Codierung des Rahmens.

- Qualität
  Die Qualität der Sprache wird mit dem von der ITU-T in [50] beschriebenen MOS (*Mean Opinion Score*) gemessen. Diese subjektive Bewertung vergleicht die Sprachqualität eines Codec mit der anderer bekannter Codec. Dabei bedeutet der Wert 1 schlecht und 5 exzellent. Der MOS-Wert von G.711, dem Codec, der in der kanalvermittelnden Telekommunikation angewandt wird, ist 4.0.

Aus Tabelle 2.2 wird deutlich, dass eine hohe Sprachqualität entweder durch eine hohe Bitrate, bei entsprechend niedriger Verzögerung (G.711, G.726), oder durch eine größere Verzögerung,

die sowohl aus der Komplexität als auch aus dem Algorithmus der Rahmenerzeugung resultiert (G.723.1, G.729), erreicht werden kann. Dabei ist aber zu beachten, dass die durch den Codec hervorgerufene Verzögerung zur Ende-zu-Ende Verzögerung beiträgt, die ein wesentliches Kriterium der Dienstgüte bei der Sprachübertragung darstellt.

Die Videocodierung für Echtzeitkommunikation erfolgt in der Regel mit den von der ITU-T standardisierten Codec nach den Empfehlungen H.261 und H.263:

- H.261
  Die Codierung nach H.261 wurde für kleine Codierungsverzögerungen und kleine Bitraten entworfen. Die zur Verfügung stehenden Bitraten sind Vielfache von 64 kbit/s im Bereich von 64 kbit/s bis zu 1.92 Mbit/s.

- H.263
  Die Codierung nach H.263 erreicht eine höhere Bildqualität als die nach H.261. Dabei können unterschiedliche Bitraten eingestellt werden. Eine interessante Eigenschaft ist der sog. *Bit-Stream Scalability Mode*, der die zu übertragenden Videodaten in verschiedene Schichten aufteilt, wobei die niedrigste Schicht ein Basisbild mit niedriger Qualität darstellt und die weiteren Schichten dieses Basisbild zu einem Bild mit entsprechend höherer Qualität ergänzen. Damit können Empfänger ein ihrer Empfangsrate angepasstes Bild empfangen, ohne dass der Sender sich auf diesen Empfänger einstellen muss. Dies ist insbesondere für Video-Konferenzen interessant, bei denen nicht alle Teilnehmer über die gleichen Empfangsraten verfügen, da z. B. einzelne Teilnehmer nur über ein Modem am Internet angeschlossen sind.

Eine genauere Beschreibung der Videocodierung kann in [70] und vor allem in [92] gefunden werden.

2.2.3 Dienstgüteunterstützung

In Abschnitt 2.1.1.4 wurde erwähnt, dass für paketvermittelnde Kommunikationsnetze, die in der Vermittlungsschicht verbindungslos arbeiten, Verfahren existieren, die eine Einhaltung einer bestimmten Dienstgüte unterstützen. In diesem Abschnitt werden zunächst in Abschnitt 2.2.3.1 die spezifischen Anforderungen und Randbedingungen bezüglich der Dienstgüte bei der Nutzdatenübertragung für VoIP vorgestellt, bevor in Abschnitt 2.2.3.2 die wichtigsten Verfahren zur Dienstgüteunterstützung beschrieben werden.

2.2.3.1 Anforderungen und Randbedingungen

Für die einzelnen Medien einer VoIP-Kommunikation existieren unterschiedliche Anforderungen an die Dienstgüte. Wie bereits in 2.2.1 erwähnt, benötigen Datenapplikationen eine äußerst

geringe Datenverlustwahrscheinlichkeit, was durch die Verwendung eines zuverlässigen Transportprotokolls wie TCP sichergestellt werden kann.

Der Austausch von Videodaten erlaubt keine großen Verzögerungen und die Verlustraten dürfen ebenfalls nicht sehr hoch werden. Ansonsten ergeben sich erhebliche Störungen des Bildes. Dies schränkt zwar die Kommunikation ein, da empfangene Videobilder entweder veraltet sind und damit nicht zu den empfangenen Sprachdaten passen oder das Bild nicht kontinuierlich erscheint, jedoch ist die Kommunikation trotzdem möglich, wenn die Sprachkommunikation zur Verfügung steht.

Die härtesten Anforderungen bezüglich der Verzögerung stellt die Sprachübertragung. Die ITU-T empfiehlt in [39], dass eine maximale Ende-zu-Ende Verzögerung, d. h. die Zeit vom Eingang des Coder bis zum Ausgang des Decoder (siehe Bild 2.7), von 150 ms nicht überschritten werden sollte. In [78] wird angegeben, dass die maximale tolerierbare Verzögerung zwischen 250 und 500 ms liegt, wobei nach [70] bereits bei einer Verzögerung von 300 ms eine Unterhaltung schwierig wird. Bei größeren Verzögerungen ähnelt die Kommunikation einer Halbduplex-Kommunikation (Walkie-Talkie-Effekt).

Im Gegensatz zur Datenkommunikation sind Datenverluste bei der Sprachkommunikation tolerierbar. Wenn einzelne Sprachpakete fehlen, wird dies ein Benutzer kaum bemerken. Erst bei Verlustraten, die größer als 5 % sind, wird nach [78] bei den meisten Codec die Sprachkommunikation erheblich gestört. Eine detailliertere Untersuchung zu den Auswirkungen von Verlusten und Verzögerungen bei der Übertragung von Sprachpaketen kann in [119] gefunden werden.

Da die Einhaltung einer akzeptablen Verzögerung bei der Sprachübertragung die härteste Anforderung an die Dienstgüte in einem Internet-basierten Netz stellt, wird diese im Folgenden etwas genauer betrachtet.

Wie aus Bild 2.7 ersichtlich setzt sich die Sprachverzögerung aus folgenden Komponenten zusammen:

- A/D- und D/A-Wandlung
  Da diese Zeitdauern im Verhältnis zu den weiteren Komponenten verschwindend klein sind, können sie vernachlässigt werden.

- Codec
  Die durch den Codec hervorgerufene Verzögerung setzt sich aus der algorithmischen Verzögerung, die in Tabelle 2.2 enthalten ist, und der Verarbeitungsverzögerung zusammen. Wie bereits erwähnt, hängt die Verarbeitungsverzögerung von der Komplexität des Codec und damit von der verwendeten Systemplattform ab.

- Jitter-Buffer
  Der Jitter-Buffer dient dazu, Laufzeitschwankungen bei der Übertragung der Sprachdaten in paketvermittelnden Netzen auszugleichen, so dass die Sprachdaten in synchronen Abständen an den Decoder weitergegeben werden. Die Synchronisierung erfolgt dabei mittels der Zeitstempel der RTP-Pakete. Bei der Dimensionierung des Jitter-Buffers ist jedoch zu beachten, dass die Ende-zu-Ende Verzögerung den maximal zulässigen Wert nicht überschreitet. Dabei muss die Verzögerung im Netz beachtet werden, die u. a. von der aktuellen Belastung des Netzes abhängt. Daher verwenden einige Applikationen Jitter-Buffer, die sich dynamisch an Änderungen des Netzzustands anpassen.

- Verzögerung im Transportnetz
  Diese Verzögerung setzt sich aus der reinen Übertragungsdauer auf den physikalischen Leitungen und der Warte- und Verarbeitungszeit in den Netzknoten zusammen. Dabei spielt die physikalische Übertragungsdauer, wenn keine Satellitenverbindungen enthalten sind, in der Regel keine Rolle. Wie bereits in Abschnitt 2.1.1.3 beschrieben, werden Konflikte beim Zugriff auf gemeinsame Ressourcen bei paketvermittelnden Netzen durch Pufferung der Anfragen gelöst. Daher kann die Verzögerung in den Netzknoten, insbesondere bei hoher Auslastung des Netzes, erheblich sein.

2.2.3.2 Dienstgüteunterstützende Verfahren

Um die in Abschnitt 2.2.3.1 angesprochenen Verzögerungen im Transportnetz für entsprechende Dienste akzeptabel zu halten, müssen in paketvermittelnden Kommunikationsnetzen Verfahren zur Unterstützung der Dienstgüte angewendet werden. In diesem Bereich wurde und wird von der wissenschaftlichen Gemeinschaft sehr intensiv geforscht. Eine gute Übersicht für dienstgüteunterstützende Verfahren für VoIP-Dienste ist beispielsweise in [6] sowie in [74] enthalten.

Im Internet ohne dienstgüteunterstützende Verfahren werden alle Pakete „so gut wie möglich“ behandelt. Man bezeichnet dies als *Best Effort* Dienst. Damit bestimmte Pakete jedoch weniger Verzögerung als andere erfahren, muss das Kommunikationsnetz eine differenzierte Behandlung dieser Pakete erlauben, so dass z. B. Sprachpakete eines VoIP-Dienstes bevorzugt gegenüber Paketen einer E-Mail behandelt werden. Daher werden zur Dienstgüteunterstützung unterschiedliche Dienstklassen definiert, so dass die einer bestimmten Dienstklasse zugehörigen Pakete entsprechend den Vorgaben dieser Dienstklasse im Netz bearbeitet und weitergeleitet werden.

Im Folgenden werden zunächst zwei Architekturvorschläge zur Dienstgüteunterstützung, *IntServ* (*Integrated Services*) und *DiffServ* (*Differentiated Services*), vorgestellt. Anschließend wird auf die Verbindungsannahmesteuerung eingegangen, die den Zugriff auf Ressourcen entsprechend beschränken soll. Schließlich werden noch weitere unterstützende Verfahren

genannt, die in den obigen Architekturen eingebunden werden können und deren Wirksamkeit evtl. vergrößern.

IntServ (Integrated Services)

In der IntServ-Architektur [7] werden drei Dienstklassen definiert. Diese orientieren sich an den Anforderungen bezüglich der maximal zulässigen Verzögerung der Anwendungen:

- *Guaranteed Service Class*
  Diese Klasse wird für Anwendungen verwendet, bei denen eine bestimmte Paketverzögerung nicht überschritten werden darf.
- *Controlled-Load Service Class*
  Bei dieser Dienstklasse wird für die mittlere Paketverzögerung eine Grenze vorgegeben, die nicht öfters überschritten werden darf, als es in einem unbelasteten Netz der Fall wäre.
- *Best Effort Service Class*
  Diese Klasse stellt im Prinzip den gleichen Dienst wie das Internet ohne dienstgüteunterstützende Verfahren zur Verfügung, wobei drei Datentransfer-Kategorien unterschieden werden: *Interactive Burst* (z. B. für WWW-Applikationen), *interactive Bulk* (z. B. für FTP) und *asynchronous* (z. B. für E-Mail).

An die Guaranteed und die Controlled-Load Dienstklassen werden quantitative Anforderungen gestellt, d. h. dass absolute Verzögerungswerte angegeben werden, die durch das Netz eingehalten werden müssen. Diese Dienstklassen werden mit Hilfe von zusätzlicher Signalisierung realisiert. Zudem wird eine Form der Verbindungsannahmesteuerung angewendet, um einerseits das Bereitstellen der entsprechenden Ressourcen und andererseits die Entscheidung, ob eine weitere Kommunikationsbeziehung zugelassen werden kann, zu ermöglichen. Meist wird für die Ressourcenverwaltung das in [8] definierte RSVP (*Resource Reservation Protocol*) angewendet. Dieses benutzt das sog. *Soft State*-Prinzip, bei dem eine periodische Erneuerung der Verbindungszustände durch entsprechende Nachrichten erfolgt. Für die Freigabe der belegten Ressourcen müssen keine Nachrichten ausgetauscht werden, da sie nach Ablauf einer gewissen Zeitspanne ohne Erneuerungsnachricht automatisch durch die verwaltende Einheit freigegeben werden.

Ein großer Nachteil von IntServ sind die Probleme bezüglich seiner Skalierbarkeit und damit seiner Anwendbarkeit in großen Kernnetzen. Da für jede Kommunikationsbeziehung der Guaranteed und der Controlled-Load Dienstklassen Zustandsinformationen in jedem beteiligten Netzknoten gehalten werden müssen, kann dies bei einer sehr großen Anzahl von Kommunikationsbeziehungen zu Problemen bezüglich der Leistungsfähigkeit dieser Netzknoten führen. Es ist zwar möglich, mehrere Kommunikationsbeziehungen aggregiert und damit gemeinsam zu behandeln, jedoch ist dies laut [74] noch nicht gründlich genug untersucht worden, um

verlässliche Aussagen treffen zu können, ob dies die genannten Probleme vollständig lösen kann.

DiffServ (Differentiated Services)

Die DiffServ-Architektur [5] definiert ebenfalls mehrere Dienstklassen, wobei die Bearbeitung der Pakete im Netz ausschließlich von der Zugehörigkeit zu einer Dienstklasse abhängt. Daher müssen im Gegensatz zu IntServ keine Informationen für einzelne Kommunikationsbeziehungen in den Netzknoten gehalten werden, wodurch eine gute Skalierbarkeit bezüglich der Anzahl der Kommunikationsbeziehungen erreicht wird.

Die Zugehörigkeit eines Pakets zu einer bestimmten Dienstklasse wird in den Randknoten eines Netzes geregelt, indem bei Paketen, die in das Netz eintreten, die Kennzeichnung einer Dienstklasse (*Differentiated Services Codepoint*) in den Kopf des IP-Pakets eingetragen wird. Über die Dienstklasse wird festgelegt, wie die Pakete in den einzelnen Netzknoten behandelt werden. Die verschiedenen Arten der Behandlung werden als PHB (*Per-Hop Behavior*) bezeichnet.

Die Anforderungen an die DiffServ-Dienstklassen werden in der Regel entweder qualitativ, wie z. B. „Dienst mit geringer Verzögerung“, oder relativ, wie z. B. „Pakete dieser Dienstklasse erfahren eine geringere Verzögerung als Pakete einer anderen Dienstklasse“, gestellt. Wie diese Vorgaben durch den Dienstanbieter realisiert werden, d. h. welches PHB einer Dienstklasse in einem Knoten zugeordnet ist, wird nicht festgelegt. Des Weiteren sind auch quantitative Anforderungen an Dienstklassen möglich, wobei sich die dadurch resultierende Komplexität an die der IntServ-Architektur annähert.

Neben der Skalierbarkeit ist ein weiterer Vorteil der DiffServ-Architektur das hierarchische Modell der Ressourcenverwaltung, das die Zusammenarbeit verschiedener Dienstanbieter unterstützt. Dabei werden zwischen Dienstanbietern und Dienstnutzern sog. SLA (*Service Level Agreements*) vereinbart, die die Eigenschaften der angebotenen Dienstklassen definieren. Welche SLA angeboten werden und wie deren Einhaltung realisiert wird, bleibt dem Dienstanbieter überlassen. Selbst die Anzahl der Dienstklassen kann jeder Dienstanbieter unabhängig bestimmen. Dieses Modell der Ressourcenverwaltung ist aber gleichzeitig ein Nachteil, da die Einhaltung von quantitativen Ende-zu-Ende Dienstgütekriterien, wie sie z. B. VoIP-Dienste benötigen, insbesondere über die Netze verschiedener Dienstanbieter, laut [74] noch nicht geklärt ist.

Da weder die IntServ- noch die DiffServ-Architektur alle geforderten Eigenschaften besitzt, ist es nicht absehbar, welche der beiden Verfahren sich durchsetzen wird. Darüber hinaus könnten beide Verfahren gemeinsam verwendet werden, um die Dienstgüteanforderungen verschiedener Anwendungen zu erfüllen. In [74] werden zwei Konzepte beschrieben, die die gemeinsame Verwendung der beiden Architekturen vorsehen:

- Nach einer Studie von 1998 [77] kommt es vor allem an den Grenzen des Internet zu Verstopfungen, die zu Paketverlusten und großen Verzögerungen führen. Die Auslastung im Kernnetz ist dagegen vergleichsweise gering. Daher bietet sich für das Kernnetz die gut skalierende DiffServ-Architektur an, während in den Zugangsnetzen IntServ angewendet werden könnte. Wenn sich die Voraussetzungen nicht grundlegend ändern, würde diese Kombination aus DiffServ und IntServ eine zufriedenstellende Dienstgüte bieten.
- Wenn die Dienstgüte Ende-zu-Ende garantiert werden muss, ist eine Lösungsmöglichkeit, die IntServ-Architektur Ende-zu-Ende anzuwenden, da DiffServ diese Garantie nicht ermöglicht. DiffServ könnte in diesem Fall die Differenzierung der Best Effort Dienstklasse unterstützen, indem die verschiedenen Kategorien dieser Dienstklasse mittels der DiffServ-Architektur differenziert behandelt werden.

Verbindungsannahmesteuerung

Die Verbindungsannahmesteuerung entscheidet, ob eine Verbindung angenommen oder abgelehnt wird. Um diese Entscheidung vorzunehmen, wird festgestellt, ob genügend Ressourcen vorhanden sind, um die notwendige Dienstgüte der neuen Verbindung und der bereits bestehenden Verbindungen sicher zu stellen. In der kanalvermittelnden Telekommunikation kann dies relativ einfach durchgeführt werden, indem überprüft wird, ob noch ein Kanal für die neue Verbindung frei ist.

Bei der paketvermittelnden Kommunikation ist dies wesentlich komplexer, da zum einen die Ressourcen gemeinsam von mehreren Kommunikationsbeziehungen verwendet werden und zum anderen verschiedene Dienste integriert werden, die sowohl unterschiedliche Verkehrscharakteristika als auch unterschiedliche Dienstgüteanforderungen besitzen.

Aufgrund seiner Verbindungsorientierung wurden gerade im ATM-Umfeld viele Methoden für die Verbindungsannahmesteuerung entwickelt und untersucht, die diese Probleme lösen. Eine Übersicht dazu kann beispielsweise in [66, 93] gefunden werden.

Da das Internet selbst verbindungslos arbeitet, spielte die Verbindungsannahmesteuerung[1] in diesem Bereich eine untergeordnete Rolle. Durch die zunehmende Verwendung von Konzepten wie z. B. *Flows*, bei denen die einzelnen Pakete einer Kommunikationsbeziehung zusammenhängend betrachtet werden, oder auch der Ende-zu-Ende Signalisierung für VoIP-Dienste, gewinnt dieses Thema zunehmend an Bedeutung [122]. Insbesondere zur Einhaltung von Dienstgütekriterien ist es notwendig, die Entscheidung, ob eine neue Kommunikationsbeziehung zugelassen werden kann, entsprechend zu steuern.

[1] Im Internet-Umfeld wird wegen der Anwendung der verbindungslosen Kommunikation meist der Begriff Annahmesteuerung (*Admission Control*) verwendet.

In [19] wird beispielsweise eine Form der Verbindungsannahmesteuerung für WWW-Server vorgestellt, die nicht einzelne TCP-Verbindungen, sondern ganze *Sessions* betrachtet. Eine Session besteht dabei aus mehreren Anfragen eines Benutzers, wie es z. B. beim Online-Handel vorkommt, wenn ein Benutzer sich zunächst über verschiedene Produkte informiert, bevor er den Kauf durchführt, der wiederum aus mehreren Anfragen bestehen kann. Wenn eine Anfrage einer neuen Session ankommt, prüft diese Verbindungsannahmesteuerung, ob noch ausreichend Ressourcen vorhanden sind. Ist dies nicht der Fall, wird die Anfrage abgelehnt. Ansonsten wird diese Anfrage und alle weiteren Anfragen, die zu dieser Session gehören, bearbeitet.

Eine weitere Form der Verbindungsannahmesteuerung wird in [10] und [58] vorgestellt. Dabei sendet ein Endpunkt, der eine Verbindung aufbauen möchte, Testpakete (*Probes*) mit der gleichen Rate, wie sie für die eigentliche Verbindung benötigt wird, zum Ziel-Endpunkt, der diese umgehend zurücksenden soll. Abhängig von der *Round Trip Time* (RTT), der Verzögerung bis ein gesendetes Paket wieder beim Sender ankommt, und den Paketverlusten wird entschieden, ob genügend Ressourcen für eine Verbindung vorhanden sind.

Schließlich seien hier noch die Untersuchungen für eine Methode zur Verbindungsannahmesteuerung für VoIP-Dienste aus [34] erwähnt: Die dort beschriebene Methode verwendet MPLS (*Multi-Protocol Label Switching*, [89]), um Pfade zwischen den Randknoten des Kernnetzes festzulegen, wobei vor der Annahme einer Verbindung geprüft wird, ob auf dem entsprechenden Pfad genügend Ressourcen zur Verfügung stehen. Neben der Verbindungsannahmesteuerung wird damit auch eine Verteilung der Last auf das Netz ermöglicht.

Weitere Verfahren

Im Folgenden werden einige Verfahren vorgestellt, die zusätzlich zu den bereits vorgestellten verwendet werden können, um eine entsprechende Dienstgüte für VoIP-Dienste zu unterstützen.

- Komprimierung der Paketköpfe (*Header Compression*)
  Wie in Abschnitt 2.2.2 beschrieben, werden in einem Sprachpaket mehrere Abtastwerte übertragen, so dass in der Regel 30 Bytes Sprachdaten in einem Paket enthalten sind. Sprachpakete werden meist mittels RTP über UDP und IP transportiert. Dadurch ergibt sich allein aus den Paketköpfen dieser Protokolle ein Datenvolumen von mindestens 40 Bytes, so dass zur Übertragung von 30 Bytes Sprachdaten mindestens 70 Bytes über das Kommunikationsnetz transportiert werden müssen. Somit ergibt sich ein zusätzlicher Aufwand (*Overhead*) von mindestens 57 %. Um den Nutzdatenanteil zu erhöhen, könnten mehr Abtastwerte in einem RTP-Paket übertragen werden. Dies würde jedoch die Verzögerung erhöhen und damit die Sprachqualität verringern. In [25] wurde untersucht, wie sich der Nutzungsgrad und die Verzögerung bei der Veränderung der Anzahl der Abtastwerte in

einem RTP-Paket verhalten. Eine zweite Möglichkeit, um den Overhead zu reduzieren, ist die Komprimierung der Paketköpfe, wie sie von der IETF in [16] vorgeschlagen wird. Dabei ergibt sich keine zusätzliche Verzögerung der Sprachpakete. Bei diesem Verfahren werden die Köpfe von RTP, UDP und IP auf 2 Bytes komprimiert. Dazu wird zunächst ein unkomprimierter Paketkopf an den Empfänger gesendet, die Paketköpfe der folgenden Pakete enthalten dann nur noch die Informationen, die unbedingt notwendig sind: Alle Felder, bei denen sich die entsprechenden Werte um konstante Werte pro Paket ändern, wie z. B. bei Reihenfolgenummern, sowie konstante Felder werden entfernt. Des Weiteren wird bei Feldern, bei denen sich der Inhalt von Paket zu Paket um einen variablen Wert ändert, wie z.B. beim Zeitstempel eines RTP-Pakets, der absolute Wert durch den Differenzwert zum vorherigen Pakets ersetzt, da dafür ein geringerer Wertebereich benötigt wird. Ein Nachteil dieses Verfahrens ist, dass alle beteiligten Komponenten dieses Verfahren unterstützen müssen. Jedoch kann seine Verwendung auf Übertragungsstrecken mit geringer Übertragungskapazität beschränkt werden, so dass dort eine möglichst große Effizienz erreicht wird.

- Multiplexen mehrerer Verbindungen
  Eine weitere Möglichkeit, um den Nutzdatenanteil bei der Übertragung von Sprachdaten zu erhöhen, wird in [114] beschrieben. Dabei wird ein Szenario betrachtet, bei dem mehrere Standorte über WAN-Strecken miteinander verbunden sind. Diese WAN-Strecken besitzen eine begrenzte Übertragungskapazität und sollen daher möglichst effizient verwendet werden. In dem vorgestellten Verfahren wird eine Kombination aus Komprimierung der Paketköpfe und Multiplexen von Sprachpaketen angewendet. Dabei werden die Sprachpakete mehrerer VoIP-Beziehungen gemeinsam in einem UDP-Paket auf den WAN-Strecken übertragen. Darüber hinaus werden die Paketköpfe der Sprachpakete komprimiert, so dass diese in den meisten Fällen eine Länge von 2 Bytes besitzen. Neben dem angesprochenen Anwendungsfall könnte dieses Verfahren von einem Dienstanbieter dazu verwendet werden, um Sprachverkehr möglichst effizient durch sein Netz zu transportieren.

Neben diesen genannten Verfahren existieren weitere, um eine ausreichende Dienstgüte für VoIP-Dienste zu unterstützen. Beispielsweise sollte in einem Netz die maximale Paketgröße so begrenzt werden, dass ein Paket mit maximaler Größe und damit maximaler Übertragungsdauer ein Sprachpaket nicht zu sehr verzögern kann. Des Weiteren existieren für die Router unterschiedliche Verfahren, wie die Pakete intern behandelt werden, so dass die Verzögerung für Sprachpakete gering gehalten wird. Einige dieser Verfahren werden beispielsweise in [6] beschrieben.

In [27] wird der Aufbau eines IP-Netzes beschrieben, das die Unterstützung einer entsprechenden Ende-zu-Ende Dienstgüte mit derzeit verfügbaren Technologien realisiert.

2.3 Signalisierung

Wie bereits in Abschnitt 2.1.1.2 erwähnt, wird unter dem Begriff Signalisierung die Steuerung von Kommunikationskomponenten in einem Netz verstanden. Die Steuerung erfolgt dabei durch den Austausch von Nachrichten zwischen den beteiligten Steuerkomponenten. Die Hauptaufgabe der Signalisierung ist die Verbindungssteuerung, die den Verbindungsauf- und -abbau beinhaltet. Des Weiteren dient sie dem Zugriff auf spezielle Komponenten, die spezifische Dienste zur Verfügung stellen, wie beispielsweise beim IN, und dem Transport von Informationen zur Verwaltung eines Netzes.

Die Signalisierung zwischen einer Endeinrichtung eines Benutzers (*Terminal*) und einer Netzkomponente (*Network Node*) wird als Teilnehmersignalisierung (*User-to-Network Signalling)* und die Signalisierung zwischen Netzkomponenten wird als Zwischenamtssignalisierung (*Interoffice Signalling*) bezeichnet.

In Abschnitt 2.3.1 wird die Signalisierung für VoIP-Dienste nach der ITU-T-Empfehlung H.323 vorgestellt, wobei auch auf das Verhalten einer H.323-basierten VoIP-Umgebung in Hochlastsituationen qualitativ eingegangen wird. Neben der Signalisierung nach H.323 existiert für die Steuerung von VoIP-Diensten das von der IETF vorgeschlagene SIP (*Session Initiation Protocol*, [90]). SIP und die Sicht der IETF auf VoIP werden u. a. in [91, 96, 97, 98] beschrieben. Sowohl SIP als auch H.323 besitzen prinzipiell die gleiche Aufgabe und stehen somit in Konkurrenz zueinander. Bisher ist noch nicht absehbar, welche der beiden sich durchsetzen wird, daher unterstützen nahezu alle namhaften Hersteller von Kommunikationskomponenten beide Signalisiersysteme. Da die weiteren Untersuchungen für eine H.323-basierte Umgebung durchgeführt werden, wird für eine kurze Vorstellung von SIP und seinen Eigenschaften auf die oben genannte Literatur verwiesen. In Abschnitt 2.3.2 werden die wichtigsten Unterschiede bei der Steuerung von VoIP-Diensten im Vergleich zur kanalvermittelnden Telefonie beschrieben.

2.3.1 ITU-T Empfehlung H.323

Zunächst werden in Abschnitt 2.3.1.1 einige einführende Bemerkungen zu H.323 gegeben, bevor in Abschnitt 2.3.1.2 die Komponenten der H.323-Architektur vorgestellt werden. Weiter werden in Abschnitt 2.3.1.3 die Signalisierprotokolle und ihr Zusammenwirken präsentiert, wobei auch auf mögliche Erweiterungen eingegangen wird. Schließlich werden in Abschnitt 2.3.1.4 die Auswirkungen von Hochlastsituationen bezüglich der Steuerung auf H.323-basierte VoIP-Netze qualitativ beschrieben.

2.3.1.1 Allgemeines

Die erste Version der Empfehlung H.323 wurde 1996 von der ITU-T unter dem Namen *Visual telephone systems and equipment for local area networks which provide a non-guaranteed Quality of Service* verabschiedet. 1998 wurde mit Version 2 der Name zu *Packet-based multimedia communications systems* geändert. Im November 2000 wurde mit Version 4 die derzeit aktuelle Version von H.323 [47] genehmigt.

Die Empfehlung H.323 stellt ein Rahmenwerk dar, das grundsätzlich beschreibt, wie Multimedia-Kommunikation über paketbasierte Kommunikationsnetze erfolgen soll. Dabei werden u. a. verschiedene Signalisierprotokolle verwendet, die in weiteren ITU-T-Empfehlungen definiert sind. H.323 selbst beschreibt die Komponenten einer H.323-Umgebung sowie die Anwendung und das Interagieren der verschiedenen Signalisierprotokolle. Des Weiteren existieren im H.323-Umfeld weitere, ergänzende ITU-T-Empfehlungen, um z. B. zusätzliche Dienste, wie sie aus dem IN bekannt sind, zu realisieren, oder das Zusammenwirken mit der kanalvermittelnden Telefonie festzulegen.

Bei der Definition von H.323 wurde auf eine nahtlose Interoperabilität mit den bestehenden kanalvermittelnden Kommunikationsnetzen, wie z. B. dem ISDN, besonderen Wert gelegt. Dies wird beispielsweise durch die Berücksichtigung entsprechender Signalisierprotokolle realisiert. Darüber hinaus werden Konferenzen und einige von privaten Telefonnetzen bekannte Dienste, wie z. B. die Anzeige, dass Nachrichten eingegangen sind (*Message Waiting Indication*), durch H.323 entsprechend unterstützt. Durch diese relative enge Beziehung zur kanalvermittelten Kommunikation und durch weitere Eigenschaften, wie z. B. die Verwendung von speziellen zentralen Komponenten, die wesentliche Steueraufgaben in einer VoIP-Umgebung wahrnehmen, wird deutlich, dass die Definition von H.323 durch die kanalvermittelnde Telekommunikation geprägt ist.

Zu H.323 gibt es eine Vielzahl einführender Literatur, wobei beispielsweise [116] einen fundierten und kompakten Überblick über H.323 und die Protokolle in seinem Umfeld liefert.

2.3.1.2 Komponenten

Im Folgenden werden zunächst die in H.323 definierten Basiskomponenten vorgestellt und anschließend weitere Komponenten, die in einer H.323-basierten VoIP-Umgebung Verwendung finden können, beschrieben.

In Bild 2.8 sind die folgenden H.323-Basiskomponenten dargestellt:

- Terminal
  Terminals stellen zusammen mit *Gateways* und *MCUs* die Endpunkte einer H.323-Umgebung dar. Im einfachst möglichen H.323-Szenario kommen ausschließlich Terminals vor, die direkt miteinander kommunizieren. Die Realisierung der Terminals reicht von einfachen

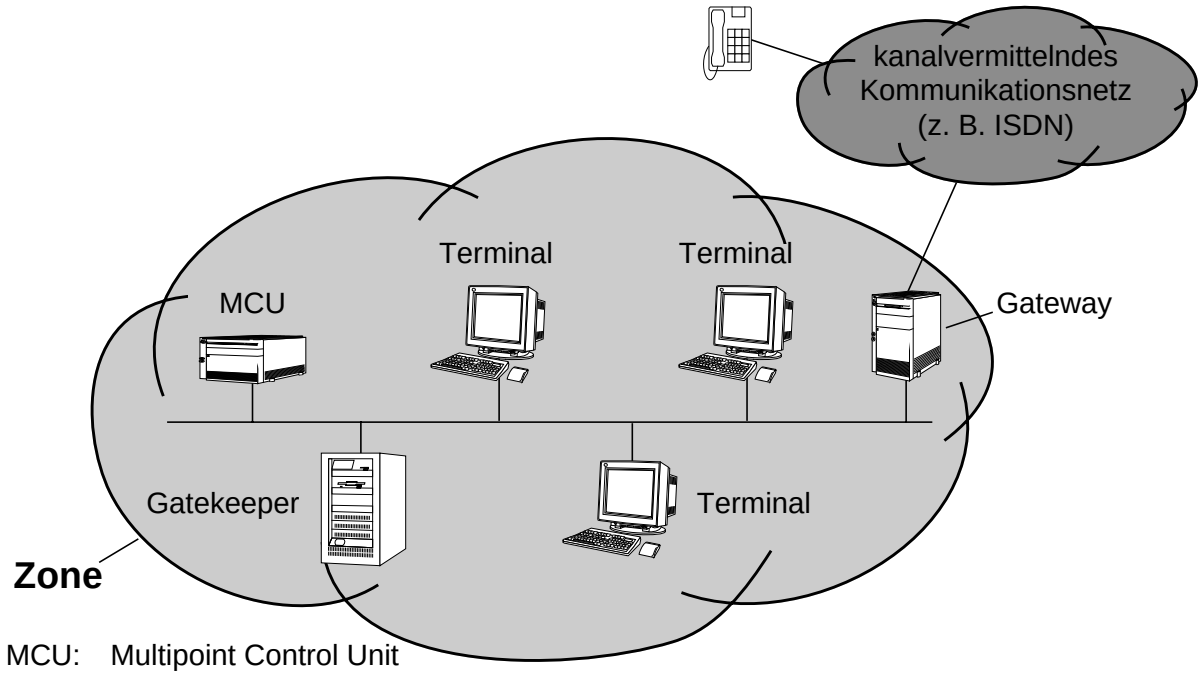


Bild 2.8: H.323-Basiskomponenten

IP-Telefonen, über spezielle Anwendungen, die auf einem Rechner ablaufen (sog. *Soft Phones*), bis zu leistungsfähigen Multimedia-Rechnern. Neben der entsprechenden Signalisierung muss ein Terminal zumindest die Sprachkommunikation ermöglichen, wobei der Codec G.711 auf jeden Fall unterstützt werden muss. Falls darüber hinaus Video-Kommunikation möglich ist, soll als Mindestanforderung die Unterstützung des Video-Codec H.261 erfüllt sein.

- Gateway
  Ein *H.323-Gateway*, im weiteren Verlauf als *Gateway* bezeichnet, erlaubt, wie in Bild 2.8 dargestellt, die Kopplung von H.323-basierten und kanalvermittelnden Kommunikationsnetzen (z. B. dem ISDN). Dazu müssen neben den Übertragungsformaten der unteren Protokollschichten auch die Kommunikationsprotokolle entsprechend transformiert werden. Dies bedeutet, dass ein Gateway die jeweiligen Signalisierprotokolle, Mediencodierung und Medienserialisierung umsetzen muss. In der ITU-T-Empfehlung H.246 [44] werden diese Transformationen spezifiziert. Wie oben erwähnt sind Gateways H.323-Endpunkte, d. h. sie verhalten sich in einer H.323-Kommunikationsbeziehung ebenso wie ein Terminal bzw. wie eine MCU (*Multipoint Control Unit*), wenn diese Funktionalität in einem Gateway realisiert sein sollte.

- Multipoint Control Unit – MCU
  Eine MCU (*Multipoint Control Unit*) unterstützt die Durchführung von Mehrpunkt-Konferenzen. Sie besteht zumindest aus einem MC (*Multipoint Controller*), wobei zusätzlich ein oder mehrere MP (*Multipoint Processor*) enthalten sein können.

- Multipoint Controller – MC
  Der MC stellt Steuerungsfunktionen für Konferenzen zwischen drei oder mehr Endpunkten zur Verfügung. Er ist für den Austausch der Informationen über die Kommunikationsfähigkeiten der Endpunkte zuständig und bestimmt dabei den ausgewählten Kommunikationsmodus (*Selected Communication Mode* – SCM), der jedoch für die einzelnen Teilnehmer einer Konferenz unterschiedlich sein kann. Ein MC kann auch in einem Terminal, einem Gateway oder einem Gatekeeper enthalten sein. Dabei wird er aber ausschließlich bei sog. *ad-hoc* Konferenzen verwendet, bei denen herkömmliche VoIP-Kommunikationsbeziehungen in Konferenzen umgewandelt werden, so dass aus 2-Punkt-Verbindungen Mehrpunktverbindungen erzeugt werden können.
- Multipoint Processor – MP
  Der MP verarbeitet empfangene Mediendaten und gibt sie anschließend an die an einer Konferenz beteiligten Endpunkte weiter. Dabei müssen z. B. die Sprachdaten geeignet gemischt und die Videodaten entsprechend kombiniert werden. Ein MP wird ausschließlich über einen MC angesprochen. Wie ein MC kann ein MP sowohl in einer MCU als auch in einem Terminal, Gateway oder Gatekeeper enthalten sein, wobei in dieser Komponente auch ein MC vorhanden sein muss. Die Kommunikation zwischen MC und MP ist nicht Bestandteil von H.323.

Konferenzen in einer H.323-Umgebung können auf verschiedene Arten durchgeführt werden. So ist es möglich, Mediendaten zentral, z. B. in einer MCU, oder dezentral in den beteiligten Endpunkten zu bearbeiten. Darüber hinaus wird die Kaskadierung von MCs unterstützt, so dass ein MC mehrere MCs steuern kann. Eine Beschreibung von Konferenzen in H.323-Umgebungen ist beispielsweise in [13] enthalten.

- Gatekeeper
  Ein *Gatekeeper* ist nach Standard optional in einer H.323-Umgebung. Jedoch ist er für eine Umgebung, die entsprechend verwaltet werden soll, notwendig, da er die dafür benötigten Dienste, wie z. B. die Administration und die Zugangssteuerung der Endpunkte, erbringt. Dabei wird zwischen Diensten, die ein Gatekeeper durchführen muss, und optionalen Diensten unterschieden. Zunächst werden die Dienste vorgestellt, die ein Gatekeeper unterstützen muss:

  - Adressauflösung
    Eine Basisfunktionalität eines Gatekeepers ist die Umsetzung von sog. *Alias*-Adressen, die beispielsweise einem Namen entsprechen können, in Transportadressen der Endpunkte. Die dazu notwendigen Informationen können beispielsweise in einer Adresstabelle enthalten sein, die bei der Registrierung der Endpunkte, die in Abschnitt 2.3.1.3 beschrieben wird, aktualisiert wird, oder über den Zugriff auf ein externes Adressver-

zeichnis (z.B. mittels des *Lightweight Directory Access Protocol* – LDAP, [121]) erhalten werden.

- Zugangssteuerung
  Mit diesem Dienst steuert der Gatekeeper die Zulassung von Kommunikationsbeziehungen der Endpunkte, wobei diese Zulassung aufgrund verschiedener Kriterien erfolgen kann. Beispielsweise können diese Kriterien auf der noch zur Verfügung stehenden Übertragungskapazität oder auf den Berechtigungen der Teilnehmer basieren.
- Unterstützung der Steuerung der zur Verfügung stehenden Übertragungskapazitäten
  Der Gatekeeper muss zumindest den Austausch der entsprechenden Signalisiernachrichten unterstützen, die Steuerung der zur Verfügung stehenden Übertragungskapazitäten selbst ist jedoch optional.
- Zonen-Verwaltung
  Eine Zone ist, wie in Bild 2.8 dargestellt, durch die Endpunkte (Terminal, Gateway und MCU), die bei einem Gatekeeper angemeldet sind, und dem Gatekeeper selbst festgelegt. Der Gatekeeper ist für die Verwaltung der Zonenmitglieder zuständig, d. h. er führt die genannten Dienste für sie durch. Das Konzept der Zonen dient der Strukturierung einer VoIP-Umgebung und erlaubt somit eine Eingrenzung der verwalteten Daten für die Steuerung der Endpunkte. In einer Zone kann zu einem Zeitpunkt genau ein Gatekeeper existieren. Jedoch ist es möglich, dass sich in einer Zone weitere Komponenten befinden, die über die Gatekeeper-Funktionalität verfügen, diese aber zu diesem Zeitpunkt nicht verwenden. Dies wird für das Konzept des alternativen Gatekeepers verwendet, bei dem eine Komponente mit Gatekeeper-Funktionalität als alternativer Gatekeeper festgelegt wird, der die Gatekeeper-Rolle beim Ausfall des ursprünglichen Gatekeeper übernehmen kann. Dabei können mehrere Komponenten als alternative Gatekeeper festgelegt werden.

Folgende optionale Dienste eines Gatekeeper sind definiert:

- Signalisierung für die Verbindungssteuerung
  Der Gatekeeper kann festlegen, ob er an der Signalisierung für die Verbindungssteuerung teilnimmt (*gatekeeper routed call signalling*, im weiteren Verlauf als Gatekeeper-geführte Signalisierung bezeichnet), oder ob die Endpunkte die entsprechenden Signalisiernachrichten direkt austauschen (*direct endpoint call signalling*). Der Vorteil der Gatekeeper-geführten Signalisierung liegt in der umfassenden Kontrolle über eine Verbindung. Dadurch ist der Gatekeeper in der Lage, die einzelnen Verbindungszustände sehr genau abzubilden, wodurch er z. B. effizient in bestehende Verbindungen eingreifen kann.
- Verbindungsberechtigungen
  Ein Gatekeeper kann die Überwachung von Berechtigungen der Teilnehmer realisieren,

so dass beispielsweise der Zugriff auf bestimmte Ressourcen, wie z. B. ein Gateway, nicht für jeden Teilnehmer zugelassen wird.

- Verwaltung der Übertragungskapazitäten
  Wie bereits erwähnt kann ein Gatekeeper die zur Verfügung stehenden Übertragungskapazitäten verwalten, indem er z. B. Verbindungsanfragen ablehnt, wenn die Übertragungskapazitäten nicht mehr ausreichen.
- Weitere optionale Dienste
  In der Empfehlung H.323 sind weitere optionale Dienste für einen Gatekeeper wie z.B. Verbindungsverwaltung und Verzeichnisdienste aufgeführt. Darüber hinaus sind noch weitere z. B. an PBX-Leistungsmerkmale (PBX – *Private Branch Exchange*, private Vermittlungsstelle) angelehnte Dienste oder die Gebührenerfassung denkbar.

Neben den vorgestellten Basiskomponenten können auch die folgenden Komponenten Bestandteil einer H.323-Umgebung sein:

- Media Gateway Controller und Media Gateway
  Um die unterschiedlichen Aufgaben eines Gateway besser unterstützen zu können, kann es in einen Steuerteil und in einen Nutzdatenteil aufgespalten werden, die jeweils in separaten Komponenten realisiert werden. Die Steuerkomponente, die als MGC (*Media Gateway Controller*) bezeichnet wird, bearbeitet dabei sowohl die H.323-Signalisiernachrichten, als auch die des kanalvermittelnden Kommunikationsnetzes. Des Weiteren übernimmt der MGC die Steuerung des MG (*Media Gateway*). Dieses führt die Transformation der Nutzdaten für das entsprechende Kommunikationsnetz durch, wobei die dafür benötigten Informationen, wie z. B. der zu verwendende Codec, durch den MGC festgelegt werden. In der Regel steuert ein MGC mehrere MG. Für die Steuerung der MG ist das sog. *Megaco* Protokoll [45] vorgesehen, das von der IETF (RFC 3015) und von der ITU-T (Empfehlung H.248) gemeinsam spezifiziert wurde.

- Border Element
  Wie in Bild 2.9 dargestellt, können mehrere Zonen zu einem gemeinsam verwalteten Bereich (*Administrative Domain* – AD) zusammengefasst werden. Damit Komponenten feststellen können, wie Endpunkte anderer ADs erreicht werden, werden sog. *Border Elements* (BE) verwendet, wobei eine AD über mehrere BEs verfügen kann. Das BE entscheidet, wie weit die Struktur der AD offen gelegt wird. Beispielsweise können die BEs einer AD so realisiert sein, dass die einzelnen Komponenten der AD von außen nur über diese BEs erreicht werden können. Damit ist die Struktur der AD von außen nicht sichtbar. Ein BE kann sowohl in einem Gatekeeper oder Gateway integriert, als auch als eigenständige Komponente realisiert sein. Das Konzept des BE und des im Folgenden beschriebenen *Clearing House* sowie die zwischen diesen Komponenten ausgetauschten Informationen werden in [41] spezifiziert.

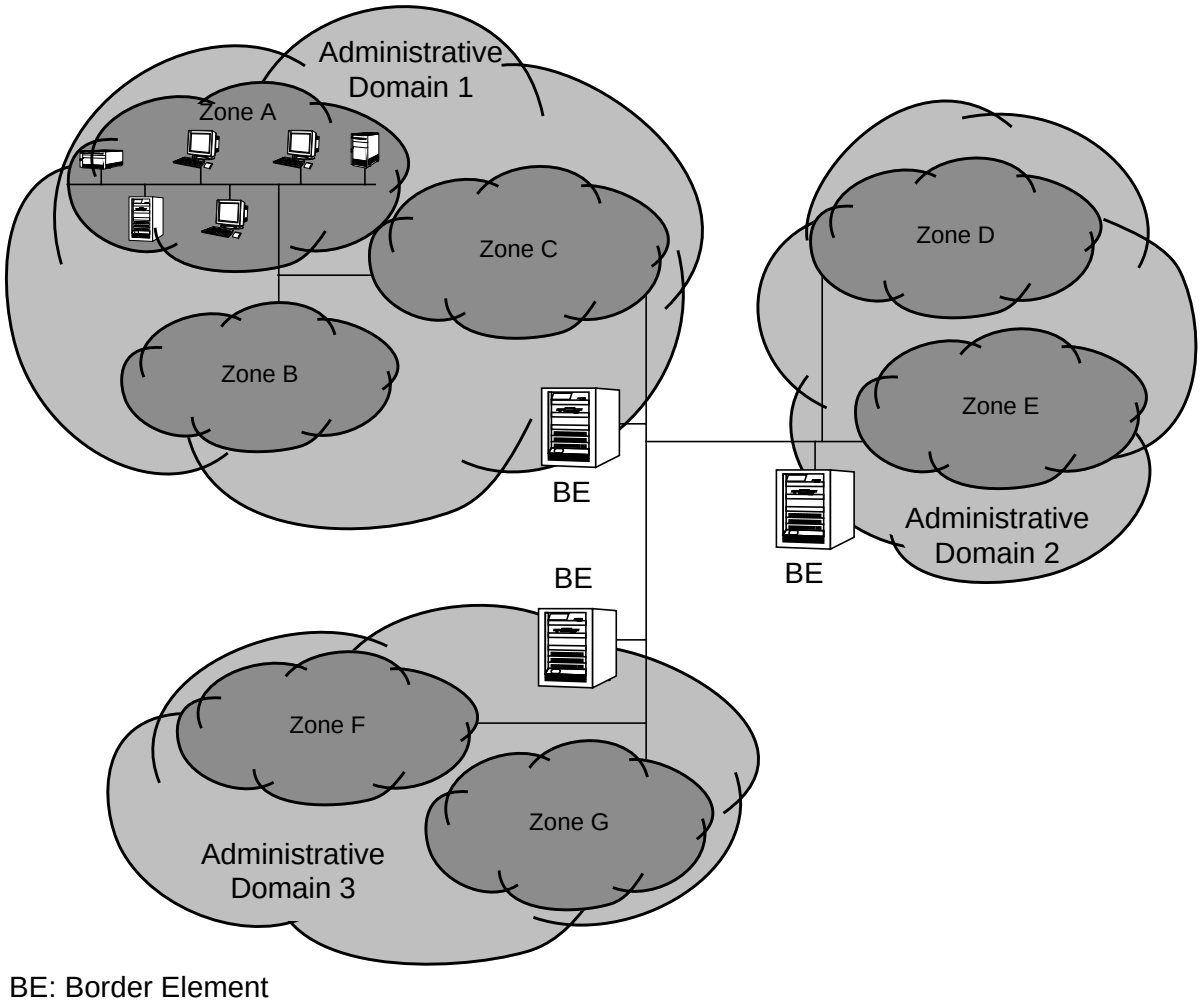


Bild 2.9: Struktur einer H.323-Umgebung

- Clearing House
  Das *Clearing House* (CH) hat prinzipiell die gleiche Aufgabe wie ein BE, wobei es nicht einer bestimmten AD zugeordnet ist, sondern übergeordnet diesen Dienst zentral für alle ADs zur Verfügung stellt. Dazu kommuniziert es mit den BEs der entsprechenden ADs, um die notwendigen Informationen zu erhalten.

2.3.1.3 Signalisierprotokolle

Die Empfehlung H.323 stellt ein Rahmenwerk dar, das verschiedene Kommunikationsprotokolle integriert, um die Multimediakommunikation über paketbasierte Netze zu realisieren. In diesem Abschnitt werden zunächst die verschiedenen Protokolle sowie ihre Anwendung in einer IP-Umgebung vorgestellt. Anschließend wird der prinzipielle Ablauf der Signalisierung, der sich in verschiedene Phasen untergliedert, beschrieben.

In Bild 2.10 sind die in H.323 verwendeten Protokolle (grau unterlegt), die im Folgenden vorgestellt werden, sowie ihre Anforderungen an die darunterliegende Transportschicht dargestellt.

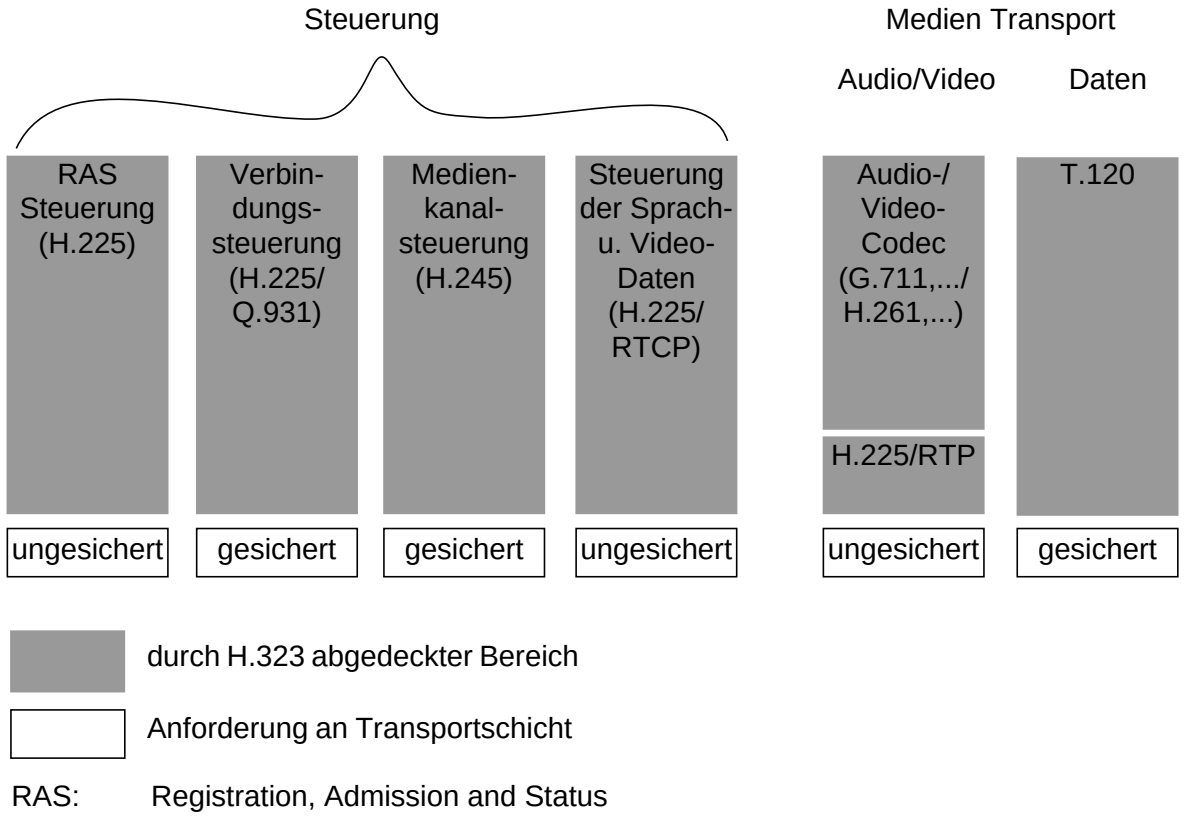


Bild 2.10: H.323-Protokolle

- RAS-Steuerung
  Die Signalisierung für die RAS-Steuerung (RAS – *Registration, Admission and Status*) findet zwischen Endpunkten und Gatekeeper statt. Ihre Aufgaben sind die Registrierung von Endpunkten, deren Zulassung für H.323-Verbindungen, die Durchführung von Anfragen für die Änderung des Bedarfs an Übertragungskapazitäten und der Austausch von Zustandsinformationen. Die Signalisiernachrichten für die RAS-Steuerung sind in der Empfehlung H.225 [40] definiert, der Ablauf der Signalisierung in der Empfehlung H.323.

- Verbindungssteuerung
  Die Signalisierung für die Verbindungssteuerung (*Call Control*) wird entweder direkt zwischen den Endpunkten oder zwischen Gatekeeper und Endpunkten bei der Gatekeepergeführten Signalisierung durchgeführt. Sie ist für die Steuerung der Verbindung als Ganzes zuständig, d. h. sie steuert den Auf- und Abbau der Verbindung, wobei die Einzelheiten der Kommunikation, wie z. B. ob und welcher Video-Codec angewandt wird, nicht Bestandteil seiner Steuerungsaufgaben sind. Die Signalisierung für die Verbindungssteuerung verwen-

det Nachrichten des ISDN-Signalisierprotokolls Q.931 [51], wobei dies in leicht abgeänderter Form, wie in H.225 definiert, erfolgt.

- Medienkanalsteuerung
  Die Signalisierung für die Medienkanalsteuerung, die in Empfehlung H.245 [43] definiert ist und daher auch als H.245-Signalisierung bezeichnet wird, hat verschiedene Aufgaben, die den Austausch der Audio-, Video- und Anwendungsdaten betreffen. Die wichtigsten Prozeduren der Medienkanalsteuerung sind:

  - Master/Slave-Bestimmung
    Um Konflikte bei der Steuerung eindeutig auflösen zu können, wird ein *Master* für eine Verbindung bestimmt, die anderen Endpunkte der Kommunikation agieren als *Slaves*. Dies ist insbesondere bei Konferenzen wichtig, um den verantwortlichen MC zu bestimmen, der z. B. die verwendeten Codecs für die einzelnen Teilnehmer der Konferenz bestimmt.
  - Austausch der Kommunikationsfähigkeiten
    Die Kommunikationsfähigkeiten (*Capabilities*) eines Endpunkts legen fest, welche Medien er unterstützt und welche Codecs dabei verwendet werden können. Da die Endpunkte über unterschiedliche Kommunikationsfähigkeiten verfügen, müssen sie sich zunächst auf einen gemeinsamen Satz einigen, der für die Kommunikation angewendet werden muss. Dies ist die Aufgabe dieser Prozedur.
  - Signalisierung für logische Kanäle
    Die Mediendaten selbst werden in sog. logischen Kanälen ausgetauscht, wobei diese für Audio- und Videodaten unidirektional sind, für Datenanwendungen jedoch bidirektional sein können. Die Aufgabe dieser Prozedur ist das Öffnen und Schließen der für die Kommunikation verwendeten logischen Kanäle. Dazu müssen z. B. die Transportadressen für die RTP- und RTCP-Kommunikationsbeziehungen ausgetauscht werden.

  Neben den genannten Prozeduren seien noch Management-Prozeduren sowie Prozeduren zur Verwaltung des H.245-Signalisierkanals und zur Bestimmung der Round Trip Time genannt.

- Steuerung der Sprach- und Videodaten
  Um Informationen über den Transport der Sprach- und Videodaten zu erhalten, wird das in Abschnitt 2.2.1 vorgestellte RTCP verwendet. Wie bereits erwähnt, erlaubt es beispielsweise die Bestimmung der Paketverlustrate und der Übertragungsverzögerung sowie der Variation der Verzögerung für einzelne Mediendaten.

- Transport von Sprach- und Videodaten
  Für den Transport von Sprach- und Videodaten wird das in Abschnitt 2.2.1 vorgestellte RTP

verwendet. Dabei wird für jedes Medium und für jede Richtung eine separate RTP-Kommunikationsbeziehung eingerichtet.

- Transport von Anwendungsdaten
  Die Unterstützung von Datenanwendungen erfolgt nach Empfehlung T.120 [52], wobei das Öffnen und Schließen der entsprechenden logischen Kanäle mittels der Medienkanalsteuerung durchgeführt wird.

Um die entsprechenden Anforderungen an die Transportschicht zu erfüllen, wird in IP-Umgebungen für die ungesicherte Übertragung in der Regel UDP als Transportprotokoll verwendet und für die gesicherte Übertragung TCP.

Des Weiteren sind für den Bereich der H.323-basierten Kommunikation weitere Dienste und Funktionen von Bedeutung, die jedoch über den Basisdienst von H.323 hinausgehen:

- Zusätzliche Dienste
  Zur Unterstützung zusätzlicher Dienste existieren für H.323 verschiedene Ansätze:

  - Der in den Empfehlungen der H.450-Serie spezifizierte Ansatz ist an das ISDN und vor allem an private Vermittlungsstellen angelehnt. Zur Diensterbringung werden spezielle Parameter in entsprechenden Signalisiernachrichten der Verbindungssteuerung integriert. Diese werden in den Endpunkten oder im Gatekeeper, falls dieser an den Transaktionen beteiligt ist, ausgewertet, so dass die notwendigen Aktionen, wie z. B. eine neue Verbindungsanfrage veranlasst werden können. Dabei müssen alle beteiligten H.323-Komponenten das Dienstmerkmal und somit die entsprechenden Empfehlungen unterstützen. In Tabelle 2.3 sind die Empfehlungen der H.450-Serie sowie die definierten zusätzlichen Dienste aufgeführt. H.450.1 [48] legt dabei das allgemeine funktionale Protokoll für die Unterstützung zusätzlicher Dienste fest. Wie aus Tabelle 2.3 ersichtlich, orientieren sich diese zusätzlichen Dienste an den bekannten Leistungsmerkmalen von privaten Vermittlungsstellen. Eine gute Übersicht zu den zusätzlichen Diensten nach H.450 ist in [65] enthalten.
  - Eine weitere Möglichkeit zur Unterstützung zusätzlicher Dienste ist die Anwendung sog. Stimulus-Signalisierung, die in Anhang L der Empfehlung H.323 beschrieben wird. Dabei senden die Endpunkte Anfragen für zusätzliche Dienste an eine geeignete Komponente. Diese Komponente wertet die Anfragen aus und führt anschließend alle notwendigen Aktionen zur Diensterbringung aus. Damit ist die Realisierung des Dienstes unabhängig von den Endpunkten, und erleichtert damit die Einführung neuer zusätzlicher Dienste. Jedoch können Interoperabilitätsprobleme auftreten, wenn beispielsweise die diensterbringende Komponente die Anfrage falsch interpretiert.
  - Die Definitionen von Anhang K der Empfehlung H.323 erlauben die Einflussnahme auf eine Verbindung über einen separaten Steuerkanal. Dabei sind die einzelnen Maßnah-

Empfehlung	Dienst
H.450.1	Allgemeines Rahmenwerk
H.450.2	Rufweiterleitung (*Call Transfer*)
H.450.3	Rufumleitung (*Call Diversion*)
H.450.4	Halten (*Call Hold*)
H.450.5	Ruf Parken und Rufübernahme (*Call Park and Pickup*)
H.450.6	Anklopfen (*Call Waiting*)
H.450.7	Anzeige über angekommene Nachrichten (*Message Waiting Indication*)
H.450.8	Anruferidentifizierung (*Name Identification*)
H.450.9	Rückruf bei Besetzt und Nichtmelden (*Call Completion*)
H.450.10	Veranlassung des Anklopfens (*Call Offer*)
H.450.11	Aufschalten in einen Ruf (*Call Intrusion*)
H.450.12	Austausch allgemeiner Informationen

Tabelle 2.3: Zusätzliche Dienste der H.450-Serie von Empfehlungen

men, die ein Benutzer zur Erbringung zusätzlicher Dienste veranlassen kann, nicht festgelegt, so dass die Einführung neuer zusätzlicher Dienste erleichtert wird.

  - Für die Unterstützung zusätzlicher Dienste kann auch das zur Steuerung von MG angewandte Megaco-Protokoll nach H.248 verwendet werden, da es ebenfalls in Form eines Stimulus-Protokolls die Steuerung von Komponenten erlaubt, wobei die Diensterbringung in einer zentralen Komponente durchgeführt wird.

- Unterstützung von Sicherheitsmaßnahmen
  In der Empfehlung H.235 [42] werden Erweiterungen für Sicherheitsdienste für H.323-basierte Kommunikation festgelegt. Dabei wird die Anwendung von Mechanismen zur Sicherung von Integrität, Vertraulichkeit und ggf. Authentizität von Signalisier- und Nutzdaten definiert.

- Zusammenwirken mit kanalvermittelnden Netzen
  Wie bereits erwähnt, spezifiziert die Empfehlung H.246, wie die Kommunikation zwischen H.323-basierten und kanalvermittelnden Kommunikationsnetzen realisiert werden kann.

Im Folgenden wird der Ablauf der Signalisierung beschrieben, wobei insbesondere der Auf- und Abbau einer H.323-Verbindung[1] vorgestellt wird. Dabei wird u. a. das Zusammenwirken der unterschiedlichen Signalisierprotokolle gezeigt.

Bei den beschriebenen Szenarien wird die Gatekeeper-geführte Signalisierung angewandt. Die an der Verbindung beteiligten Endpunkte seien Mitglieder einer Zone und kommunizieren somit mit dem gleichen Gatekeeper.

Die H.323-Signalisierprozeduren werden in die in Bild 2.11 dargestellten Phasen unterteilt, wobei diese Aufteilung in weiten Teilen der in [116] vorgestellten entspricht. Die einzelnen Phasen sowie weitere wichtige Prozeduren werden im Folgenden beschrieben:

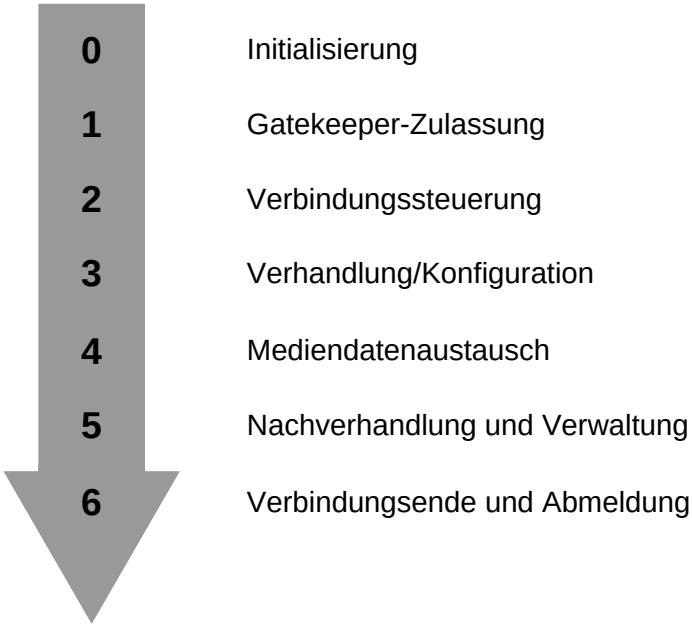


Bild 2.11: H.323 Protokoll-Phasen

- Phase 0 – Initialisierung
  In der Initialisierungsphase ermittelt ein Endpunkt zunächst seinen zuständigen Gatekeeper und registriert sich anschließend bei ihm. Die Ermittlung des Gatekeepers kann entweder manuell erfolgen, z. B. durch Konfigurationsdaten, oder durch Ausführung einer entsprechenden Signalisierprozedur. Um die Dienste des Gatekeepers in Anspruch nehmen zu können, muss sich der Endpunkt anschließend mittels einer weiteren Signalisierprozedur beim Gatekeeper registrieren. Dazu sendet er ihm eine entsprechende RAS-Signalisiernachricht, die der Gatekeeper mit der dazugehörigen Bestätigungsnachricht beantwortet, falls er die Registrierung akzeptiert.

- Phase 1 – Gatekeeper-Zulassung
  Bevor ein Endpunkt eine Verbindung aufbaut oder annimmt, beantragt er die Zulassung für

[1] Der Begriff H.323-Verbindung wird hier und im weiteren Verlauf der Arbeit für eine Kommunikationsbeziehung verwendet, die gemäß der Regeln der Empfehlung H.323 realisiert wurde.

diese Verbindung beim Gatekeeper mit der RAS-Signalisiernachricht ARQ (*Admission Request*). Dabei wird neben dem Ziel der Verbindung die benötigte Übertragungskapazität angegeben. Die Zulassung eines Endpunkts für eine Verbindung wird durch den Gatekeeper mit der Nachricht ACF (*Admission Confirm*) bestätigt, wobei er dabei die beantragte Übertragungskapazität reduzieren kann. Neben der zugelassenen Übertragungskapazität gibt der Gatekeeper an, ob die direkte oder die Gatekeeper-geführte Signalisierung für die Verbindungssteuerung durchgeführt werden soll. Abhängig davon wird entweder die Transportadresse für die Signalisierung zur Verbindungssteuerung des Zielendpunkts oder die des Gatekeepers in der ACF Nachricht angegeben. Wenn der Gatekeeper die Zulassung ablehnt, wird dies dem Endpunkt mit der Nachricht ARJ (*Admission Reject*) angezeigt, die neben dem Grund der Ablehnung auch die Adresse eines alternativen Gatekeepers enthalten kann. Bild 2.12 enthält u. a. die Signalisierprozedur für die Gatekeeper-Zulassung.

- Phase 2 – Verbindungssteuerung
  Die Signalisierung zur Verbindungssteuerung ist an der Empfehlung Q.931 angelehnt. Bild 2.12 enthält ein Beispiel-Szenario für einen erfolgreichen Verbindungsaufbau für die RAS- und die Verbindungssteuerungssignalisierung, wobei die Gatekeeper-geführte Signalisierung angewendet wird. Nachdem Endpunkt A die Zulassung zum Aufbau einer Verbindung zu Endpunkt B erhalten hat, sendet er die Signalisiernachricht *Setup* zum Gatekeeper. Dieser sendet eine entsprechende *Setup*-Nachricht zu Endpunkt B und *Call Proceeding* zu Endpunkt A. Damit zeigt der Gatekeeper an, dass der Verbindungsaufbau gerade durchgeführt wird. Endpunkt B sendet als Antwort auf die *Setup*-Nachricht ebenfalls ein *Call Proceeding* zum Gatekeeper und beantragt anschließend die Zulassung zu dieser Verbindung. Wenn der Gatekeeper die Zulassung bestätigt, zeigt Endpunkt B mit *Alerting* an, dass er grundsätzlich bereit ist, den Ruf anzunehmen. Diese Nachricht wird vom Gatekeeper an Endpunkt A weitergegeben. Mit der Nachricht *Connect* nimmt Endpunkt B schließlich die Verbindung an. Der Gatekeeper gibt diese Information mittels eines *Connect* an Endpunkt A weiter. Die *Connect*-Nachricht enthält die jeweils zu verwendende Adresse für die Signalisierung zur Medienkanalsteuerung. Dabei kann der Gatekeeper, ebenso wie für die Verbindungssteuerung, festlegen, ob diese Signalisierung über ihn geführt wird, oder direkt zwischen den Endpunkten stattfindet.

- Phase 3 – Verhandlung/Konfiguration
  In dieser Phase werden die Prozeduren der H.245-Signalisierung durchgeführt, d. h. es erfolgt die Master/Slave-Bestimmung, der Austausch der Kommunikationsfähigkeiten und das Öffnen der logischen Kanäle für den Mediendatenaustausch. Neben der Verwendung einer separaten Kommunikationsbeziehung können die H.245-Signalisiernachrichten auch mittels des sog. *Tunneling* übertragen werden. Bei diesem Verfahren werden die H.245-Signalisiernachrichten gemeinsam mit Signalisiernachrichten der Verbindungssteuerung übertragen. Falls gerade keine Nachrichten zur Verbindungssteuerung zu senden sind, obwohl

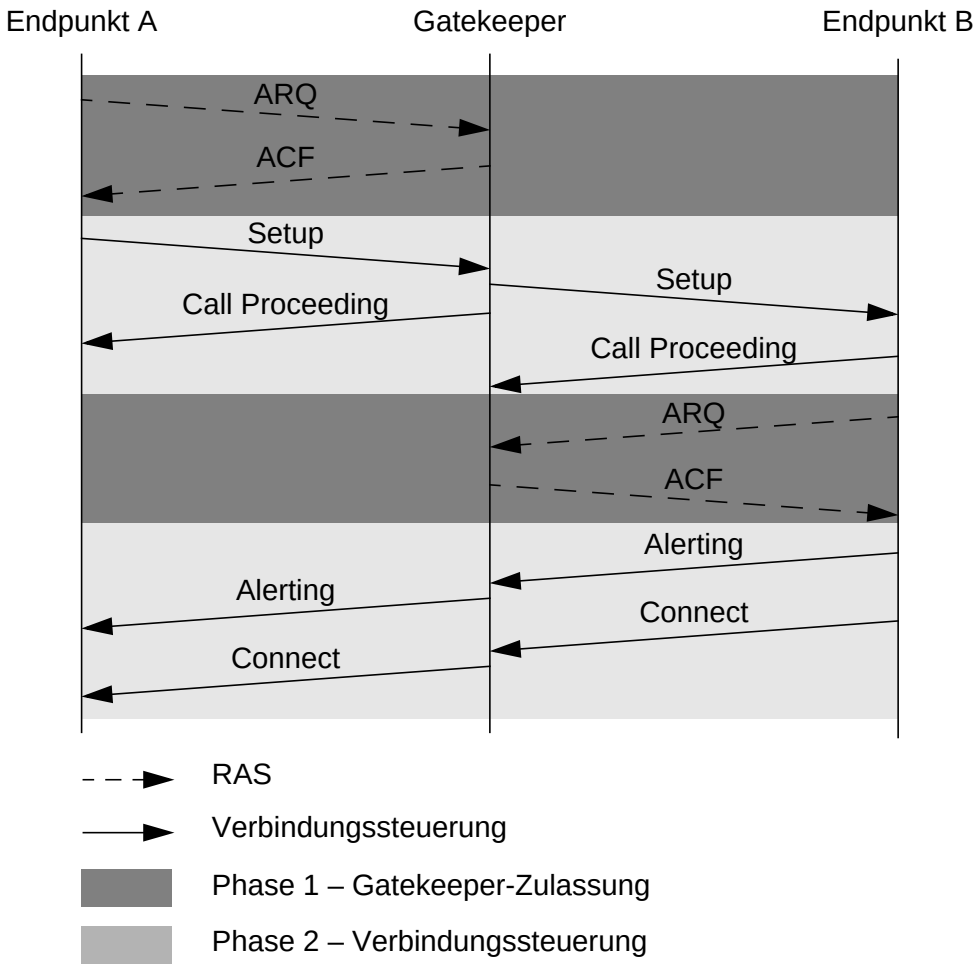


Bild 2.12: Signalisierprozeduren für den Verbindungsaufbau bei H.323 (RAS- und Verbindungssteuerungssignalisierung)

eine H.245-Signalisiernachricht zu übertragen wäre, wird die Signalisiernachricht *Facility* mit der entsprechenden H.245-Signalisiernachricht versendet.

- Phase 4 – Mediendatenaustausch
  Die Mediendaten werden direkt zwischen den Endpunkten ohne Beteiligung des Gatekeepers ausgetauscht. Wie bereits vorgestellt, werden für die Sprach- und Videodaten separate logische RTP-Kanäle für jede Richtung verwendet.

- Phase 5 – Nachverhandlung und Verwaltung
  Im Laufe einer Verbindung können weitere Dienste angewendet werden. Beispielsweise kann eine Änderung der Übertragungskapazität für eine Verbindung beantragt werden. Dazu wird dies zunächst beim Gatekeeper mittels einer entsprechenden RAS-Signalisiernachricht angefragt, bevor die betroffenen logischen Kanäle geschlossen und anschließend mit den neuen Parametern wieder geöffnet werden. Des Weiteren können zusätzliche Dienste entsprechend den Empfehlungen der H.450-Serie durchgeführt werden und der

Gatekeeper kann durch entsprechende RAS-Signalisiernachrichten Informationen über den Zustand der H.323-Verbindung erhalten.

- Phase 6 – Verbindungsende und Abmeldung
  Das Beenden einer H.323-Verbindung kann sowohl durch die beiden Endpunkte als auch durch den Gatekeeper initiiert werden. Wie in Bild 2.13 dargestellt, werden dabei zunächst die logischen Kanäle geschlossen (*CloseLogicalChannel*, *CloseLogicalChannelAck*), anschließend werden die Signalisierbeziehungen für die H.245-Signalisierung (*EndSessionCommand*) und die Verbindungssteuerung (*Release Complete*) beendet und schließlich wird das Verbindungsende dem Gatekeeper mit der RAS-Signalisiernachricht DRQ (*Disengage Request*) angezeigt, der dies mit DCF (*Disengage Confirm*) bestätigt. Der Gatekeeper kann den Verbindungsabbau mit DRQ initiieren, wobei der Endpunkt den Abbau mit DCF bestätigt, sobald die entsprechenden Kanäle geschlossen wurden. Wenn ein Endpunkt die Dienste eines Gatekeepers nicht mehr benötigt, z. B. wenn er ausgeschalten wird, erfolgt die Abmeldung beim Gatekeeper durch eine entsprechende RAS-Signalisierprozedur.

- Fast Connect-Prozedur
  Um die Dauer des Verbindungsaufbaus abzukürzen, wurde in der Empfehlung H.323 die sog. *Fast Connect*-Prozedur definiert. Dabei werden in der ersten Verbindungssteuerungsnachricht Elemente der H.245-Signalisierung eingefügt, die logische Kanäle beschreiben, die der Sender sowohl zum Empfang als auch zum Senden von Mediendaten unterstützt. Damit kann der Empfänger die für ihn verwendbaren logischen Kanäle auswählen und dies dem Sender der Nachricht in einer weiteren Verbindungssteuerungsnachricht anzeigen. Anschließend werden die so bestätigten logischen Kanäle als geöffnet betrachtet und der Mediendatenaustausch kann beginnen.

- Fehlerbehandlung
  Auftretende Fehler im Verlauf einer Signalisierprozedur, wie z. B. das Verlieren von Signalisiernachrichten oder der Empfang unerwarteter Signalisiernachrichten, werden in den einzelnen Signalisierprotokollen unterschiedlich behandelt. Wie bereits erwähnt, verwendet die H.245-Signalisierung ein gesichertes Transportprotokoll und daher werden verloren gegangene oder verspätete Nachrichten als schwerwiegender Fehler betrachtet. Wenn ein derartiger Fehler auftritt, wird die gesamte H.323-Verbindung beendet. Die Signalisierung für die Verbindungssteuerung verwendet ebenfalls ein gesichertes Transportprotokoll, wobei dort unterschieden wird, ob der Gatekeeper oder der Endpunkt den aufgetretenen Fehler entdeckt. Wenn der Gatekeeper den Fehler entdeckt, wird versucht, über den erneuten Aufbau der Kommunikationsbeziehung für die Verbindungssteuerung den Fehler zu beheben, wobei die Verbindungszustände erhalten bleiben. Der Endpunkt kann ebenso vorgehen oder aber die gesamte H.323-Verbindung beenden. Da die RAS-Signalisierung ein

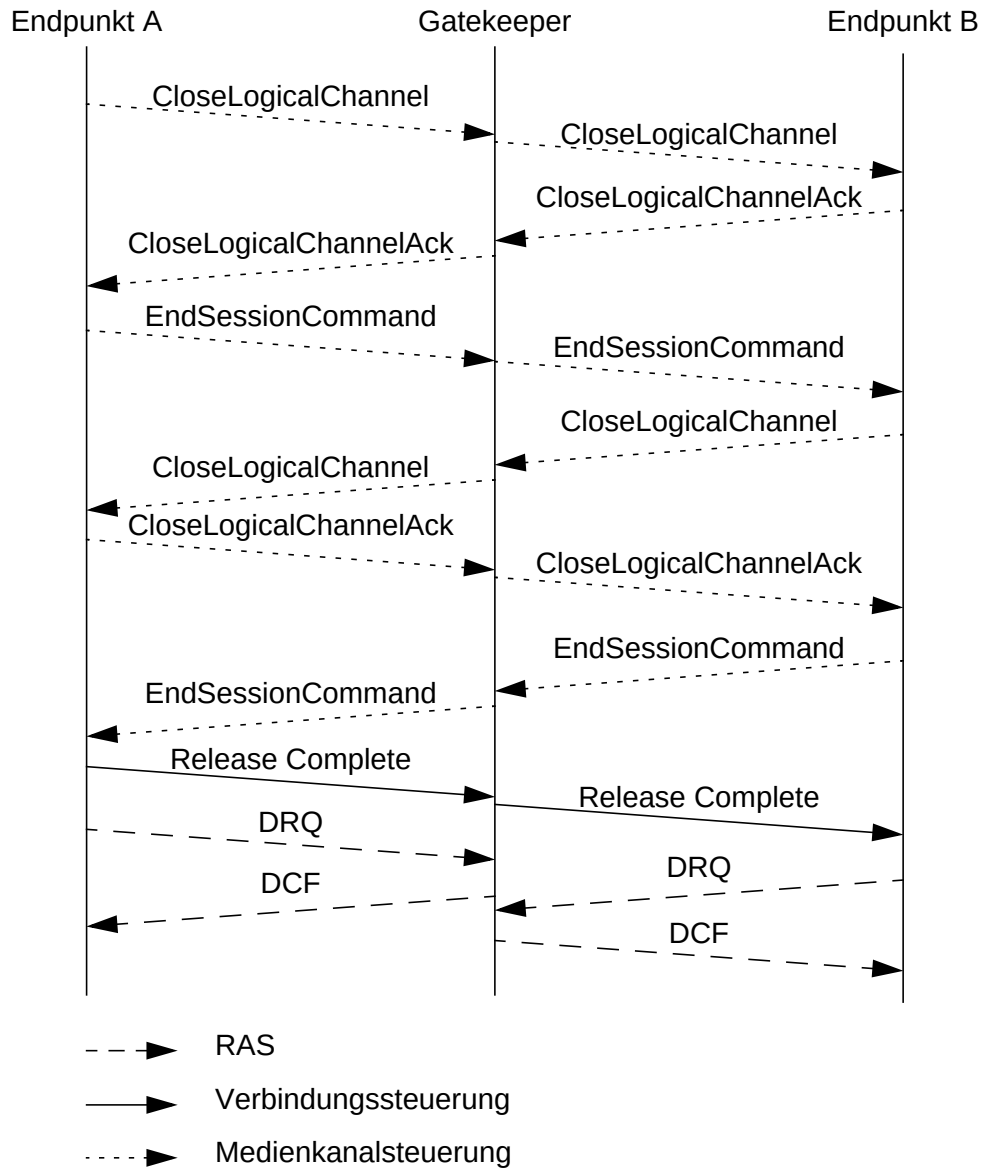


Bild 2.13: Signalisierprozeduren für den Verbindungsabbau bei H.323

ungesichertes Transportprotokoll verwendet, sind Wiederholungen von Nachrichten im Fehlerfall vorgesehen.

2.3.1.4 Qualitative Betrachtungen für Hoch- und Überlastsituationen

In diesem Abschnitt wird beschrieben, welche Vorkehrungen für den Hoch- und Überlastfall bei den H.323-Signalisierprotokollen vorgesehen sind.

Eine Hochlastsituation liegt vor, wenn sich die betrachtete Komponente nahe ihrer maximal vorgesehenen Belastung befindet. In einer Überlastsituation wurde diese Maximalbelastung überschritten, so dass es zu Störungen durch erhebliche Verzögerungen oder Verlust von Nachrichten kommen kann.

Es werden zwei Fälle von Hoch- und Überlast unterschieden:

- Hoch- und Überlast im Netz
  In diesem Fall befinden sich das Netz bzw. einzelne Netzknoten in Hoch- oder Überlast. Beispielsweise könnte die Kapazität eines Transportpfades oder eines Netzknotens nicht ausreichen. In diesem Fall würden einzelnen Nachrichten nicht rechtzeitig beim Ziel ankommen, da sie stark verzögert werden, oder es könnten Nachrichten während des Transports von einer H.323-Komponente zu einer anderen verloren gehen. Solche Fehlerfälle werden von einem gesicherten Transportprotokoll, wie es für die Verbindungssteuerung und die Medienkanalsteuerung verwendet wird, bemerkt und behoben, sofern die wiederholten Nachrichten im erlaubten Zeitfenster des entsprechenden Signalisierprotokolls ankommen. Bei der RAS-Signalisierung, die ein ungesichertes Transportprotokoll verwendet, das derartige Fehlerfälle nicht erkennt, werden für die einzelnen Anfragen Zeitüberwachungen eingesetzt. Wenn eine Anfrage nicht innerhalb der vorgegebenen Zeit beantwortet wird, wird die Anfrage wiederholt.

- Hoch- und Überlast in einzelnen H.323-Komponenten
  Wenn sich nicht das Netz, sondern einzelne H.323-Komponenten in Hoch- oder Überlast befinden, äußert sich das dadurch, dass Nachrichten nicht rechtzeitig oder gar nicht in der jeweiligen Komponente bearbeitet werden und somit die sendende Komponente eine verspätete oder gar keine Antwort erhält. Wenn der Empfang einer Nachricht von der Bearbeitung innerhalb der Komponente entkoppelt ist, werden derartige Fehlerfälle durch ein gesichertes Transportprotokoll nicht behoben. Dies ist z. B. der Fall, wenn das Transportprotokoll den Empfang einer Nachricht bestätigt, diese Nachricht an die darüberliegende Schicht weitergibt, diese jedoch die Nachricht verwerfen muss, da sie überlastet ist. Um diese Fehler zu beheben, müssten die darüberliegenden Protokollschichten entsprechende Verfahren, wie Zeitüberwachungen und Wiederholungen von Anfragen unterstützen. Bei der H.245-Signalisierung werden zwar Zeitüberwachungen durchgeführt, jedoch sind keine Wiederholungen von Anfragen vorgesehen. In der Empfehlung Q.931, auf der die Signalisierung für die Verbindungssteuerung basiert, sind ebenfalls Zeitüberwachungen definiert, wobei Wiederholungen nicht für alle Nachrichten vorgesehen sind. Bei einem ungesicherten Transportprotokoll äußert sich die Hoch- und Überlast in einzelnen Komponenten ebenso wie eine Hoch- und Überlastsituation im Netz, da die über der Transportschicht liegenden Protokolle für die Fehlererkennung und -behebung zuständig sind. Somit ergeben sich für die RAS-Signalisierung die gleichen Anmerkungen wie für den Fall der Hoch- und Überlast im Netz.

Bei der Betrachtung der einzelnen H.323-Signalisierprotokolle in Hoch- und Überlastsituationen lässt sich Folgendes feststellen:

- Bei der RAS-Signalisierung werden verspätete oder verloren gegangene Signalisiernachrichten durch Zeitüberwachungen erkannt. Dies führt zu einer Wiederholung der entsprechenden Anfrage, wobei die Anzahl der Wiederholungen begrenzt ist. Wenn die maximale Anzahl erreicht wird und die Zeitüberwachung abläuft, wird neben der Anfrage meist auch die gesamte H.323-Verbindung abgebrochen. Beispielsweise wird in [40] für die Gatekeeper-Zulassung für eine H.323-Verbindung die Zeit, bis eine Nachricht als verloren gegangen betrachtet wird, auf 3 Sekunden und die maximale Anzahl von Wiederholungen der Anfrage auf 2 festgelegt.
- Die Signalisierung zur Verbindungssteuerung verwendet ebenfalls Zeitüberwachungen, um die Einhaltung von Antwortzeiten zu regeln, wobei nur für die *Setup*-Nachricht eine Wiederholung vorgesehen ist. Wenn die wiederholte Nachricht ebenfalls nicht erfolgreich bearbeitet wird und somit keine entsprechende Antwort (z. B. *Call Proceeding*) erzeugt wird, wird der Verbindungsaufbau abgebrochen. Des Weiteren werden Nachrichten wie z. B. *Connect* und *Release Complete* nicht bestätigt, so dass deren Verlust durch Hoch- oder Überlast in einer Komponente nicht unbedingt erkannt werden würde, falls sie durch das Transportprotokoll als korrekt zugestellt betrachtet werden. Dies könnte zu inkonsistenten Verbindungszuständen in den beteiligten Komponenten führen.
- Bei der H.245-Signalisierung schließlich werden ebenfalls Zeitüberwachungen angewendet, um verspätete oder verloren gegangene Anfragen zu erkennen. Dabei sind jedoch keine Wiederholungen vorgesehen, so dass in diesem Fall die Verbindung beendet werden würde.

Wie aus diesen Ausführungen ersichtlich ist, kann bereits der Verlust bzw. die Verspätung einzelner Nachrichten das Beenden einer H.323-Verbindung hervorrufen, für die bereits Ressourcen verwendet wurden. Des Weiteren werden für den Auf- und Abbau von H.323-Verbindungen im Vergleich zu anderen Protokollen, wie z. B. bei der ISDN-Signalisierung, recht viele Nachrichten ausgetauscht. Dies führt dazu, dass im Falle von Hoch- und Überlastsituationen die Wahrscheinlichkeit, dass alle Nachrichten rechtzeitig bearbeitet werden, um somit einen erfolgreichen Verbindungsaufbau zu realisieren, geringer ist.

2.3.2 Unterschiede zur Steuerung in der kanalvermittelnden Telefonie

Nach der Vorstellung der H.323-basierten VoIP-Signalisierung werden in diesem Abschnitt die prinzipiellen Unterschiede der Steuerung für VoIP im Vergleich zur Steuerung in der kanalvermittelnden Telekommunikation vorgestellt. Dabei werden die für die Steuerung relevanten Unterschiede in den Bereichen Struktur einer entsprechenden Umgebung, Realisierung der Signalisierung sowie Steuerungsaufwand aufgezeigt.

Bei der Betrachtung der Struktur einer VoIP-Umgebung wird deutlich, dass eine logische Zuordnung zwischen den Endpunkten und den verwaltenden Komponenten besteht. Diese

logische Zuordnung kann durch entsprechende Signalisiertransaktionen geändert werden, um z. B. eine Entlastung zentraler Komponenten zu erreichen. Wenn man dazu beispielsweise die Struktur eines privaten Telefonsystems wie in Bild 2.14 dargestellt betrachtet, wird deutlich, dass dort eine statische Zuordnung zwischen den Endpunkten und der PBX bzw. den PBX-Modulen besteht. Eine Veränderung der Zuordnung ist somit kaum bzw. nur mit beträchtlichem Aufwand möglich.

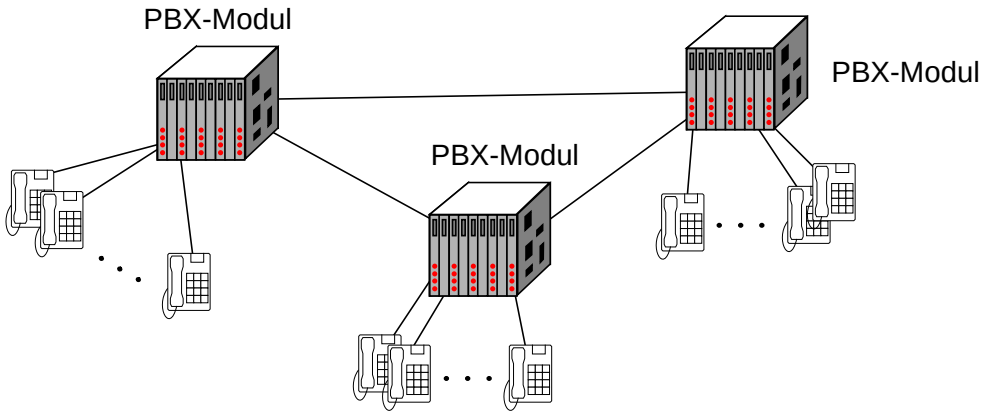


Bild 2.14: Struktur eines privaten Telefonsystems

Zur Realisierung der Signalisierung bei der kanalvermittelnden Telekommunikation wird beim ISDN die Außerband-Signalisierung angewandt, so dass an jedem Anschluss ein separater Signalisierkanal zur Verfügung steht, der unabhängig vom Nutzkanal ist. Durch diese strikte Trennung von Signalisier- und Nutzdaten können auch Signalisiertransaktionen im Falle belegter Nutzkanäle durchgeführt werden. Des Weiteren werden die Signalisiernachrichten für die einzelnen Übertragungsabschnitte zwischen den einzelnen Netzknoten gesichert übertragen, so dass verspätete oder verloren gegangene Nachrichten schnell erkannt und lokal wiederholt werden können. Im Gegensatz dazu wird für die VoIP-Signalisierung eine virtuelle Ende-zu-Ende Signalisierbeziehung eingerichtet.

Wie bei paketvermittelnden Netzen üblich, wird die zur Verfügung stehende Übertragungskapazität gemeinsam mit den Nutzdaten und mit weiteren Anwendungen verwendet. Jedoch kann durch geeignete Maßnahmen eine Priorisierung der Signalisiernachrichten gegenüber diesen anderen Paketen erreicht werden, so dass sich ähnliche Eigenschaften wie bei der Außerband-Signalisierung ergeben. Wie bereits beschrieben, verwendet die VoIP-Signalisierung entweder ein ungesichertes Transportprotokoll, so dass die Signalisierprotokolle selbst das Antwortverhalten überwachen müssen, oder ein gesichertes Ende-zu-Ende Transportprotokoll, das jedoch erhebliche Verzögerungen bei der Erkennung und Behebung verspäteter oder verloren gegangener Nachrichten aufweist.

Bei der Betrachtung des für die Steuerung notwendigen Aufwands ist von Bedeutung, dass bei der kanalvermittelnden Telefonie in der Regel nur ein Basisdienst angewendet wird, bei dem

sich die Parameter meist nicht ändern, und somit die Realisierung der Steuerung für diesen Dienst optimiert werden konnte. Dagegen können bei VoIP unterschiedliche Medien und bei diesen wiederum verschiedene Parameter, wie z. B. Codec, verwendet werden. Daher muss neben dem Verbindungsaufbau auch die Bestimmung der für die Kommunikation verwendeten Medien durchgeführt werden. Dabei sollten auch die zur Verfügung stehenden Ressourcen beachtet werden, damit zum einen die neue Kommunikationsbeziehung eine ausreichende Dienstgüte erfährt und zum anderen die Dienstgüte bestehender Kommunikationsbeziehungen nicht verringert wird. Darüber hinaus sind während einer Kommunikationsbeziehung Nachverhandlungen bezüglich der Kommunikationsparameter möglich, um z. B. die zur Verfügung stehende Übertragungskapazität zu erhöhen. Dies führt zu einem deutlich höheren Steuerungsaufwand bei VoIP im Vergleich zur kanalvermittelnden Telefonie, insbesondere wenn eine entsprechende Dienstgüte gewährleistet werden soll. Des Weiteren wird die Anzahl der verfügbaren Dienste bei VoIP weiter zunehmen und da der Steuerungsaufwand für die einzelnen Dienste unterschiedlich ist, ergibt sich eine inhomogenere Steuerlast als es bei der kanalvermittelnden Telefonie der Fall ist.

Wenn während einer bestehenden Kommunikationsbeziehung keine Änderungen der Kommunikationsparameter durchgeführt und keine zusätzlichen Dienste angewendet werden, ist der Steuerungsaufwand in dieser Phase bei VoIP niedriger, da die zentralen Steuerkomponenten, wie z. B. bei H.323 der Gatekeeper, beim Nutzdatenaustausch in der Regel nicht beteiligt sind, da dieser direkt zwischen den Endpunkten durchgeführt wird. Bei der kanalvermittelnden Telefonie werden die Nutzkanäle über die Vermittlungsstelle geführt, so dass diese während einer Verbindung zumindest die Weiterleitung der entsprechenden Nutzdaten durchführen muss.

Zusammenfassend lässt sich feststellen, dass die Hauptunterschiede der Steuerung bei VoIP gegenüber der kanalvermittelnden Telefonie im höheren und inhomogeneren Steuerungsaufwand während der Signalisierphasen und in der dynamischen Veränderbarkeit der Struktur der VoIP-Umgebungen liegen. Darüber hinaus sind die zentralen VoIP-Steuerkomponenten vom Nutzdatenaustausch selbst nicht betroffen. Diese Merkmale sollen für die Betrachtungen und Untersuchungen der folgenden Kapitel verwendet werden, um die Leistung einer VoIP-Umgebung bezüglich der Steuerung zu optimieren.

Kapitel 3

Optimierte Steuerung für H.323-basierte VoIP-Kommunikationsnetze

In diesem Kapitel wird beschrieben, wie durch eine optimierte Steuerung für H.323-basierte VoIP-Kommunikationsnetze eine möglichst effiziente Nutzung der Ressourcen erreicht werden kann. Damit soll eine maximale Leistung dieses Netzes über große Lastbereiche hinweg und in unterschiedlichen Szenarien erzielt werden. Die vorgestellten Verfahren werden für eine H.323-basierte Umgebung abgeleitet, wobei sie zumindest teilweise auch in einer SIP-Umgebung Anwendung finden könnten.

Um eine optimierte Steuerung durchzuführen, muss zunächst festgelegt werden, was optimiert werden soll, d. h. welche Leistungsgrößen Ziel der entsprechenden Maßnahmen sein sollen. Daher werden in Abschnitt 3.1 mögliche Leistungsdefinitionen für VoIP-Umgebungen vorgestellt. Die prinzipielle Durchführung der Optimierung, d. h. wie und welche grundsätzlichen Maßnahmen angewendet werden, wird in Abschnitt 3.2 beschrieben. In Abschnitt 3.3 werden die in dieser Arbeit vorgestellten Maßnahmen und Verfahren eingeordnet und weitere Untersuchungen, die im Bezug zu dieser Arbeit stehen, präsentiert. In Abschnitt 3.4 werden Maßnahmen der Steuerung für die Optimierung der Nutzung verschiedener Ressourcen, wie z. B. der Gateways oder MCUs einer VoIP-Umgebung, beschrieben. Ein Schwerpunkt dieser Arbeit ist die Optimierung der Nutzung der Steuerungsressourcen und insbesondere des Gatekeepers, der eine zentrale Ressource in einer H.323-basierten VoIP-Umgebung darstellt. Die dabei anwendbaren Verfahren und Maßnahmen werden in Abschnitt 3.5 vorgestellt. Schließlich werden in Abschnitt 3.6 mögliche Verfahren zur optimierten Steuerung eines integriert verwalteten Unternehmensnetzes, bei dem die VoIP-Kommunikation und die Datenkommunikation gemeinsam gesteuert werden, beschrieben.

3.1 Leistungsdefinition

Im Folgenden werden verschiedene Leistungsgrößen vorgestellt, die als Zielkriterien für die optimierte Steuerung Anwendung finden können. Dabei werden zunächst elementare Kenngrößen für die Leistungsdefinition beschrieben, bevor anschließend kombinierte Kenngrößen aus diesen elementaren abgeleitet werden. Des Weiteren kann die Bestimmung der Kenngrößen getrennt nach Klassen erfolgen, wobei eine Klasse z. B. durch einen Dienst oder durch eine Benutzergruppe festgelegt ist. Damit kann die Steuerungsoptimierung auf einer differenzierten Leistungsdefinition basierend erfolgen, so dass z. B. die einzelnen Klassen eine unterschiedliche Gewichtung erhalten und entsprechend unterschiedlich durch die Steuerung behandelt werden. Eine Übersicht über verschiedene Leistungsdefinitionen kann z. B. in [31] gefunden werden.

- Durchsatzrate (*Throughput*)
  Für einen Netzbetreiber ist die Durchsatzrate von Dienstanforderungen eine wichtige Kenngröße. Dabei wird die Anzahl der erfolgreich bearbeiteten Anforderungen pro Zeiteinheit und pro Klasse bestimmt. Die Gesamtdurchsatzrate errechnet sich aus der Summe der Durchsatzraten der einzelnen Klassen.

- Antwortverzögerung (*Delay*)
  Im Gegensatz zur Durchsatzrate ist für den einzelnen Benutzer die Antwortverzögerung auf Dienstanforderungen von Bedeutung. Wenn die Antwortverzögerung zu groß wird, könnte der Benutzer dies als Systemfehler oder gar als Systemausfall interpretieren. Als Leistungsgröße wird der arithmetische Mittelwert der Antwortverzögerungen einer Klasse verwendet. Für die Gesamtantwortverzögerung aller Klassen kann beispielsweise der arithmetische Mittelwert über den Antwortverzögerungen der einzelnen Klassen berechnet werden.

- Verlustrate (*Loss Rate*)
  Eine weitere Kenngröße ist die Verlustrate, die die Anzahl der fehlgeschlagenen Dienstanforderungen pro Zeiteinheit angibt. Dabei werden jedoch die durch entsprechende Maßnahmen abgelehnten Anforderungen nicht miteinbezogen, sondern ausschließlich die Anforderungen betrachtet, die wegen zu großer Last nicht oder nicht rechtzeitig bearbeitet werden konnten. Diese Leistungsgröße wird nicht so oft verwendet wie die Durchsatzrate und die Antwortverzögerung, wobei sie aber trotzdem von Interesse ist, da bei einer fehlgeschlagenen Anforderung die Unzufriedenheit des Benutzers höchstwahrscheinlich größer ist, als wenn er für eine abgelehnte Anforderung eine Mitteilung über eine vorübergehende Störung erhält. Daher kann die Minimierung der Verlustrate ein Ziel der Steuerungsoptimierung sein. Zur Bestimmung der Gesamtverlustrate über alle Klassen hinweg können die Verlustraten der einzelnen Klassen gewichtet aufsummiert werden.

Neben den genannten elementaren Leistungsgrößen existieren noch weitere, wie z. B. in [73] beschrieben, die hier jedoch nicht weiter betrachtet werden sollen, da sie für die weiteren Untersuchungen nicht relevant sind.

Wie bereits erwähnt, sind für Benutzer und Netzbetreiber unterschiedliche elementare Leistungsgrößen von Bedeutung, die jedoch nicht immer gemeinsam optimiert werden können. So ergibt sich z. B. bei einem Wartesystem eine relativ hohe mittlere Antwortverzögerung, wenn die Durchsatzrate maximal ist, da für eine maximale Durchsatzrate immer eine Anforderung, die auf ihre Bearbeitung wartet, zur Verfügung stehen muss, wenn die Bearbeitung der vorhergehenden beendet wurde. Um einen geeigneten Kompromiss zu finden, werden daher entsprechende Kombinationen aus den elementaren Leistungsgrößen gebildet.

- Kosten- bzw. Gewinnfunktion (*Cost* bzw. *Benefit of Operation*)
  Diese Leistungsdefinition ist von der Unternehmensforschung (*Operations Research*) abgeleitet. Dabei werden nach [113] die *Kosten* für die Bearbeitung der Anforderungen dem *Gewinn* durch den erfolgreichen Abschluss der Bearbeitung gegenübergestellt:

$$C = \lambda_x \cdot \alpha - \lambda_y \cdot \beta \tag{3.1}$$

  C Kostendichte (Kosten pro Zeiteinheit)
  λ_x Mittlere Anzahl eintreffender Anforderungen pro Zeiteinheit (Angebotsrate)
  λ_y Mittlere Anzahl erfolgreicher Anforderungen pro Zeiteinheit (Durchsatzrate)
  α Kosten pro Anforderung
  β Gewinn je erfolgreicher Anforderung

  Um diese Leistungsgröße zu optimieren, muss die Funktion nach Gl. (3.1) minimiert werden. Darüber hinaus können weitere Kosten miteinbezogen werden, wie z. B. die Kosten für fehlgeschlagene oder für abgelehnte Anforderungen. Dabei erfolgt eine entsprechende Gewichtung der Kosten durch die Definition der Kostenfaktoren. Zur Realisierung verschiedener Klassen können jeweils unterschiedliche Kostenfaktoren festgelegt werden. Beispielsweise kann der Gewinn für die Bearbeitung der Anforderungen einer Klasse höher festgelegt werden, als für eine andere, so dass die Anforderungen der Klasse mit dem höheren Gewinn mit einer größeren Wahrscheinlichkeit bearbeitet werden.

- Power
  Die Leistungsgröße *Power* wurde u. a. in [60] und [99] verwendet, um einen möglichst guten Kompromiss zwischen der Minimierung der Antwortverzögerungen und der Maximierung der Durchsatzrate zu erreichen. Sie wird aus dem Quotient der mittleren Durchsatzrate λ_y und der mittleren Antwortverzögerung $\overline{D}$ gebildet:

$$P = \frac{\lambda_y}{\overline{D}} \tag{3.2}$$

Falls keine Antwortverzögerungen vorliegen sollten, kann entweder die Größe Power nicht bestimmt werden, oder es muss ein Schätzwert verwendet werden.

In [61] wird eine Erweiterung der Power-Definition vorgenommen, bei der auch fehlgeschlagene Anforderungen betrachtet werden. Dabei wird neben der Verlustwahrscheinlichkeit B die mittlere Bedienzeit h und die Auslastung $\rho = \lambda_y \cdot h / m$ (m entspricht der Anzahl der Bedieneinheiten) miteinbezogen:

$$P_B = \frac{\rho \cdot (1 - B)}{\overline{D} / h} \tag{3.3}$$

Die Größe Power kann auch für einzelne Klassen berechnet werden. Um eine gemeinsame Leistungsgröße aus diesen Werten zu bestimmen, werden nach [99] entweder das Power-Produkt nach Gl. (3.4) oder die Power-Summe nach Gl. (3.5) verwendet. Dabei ist zu beachten, dass beim Power-Produkt keine Anteile verwendet werden, bei denen keine Anforderung der entsprechenden Klasse erfolgreich bearbeitet wurde, da der entsprechende Power-Wert unbestimmt ist. Bei der Power-Summe tritt dieses Problem nicht auf. Des Weiteren können Gewichtungen der einzelnen Summanden (g_ν) und damit der entsprechenden Klassen vorgenommen werden.

$$P_P = \prod_{\nu \mid \lambda_{y\nu} > 0} \frac{\lambda_{y\nu}}{\overline{D}_\nu} = \prod_{\nu \mid \lambda_{y\nu} > 0} P_\nu \tag{3.4}$$

$$P_S = \sum_\nu g_\nu \cdot \frac{\lambda_{y\nu}}{\overline{D}_\nu} = \sum_\nu g_\nu \cdot P_\nu \tag{3.5}$$

Die bisher genannten Leistungsgrößen stellen nur Mittelwerte über einen größeren Zeitbereich dar, die für stationäre Betrachtungen verwendet werden können. Für die Bestimmung des Reaktionsverhaltens eines Systems sind darüber hinaus instationäre Betrachtungen notwendig, die die Untersuchung transienter Vorgänge erlauben. In [73] werden die dazu notwendigen Erweiterungen für die Bestimmung der elementaren Leistungsgrößen beschrieben: Prinzipiell werden die einzelnen Leistungsgrößen als zeitabhängig betrachtet. Daher werden sie jeweils für entsprechend kleine Zeitintervalle bestimmt. Die Intervalldauer darf dabei nicht zu klein gewählt werden, damit der Aufwand zur Bestimmung der Leistungsgrößen nicht zu groß wird. Jedoch darf die Intervalldauer auch nicht zu groß sein, da ansonsten die instationären Effekte nicht deutlich werden. Diese über die Intervalle gemittelten Leistungsgrößen können dann zur Untersuchung des instationären Verhaltens, z. B. als Reaktion auf einen sprunghaften Anstieg der Systemlast, verwendet werden.

3.2 Prinzipieller Ablauf der Steuerungsoptimierung

In diesem Abschnitt wird das prinzipielle Vorgehen zur Steuerungsoptimierung in einer VoIP-Umgebung beschrieben. Dazu wird eine entsprechende Steuerung der anfallenden Last, die aus Signalisier- und Nutzdaten bestehen kann, durchgeführt. Der Ablauf für diese Steuerung gliedert sich in drei Teile, die in den folgenden Abschnitten beschrieben werden:

- Zunächst muss, wie in Abschnitt 3.2.1 vorgestellt wird, die aktuelle Belastung der betrachteten Komponenten mittels geeigneter Lastindikatoren ermittelt werden.
- Wenn mehrere Komponenten zur Verfügung stehen, die eine Anforderung bedienen können, ist eine Verteilung der entsprechenden Last, wie in Abschnitt 3.2.2 beschrieben, auf diese Komponenten lohnenswert.
- Wenn schließlich die Last so groß ist, dass eine erfolgreiche Bearbeitung aller Anforderungen nicht möglich ist, müssen Überlastabwehrmaßnahmen, wie sie in Abschnitt 3.2.3 beschreiben werden, angewendet werden.

Eine gemeinsame Anforderung an die einzelnen Verfahren dieser drei Teile ist ein möglichst geringer Ressourcenverbrauch durch die Verfahren selbst, da sie gerade in Hoch- und Überlastsituationen wirken sollen, in denen die wenigen freien Ressourcen nicht durch die Steuerungsoptimierung belegt sein sollten. Darüber hinaus ist die Reaktionsfähigkeit von Bedeutung, so dass schnell genug auf Laständerungen reagiert wird, wobei kurzzeitige Lastimpulse nicht oder nur kaum beachtet werden sollten.

3.2.1 Bestimmung des aktuellen Lastzustands - Lastindikatoren

Bevor entsprechende Maßnahmen zur Lastverteilung oder zur Überlastabwehr ergriffen werden, muss zunächst der aktuelle Lastzustand einer Komponente ermittelt werden. Dies erfolgt über Lastindikatoren, die beispielsweise aus Systeminformationen den Zustand der Komponente bezüglich ihrer Belastung ableiten. Im Folgenden wird dazu zunächst in Abschnitt 3.2.1.1 das Prinzip der Lastzustandsermittlung vorgestellt, bevor in Abschnitt 3.2.1.2 die Filterung von Kenngrößen beschrieben wird.

3.2.1.1 Prinzip

Zur Bestimmung der aktuellen Belastung einer Komponente werden zur Laufzeit Indikatorwerte ermittelt, die z. B. aus Messungen gewonnen werden können. Grundsätzlich stehen nach [3] folgende Verfahren zu ihrer Ermittlung zur Verfügung:

- Messungen im Kommunikationsnetz und in der Komponente,
- Auswertung interner Zustandsinformationen der Komponente,

- Auswertung von Meldungen anderer Komponenten.

Die so ermittelten Indikatorwerte werden entweder direkt zur Bestimmung des Lastzustands herangezogen, oder sie werden mit anderen Größen geeignet zu indirekten bzw. abgeleiteten Indikatoren verknüpft. Direkte Lastindikatoren sind z. B. die Anzahl der Nachrichten in einer Warteschlange oder die Antwortverzögerung auf eine Anfrage, ein indirekter Lastindikator ist z. B. der Gradient einer Warteschlangenbelegung, da dieser aus Warteschlangenbelegung und entsprechender Dauer dieser Belegung ermittelt wird.

Lastindikatoren, wie z. B. die Anzahl der Nachrichtenwiederholungen oder die Antwortverzögerung werden in der Regel über ein Zeitintervall gemittelt angegeben. Diese zeitgesteuerte Lastindikatorermittlung führt zu Ungenauigkeiten bei der Bestimmung des aktuellen Systemzustands, die abhängig von der Länge des Intervalls sind. Des Weiteren können Indikatoren ereignisgesteuert ermittelt werden, d. h. der Indikatorwert wird beim Auftreten eines Ereignisses, z. B. beim Empfang einer Nachricht, aktualisiert. Ein Beispiel für eine ereignisgesteuerte Ermittlung eines Lastindikators ist die Ermittlung der Anzahl der Nachrichten in einer Warteschlange. Verschiedene Verfahren der zeit- und ereignisgesteuerten Indikatorermittlung werden beispielsweise in [99] vorgestellt.

In der Regel wird der gesamte Lastbereich in mehrere Laststufen unterteilt, denen jeweils ein Lastzustand zugeordnet ist. Dieser wird abhängig vom Wert des Lastindikators eingenommen. Dazu werden die einzelnen Lastzustände in der Regel nummeriert angegeben, wobei in dieser Arbeit der Lastzustand *Null* keine Überlastung anzeigt und höhere Belastungen höheren Nummern der Lastzustände entsprechen. Die Maßnahmen der Lastverteilung und der Überlastabwehr erfolgen dann entsprechend dieser Lastzustände. Beispielsweise könnte die Überlastabwehrmaßnahme einer Komponente, die sich im höchsten Lastzustand befindet, alle Verbindungsanforderungen ablehnen.

Um die Oszillation der Lastzustände gering zu halten, werden die Wertebereiche der einzelnen Laststufen meist überlappend definiert, so dass Hysterese-ähnliche Effekte erzielt werden. Dies wird in Bild 3.1 anhand eines Beispieles dargestellt.

3.2.1.2 Filterung von Kenngrößen

Um die Auswirkungen von statistischen Schwankungen der Belastung zu minimieren und um Schätzungen für den Verlauf der Belastung einer Komponente vorzunehmen, können Schätz- und Filterverfahren angewendet werden, wie es z. B. in [100] beschrieben wird. Diese Verfahren sollen einen Kompromiss zwischen der *Reagibilität*, d. h. der Eigenschaft, Laständerungen möglichst schnell anzuzeigen, und der *Stabilität* der Lastzustände, d. h. dass kleine oder kurzzeitige Laständerungen, die keine Bedeutung für den Systemzustand haben, ignoriert werden, erreichen. Des Weiteren darf die Ausführung der Filterung wenig Ressourcen benötigen, um das System nicht weiter zu belasten.

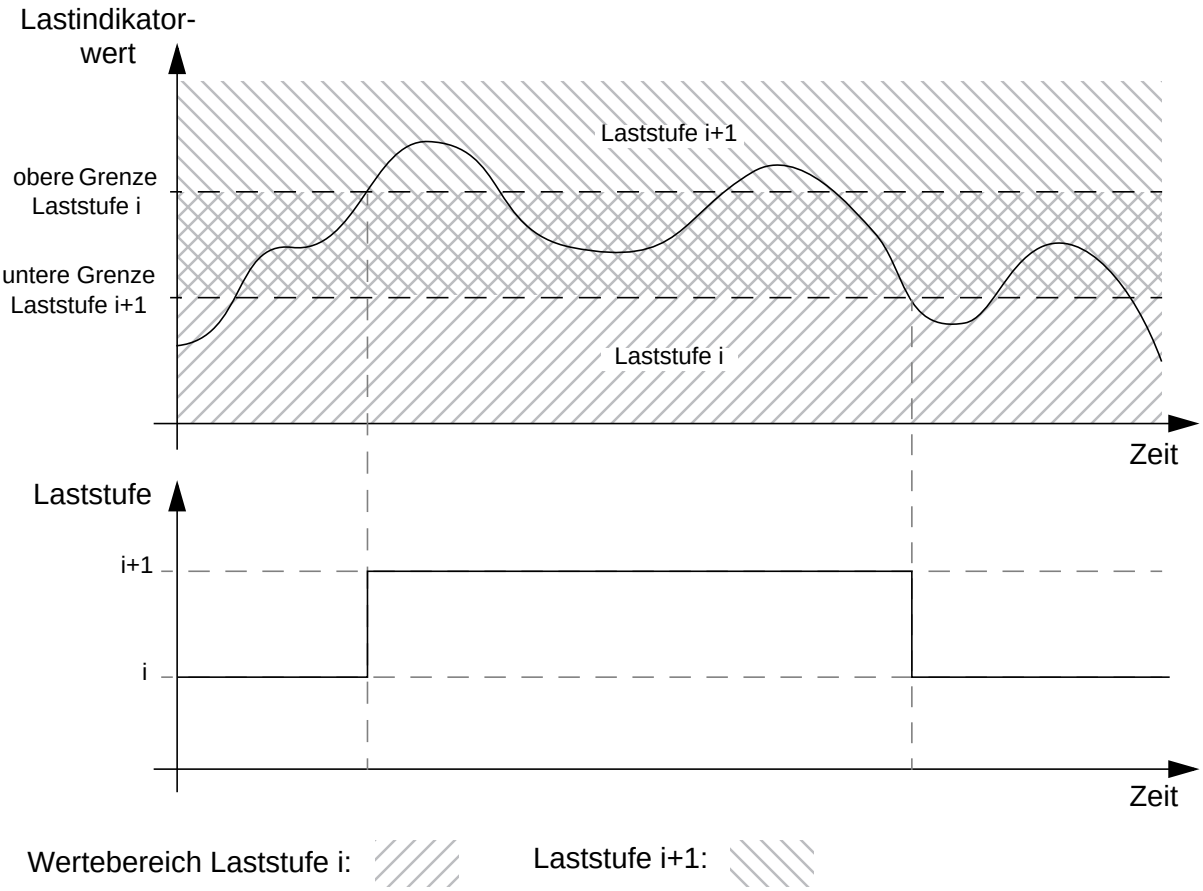


Bild 3.1: Hysterese-Effekt bei der Zuordnung von Lastindikatorwerten zu Laststufen

In der Kommunikationstechnik sind vor allem die im Folgenden beschriebenen Filterverfahren von Bedeutung. In den nachfolgenden Gleichungen stellt $\hat{x}(k)$ den Lastindikatorwert des Intervalls k und $\tilde{x}(i)$ den Messwert im Intervall i dar.

- Gleitender Mittelwert (*Moving Average*) und gewichteter gleitender Mittelwert (*Weighted Moving Average*)

  Bei diesen beiden Verfahren werden die letzten N Messwerte für die Indikatorermittlung herangezogen. Beim gleitenden Mittelwert nach Gl. (3.6) werden die einzelnen Messwerte gleichgewichtet, während beim gewichteten gleitenden Mittelwert, Gl. (3.7), den einzelnen Messwerten jeweils ein Gewichtungsfaktor zugeordnet ist.

$$\hat{x}(k) = \frac{1}{N} \cdot \sum_{i = k - N + 1}^{k} \tilde{x}(i) \text{ bzw.} \tag{3.6}$$

$$\hat{x}(k) = \sum_{i = k - N + 1}^{k} a_{i - k + N} \cdot \tilde{x}(i) \text{ mit } \sum_{j = 1}^{N} a_j = 1 \tag{3.7}$$

- Exponentielles Glätten (*Exponential Smoothing*)
  Das exponentielle Glätten zählt zu den Verfahren der Mittelwertbildung mit unendlichem, nachlassendem Gedächtnis. Der Lastindikatorwert kann dabei mittels der folgenden Gleichung (3.8) bestimmt werden.

$$\hat{x}(k) = \xi \cdot \tilde{x}(i) + (1 - \xi) \cdot \hat{x}(k-1) \text{ mit Glättungsfaktor (}\textit{Smoothing Factor}\text{) } \xi \qquad (3.8)$$

Da die Eigenschaften des exponentiellen Glättens sehr gut sind und seine Realisierung sowie seine Ausführung keinen großen Aufwand benötigen, wird dieses Filterverfahren in vielen Anwendungen benutzt ([99]). Wenn jedoch länger anhaltende Überlastsituationen erkannt werden sollen, wird zur Filterung eher die Mittelwertbildung mit großem N verwendet.

3.2.2 Lastverteilung

Durch die Lastverteilung soll eine möglichst effiziente Nutzung der zur Verfügung stehenden Ressourcen erreicht werden, indem die auftretende Last auf Komponenten verteilt wird, die bezüglich ihrer Funktionalität in der Lage sind, diese Last zu bearbeiten. Damit können z. B. mehrere weniger leistungsfähige Komponenten zu einer sehr leistungsfähigen virtuellen Plattform zusammengefasst werden, wie es beispielsweise bei der Bildung sog. *Cluster* von Web-Servern durchgeführt wird. Eine sehr gute Übersicht über Lastverteilung und dabei angewandte Verfahren stellt [103] dar.

Eine Klassifizierung der bei der Lastverteilung angewandten Verfahren kann anhand der in [15] vorgestellten Kriterien erfolgen. Dabei wird zunächst zwischen statischen und dynamischen Lastverteilungsverfahren unterschieden.

Bei statischen Verfahren werden vor dem Start des Systems die Entscheidungen getroffen, wie die lastverursachenden Komponenten den lastaufnehmenden Komponenten zugeordnet sind. Diese Zuordnung wird während der Systemlaufzeit nicht mehr geändert. Um die Last möglichst effektiv zu verteilen, werden für die statischen Verfahren aufwendige Berechnungen zur Bestimmung einer geeigneten Zuordnung angewandt. Eine Untersuchung eines statischen Lastverteilungsverfahrens ist z. B. in [104] zu finden. Ein Vorteil statischer Verfahren ist, dass nahezu keinerlei Ressourcen während ihrer Anwendung benötigt werden.

Bei dynamischen Lastverteilungsverfahren wird während der Systemlaufzeit entschieden, wie die zu bearbeitenden Anforderungen auf die lastaufnehmenden Komponenten verteilt werden. Daher dürfen die notwendigen Berechnungen nicht zu aufwendig sein, um das System nicht zu sehr zusätzlich zu belasten. Der Vorteil der dynamischen Verfahren liegt in ihrer Anpassungsfähigkeit an sich dynamisch ändernde Situationen, die sich z. B. durch Laständerungen oder auch durch Ausfall einer lastaufnehmenden Komponente ergeben können. Des Weiteren ist wenig a priori Wissen über die zu verteilende Last notwendig.

Die dynamischen Lastverteilungsverfahren können weiter in zentrale und verteilte Verfahren unterteilt werden. Bei den zentralen Verfahren verteilt eine zentrale Instanz die Last auf die Komponenten. Dabei kann sie Informationen dieser Komponenten verwenden, um deren aktuellen Lastzustand abzuleiten. Bei den verteilten Verfahren erfolgt die Lastverteilung zwischen den Komponenten ohne eine zentrale Instanz. Ein Vorteil der zentralen Verfahren ist, dass die lastaufnehmenden Komponenten meist nichts oder sehr wenig über die Lastverteilung selbst wissen müssen. So können z. B. bei der Bildung von Web-Server-Cluster mit zentraler Lastverteilung Standard-Web-Server verwendet werden, die über keinerlei zusätzliche Funktionalität bezüglich der Lastverteilung verfügen. Ein Nachteil der zentralen Verfahren ist jedoch die zusätzlich benötigte Instanz zur Lastverteilung. Zum einen stellt sie einen möglichen Engpass (*Bottleneck*) für die Bearbeitung der Anfragen dar, da die gesamte Last durch sie verteilt werden muss. Zum anderen würde beim Ausfall dieser zentralen Instanz das ganze System ausfallen. Ein bekanntes zentrales Verfahren ist das *Round-Robin*-Verfahren, bei dem die lastaufnehmenden Komponenten zyklisch bei jeder ankommenden Anforderung gewechselt werden.

Schließlich können die verteilten dynamischen Lastverteilungsverfahren noch in kooperierende (*cooperative*) und nicht-kooperierende (*non-cooperative*) Verfahren unterschieden werden. Bei den kooperierenden Verfahren tauschen die lastaufnehmenden Komponenten Informationen aus, so dass der Lastzustand aller Komponenten bei der Lastverteilung berücksichtigt werden kann. Im Gegensatz dazu entscheiden bei den nicht-kooperierenden Verfahren die Komponenten über die Lastverteilung, ohne die Zustände der jeweils anderen Komponenten miteinzubeziehen. Der Nachteil der kooperierenden Verfahren liegt in dem notwendigen Informationsaustausch zwischen den lastaufnehmenden Komponenten. Jedoch ist dadurch eine näher dem Optimum liegende Lastverteilung möglich, da der aktuelle Lastzustand der anderen Komponenten entsprechend berücksichtigt werden kann. Ein Beispiel für ein kooperierendes Verfahren ist das *Receiver-initiated*-Verfahren, bei dem eine lastaufnehmende Komponente den anderen mitteilt, wenn sie durch eine entsprechend niedrige Auslastung in der Lage ist, weitere Anforderungen zu bearbeiten. Wenn eine Anforderung von einer stark belasteten, lastaufnehmenden Komponente weitergegeben werden soll, erfolgt dies nur an eine dieser *Receiver*-Komponenten. Das *Random*-Verfahren ist dagegen ein Beispiel für ein nicht-kooperierendes Verfahren, bei dem eine stark belastete, lastaufnehmende Komponente Anforderungen an jeweils zufällig ausgewählte andere Komponenten weitergibt.

Wenn das Ziel eines Lastverteilungsverfahrens eine möglichst gleichmäßige Belastung der einzelnen lastaufnehmenden Komponenten ist, wird dies als Lastausgleichsverfahren (*Load Balancing*) bezeichnet. Wenn nur eine Aufteilung der Last Ziel des Verfahrens ist, wird dies als Lastteilung (*Load Sharing*) bezeichnet.

Ein Problem der Lastverteilung ist der Zugriff auf Daten, die zur Bearbeitung der Anforderung notwendig sind, die nicht in der Anforderung selbst enthalten sind, und die Sicherstellung der

Konsistenz dieser Daten. Beispielsweise muss bei einer Verbindungsaufbauanforderung die Berechtigung des Teilnehmers für diese Anforderung überprüft werden. Des Weiteren werden für die Bearbeitung der Signalisiernachrichten die Zustände der jeweiligen Protokollautomaten benötigt, da sie ansonsten nicht korrekt interpretiert werden können. Zur Lösung dieses Problems können z. B. entsprechende Datenbanken verwendet werden oder Verfahren, die in gemeinsam genutzten Dateisysteme [102] oder verteilten *Web-Cache*-Architekturen [29, 127] Verwendung finden, in adaptierter Form benutzt werden.

3.2.3 Überlastabwehr

Eine Überlastung einer Komponente liegt vor, wenn die zu bearbeitende Last so groß ist, dass sie nicht vollständig erfolgreich bearbeitet werden kann. Wenn diese Überlastung länger anhält, so dass auch durch entsprechende Pufferung der ankommenden Anforderungen die Überlastsituation nicht aufgelöst werden kann, sind Überlastabwehrmaßnahmen sinnvoll, die gegebenenfalls Anforderungen ablehnen oder verwerfen, so dass die Komponente entlastet wird.

Wenn bei zu großer Last keine Überlastabwehrmaßnahmen durchgeführt werden, werden Ressourcen für Anforderungen aufgewendet, die im weiteren Verlauf fehlschlagen oder durch die erhebliche Verzögerung von der anfordernden Komponente nicht mehr akzeptiert werden. Dies wird als Blindlast bezeichnet, da die dabei aufgewendeten Ressourcen verschwendet wurden. Das Ziel der Überlastabwehr ist es, diese Blindlast möglichst gering zu halten, indem Anforderungen entweder möglichst früh im Laufe ihrer Bearbeitung abgelehnt oder komplett und rechtzeitig bearbeitet werden. Dieser Sachverhalt ist in Bild 3.2 dargestellt.

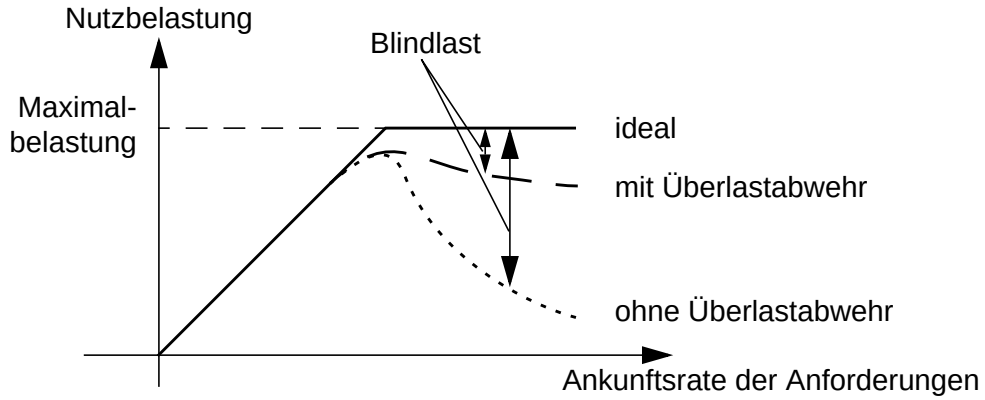


Bild 3.2: Exemplarischer Verlauf der Nutzbelastung über der Ankunftsrate mit und ohne Anwendung von Überlastabwehrmaßnahmen

Neben der Effektivität der Überlastabwehrmaßnahmen ist ein weiteres Kriterium ihre Reaktionsschnelligkeit. Dies bedeutet, dass sie angemessen schnell auf Änderungen der Lastsituation reagieren müssen, damit keine Ressourcen für später abgelehnte Verbindungen verschwendet

werden und die Überlastsituation in der Komponente nicht länger als notwendig anhält, da beispielsweise zu spät damit begonnen wurde, Anforderungen abzulehnen.

Schließlich sollen die Überlastabwehrmaßnahmen fair gegenüber den lastverursachenden Komponenten und verschiedenen Diensten sein. Dabei gibt es unterschiedliche Interpretationen von Fairness: Eine Interpretation wäre eine möglichst gleichmäßige Verteilung der zur Verfügung stehenden Ressourcen, so dass z. B. allen Diensten ein gleich großer Anteil zur Verfügung steht. Eine andere Interpretation wäre eine gleichmäßige Ablehnungswahrscheinlichkeit für ankommende Anforderungen. In [59] wird vorgeschlagen, dass nur Anforderungen des für die Überlastsituation verantwortlichen Dienstes bzw. der verantwortlichen Komponente abgelehnt werden, so dass die anderen Dienste bzw. Komponenten von diesen Ablehnungen nicht betroffen sind. Des Weiteren können verschiedene Klassen von Diensten oder Teilnehmern definiert werden, die entsprechend unterschiedlich von einer Überlastabwehrmaßnahme behandelt werden, so dass z. B. für eine bestimmte Klasse die Ablehnungswahrscheinlichkeit in Überlastsituationen geringer als bei anderen Klassen ist, um sicherheitskritische Dienste (z. B. für Notrufe) zu realisieren. Bei der Unterstützung dieser Fairnesskriterien ist zu beachten, dass die entsprechenden Verfahren mehr Ressourcen während der Ausführung benötigen, da komplexere Algorithmen bearbeitet und mehr Daten über die einzelnen Dienste und Komponenten gespeichert werden müssen.

Beispiele für Überlastabwehrmaßnahmen sind z. B. die „Prozentuale Drosselung“ (*Percentage Throttling*), bei der je nach Lastzustand der Komponente ein bestimmter Prozentsatz der Anforderungen abgelehnt wird, oder das *Automatic Call Gapping* (ACG), bei dem nach jeder bearbeiteten Anforderung für ein vorgegebenes Zeitintervall alle weiteren Anforderungen abgelehnt werden.

3.3 Einordnung der Arbeit

In diesem Abschnitt werden die im Verlauf dieser Arbeit abgeleiteten Verfahren und Untersuchungen eingeordnet. Dabei werden verwandte Fragestellungen sowie bestehende Untersuchungen, die Teilaspekte dieser Arbeit betreffen, vorgestellt.

Da VoIP aus der Konvergenz der Tele- und der Datenkommunikation entstand, können Erkenntnisse aus beiden Bereichen Verwendung finden. Daher werden im Folgenden bekannte Ansätze, die sowohl in der Tele- als auch in der Datenkommunikation entstanden, für die Lastverteilung und für die Überlastabwehr präsentiert. Des Weiteren werden für VoIP-Umgebungen entwickelte Ansätze und Untersuchungen vorgestellt.

- Lastverteilung in der Telekommunikation
  In der Telekommunikation werden Lastverteilungsverfahren vor allem innerhalb einzelner komplexerer Komponenten und Systeme angewandt. Dabei werden einzelne Aufgaben,

z. B. der Steuerung, auf entsprechende Prozessoren dieser Komponente verteilt [67, 99]. Da dies innerhalb eines physikalischen Systems stattfindet, ist nur eine begrenzte Skalierbarkeit gegeben. Eine Ausnahme bildet das u. a. in [57] beschriebene *System 12* der Firma *Alcatel*, das eine verteilte Steuerung realisiert, bei dem z. B. erst bei Bedarf ein Modul zur Bearbeitung einer Anforderung bestimmt wird.

Innerhalb des IN senden die *Service Switching Points* (SSP), die sich in entsprechenden Vermittlungsstellen befinden, Anfragen an die dienstbearbeitenden Komponenten, die *Service Control Points* (SCP). Diese bearbeiten die Anfrage, wobei dazu in der Regel weitere Nachrichten zwischen SSP und SCP ausgetauscht werden müssen. Da mehrere SCP für den gleichen Dienst zuständig sein können, kann der SSP eine entsprechende Lastverteilung durchführen. In [55] wird dazu ein Verfahren für eine optimierte Steuerung mehrerer IN-Ressourcen vorgestellt, wobei neben einer Lastverteilung der Anforderungen auf mehrere SCPs eine Überlastabwehrmaßnahme für die SCPs angewendet wird.

- Überlastabwehr in der Telekommunikation
  In der Telekommunikation sind Überlastabwehrmaßnahmen sehr verbreitet. Vor allem in öffentlichen Telekommunikationsnetzen verfügen nahezu alle Systeme über eine entsprechende Funktionalität, um die Diensterbringung zu gewährleisten und das Netz robust gegenüber Lastspitzen zu machen [99]. So wird in [23, 33] beispielsweise ein Verfahren vorgestellt, das zwei kooperierende Regelkreise für die Überlastabwehr verwendet, wobei der eine für die Rufablehnung in der Peripherie zuständig ist und der andere für die Anpassung der Indikatorwerte. Obwohl die Überlastabwehr in privaten Telefonsystemen nicht diese Bedeutung hat, werden auch dort entsprechende Maßnahmen angewendet, wie es z. B. in [35] vorgestellt wird.

  Für die Steuerung und insbesondere für die Signalisiernetze, die den Austausch der Steuerinformationen durchführen und daher sehr stabil sein müssen, hat die Überlastabwehr eine große Bedeutung, was sich auch in der Zahl der Untersuchungen und entsprechenden Maßnahmen niederschlägt. Beispiele dafür sind u. a. in [56, 62, 69, 80, 99, 128] gegeben. Des Weiteren wurde für das Signalisiersystem Nr. 7 (SS7) von der ITU-T ein Rahmenwerk für Überlastabwehrmaßnahmen vorgegeben [38]. In [62] wird z. B. ein Verfahren vorgestellt, das die Effektivität der Überlastabwehr für SS7 verbessert, wobei die standardisierten Schnittstellen zwischen den einzelnen Komponenten nicht verändert werden, so dass dieses Verfahren ohne weitere Veränderungen in das bestehende Netz integriert werden kann. Das in [80] beschriebene Verfahren verwendet Methoden der Entscheidungstheorie, um festzustellen, ob eine Verbindungsanforderung angenommen oder abgelehnt werden soll. Dazu wird aus den vorangegangenen Verbindungsanforderungen abgeschätzt, ob die Anforderung vollständig in der vorgegebenen Zeit bearbeitet werden kann. Durch Optimierung einer

Kostenfunktion wird schließlich die Entscheidung über Annahme oder Ablehnung der Verbindungsanforderung getroffen.

Da beim IN zentrale Komponenten (SCP) die Dienstbearbeitung durchführen, sind diese besonders von einer Überlastung gefährdet. Da außerdem die Wirkbreite einer solchen Überlastung, d. h. die Anzahl der betroffenen Teilnehmer, sehr groß wäre, ist die Anwendung entsprechender Überlastabwehrmaßnahmen für diese zentralen Komponenten sehr sinnvoll. Einige Untersuchungen zur Überlastabwehr im IN sind z. B. in [2, 55, 59, 76, 81, 108, 109] enthalten. So wird z. B. in [108, 109] ein integriertes Überlastabwehr- und Lastverteilungsverfahren vorgestellt und untersucht, bei dem die SCPs zur Überlastabwehr die einzelnen SSPs zur Drosselung ihrer Dienstanforderungen veranlassen. Des Weiteren ist eine Lastverteilung vorgesehen, indem die Dienstbearbeitung teilweise in die SSPs mittels *mobiler Agenten* ausgelagert wird. Bei der in [81] vorgestellten Überlastabwehrmaßnahme wird in den SSPs ein aus der Flusskontrolle bekanntes Fenster-Verfahren angewendet, so dass die SCPs selbst nicht durch die Ausführung der Überlastabwehr belastet werden. Da dabei die Steuerung nicht durch den SCP erfolgt, ist es jedoch nicht möglich, die Last einzelner SSPs individuell zu steuern.

- Lastverteilung in der Datenkommunikation
  In der Datenkommunikation sind Lastverteilungsverfahren weit verbreitet. So werden sie beispielsweise für verteilte Anwendungen (z. B. in [117]), die eine schnellere Lösung komplexer Probleme ermöglichen sollen, und für unterschiedliche Web-Dienste angewendet. In der Regel werden dazu mehrere Komponenten in einem Cluster zusammengefasst, der nach außen wie ein einzelnes System wirkt. Beispiele dazu können u. a. in [1, 11, 14, 20, 75, 79, 94, 101, 129] gefunden werden. Bei einem Cluster können die internen Komponenten räumlich weit verteilt sein. Beispielsweise wird in [75] ein System vorgestellt, das die Inhalte eines Web-Servers auf weitere Server verteilt, und das die Last, die durch die Zugriffe auf diese Inhalte entsteht, entsprechend auf diese Server verteilt. Dabei wird die Belastung der Server durch eine zentrale Komponente überwacht, so dass die Lastverteilung auf der aktuellen Belastung der Server basiert. Darüber hinaus wird bei Bedarf der Inhalt des ursprünglichen Web-Servers auf weitere Server verteilt.

  Einige der genannten Untersuchungen für Web-Server-Cluster optimieren die Leistung der Cluster, indem die Anfragen abhängig vom Inhalt auf die einzelnen Server verteilt werden. Dabei sollen möglichst viele Anfragen aus dem *Cache*-Speicher der Server beantwortet werden, so dass möglichst wenig der zeitintensiveren Zugriffe auf den Hauptspeicher oder die Festplatte notwendig sind. Daher sollen gleiche Anfragen jeweils zum gleichen Server weitergeleitet werden, wobei die einzelnen Server möglichst gleichmäßig ausgelastet werden sollen. In [20] wird z. B. vorgeschlagen, die Anfragen nach der Größe der angefragten Inhalte zu ordnen und anschließend entsprechend zu verteilen. Dabei ist die Zuordnung

zwischen Größe der Inhalte und bearbeitendem Server ein kritischer Punkt, um eine gleichmäßige Belastung der Server zu erreichen. Dazu wird ebenfalls ein Verfahren vorgeschlagen, das eine dynamische Anpassung bei sich ändernden Lastcharakteristika erlaubt.

Bei den Lastverteilungsverfahren für Web-Server-Cluster werden in der Regel nur einzelne Anfragen betrachtet. Wenn mehrere Anfragen zusammen gehören sollten, wie es z. B. beim Online-Handel vorkommt, wird dies meist nicht in die Lastverteilung miteinbezogen [126]. Daher müssen bei der Änderung dynamischer Inhalte diese Änderungen sofort dem ganzen Cluster verfügbar gemacht werden, damit bei der nächsten Anfrage, die evtl. von einem anderen Server des Clusters bearbeitet wird, die aktuelle Version des Inhalts bearbeitet wird. Wenn bei den Inhalten keine strenge Konsistenz der Daten notwendig ist, da die Inhalte entweder statisch sind oder sich nur selten ändern, können die Inhalte vervielfältigt und an die Server weitergegeben werden. Dieses Prinzip wird z. B. bei Web-Cache-Architekturen angewandt [127], wobei in diesem Bereich auch Methoden zur Erhöhung der Konsistenz für sich ändernde Daten vorgeschlagen werden.

- Überlastabwehr in der Datenkommunikation
  Während bei der Überlastabwehr in der Telekommunikation meist die Effektivität einzelner zentraler Netzkomponenten optimiert wird, wurde bei der Datenkommunikation und insbesondere beim Internet der Schwerpunkt auf einen möglichst großen Durchsatz des gesamten Netzes unter Einhaltung der Fairness gegenüber allen Benutzern gelegt, wobei die Einhaltung bestimmter Dienstgütekriterien, wie z. B. der Antwortverzögerung, eine untergeordnete Rolle gespielt hat. Dies spiegelt sich z. B. auch in der Definition von TCP für Internet-basierte Umgebungen wider. Durch die zunehmende Bedeutung von Anwendungen, die entsprechend auf eine schlechte Dienstgüte, wie z. B. lange Verzögerungen, reagierten, wurden Verfahren untersucht, die eine Differenzierung von Diensten über das Best Effort-Modell hinaus ermöglichten, was u. a. zur Definition von IntServ und DiffServ führte (siehe dazu auch Abschnitt 2.2.3.2).

  Neben der Verbesserung der Effektivität des Netzes wurde es durch die zunehmende Bedeutung von Web-Diensten notwendig, die Verfügbarkeit der entsprechenden Web-Server zu gewährleisten. Dazu wurden außer den bereits genannten Lastverteilungsverfahren in Web-Server-Cluster auch verschiedene dienstgüteunterstützende Verfahren vorgeschlagen, die auch Überlastabwehrmaßnahmen beinhalten [4, 18, 19, 54, 120, 123]. Dabei werden meist die einzelnen Anfragen nicht mehr separat behandelt, sondern die Maßnahmen werden auf ganze *Sessions* angewandt, d. h. dass mehrere, zusammenhängende Anfragen, wie sie z. B. beim Online-Handel vorkommen, entweder komplett abgelehnt oder vollständig bearbeitet werden. So werden z. B. in [18, 19] unterschiedliche Überlastabwehrmaßnahmen untersucht, um den Durchsatz an Sessions eines Web-Servers zu optimieren. Der Schwerpunkt der Untersuchungen lag, neben der Einführung von Session-basierten Überlastabwehrmaß-

nahmen für Web-Server, auf der Optimierung und der dynamischen Anpassung von Parametern der Überlastabwehrmaßnahmen. In [120] wird ein kombiniertes Verfahren vorgestellt, das aus einer Session-basierten Verbindungsannahmesteuerung, einer Priorisierung der Annahme von TCP-Verbindungen und einer TCP-basierten Überlastabwehrmaßnahme besteht. Bei der Session-basierten Verbindungsannahmesteuerung werden Daten aus den Köpfen von HTTP-Paketen ausgewertet und für die Steuerung der Annahme einer Session verwendet. Die Priorisierung der Annahme der TCP-Verbindungen wird nach dem Ende des Verbindungsaufbaus der TCP-Verbindung ausgeführt und bestimmt, mit welcher Priorität die über der TCP-Schicht liegende Anwendungsschicht die Verbindung annehmen wird, bevor die Anwendungsdaten ausgetauscht werden. Damit kann für höherpriore Anwendungen eine kleinere Antwortverzögerung erreicht werden. Schließlich wird bei der TCP-basierten Überlastabwehrmaßnahme die Rate der angenommenen TCP-Verbindungsaufbauanforderungen mit einem *Token*-basierten Verfahren gesteuert. Bei dieser Überlastabwehrmaßnahme wird für jede weitergeleitete Anforderung ein Token verbraucht. Wenn kein Token mehr zur Verfügung steht, wird die entsprechende Anforderung abgelehnt. Tokens werden kontinuierlich mit einer konstanten Rate erzeugt, wobei eine maximale Anzahl von zur Verfügung stehenden Tokens nicht überschritten werden darf. Durch die Einstellung der Rate zur Erzeugung der Tokens und der maximalen Anzahl von Tokens kann die mittlere Rate von weitergeleiteten Anforderungen sowie die maximale Größe einer kurzzeitigen Lastspitze von Anforderungen festgelegt werden. Diese Überlastabwehrmaßnahme kann in dem vorgestellten Verfahren auch für einzelne Dienstklassen jeweils separat durchgeführt werden.

Die bisher beschriebenen Untersuchungen lassen sich teilweise auf VoIP-Umgebungen übertragen, wobei insbesondere bei der Datenkommunikation die Erweiterung der Granularität von einzelnen Anforderungen, auf ganze Sessions von Interesse ist, da dabei der Zusammenhang mehrerer Anforderungen berücksichtigt wird. Dies ist auch bei der Steuerung von VoIP-Diensten notwendig, da beispielsweise der Verbindungsaufbau bei H.323, wie in Abschnitt 2.3.1.3 beschrieben, aus mehreren Anforderungen besteht. Für VoIP-Umgebungen speziell wurden jedoch bisher sehr wenig Untersuchungen durchgeführt. In [34] wird ein Verfahren zur Verbindungsannahmesteuerung und zur Lastverteilung für VoIP vorgestellt, wobei dort die Optimierung des Durchsatzes des Netzes bei Einhaltung der geforderten Dienstgüte das Ziel ist. Die Belastung einzelner VoIP-Steuerungskomponenten wird dabei nicht betrachtet, die Lastverteilung bezieht sich auf die Übertragungsabschnitte.

Zur Sicherstellung der Funktion von Media Gateways (MG) wurde von der ITU-T eine Empfehlung zur Überlastabwehr dieser Komponenten definiert [46]. Die Überlastabwehr selbst wird dabei durch den Media Gateway Controller (MGC) durchgeführt, wobei je nach Lastzustand des MGs die Rate der durch den MG zu bearbeitenden Rufe angepasst wird. Die MGs

zeigen ihren Lastzustand dem MGC durch entsprechende Nachrichten an, so dass dieser die Belastung des MGs ableiten kann.

Bei der VoIP-Architektur der Firma *Cisco* spielt der sog. *CallManager* eine zentrale Rolle, da er für die Steuerung der VoIP-Verbindungen zuständig ist. Da bei einer entsprechend großen Umgebung Skalierungsprobleme auftreten könnten und um die Ausfallsicherheit zu erhöhen, kann wie in [21] beschrieben ein Cluster von CallManagern betrieben werden. Bei diesem Cluster wird ein statisches Lastverteilungsverfahren angewandt, bei dem die einzelnen Teilnehmer einem CallManager des Clusters zugeordnet werden. Des Weiteren werden für jeden Teilnehmer Reserve-CallManager festgelegt, die ebenfalls Mitglied des Clusters sind. Beim Ausfall des ersten CallManagers wird dann der erste Reserve-CallManager verwendet, wenn dieser auch ausfallen sollte wird auf den zweiten Reserve-CallManager übergewechselt. Zur Sicherstellung der Datenkonsistenz innerhalb eines Clusters verwaltet ein dedizierter CallManager die Konfigurationsdaten der Teilnehmer und verteilt jeweils eine Kopie dieser Daten an die anderen Cluster-Mitglieder. Nach einer Änderung, die nur über den verwaltenden CallManager durchgeführt werden kann, verteilt dieser die geänderten Konfigurationsdaten an die anderen CallManager. Daten, die öfters geändert werden, wie z. B. Registrierinformationen von Endgeräten oder Gateways, werden in einem verteilten Dateisystem, das allen Cluster-Mitgliedern zugänglich ist, gespeichert.

Der Schwerpunkt dieser Arbeit liegt auf der Untersuchung von Verfahren zur Lastverteilung und zur Überlastabwehr für die optimierte Steuerung von VoIP-Diensten. Dabei werden Cluster von Gatekeepern eingeführt und untersucht, wobei insbesondere auf die Problematik der Granularität der Lastverteilung bei dynamischen Verfahren eingegangen wird. In Anhang R der Empfehlung H.323 werden zwar Cluster von Gatekeepern erwähnt, jedoch werden dort ausschließlich mögliche Verfahren zur Optimierung der Ausfallsicherheit vorgestellt. Des Weiteren wurde die Lastverteilung über Zonengrenzen hinweg, die in dieser Arbeit untersucht wird, bisher nicht betrachtet. Schließlich ist die Anwendung von Überlastabwehrmaßnahmen für die Steuerung von VoIP-Diensten, wie sie aus der kanalvermittelnden Telefonie bekannt sind, bisher noch nicht umfassend untersucht worden, wobei sicherlich Gemeinsamkeiten zur Steuerung von digitalen Vermittlungssystemen und zum IN vorhanden sind.

3.4 Steuerungsoptimierung für verschiedene Ressourcen

In diesem Abschnitt werden Verfahren zur optimierten Steuerung für unterschiedliche Ressourcen einer H.323-basierten VoIP-Umgebung vorgestellt, wobei eine Optimierung für Gatekeeper zunächst nicht betrachtet wird. Dies wird in Abschnitt 3.5 im Detail abgeleitet und ausführlich beschrieben. In den folgenden Abschnitten werden Steuerungsverfahren zur Optimierung der Verwendung der Übertragungskapazität auf dem Transportpfad (3.4.1), der Verwendung von Gateways (3.4.2) und zur Verwendung spezieller Komponenten (3.4.3), wie

z. B. MCUs vorgestellt. Dabei wird jeweils erläutert, warum eine optimierte Steuerung für die entsprechende Ressource interessant ist. Des Weiteren werden jeweils mögliche Lastindikatoren, Lastverteilungsverfahren und entsprechende Überlastabwehrmaßnahmen vorgestellt. Schließlich wird beschrieben, welche Komponenten die einzelnen Maßnahmen durchführen könnten.

3.4.1 Übertragungskapazität auf dem Transportpfad

Wie in Abschnitt 2.1.1.4 beschrieben, kann es in paketvermittelnden Netzen bei zu großer Belastung zu erheblichen Verzögerungen von Paketen oder gar zu Paketverlusten kommen. Da dies jedoch die Dienstgüte von VoIP-Diensten soweit verringern könnte, dass eine Kommunikation nicht mehr akzeptabel wäre, ist es notwendig, die Verwendung der zur Verfügung stehenden Übertragungskapazitäten entsprechend zu steuern.

Um die Dienstgüte für VoIP-Dienste einzuhalten, existieren verschiedene Verfahren, die innerhalb des Netzes und seinen Komponenten agieren und somit nicht oder nur kaum durch die VoIP-Steuerung beeinflusst werden können. Dazu zählt die in 2.2.3.2 vorgestellte DiffServ-Architektur sowie die entsprechenden Router-Architekturen, die eine differenzierte Bearbeitung der einzelnen Pakete erlauben. Des Weiteren ist es möglich, durch entsprechende Verkehrslenkung (*Routing*) die einzelnen Übertragungsabschnitte so auszulasten, dass die einzelnen Abschnitte gleichmäßig ausgelastet werden. Die Verkehrslenkungsverfahren zur Unterstützung der Einhaltung der Dienstgüte werden als *QoS-Routing* bezeichnet. Diese Verfahren werden z. B. in [12] untersucht und bewertet.

Da der Gatekeeper, wie in Abschnitt 2.3.1.2 beschrieben, für die Verbindungsannahmesteuerung zuständig ist und an der Steuerung aller VoIP-Verbindungen in seiner Zone beteiligt ist, kann er die Steuerung der in seiner Zone zur Verfügung stehenden Übertragungskapazitäten durchführen.

Wenn eine IntServ-Architektur zu Grunde liegt, wird für VoIP-Dienste in der Regel die Guaranteed Service Class verwendet. Dabei kann der Gatekeeper die Reservierung der für die Verbindung notwendigen Ressourcen mittels RSVP durchführen. Wenn die Reservierung fehlschlagen sollte, würde die Verbindung durch den Gatekeeper abgelehnt werden. Bei dieser Lösung verlässt sich der Gatekeeper vollständig auf die IntServ-Architektur bzw. auf die RSVP-Signalisierung und er selbst führt im Prinzip nur die entsprechenden Signalisierprozeduren durch.

Eine weitere Möglichkeit zur Steuerung der Verwendung der Übertragungskapazitäten, die ausschließlich durch den Gatekeeper durchgeführt wird, kann vom prinzipiellen Vorgehen, das in Abschnitt 3.2 vorgestellt wurde, abgeleitet werden: Dabei bestimmt der Gatekeeper zunächst die Belastung der einzelnen Übertragungsabschnitte seiner Zone, anschließend ver-

teilt er die einzelnen VoIP-Verbindungen möglichst effektiv auf die einzelnen Abschnitte. Wenn jedoch trotz Lastverteilung die Belastung eines Übertragungsabschnittes für eine neue VoIP-Verbindung zu groß werden würde, wird diese neue Verbindung abgelehnt.

- Bestimmung der Belastung der einzelnen Übertragungsabschnitte
  Bei dieser Vorgehensweise ist insbesondere die Bestimmung der Belastung der einzelnen Übertragungsabschnitte sehr kritisch, da die einzelnen Medienströme nicht mit einer konstanten Paketrate versendet werden, sondern teilweise sehr büschelförmig auftreten. Wenn für die Verbindungsannahme nur der Mittelwert der Übertragungsrate betrachtet werden würde, käme es während und nach diesen kurzzeitigen Büscheln von Paketen zu großen Verzögerungen oder zum Verlust von Paketen. Wenn jedoch die Verbindungsannahme auf der maximalen Übertragungsrate basiert, würden die einzelnen Übertragungsabschnitte sehr ineffektiv verwendet werden. Um dieses Problem zu lösen, wurde und wird intensiv geforscht, wobei die dabei gewonnenen Erkenntnisse in eine entsprechende Gatekeeper-Realisierung einfliessen können. Beispielsweise wird in [30] ein Verfahren vorgeschlagen, das die *effektive Bandbreite*, die eine Verbindung benötigt, und die *effektive Belastung* der Übertragungsabschnitte bestimmt, um damit die Verbindungsannahme entsprechend zu steuern. Des Weiteren muss beachtet werden, welche Informationen dem Gatekeeper über die Belastung der Übertragungsabschnitte zur Verfügung stehen, da in der Regel nicht nur VoIP-Anwendungen das IP-basierte Netz verwenden, sondern auch andere, die nicht durch den Gatekeeper gesteuert werden. Jedoch kann durch eine entsprechende Differenzierung der Dienste im Netz erreicht werden, dass den VoIP-Diensten eine vorgegebene Kapazität zur Verfügung steht, die durch den Gatekeeper verwaltet wird.

- Lastverteilung
  Um eine Verteilung der Last für die einzelnen Übertragungsabschnitte durchführen zu können, muss der Gatekeeper auf den Weg, den die Nutzdaten der VoIP-Verbindungen zwischen den beteiligten Endpunkten verwenden, Einfluss nehmen. Dazu wäre im Prinzip das sog. *Source Routing* geeignet, bei dem der Sender eines Pakets den Pfad angibt, den das Paket zum Empfänger verwenden soll. Diese Option wird jedoch kaum in heutigen IP-Netzen angewandt. Eine weitere Möglichkeit besteht, wenn ein Endpunkt einer Verbindung ein Gateway zu einem anderen Netz wäre und wenn mehrere Gateways zum Zielnetz existieren. In diesem Fall könnte der Gatekeeper das Gateway, bei dem die Übertragungsabschnitte zwischen Start- und Zielendpunkt am wenigsten belastet sind, als Zielendpunkt verwenden.

- Ablehnung von Verbindungen
  Eine neu hinzukommende Verbindung sollte abgelehnt werden, wenn die Belastung der Übertragungsabschnitte so groß werden würde, dass die Dienstgüte einzelner Verbindungen nicht mehr gewährleistet wäre. Dabei sind unterschiedliche Vorgehensweisen denkbar. Beispielsweise könnte man eine Verbindung, die im Vergleich zu anderen einen großen Bedarf

an Übertragungskapazitäten hat, bereits in einer Phase ablehnen, in der noch genügend Kapazitäten vorhanden gewesen wären, damit anschließend eine größere Anzahl von Verbindungen mit niedrigerem Bedarf an Übertragungskapazitäten zugelassen werden kann. Umgekehrt kann für einen Netzbetreiber auch eine frühe Ablehnung der Verbindungen mit wenig Bedarf an Kapazitäten sinnvoll sein, um möglichst viele Verbindungen mit hohem Bedarf zulassen zu können, da diese evtl. deutlich höhere Einnahmen erzeugen. Schließlich ist auch eine Priorisierung einzelner Dienste oder Benutzergruppen denkbar, wobei, wie im vorigen Beispiel beschrieben, Anforderungen niederpriorer Dienste und Benutzer früher abgelehnt werden als höherpriore Anforderungen.

3.4.2 Gateway

Ein Gateway unterstützt in der Regel gleichzeitig viele Verbindungen und stellt seinen Dienst vielen Endpunkten zur Verfügung. Somit stellt es eine zentrale Komponente in einer H.323-basierten VoIP-Umgebung dar und ist daher ein möglicher Ort, an dem Überlast auftreten kann. Des Weiteren hat ein Gateway eine große Wirkbreite, da viele Benutzer von einer Überlastung betroffen sein könnten.

Bei einem Gateway müssen zwei Arten der Belastung unterschieden werden: Zum einen kann eine überhöhte Belastung bezüglich seiner Steuerung auftreten. Dies kann sowohl die Umsetzung der einzelnen Steuernachrichten zwischen den Netztypen, die das Gateway verbindet, als auch die Steuerung des Gateways selbst, wie z. B. die Überwachung seiner Ressourcen, umfassen. Zum anderen kann ein Gateway auch bezüglich der Transformation der Nutzdaten übermäßig belastet werden, wenn zu viele aktive Verbindungen über ein Gateway geführt werden.

Des Weiteren kann ein Gateway, wie bereits in Abschnitt 2.3.1.2 beschrieben, in die separaten Komponenten Media Gateway Controller (MGC) und Media Gateway (MG) aufgespalten sein. Im Folgenden werden für beide Realisierungsformen von Gateways mögliche Steuerungsoptimierungen vorgestellt. Die dabei verwendeten Verfahren sind in der Regel für beide Realisierungsformen gleich, jedoch werden auch entsprechende Erweiterungen für die Kombination MGC/MG beschrieben, wenn sie sinnvoll bzw. notwendig sind.

- Steuerungsoptimierung bezüglich der Nutzdaten
  Da die Kapazitäten eines Gateways/MGs zur Transformation der Nutzdaten für das jeweils andere Netz begrenzt sind, müssen entsprechende Verfahren angewendet werden, damit diese Komponenten nicht überlastet werden und somit die Dienstgüte einer Verbindung verschlechtern.

  - Bestimmung des aktuellen Lastzustands
    Um die Belastung eines Gateways/MGs bezüglich der Nutzdaten festzustellen, kann eine steuernde Komponente, wie z. B. der Gatekeeper beim Gateway oder der MGC beim

MG, die Anzahl und die Art der Verbindungen ermitteln, die im Augenblick aktiv sind. Damit die steuernde Komponente daraus die Auslastung des Gateways/MGs ableiten kann, müssen die jeweiligen maximal zulässigen Belastungen bezüglich der Verbindungsanzahl dieser Komponenten bekannt sein. Des Weiteren besteht für ein Gateway die Möglichkeit, seine aktuelle Belastung mittels der RAS-Signalisiernachricht RAI (*Resources Available Indicate*) dem Gatekeeper anzuzeigen. Dabei werden u. a. die aktuellen Datenraten für die unterstützten Standards sowie eine Anzeige, ob sich das Gateway nahe der Maximalbelastung befindet, angegeben. Der Gatekeeper bestätigt den korrekten Empfang dieser Nachricht mit RAC (*Resources Available Confirm*).

- Lastverteilung
  Wenn der Gatekeeper bzw. der MGC die Belastung seiner zu verwaltenden Gateways/MGs kennt, kann er für neu hinzukommende Verbindungen das jeweils am geringsten belastete Gateway/MG auswählen, um eine gleichmäßige Auslastung der einzelnen Gateways/MGs zu erreichen. Um eine derartige Verteilung durchzuführen, ist es jedoch notwendig, dass mehrere Gateways/MGs den Zugang zu den gleichen Netzen ermöglichen. Für die Lastverteilung selbst sind verschiedene Verfahren, wie sie in Abschnitt 3.2.2 eingeführt wurden, anwendbar. Des Weiteren wäre auch eine Weiterleitung der Verbindungsanfrage an Gateways/MGs anderer Zonen denkbar, so dass die dort zur Verfügung stehenden Ressourcen ebenfalls verwendet werden können. Bei der MGC/MG-Realisierung müsste der Gatekeeper dazu die Verbindung an einen MGC einer anderen Zone weiterleiten, der dann ein entsprechendes MG dieser Zone auswählt. Die Weiterleitung an andere Zonen birgt jedoch die Gefahr, dass eine lokale Überlastsituation auf weitere Zonen verbreitet wird, so dass die Gateways auch in diesen weiteren Zonen überlastet werden. Eine weitere Möglichkeit der Lastverteilung ist die Umleitung der Verbindung über ein drittes Netz, das dann den Zugang zum Zielnetz über ein weiteres Gateway erlaubt. Dies wird in Bild 3.3 dargestellt. Dabei ist aber zu beachten, dass für diese Verbindung mehr Ressourcen benötigt werden als bei einer direkten Verbindung von Ursprungs- und Zielnetz über ein einzelnes Gateway. Darüber hinaus trägt dies ebenfalls zu einer Verbreitung von lokalen Überlastsituationen bei.

- Überlastabwehr
  Falls die Gateways/MGs bereits voll ausgelastet sind, so dass keine weiteren Verbindungen mehr bearbeitet werden können, müssen neu ankommende Verbindungsanforderungen, die über eines dieser Gateways/MGs geführt werden müssten, abgelehnt werden. Dabei gelten die gleichen Anmerkungen, wie sie für die Ablehnung von Verbindungen im Abschnitt 3.4.1 aufgeführt wurden, d. h. dass beispielsweise eine Priorisierung von Diensten und Benutzergruppen möglich ist, indem ein vorgegebener Anteil der Kapazitäten der Gateways/MGs für höherpriore Verbindungen reserviert wird.

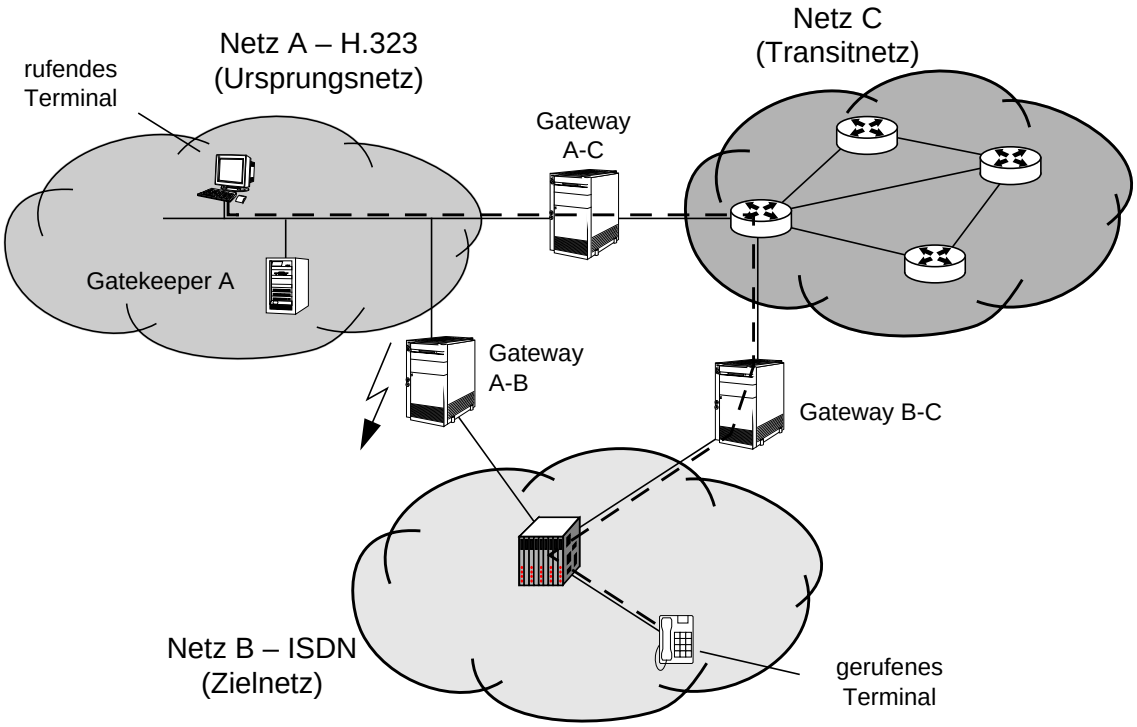


Bild 3.3: Umleitung einer Verbindung über ein anderes Netz wegen überlastetem Gateway A-B

- Steuerungsoptimierung bezüglich der Steuerung
  Der Steuerungsanteil eines Gateways selbst kann ebenfalls zu hoch belastet werden, so dass Verbindungsanforderungen nicht bearbeitet werden können, obwohl die Kapazitäten für den Nutzdatenanteil ausreichend wären. Insbesondere ein MGC, der sowohl Aufträge von einem Gatekeeper für die Verbindungssteuerung erhält als auch die Steuerung der MGs übernimmt, könnte dabei zu stark belastet werden. Des Weiteren kann auch der Steuerungsanteil eines MGs, der die Steuerungsnachrichten des MGCs bearbeiten muss, überlastet werden. Für diesen Fall wurde von der ITU-T in [46] ein Verfahren spezifiziert, bei dem die MGs Lastanzeige-Nachrichten an den MGC senden, der aus der Rate dieser Nachrichten die aktuelle Belastung der MGs ableiten kann, und für die Überlastabwehr das *Leaky Bucket*-Verfahren anwendet, das in Abschnitt 3.5.3 vorgestellt wird. Im Folgenden werden mögliche Verfahren für die optimierte Steuerung von Gateways und MGCs vorgestellt, die Schnittstelle zwischen MGC und MG wird nicht weiter betrachtet. Diese Verfahren werden durch den Gatekeeper durchgeführt, da er als einzige Komponente in einer Zone über alle notwendigen Informationen für die Steuerung der einzelnen Komponenten verfügt. Es ist auch möglich, dass die Überlastabwehrmaßnahmen durch die Gateways und MGCs selbst ausgeführt werden, wobei die in Abschnitt 3.5.3 vorgestellten Verfahren in adaptierter Form

angewendet werden können, jedoch besitzen diese Komponenten in der Regel nicht die Sicht auf die ganze Zone, um beispielsweise eine sinnvolle Lastverteilung durchzuführen.

- Bestimmung des aktuellen Lastzustands
  Da der Gatekeeper keinen Zugriff auf interne Daten der Gateways/MGCs hat, muss er den aktuellen Lastzustand bezüglich der Steuerung aus Messungen ableiten, die bei ihm selbst ablaufen. Folgende Messwerte können dabei Verwendung finden:
  - Anzahl der aktiven Signalisierverbindungen
    Bei diesem Messwert wird der aktuelle Lastzustand eines Gateways/MGCs abgeleitet, indem die Anzahl der aktiven Signalisierverbindungen ermittelt wird. Dabei ist es sinnvoll, eine Unterteilung in Signalisierphasen vorzunehmen, so dass der Lastzustand aus der Anzahl der im Auf- und Abbau sowie in der Ausführung zusätzlicher Dienste befindlichen Verbindungen bestimmt wird.
  - Anzahl der ausstehenden Antworten
    Aus diesem Messwert kann die Länge der Eingangswarteschlange des Gateways/MGCs abgeschätzt werden, so dass daraus der aktuelle Lastzustand abgeleitet werden kann.
  - Antwortverzögerungen
    Bei einer hohen Belastung eines Gateways/MGCs steigen die Antwortverzögerungen für die Anfragen des Gatekeepers an, so dass aus diesen Werten ebenfalls der aktuelle Lastzustand abgeleitet werden kann. Dabei muss jedoch beachtet werden, dass unterschiedliche Anfragen die einzelnen Ressourcen unterschiedlich beanspruchen und dass daher die Antwortverzögerungen entweder gemittelt über einen bestimmten Zeitraum betrachtet oder entsprechend gewichtet werden müssen, um den aktuellen Lastzustand des Gateways/MGCs zu ermitteln.

  Des Weiteren könnten Gateways/MGCs ihre Lastzustände dem Gatekeeper explizit mittels entsprechender Nachrichten mitteilen, wobei dies bisher in den einzelnen Standards nicht vorgesehen ist.

- Lastverteilung
  Um die Steuerungslast auf die einzelnen Gateways/MGCs zu verteilen, kann der Gatekeeper verschiedene Verfahren anwenden, wie sie in Abschnitt 3.2.2 eingeführt wurden. Dabei sollten vollständige Verbindungen verteilt werden, d. h. dass die Komponente, die die erste Nachricht einer Verbindung bearbeitet, für alle folgenden Nachrichten dieser Verbindung zuständig ist. Eine weitere Unterteilung in Verbindungsphasen, so dass beispielsweise der Verbindungsabbau von einem anderen Gateway/MGC gesteuert wird als der Verbindungsaufbau, erscheint wenig sinnvoll, da zum einen die Verbindungsdaten für die entsprechenden Gateways/MGCs verfügbar gemacht werden müssten und zum

anderen die Zuständigkeiten für die einzelnen Ressourcen, wie z. B. MG oder Kanäle der kanalvermittelnden Kommunikation, beachtet werden müssten.

- Überlastabwehr
  Wenn die Belastung der Steueranteile der Gateways/MGCs zu groß wird, müssen entsprechende Überlastabwehrmaßnahmen angewandt werden. Dabei können Verfahren aus der Telekommunikation, wie sie für Signalisiersysteme entwickelt wurden (wie z. B. in [99, 128]), verwendet werden, wobei der Gatekeeper diese Verfahren für die einzelnen Gateways/MGCs anwendet. Dies entspricht dem Vorgehen des MGCs bei der Überlastabwehr für die MGs, wie es in [46] von der ITU-T vorgeschlagen wurde.

3.4.3 Spezielle Komponenten

In einer H.323-basierten VoIP-Umgebung können Komponenten mit spezifischen Funktionen und Aufgaben vorhanden sein, die von vielen Benutzern verwendet werden. Beispiele für derartige Komponenten sind MCUs, IVR-Systeme (IVR – *Interactive Voice Response*), die den Zugriff auf Informationen über Sprach- und Tasteneingaben erlauben, oder intelligente Anrufbeantworter, die z. B. abhängig vom rufenden Teilnehmer entsprechende Ansagen machen oder im Rahmen einer *Unified Messaging*-Lösung eine E-Mail erzeugen. Da diese Komponenten in der Regel zentral ihre Dienste zur Verfügung stellen, ist die Gefahr einer Überlastung gegeben, wobei eine Einschränkung oder gar ein Ausfall der Diensterbringung viele Benutzer betreffen würde.

Im Prinzip können für diese speziellen Komponenten die gleichen Verfahren für die Steuerungsoptimierung angewandt werden, wie für Gateways. Daher wird für die einzelnen Verfahren auf diesen Abschnitt verwiesen. Im Folgenden werden daher nur die Unterschiede zu den dort beschriebenen Verfahren, die für diese speziellen Komponenten notwenig sind, erläutert.

Im Gegensatz zu den Gateways sind die Steuerungsaufgaben in diesen speziellen Komponenten weniger komplex, so dass davon ausgegangen werden kann, dass eine Überlastung bezüglich der Steuerung nur eine untergeordnete Rolle spielt. Eine Überlastung für die Bearbeitung der Nutzdaten ist jedoch durchaus möglich, da die einzelnen Komponenten vor allem während der Nutzdatenaustauschphase ihre speziellen Dienste erbringen. Beispielsweise führt die MCU das Mischen der Audiodaten durch und erzeugt aus den einzelnen Videoströmen einen gemeinsamen Videostrom für die Teilnehmer der Konferenz. Daher hängt die Belastung einer MCU von der Anzahl der Konferenzen und der Anzahl der Teilnehmer der jeweiligen Konferenzen ab, so dass bei einer Verbindungsanforderung unterschieden werden muss, ob es sich um eine neue Konferenz oder um einen weiteren Teilnehmer einer bestehenden Konferenz handelt. Des Weiteren ist für diese speziellen Komponenten kein Nachrichtenaustausch zur Anzeige des aktuellen Lastzustands beim Gatekeeper in den Standards vorgesehen, so dass eine derartige Lastanzeige nur über proprietäre Erweiterungen realisiert werden könnte.

Bei den IVR- und Anrufbeantworter-Systemen kann eine kurzzeitige Pufferung von Verbindungen in speziellen Warteschleifen sinnvoll sein. Dabei wird z. B. für den Teilnehmer ein entsprechender Ansagetext abgespielt, der auch die geschätzte Wartezeit enthalten kann, bis die gewünschte Komponente verfügbar ist. Da ein Teilnehmer nur eine bestimmte Zeit warten will, sollte eine maximale Wartezeit nicht überschritten werden, weil ansonsten die Wahrscheinlichkeit groß wird, dass der Teilnehmer die Verbindung beendet. Dazu wurden z. B. in [9] Untersuchungen durchgeführt.

3.5 Steuerungsoptimierung für Gatekeeper

Ein zentraler Punkt dieser Arbeit ist die optimierte Nutzung der Steuerressourcen des Gatekeepers. Wie in Abschnitt 2.3.1.2 beschrieben, ist der Gatekeeper für die Verwaltung einer Zone zuständig. Daher ist er einerseits eine mögliche Komponente, bei der eine Überlastung auftreten kann, und andererseits kann er die Steuerung bezüglich der Lastverteilung und der Überlastabwehr für die anderen Komponenten seiner Zone durchführen. Da eine Überlastung oder ein Ausfall des Gatekeepers die ganze Zone betreffen würde, ist es notwendig, entsprechende Maßnahmen durchzuführen, die die Diensterbringung des Gatekeepers auch in Hoch- und Überlastsituationen sicherstellen.

In diesem Abschnitt werden Verfahren zur optimierten Steuerung für Gatekeeper vorgestellt. Dazu werden in Abschnitt 3.5.1 Lastindikatoren beschrieben, die eine Ableitung des aktuellen Lastzustands eines Gatekeepers erlauben, wobei auch ein neuer, in der Literatur bisher nicht beschriebener Indikator eingeführt wird. In Abschnitt 3.5.2 werden Verfahren zur Lastverteilung vorgestellt, wobei zunächst ein Cluster von Gatekeepern für die Verwaltung einer Zone eingeführt wird. Dabei wird insbesondere auf die Problematik der Granularität der Lastverteilung eingegangen. Des Weiteren werden mögliche Verfahren für die Lastverteilung über Zonengrenzen hinweg vorgestellt. Anschließend beschreibt Abschnitt 3.5.3 Überlastabwehrmaßnahmen für Gatekeeper. Schließlich werden in Abschnitt 3.5.4 Realisierungsaspekte der beschriebenen Verfahren, die Auswirkungen auf die Steuerungsoptimierung haben, vorgestellt.

3.5.1 Lastindikatoren

Im folgenden Abschnitt 3.5.1.1 werden relevante Lastindikatoren sowie ihre Bestimmung im Gatekeeper beschrieben. Anschließend werden in Abschnitt 3.5.1.2 Kombinationen von Lastindikatoren, die die interessierenden Eigenschaften der einzelnen Lastindikatoren entsprechend vereinen sollen, vorgestellt.

3.5.1.1 Bestimmung von Lastindikatoren

Um Lastindikatoren für Gatekeeper zu bestimmen, können verschiedene Werte und Größen herangezogen werden. In diesem Abschnitt werden dazu einige relevante Kenngrössen und Verfahren beschrieben, die eine Ableitung des aktuellen Lastzustands eines Gatekeepers erlauben. Dabei wird auch ein neues Verfahren vorgestellt, das den Ressourcenbedarf für die einzelnen Phasen einer VoIP-Verbindung miteinbezieht.

Die folgenden Lastindikatoren können zur Bestimmung des aktuellen Lastzustands für Gatekeeper angewandt werden:

- Warteschlangenlänge
  Die Länge der Eingangswarteschlange ist eine weit verbreitete und einfach zu bestimmende Größe, die direkt für die Lastindikatorbestimmung herangezogen werden kann. Dabei werden den einzelnen Lastzuständen entsprechende Bereiche der Warteschlangenbelegung zugeordnet. Dieser Lastindikator wird meist ereignisgesteuert angewendet, d. h. bei jeder Veränderung der Warteschlangenlänge wird der Lastindikatorwert aktualisiert. Ein Vorteil dieses Indikators ist, dass er die Nachrichten betrachtet, bevor sie durch die verarbeitende Einheit, den Prozessor, bearbeitet werden. D. h. es wird die aktuell noch zu bearbeitende Last der Komponente angezeigt. Eine Differenzierung der Nachrichten und damit des erwarteten Ressourcenverbrauchs für die einzelnen Nachrichten ist jedoch nur durch aufwendigere Verfahren möglich, indem beispielsweise verschiedene Warteschlangen für unterschiedliche Nachrichtentypen verwendet werden, wobei in diesem Fall eine entsprechende Synchronisierung der einzelnen Warteschlangen notwendig wäre.

- Gradient der Warteschlangenlänge
  Bei diesem indirekten Lastindikator wird aus der Änderung der Warteschlangenlänge der aktuelle Lastzustand ermittelt. Die Bestimmung des Lastindikators erfolgt dabei zeitgesteuert, d. h. der Gradient wird jeweils für ein Messintervall ermittelt, z. B. indem die Differenz der Warteschlangenlängen an den Messintervallgrenzen durch die Intervalldauer geteilt wird. Abhängig vom jeweiligen Indikatorwert wird der aktuelle Lastzustand angepasst, d. h. erhöht oder verringert. Dazu kann der Indikatorwert mittels entsprechender Filterverfahren zunächst vorverarbeitet werden. In [100] wird beispielsweise ein Verfahren beschrieben, dass die Ergebnisse verschiedener Filter geeignet kombiniert, um den aktuellen Lastzustand zu bestimmen.

- Rufankunftsrate
  Die Rufankunftsrate kann direkt zur Lastindikatorermittlung verwendet werden. Sie wird gemittelt über ein Messintervall bestimmt, wobei entweder die Anzahl von Rufankünften für ein vorgegebenes Messintervall oder die Zeitdauer, bis eine bestimmte Anzahl von Verbindungsanforderungen angekommen ist, gemessen wird. Da insbesondere bei diesem

Messwert die aktuelle Belastung des Gatekeepers von der Ankunftsrate der letzten Intervalle abhängt, werden hierbei entsprechende Filter angewandt, um die Bearbeitung der letzten Rufankünfte miteinzubeziehen. Diesen gefilterten Messwerten werden die einzelnen Lastzustände zugeordnet. Da nicht jede beim Gatekeeper ankommende Nachricht einer neuen Rufankunft entspricht, kann dieser Indikator im Gegensatz zur „Warteschlangenlänge“ erst nach der Auswertung der entsprechenden Nachrichten in der verarbeitenden Einheit bestimmt werden.

- Prozessorauslastung
  Ein Prozessor kann die beiden Zustände *belegt* und *frei* einnehmen. Zur Bestimmung des Lastindikators wird die mittlere Auslastung in einem Intervall ermittelt. Damit kurzzeitige Lastspitzen keine oder nur geringe Auswirkungen haben, können bei diesem Messwert ebenfalls Filter angewandt werden. Die Bestimmung des aktuellen Lastzustands erfolgt durch eine entsprechende Zuordnung zu einem Auslastungswert. Dieser Messwert erlaubt die Ableitung der aktuellen Belastung des Gatekeepers, jedoch ist die Bestimmung verschiedener Überlaststufen schwierig, da die Unterscheidung zwischen voller Auslastung und schwerer Überlastung nicht möglich ist, weil in beiden Situationen die Prozessorauslastung den Maximalwert einnimmt.

- Anzahl offener Anfragen
  Zur Verwendung dieses Lastindikators wird eine H.323-basierte VoIP-Verbindung, wie in Bild 3.4 dargestellt, in verschiedene Phasen unterteilt. Beim einfachen Basisdienst, der Punkt-zu-Punkt Sprachkommunikation, ist dabei die Verbindungsaufbauphase und die Verbindungsabbauphase für den Gatekeeper von Interesse. Des Weiteren können noch Phasen zur Erbringung zusätzlicher Dienste oder zur Veränderung der Verbindungsparameter miteinbezogen werden. Zur Bestimmung des Lastindikators wird ein Zähler verwendet, der die Anzahl der Verbindungen, die sich in einer der genannten Phasen befinden, anzeigt. Um den Ressourcenbedarf in den einzelnen Verbindungsphasen einzubeziehen, können den Verbindungsphasen unterschiedliche Gewichte zugeordnet werden, so dass z. B. eine Verbindung, die in die Aufbauphase eintritt, den Zähler mehr erhöht, als eine Verbindung die abgebaut wird. Wenn eine Phase einer Verbindung beendet ist, wird der Zähler entsprechend verringert. Die Bestimmung des aktuellen Lastzustands kann ereignis- oder zeitgesteuert erfolgen, wobei im zweiten Fall die einzelnen Lastindikatorwerte ebenfalls durch geeignete Filter bearbeitet werden können. Zur Ermittlung der Lastzustände werden diesen entsprechende Indikatorwerte zugeordnet. Im Gegensatz zum Lastindikator „Warteschlangenlänge“ kann bei diesem Indikator der unterschiedliche Ressourcenbedarf für die einzelnen Verbindungsphasen integriert werden. Jedoch erfolgt die Bestimmung des Indikators erst bei der Nachrichtenbearbeitung im Prozessor, da u. a. festgestellt werden muss, ob eine Nachricht eine neue Verbindungsphase auslöst oder innerhalb einer bestehenden Verbindungsphase ausgetauscht wird.

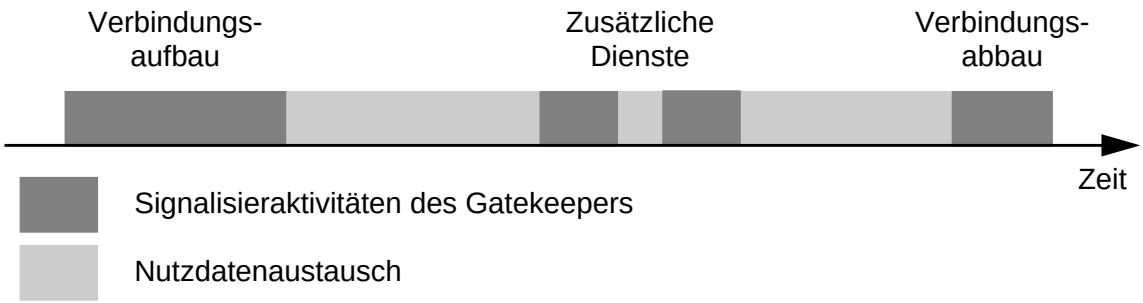


Bild 3.4: Beispiel für Signalisierphasen in einer VoIP-Verbindung

- Gewichtete Verbindungszustände
  Ein Nachteil des Lastindikators „Anzahl offener Anfragen“ ist die fehlende Anpassung des Ressourcenbedarfs im weiteren Verlauf der einzelnen Verbindungsphasen. Beispielsweise wird eine Verbindung, deren Verbindungsaufbau nahezu abgeschlossen ist, gleich bewertet wie eine Verbindung, bei der gerade die erste Nachricht des Verbindungsaufbaus bearbeitet wurde. Um diese Ungenauigkeiten zu minimieren, wird in dieser Arbeit der Lastindikator „Gewichtete Verbindungszustände“ eingeführt. Dieser Lastindikator ordnet jedem Verbindungszustand S des Gatekeepers ein Gewicht $W(S)$ zu. Bei jeder ankommenden Signalisiernachricht wird der Lastindikatorwert LIV_{WCS} (*LIV – Load Indicator Value*, *WCS – Weighted Connection States*) wie in Gl. (3.9) dargestellt aktualisiert.

$$LIV_{WCS}(new) = LIV_{WCS}(old) - W(S_{old}) + W(S_{new}) \tag{3.9}$$

  Damit bildet LIV_{WCS} die Summe der Gewichte aller derzeit durch den Gatekeeper bearbeiteten Verbindungen. Dieser Lastindikator erlaubt die Abschätzung des zukünftigen Ressourcenbedarfs, indem dies bei der Festlegung der Gewichte der Verbindungszustände beachtet wird. Darüber hinaus können auch Verbindungen in der Nutzdatenaustauschphase einbezogen werden, die zwar zu diesem Zeitpunkt nahezu keine Ressourcen benötigen, jedoch zum einen überwacht werden müssen, z. B. zur Gebührenerfassung, und zum anderen jederzeit in die Verbindungsabbauphase eintreten können. Bild 3.5 zeigt einen möglichen zeitlichen Verlauf der Verbindungsgewichte während des Verbindungsaufbaus sowie die entsprechenden Nachrichten, die die einzelnen Zustandsübergänge auslösen. Wie beim Lastindikator „Anzahl offener Anfragen“ kann die Lastzustandsbestimmung ereignis- oder zeitgesteuert erfolgen. Dabei werden den einzelnen Lastzuständen entsprechende Wertebereiche des Lastindikatorwerts LIV_{WCS} zugeordnet.

- Weitere Lastindikatoren
  Zur Ableitung der aktuellen Belastung des Gatekeepers sind grundsätzlich auch die Kenngrössen „Anzahl aktiver Timer“, „Rate der Nachrichtenwiederholungen“, „Verlustwahrscheinlichkeit von Nachrichten oder Anfragen“ sowie „Speicherbedarf für Verbindungen“ verwendbar. Da diese Messwerte entweder nur sehr hohe Lastbereiche anzeigen („Rate der

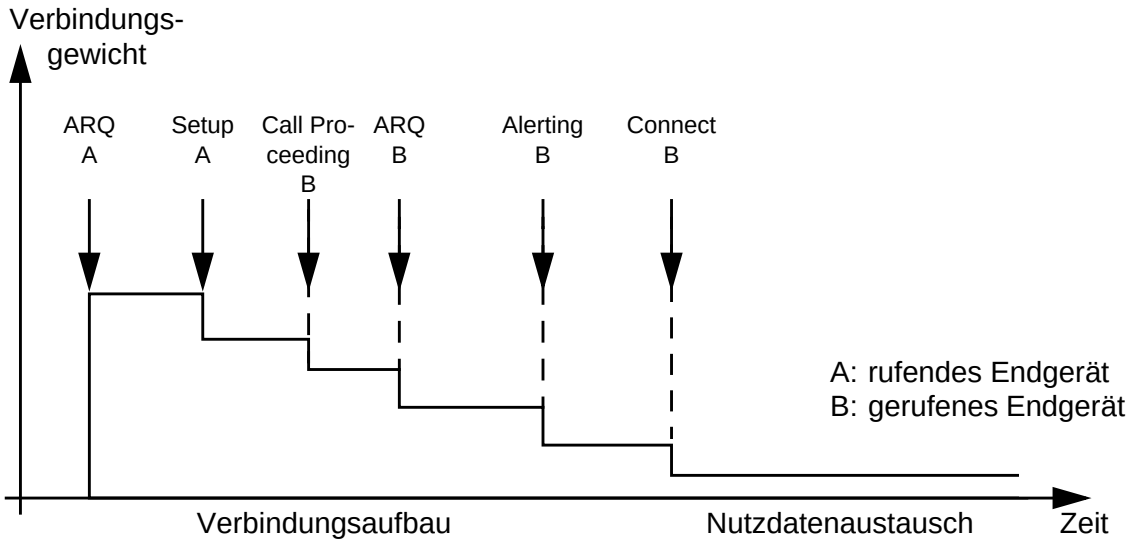


Bild 3.5: Beispielhafter zeitlicher Verlauf der Verbindungsgewichte beim Verbindungsaufbau (Signalisierung für die Medienkanalsteuerung nicht enthalten)

Nachrichtenwiederholungen“, „Verlustwahrscheinlichkeiten“), Ressourcen überwachen, die in heutigen Systemen keinen Engpass mehr darstellen („Anzahl aktiver Timer“), oder den Lastzustand ungenauer bestimmen, als bereits vorgestellte Verfahren („Speicherbedarf“ gegenüber „Anzahl offener Anfragen“ und „Gewichtete Verbindungszustände“), werden diese Indikatoren in dieser Arbeit nicht weiter untersucht.

Bei den vorgestellten Lastindikatoren ist zu beachten, dass diese von der Leistungsfähigkeit des jeweiligen Gatekeepers abhängen können, d. h. dass beispielsweise eine bestimmte Warteschlangenlänge für einen Gatekeeper einen Überlastzustand anzeigt, während sie für einen wesentlich leistungsfähigeren Gatekeeper eine normale Arbeitsbelastung bedeuten kann. Daher muss in der Regel die Zuordnung der Lastindikatorwerte zu den einzelnen Lastzuständen an die jeweiligen Gatekeeper-Realisierungen angepasst werden.

Damit eine Lastverteilung über Zonengrenzen hinweg, wie sie in Abschnitt 3.5.2.2 vorgestellt wird, sinnvoll durchgeführt werden kann, ist die Verwendung von Lastindikatoren notwendig, die Lastzustände ganzer Zonen anzeigen. Dabei ist insbesondere die Lastanzeige für einen Cluster von Gatekeepern, bei dem ein Verbund von Gatekeepern die Steuerung einer Zone übernimmt, von Interesse. Bei diesen Lastindikatoren ist die Stabilität von großer Bedeutung, da die bei der Zonen-überschreitenden Steuerungsoptimierung angewandten Verfahren aufwendig sind und relativ viel Zeit benötigen, bis sie wirksam werden. Dies kann durch entsprechende Filterverfahren, wie z. B. dem gleitenden Mittelwert, realisiert werden. Des Weiteren sollen die zonenüberschreitenden Verfahren nur bei sehr hoher Belastung angewendet werden, um den entsprechenden Aufwand zu rechtfertigen.

Eine Möglichkeit zur Bestimmung des Lastzustands eines Clusters von Gatekeepern ist die Bildung des Mittelwerts der Lastzustände aller Gatekeeper des Clusters. Für eine möglichst genaue Ermittlung des aktuellen Lastzustands in sehr hohen Lastbereichen ist jedoch die Rate der durch Gatekeeper abgelehnten oder fehlgeschlagenen Verbindungsanforderungen als Lastindikator geeigneter, da hiermit eine Differenzierung verschiedener Überlastzustände in den Lastbereichen möglich ist, in denen die Lastindikatoren der einzelnen Gatekeeper jeweils bereits den Maximalwert anzeigen.

3.5.1.2 Kombinationen von Lastindikatoren

Neben der Verwendung einzelner Lastindikatoren kann es sinnvoll sein, mehrere Lastindikatoren entsprechend zu kombinieren. Damit können spezielle Eigenschaften einzelner Indikatoren geeignet verknüpft werden, so dass sie den gewünschten Eigenschaften möglichst nahe kommen.

So sollte z. B. der Lastindikator „Gradient der Warteschlangenlänge“ mit anderen Lastindikatoren kombiniert werden, da er ein entsprechend steiles Ansteigen oder Abfallen der Warteschlangenlänge sicher erkennt, aber bei langsamen Laständerungen nicht reagieren und somit keine Lastzustandsänderung hervorrufen würde.

Für die Bestimmung des aktuellen Lastzustands eines Clusters von Gatekeepern ist ebenfalls eine Kombination von Lastindikatoren sinnvoll. Dabei können für unterschiedliche Lastbereiche verschiedene Lastindikatoren angewandt werden: Für den unteren Lastbereich wird dabei der Mittelwert der Lastzustände der einzelnen Gatekeeper verwendet. Sobald die Rate der durch Gatekeeper abgelehnten oder fehlgeschlagenen Verbindungsanforderungen einen bestimmten Wert überschreitet, wird für die Ermittlung des Lastzustands des Clusters diese Rate benutzt. Dies erlaubt eine differenzierte Bestimmung der Belastung des Clusters sowohl in den unteren Lastbereichen als auch im Überlastbereich.

3.5.2 Lastverteilung

Im Folgenden werden Verfahren zur Verteilung der Steuerungslast auf mehrere Gatekeeper vorgestellt. Dabei werden in Abschnitt 3.5.2.1 zunächst *Intrazonen*-Lastverteilungsverfahren beschrieben, die die Verteilung innerhalb einer Zone erlauben. Anschließend werden in Abschnitt 3.5.2.2 *Interzonen*-Verteilungsverfahren vorgestellt, die die Lastverteilung über Zonengrenzen hinweg und somit zwischen verschiedenen Zonen ermöglichen.

3.5.2.1 Intrazonen-Lastverteilung

Um die Steuerungslast innerhalb einer Zone geeignet verteilen zu können, wird im folgenden Abschnitt zunächst der *Gatekeeper-Cluster* definiert. Anschließend wird beschrieben, wie ein derartiger Gatekeeper-Cluster realisiert werden kann. Des Weiteren wird auf die Problematik

der Granularität der Lastverteilung eingegangen, und es werden Betrachtungen zur Datenverwaltung innerhalb eines Gatekeeper-Clusters und zur Konsistenz dieser Daten vorgenommen. Abschließend werden Lastverteilungsverfahren, die in H.323-basierten VoIP-Umgebungen von Interesse sind, vorgestellt.

Definition Gatekeeper-Cluster

Ein Gatekeeper-Cluster besteht aus mehreren Gatekeepern, die gemeinsam die Verwaltung einer Zone durchführen. Dazu sind die Gatekeeper leistungsfähig miteinander vernetzt, so dass die Kommunikation zwischen den einzelnen Cluster-Mitgliedern mit minimalen Verzögerungen erfolgt. In Bild 3.6 ist ein Beispiel für eine Zone, die durch einen Gatekeeper-Cluster verwaltet wird, dargestellt. Die Diensterbringung des Gatekeeper-Clusters kann transparent durchgeführt werden, d. h. die einzelnen Zonenmitglieder sehen den Cluster als eine Einheit, ohne dass die Realisierung und somit die einzelnen Gatekeeper des Clusters sichtbar werden oder beachtet werden müssen.

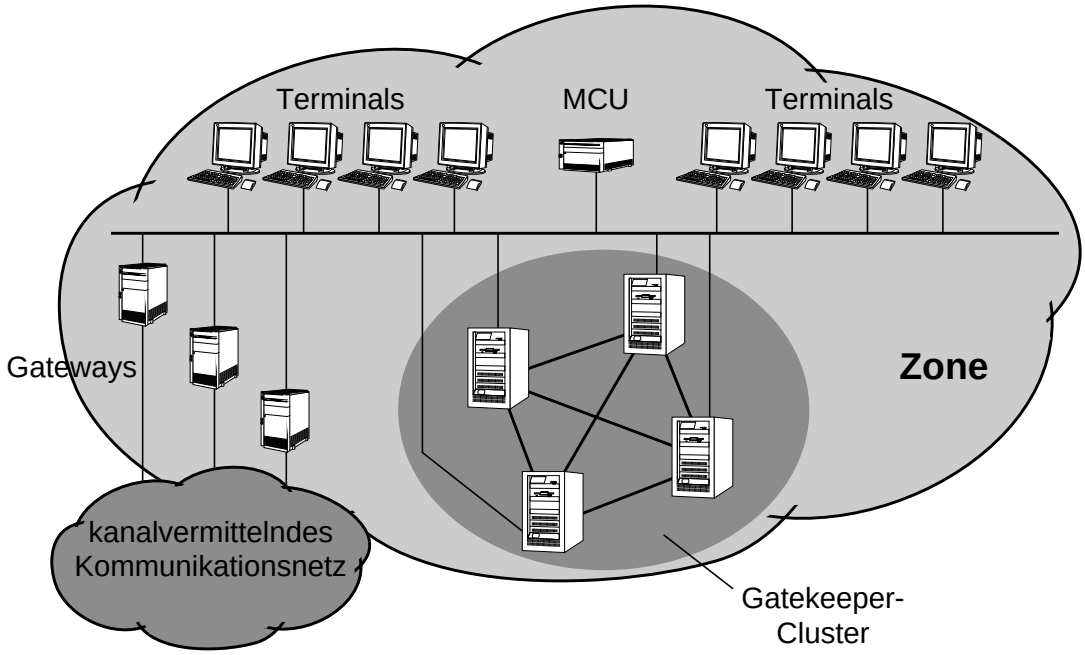


Bild 3.6: Gatekeeper-Cluster in einer H.323-Zone

Im Folgenden werden die wichtigsten Vorteile eines Clusters im Vergleich zu einer einzelnen Komponente vorgestellt:

- Leistungsfähigkeit
  Ein Cluster von Komponenten kann in der Regel eine wesentlich höhere Leistungsfähigkeit als eine alleinstehende Komponente erreichen, da die Cluster-Mitglieder gemeinsam den entsprechenden Dienst erbringen und daher eine Verteilung der anfallenden Last durchgeführt werden kann. Insbesondere können innerhalb eines Clusters weniger leistungsfähige

Komponenten verwendet werden, so dass die Kosten für einen derartigen Cluster geringer ausfallen können als für eine ähnlich leistungsfähige, einzelne Komponente [28].

- Skalierbarkeit
  Um die Leistungsfähigkeit eines Clusters zu erhöhen, so dass eine größere Anzahl von Anforderungen bedient werden kann, genügt es in der Regel, die Anzahl der Cluster-Mitglieder zu erhöhen. Gegebenenfalls müssen die Lastverteilungsverfahren entsprechend angepasst werden. Dabei sind heterogene Cluster möglich, d. h. die einzelnen Cluster-Mitglieder können unterschiedlich realisiert sein und somit verschieden leistungsfähig sein. Es ist jedoch zu beachten, dass die Anzahl der Cluster-Mitglieder nicht beliebig groß werden kann, da der Leistungsgewinn durch weitere Cluster-Mitglieder geringer als der dabei notwendige Verwaltungsaufwand werden kann. Im Gegensatz dazu muss bei einer einzelnen Komponente zur Erhöhung der Leistungsfähigkeit diese entweder vollständig ausgetauscht werden oder es müssen einzelne Bestandteile durch entsprechend leistungsfähigere ersetzt werden.

- Ausfallsicherheit
  Da bei einem Cluster mehrere Komponenten einen Dienst erbringen, können bei Ausfall einer Komponente die anderen die Dienstbearbeitung für die ausgefallene Komponente übernehmen. Dabei hängt es von der Realisierung der Steuerung des Clusters ab, ob Anforderungen, die bereits in Bearbeitung bei der ausgefallenen Komponenten waren, ebenfalls übernommen werden oder ob sie verloren gehen. Wenn dagegen eine einzelne, alleinstehende Komponente ausfällt, kann der entsprechende Dienst nicht erbracht werden. Die einzelne Komponente selbst sollte somit sehr ausfallsicher sein, um eine hohe Systemverfügbarkeit zu erreichen.

Bei der Verwendung von Cluster müssen jedoch zusätzliche Aufgaben gelöst werden, die bei einer einzelnen Komponente nicht oder in wesentlich einfacherer Form auftreten. In [28] werden beispielsweise die folgenden genannt:

- Verwaltung
  Die Verwaltung eines verteilten Systems ist wesentlich komplexer als einer einzelnen Komponente. Beispielsweise müssen die einzelnen Komponenten des Clusters überwacht werden, um deren korrekte Funktionsweise sicherzustellen. Dies kann z. B. durch den Austausch von Statusmeldungen durchgeführt werden.

- Lastverteilung
  Bei der Verteilung der Dienstbearbeitung muss zunächst festgestellt werden, ob die einzelnen Cluster-Mitglieder den Dienst vollständig bearbeiten können, oder ob die Bearbeitung weiter untergliedert und auf mehrere Komponenten aufgeteilt werden muss. Des Weiteren kann in einem Cluster eine Funktionsteilung vorgenommen werden, so dass für einzelne Dienste spezialisierte Komponenten existieren. In diesem Fall muss bei der Verteilung der

Last der zu erbringende Dienst beachtet werden. Schließlich muss die Verteilung der Last auf die einzelnen Komponenten geeignet durchgeführt werden, so dass die Leistungsfähigkeit dieser Komponenten möglichst effizient ausgenutzt wird.

- Reaktion auf teilweisen Ausfall
  Damit der Ausfall einzelner Komponenten die Diensterbringung des Clusters möglichst wenig beeinträchtigt, muss darauf entsprechend reagiert werden. So muss zunächst ein Ausfall erkannt werden, um anschließend die Dienstbearbeitung auf die anderen geeigneten Cluster-Mitglieder zu verteilen.

- Gemeinsame Verwendung von Daten
  Bei der Bearbeitung der Dienste werden bestimmte Daten von allen Cluster-Mitgliedern gemeinsam verwendet. Wenn diese Daten konstant sind oder sich nur selten ändern, können Kopien dieser Daten an die einzelnen Komponenten des Clusters verteilt werden. Bei sich schnell ändernden Daten, wie z. B. Zustände von Teilnehmern oder Endgeräten, müssen spezielle Verfahren zur Sicherstellung der Konsistenz dieser Daten angewandt werden. Um die Probleme, die durch die gemeinsame Verwendung von Daten entstehen, möglichst zu umgehen, erfolgt die Lastverteilung bei Diensten, die aus mehreren Anfragen bestehen meist statisch. Dabei ist ein Endgerät einem bestimmten Cluster-Mitglied fest zugeordnet, wie es z. B. bei der VoIP-Architektur der Firma *CISCO* durchgeführt wird. Des Weiteren ist dies auch ein Grund für die weite Verbreitung der Cluster-Verfahren bei Web-Servern, da dort in der Regel die einzelnen Anfragen unabhängig voneinander behandelt werden können [11, 14, 20, 94, 101].

Realisierungsformen für Gatekeeper-Cluster

Zur Realisierung eines Gatekeeper-Clusters können unterschiedliche Kriterien angewandt werden, die zu entsprechenden Realisierungsformen führen.

Ein Kriterium für die Realisierung eines Gatekeeper-Clusters ist das angewandte Steuerungsprinzip. Dabei wird zwischen zentraler und verteilter Steuerung unterschieden.

Bei der zentralen Steuerung ist eine ausgezeichnete Komponente für die Steuerung des Clusters zuständig und führt die Lastverteilung auf die einzelnen Cluster-Mitglieder durch. Dabei sollte sichergestellt werden, dass eine andere Komponente die Steuerungsaufgaben dieser zentralen Komponente übernehmen kann, da ansonsten bei ihrem Ausfall der Gatekeeper-Cluster vollständig ausfällt. Des Weiteren kann diese zentrale Komponente die Leistungsfähigkeit des gesamten Clusters einschränken, da alle ankommenden Anforderungen von ihr weitergeleitet werden müssen. Daher muss diese Komponente entsprechend leistungsfähig realisiert sein.

Im Gegensatz zur zentralen Steuerung wird bei der verteilten Steuerung der Gatekeeper-Cluster von allen Mitgliedern gemeinsam gesteuert. Dabei entscheidet jede Komponente, ob sie

weitere Anforderungen bearbeiten kann oder ob neue Anforderungen an andere Mitglieder weitergeleitet werden sollen. Der Vorteil dieses Verfahrens liegt in dem Fehlen einer zentralen Instanz, die bei einem Ausfall den gesamten Cluster funktionsunfähig machen würde und darüber hinaus auch einen Engpass für den Cluster darstellen könnte. Jedoch muss das angewandte Verfahren zur Lastverteilung in allen Komponenten des Clusters realisiert sein.

Ein weiteres Kriterium zur Realisierung eines Gatekeeper-Clusters ist seine Transparenz gegenüber den Endpunkten, d. h. ob die einzelnen Cluster-Mitglieder für die Endpunkte sichtbar sind und sie damit an den Lastverteilungsverfahren beteiligt sind, oder ob die Endpunkte den Cluster als eine Einheit betrachten und somit vollständig unabhängig von den Lastverteilungsverfahren agieren. Dazu werden im Folgenden mögliche Realisierungsformen mit unterschiedlicher Beteiligung der Endpunkte vorgestellt:

- Statische Konfiguration in den Endpunkten
  Eine einfache Realisierungsform eines Gatekeeper-Clusters ist die statische Aufteilung der Endpunkte auf die Cluster-Mitglieder. Dabei wird jeder Endpunkt mit der Adresse seines verwaltenden Cluster-Mitglieds konfiguriert. Zur Erhöhung der Ausfallsicherheit können Adressen von Reserve-Cluster-Mitgliedern festgelegt werden, so dass diese beim Ausfall des ursprünglichen Cluster-Mitglieds von den Endpunkten als Gatekeeper verwendet werden. Bei dieser Realisierungsform ist eine starke Beteiligung der Endpunkte notwendig und bei einer Änderung der Struktur des Clusters muss die Konfiguration der Endpunkte entsprechend angepasst werden. Des Weiteren entspricht diese Lösung, wenn man von den Reserve-Cluster-Mitgliedern absieht, weitestgehend dem H.323-Zonen-Konzept, wobei jedes Cluster-Mitglied seine eigene Zone aufspannt.

- Konzept des alternativen Gatekeepers
  Das in Version 4 von H.323 beschriebene und in [82] genauer untersuchte Konzept des alternativen Gatekeepers definiert Prozeduren für den Wechsel eines Endpunkts zu einem anderen Gatekeeper. Dieser Wechsel kann sowohl durch den ursprünglichen Gatekeeper selbst initiiert als auch vom Endpunkt bei einem Fehlerfall veranlasst werden. In beiden Fällen erhält der Endpunkt eine Liste möglicher alternativer Gatekeeper durch den ursprünglichen Gatekeeper. Sie wird entweder bei der Registrierung oder bei einer ablehnenden RAS-Nachricht des Gatekeepers an den Endpunkt gesendet. Die Umleitung auf einen alternativen Gatekeeper kann für einzelne Anfragen oder permanent erfolgen. Dieses Konzept kann für die Realisierung eines Gatekeeper-Clusters verwendet werden, indem die Lastverteilung mittels Umleitung der Anfragen oder der Endpunkte durchgeführt wird. Im Gegensatz zur statischen Konfiguration der Endpunkte kann bei dieser Realisierungsform ein dynamisches Lastverteilungsverfahren angewandt werden, wobei dies aber mit der Beteiligung der Endpunkte abläuft, die das alternative Gatekeeper-Konzept entsprechend

unterstützen müssen. Des Weiteren ergibt sich durch den zusätzlich notwendigen Nachrichtenaustausch eine größere Verzögerung der Diensterbringung.

- Aggregierendes Server-Zugriffsprotokoll (*Aggregate Server Access Protocol*)
  Das von der IETF in [112] definierte *Aggregate Server Access Protocol* (ASAP) wird zusammen mit dem *Endpoint Name Resolution Protocol* (ENRP, [125]) und dem Transportprotokoll *Stream Control Transmission Protocol* (SCTP, [111]) verwendet. Es erlaubt die Trennung von logischen Kommunikationsendpunkten und ihren physikalischen Adressen, so dass ohne Mitwirkung der Anwendungen verschiedene Komponenten unter einer logischen Adresse erreicht werden können und somit für einen Dienst zur Verfügung stehen. Zur Realisierung dieser Funktionalität wird, wie in Bild 3.7 dargestellt, eine Zwischenschicht zwischen der Anwendungs- und der Transportschicht eingefügt. Diese Zwischenschicht, in der das ASAP implementiert ist, verwendet das ENRP, das die Verwaltung der physikalischen Adressen der diensterbringenden Komponenten durchführt. Wenn eine Anwendung eine Anfrage an eine Komponente sendet, verwendet sie eine logische Adresse. Diese wird durch ASAP und ENRP in eine entsprechende physikalische Adresse umgesetzt, wobei das ENRP eine Liste aller möglichen Adressen ermittelt und das ASAP eine Adresse aus dieser Liste bestimmt, so dass die Anfrage an die ausgewählte Komponente gesendet wird. Mit diesem Verfahren kann ein Gatekeeper-Cluster realisiert werden, ohne dass dies für die H.323-Anwendungen in den Endpunkten sichtbar wird. Jedoch müssen alle Endpunkte über die ASAP-Zwischenschicht verfügen, damit eine sinnvolle Verwendung des Gatekeeper-Clusters erzielt wird. Des Weiteren erfolgt die Bestimmung der Liste der möglichen Adressen für eine Anfrage durch Nachrichtenaustausch zwischen der entsprechenden ENRP-Instanz des Endpunkts und einem ENRP-Server, so dass es dadurch zu einer Verzögerung der Diensterbringung kommt.

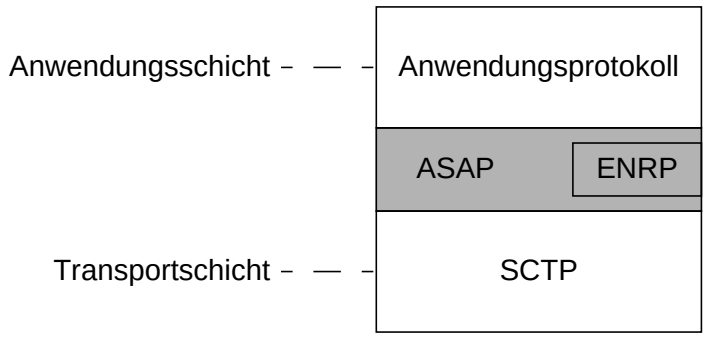


Bild 3.7: Zwischenschicht bei der Verwendung von ASAP und ENRP

- Einheitlicher Zugangspunkt
  Eine für die Endpunkte vollständig transparente Realisierung eines Gatekeeper-Clusters kann erreicht werden, indem, wie in Bild 3.8 dargestellt, eine zentrale Komponente die ankommenden Anforderungen auf die einzelnen Cluster-Mitglieder verteilt. Diese Kompo-

nente wird als *Dispatcher* bezeichnet. Je nach Realisierung können die Antworten der einzelnen Cluster-Mitglieder wieder über den Dispatcher geführt oder direkt an die entsprechenden Endpunkte gesendet werden. Bei dieser Lösung wird der gesamte Cluster unter einer Adresse erreicht, so dass die Struktur des Clusters von aussen nicht sichtbar wird. Diese Realisierungsform legt zwar eine zentrale Steuerung des Clusters nahe, jedoch ist dies nicht zwingend: Beispielsweise könnte der Dispatcher die einzelnen Anforderungen statisch an die Cluster-Mitglieder weiterleiten und diese führen dann die entsprechenden Steuerprozeduren für die Lastverteilung durch.

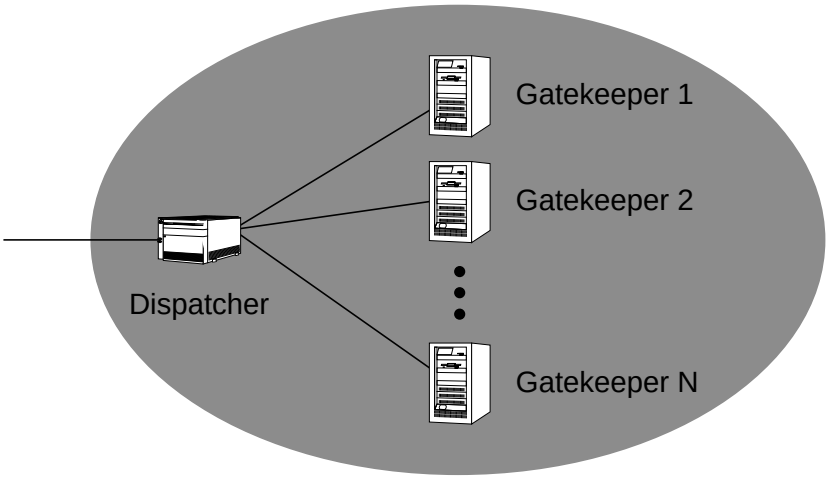


Bild 3.8: Schematische Darstellung einer vollständig transparenten Cluster-Realisierung mittels eines einheitlichen Zugangspunkts

Granularität der Lastverteilung

Ein wesentlicher Unterschied des Gatekeeper-Clusters im Vergleich zu einem Web-Server-Cluster besteht darin, dass zur Erbringung eines Dienstes mehrere, zusammenhängende Anfragen bearbeitet werden müssen, während bei einem Web-Server-Cluster in der Regel jede Anfrage separat betrachtet und bearbeitet werden kann. Daher spielt die Granularität der Lastverteilung eine wesentliche Rolle, insbesondere bezüglich der im nachfolgenden Abschnitt vorgenommenen Betrachtungen zur Datenverwaltung. Folgende Ebenen der Lastverteilung sind für einen Gatekeeper-Cluster relevant:

- Endpunkt-Ebene
  Dies stellt die gröbste Granularität der Lastverteilung dar, bei dem für jeden Endpunkt das zuständige Cluster-Mitglied festgelegt wird. Da diese Zuordnung nur bei einer Umstrukturierung der Zone oder des Clusters geändert wird, entspricht dies einem statischen Lastverteilungsverfahren.

- Verbindungs-Ebene
  Erfolgt die Lastverteilung auf der Verbindungsebene, wird für jede neu ankommende Ver-

bindung ein Cluster-Mitglied bestimmt, das die Verbindungsbearbeitung vollständig bis zum Verbindungsende durchführt.

- Verbindungsphasen-Ebene
  Wie in Abschnitt 3.5.1.1 erwähnt, kann eine Verbindung in verschiedene Phasen wie z. B. Verbindungsaufbau und Verbindungsabbau untergliedert werden. Bei der Lastverteilung, die auf dieser Verbindungsphasen-Ebene basiert, wird für jede Verbindungsphase ein Cluster-Mitglied festgelegt, das für die Bearbeitung der Verbindung in dieser Phase zuständig ist. Tritt die Verbindung in eine neue Phase ein, wird erneut ein Cluster-Mitglied für die Verbindungsbearbeitung in dieser Phase bestimmt.

- Nachrichten-Ebene
  Die feinste Granularität der Lastverteilung stellt die Verteilung auf der Nachrichten-Ebene dar, bei der für jede ankommende Nachricht ein Cluster-Mitglied für die Bearbeitung dieser Nachricht bestimmt wird.

Bei der Bewertung der Granularität der Lastverteilung muss zum einen beachtet werden, wie effektiv die Lastverteilung durchgeführt wird, d. h. wie gleichmäßig die Cluster-Mitglieder belastet werden. Beispielsweise sollte es nicht vorkommen, dass ein Cluster-Mitglied in Überlast gerät, während ein anderes noch über wesentliche Kapazitäten verfügt. In diesem Fall ist eine feine Granularität wünschenswert. Zum anderen muss der Aufwand für die Durchführung der Lastverteilung beachtet werden. Dabei ist z. B. die Häufigkeit der Ausführung der Lastverteilungsalgorithmen oder des Zugriffs auf gemeinsame Daten, wie er im folgenden Abschnitt beschrieben wird, relevant.

Datenverwaltung und Konsistenzbetrachtungen

Zur Erbringung der vorgesehenen Dienste benötigen die Gatekeeper entsprechende Daten. Dabei kann zwischen nahezu konstanten Konfigurationsdaten und sich häufig ändernden Zustandsdaten unterschieden werden. Zu den Konfigurationsdaten zählen beispielsweise Endpunktadressen, Berechtigungen der einzelnen Teilnehmer, Lage und Funktionalität von Gateways sowie Gebührentabellen. Beispiele für Zustandsdaten sind die Belegung von speziellen Komponenten und Gateways, Verbindungszustände der einzelnen H.323-Verbindungen sowie der aktuelle Bedarf an Übertragungskapazität auf den einzelnen Übertragungsabschnitten.

In einem Gatekeeper-Cluster erbringen die einzelnen Cluster-Mitglieder die Dienste gemeinsam. Dazu ist es notwendig, dass alle Cluster-Mitglieder auf diese Daten Zugriff haben, sobald sie benötigt werden, und dass die Konsistenz dieser Daten gewährleistet ist. Bei den nahezu konstanten Daten könnten diese auf die Cluster-Mitglieder *gespiegelt* werden, d. h. jedes Cluster-Mitglied erhält eine Kopie dieser Daten. Falls diese Daten geändert werden müssen, wird diese Änderung schnellstmöglich den Cluster-Mitglieder mitgeteilt. Zur Vermeidung von Inkonsistenzen können beispielsweise die zu ändernden Daten zunächst bei allen Cluster-Mit-

gliedern gesperrt und erst nach der Änderung wieder frei gegeben werden. Eine weitere Möglichkeit ist die Verwendung einer speziellen Komponente, auf der eine entsprechende Datenbank realisiert ist. Auf diese können die einzelnen Cluster-Mitglieder beispielsweise mittels des LDAP zugreifen.

Da sich die Zustandsdaten sehr schnell ändern und deren Konsistenz, insbesondere bei den Verbindungszuständen, absolut sichergestellt sein muss, sollten entsprechend effiziente Verfahren angewandt werden, die dies berücksichtigen. Eine Möglichkeit ist die Verwendung eines gemeinsamen Speichers durch die Cluster-Mitglieder, wobei der Zugriff auf die einzelnen Daten geeignet gesteuert werden muss, damit keine gleichzeitigen Schreib- und Lesezugriffe erfolgen, die zu fehlerhaften Daten führen könnten. Derartige Verfahren werden beispielsweise bei fehlertoleranten Systemen verwendet, damit im Fehlerfall eine Reserve-Komponente die Aufgaben der fehlerhaften Komponente übernehmen kann, ohne dass es zu Beeinträchtigungen bei der Diensterbringung kommt. Eine weitere Möglichkeit ist die Weitergabe der Zustandsdaten mittels Cluster-interner Nachrichten. Dabei können die Zustandsdaten entweder direkt zwischen den Cluster-Mitglieder oder über eine zentrale Instanz ausgetauscht werden. Die Sicherstellung der Konsistenz der Daten kann damit wie folgt realisiert werden: Für den Lesezugriff verfügt bei der direkten Weitergabe der Daten jedes Cluster-Mitglied über eine aktuelle Kopie der Daten, während bei der Verwendung einer zentralen Instanz die notwendigen Daten zunächst bei dieser zentralen Instanz angefordert und erhalten werden müssen. Für einen Schreib-Zugriff müssen die entsprechenden Daten jeweils zunächst gesperrt werden, um sie nach dem Schreiben wieder frei zu geben. Dabei muss bei der direkten Weitergabe das Sperren und Freigeben für jedes Cluster-Mitglied einzeln durchgeführt werden, während bei der Verwendung einer zentralen Instanz die Daten nur bei dieser zentralen Instanz gesperrt und nach der Aktualisierung wieder frei gegeben werden müssen.

Wie aus diesen Ausführungen ersichtlich wird, sind die Schreibe- und Leseoperationen auf gemeinsame Daten relativ aufwendig. Daher ist es sinnvoll, den gemeinsam verwendeten Datenanteil möglichst gering zu halten. Dies kann beispielsweise durch eine gröbere Granularität der Lastverteilung erreicht werden, da nur diejenigen Daten allen Cluster-Mitgliedern zugänglich gemacht werden müssen, die von einem Cluster-Mitglied bei der Übernahme der Bearbeitung benötigt werden. Erfolgt z. B. die Lastverteilung auf der Verbindungs-Ebene, können die H.323-Verbindungszustände lokal bei dem bearbeitenden Cluster-Mitglied gehalten werden, da er den kompletten Ruf bearbeiten wird. Erst nach Verbindungsende müssen die entsprechenden Daten, wie z. B. die Freigabe der benötigten Ressourcen und der neue Belegungszustand des Teilnehmers dem ganzen Cluster zur Verfügung gestellt werden. Um bei einem Verfahren mit gröberer Granularität die Ausfallsicherheit zu erhöhen, wird in Anhang R der Empfehlung H.323 vorgeschlagen, das Erreichen spezieller Verbindungszustände (*Checkpoints*) den anderen Cluster-Mitgliedern mitzuteilen, damit bei einem Ausfall des ursprünglichen Cluster-Mitglieds ein anderes die Verbindungsbearbeitung übernehmen kann.

Lastverteilungsverfahren

Im Folgenden werden einige Lastverteilungsverfahren vorgestellt, die innerhalb eines Gatekeeper-Clusters Anwendung finden können. Dazu werden die einzelnen Verfahren zunächst klassifiziert, bevor das Verfahren der Lastverteilung selbst beschrieben wird. Die Klassifizierung erfolgt dabei nach den Kriterien aus [15]. Des Weiteren werden prinzipielle Vor- und Nachteile der Verfahren vorgestellt.

- Statische Lastverteilung durch Konfiguration der Endpunkte
  Bei diesem Verfahren handelt es sich um ein statisches Lastverteilungsverfahren. Dabei wird während der Konfiguration jedem Endpunkt ein Cluster-Mitglied zugewiesen, das die Verwaltung des Endpunkts durchführt. Durch geeignete Konfiguration kann damit die Last auf die einzelnen Cluster-Mitglieder verteilt werden. Der Vorteil des Verfahrens liegt in seiner Einfachheit, da zur Ausführung der Lastverteilung keinerlei Ressourcen benötigt werden. Da die Konfiguration in der Regel während der Laufzeit nicht geändert wird, kann jedoch auf dynamische Veränderungen nicht reagiert werden. Des Weiteren besteht die Möglichkeit, dass einzelne Cluster-Mitglieder wesentlich stärker belastet werden als andere, falls die entsprechenden, zugeordneten Endpunkte sehr aktiv sind. Es entsteht somit kein Bündelungsgewinn (*Economy of scale*), da jedes Cluster-Mitglied im Prinzip seine eigene H.323-Zone aufspannt. Um bei Ausfall eines Cluster-Mitglieds die Diensterbringung zu gewährleisten, können Reserve-Cluster-Mitglieder für die Endpunkte konfiguriert werden, die im Fehlerfall die Verwaltung dieser Endpunkte übernehmen. Dieses Vorgehen wird z. B. bei der VoIP-Architektur der Firma *CISCO* angewandt, bei der bis zu zwei Reserve-Cluster-Mitglieder definiert werden, um eine hohe Ausfallsicherheit zu erreichen.

- Round-Robin
  Das „Round-Robin“-Verfahren zählt zu den dynamischen, nicht-kooperierenden Lastverteilungsverfahren, das in der Regel zentral gesteuert wird. Neu ankommende Anforderungen werden dabei zyklisch an die Cluster-Mitglieder weitergegeben. Im Gegensatz zu dem vorigen Verfahren können alle Cluster-Mitglieder von einem Endpunkt aus verwendet werden, so dass ein entsprechender Bündelungsgewinn erzielt wird. Des Weiteren benötigt die Ausführung der Lastverteilung wenig Ressourcen, da das Verfahren ebenfalls sehr einfach ist. Das Verfahren schneidet insbesondere bei homogener Last, d. h. dass die einzelnen Anfragen ähnlich viel Ressourcen beanspruchen, und homogenem Cluster sehr gut ab. Falls die Leistungsfähigkeit der einzelnen Cluster-Mitglieder sehr unterschiedlich ist, kann das „Weighted Round-Robin“-Verfahren angewandt werden, bei dem jedem Cluster-Mitglied ein Gewicht zugeordnet wird und die Last gemäß diesem Gewicht verteilt wird. Da das „Round-Robin“-Verfahren keine Information über die aktuelle Belastung der Cluster-Mitglieder verwendet, sinkt seine Effektivität jedoch bei inhomogener Last, d. h. wenn der Ressourcenbedarf für die Bearbeitung der einzelnen Anfragen sehr unterschiedlich ist.

- Zufällige Auswahl
  Ein weiteres dynamisches, nicht-kooperierendes Lastverteilungsverfahren ist die zufällige Auswahl eines Cluster-Mitglieds zur Bearbeitung einer Anfrage (*Random*-Verfahren). Dieses Verfahren kann sowohl zentral als auch verteilt gesteuert angewandt werden. Ansonsten sind die gleichen Bemerkungen wie beim „Round-Robin“-Verfahren gültig. Insbesondere kann ebenfalls eine Gewichtung der einzelnen Cluster-Mitglieder vorgenommen werden.

- Lastzustandsabhängige Auswahl
  Bei der lastzustandsabhängigen Auswahl der bearbeitenden Komponente handelt es sich um ein dynamisches, kooperierendes Lastverteilungsverfahren. Eine neue Anforderung wird an das Cluster-Mitglied weitergegeben, das durch seinen Lastzustand, den es mittels geeigneter Lastindikatoren bestimmt hat, anzeigt, dass es am wenigsten belastet ist. Der Vorteil des Verfahrens liegt in seiner Effizienz auch bei inhomogener Last und bei einem inhomogenen Gatekeeper-Cluster, da es auf die aktuelle Belastung der einzelnen Cluster-Mitglieder reagiert. Ein Nachteil ist jedoch die dabei notwendige Verwaltung der Lastzustände sowie der Austausch der Lastzustandsinformationen. Wie bereits im vorigen Abschnitt über die Datenverwaltung innerhalb eines Clusters beschrieben, kann der Informationsaustausch über einen Speicher oder über das Senden entsprechender Nachrichten erfolgen. Dabei kann sowohl eine zentrale Verwaltung der Daten als auch eine verteilte angewandt werden:

  - Zentrale Lastzustandsverwaltung
    Wird eine zentrale Verwaltung der Lastzustandsinformationen durchgeführt, teilen alle Cluster-Mitglieder ihren aktuellen Lastzustand einer zentralen Komponente mit. Dabei kann der Austausch der Lastzustandsinformationen zeit- oder zustandsgesteuert erfolgen. Der Vorteil der zeitgesteuerten Variante ist das frühzeitige Erkennen einer ausgefallenen Komponente, da diese keine Zustandsinformationen mehr liefert. Dabei darf die Intervallgröße zwischen zwei Zustandsanzeigen nicht zu groß gewählt werden, da die Informationen ansonsten schon veraltet sein könnten. Jedoch kann bei einer zu kleinen Intervallgröße die Belastung durch diese Prozedur selbst zu groß werden. Beim zustandsgesteuerten Austausch wird dagegen der aktuelle Zustand nur nach einer Lastzustandsänderung angezeigt, so dass die Anzahl der notwendigen Zustandsanzeigen auf ein Minimum reduziert ist.

  - Verteilte Lastzustandsverwaltung
    Bei der verteilten Lastzustandsverwaltung verfügt jedes Cluster-Mitglied über die entsprechenden Lastzustandsinformationen der anderen Cluster-Mitglieder. Daher müssen die einzelnen Lastzustände jeweils allen Cluster-Mitgliedern angezeigt werden. Dies kann insbesondere beim Austausch der Informationen mittels Nachrichten und bei einer entsprechenden Anzahl von Cluster-Mitgliedern zu einer erheblichen Belastung des Clusters führen. Ebenso wie bei der zentralen Lastzustandsverwaltung kann der Informa-

tionsaustausch zeit- oder zustandsgesteuert erfolgen, wobei die dort genannten Eigenschaften auch für die verteilte Lastzustandsverwaltung gelten.

Zu den dynamischen, kooperierenden Lastverteilungsverfahren mit verteilter Steuerung zählt das „Sender-Receiver“-Verfahren (Sender-Empfänger), das auf einem in [105] vorgestellten Verfahren basiert. Dabei kann jedes Cluster-Mitglied die Zustände *Sender*, *Receiver* oder *Ok* einnehmen. Im *Sender*-Zustand sollen möglichst alle Anforderungen an ein anderes Cluster-Mitglied weitergegeben werden, während der *Receiver*-Zustand anzeigt, dass noch Kapazitäten für Anforderungen anderer Cluster-Mitglieder zur Verfügung stehen. Der Zustand *Ok* zeigt eine Belastung an, die zwischen der des *Sender*- und des *Receiver*-Zustands liegt, wobei auch noch in diesem Zustand Anforderungen anderer Cluster-Mitglieder bearbeitet werden können. Jedes Cluster-Mitglied verfügt über jeweils eine Liste der Cluster-Mitglieder im *Receiver*- und im *Ok*-Zustand. Um den Informationsaustausch zwischen den Cluster-Mitglieder zu minimieren, werden nur Änderungen der oben genannten Zustände den anderen Cluster-Mitgliedern angezeigt. Dabei werden neue Zustandsinformationen an den Beginn der jeweiligen Liste gestellt, so dass die aktuellste Information vorne in der Liste steht. Wenn eine Anforderung bei einem Cluster-Mitglied ankommt, das sich im *Sender*-Zustand befindet, bestimmt es das erste Element der *Receiver*-Liste, an das die Anforderung anschließend weitergeleitet wird. Anschließend wird dieses Element an das Ende der *Receiver*-Liste gestellt. Falls die *Receiver*-Liste leer ist, wird nach dem gleichen Prinzip mit der *Ok*-Liste verfahren. Wenn diese ebenfalls leer sein sollte, wird die Anforderung lokal durch dieses Cluster-Mitglied bearbeitet. Ein Vorteil des Verfahrens liegt in der effizienten Lastverteilung auch bei inhomogener Last in einem inhomogenen Cluster. Des Weiteren skaliert das Verfahren sehr gut, da der Steuerungsaufwand auch bei einer großen Anzahl von Cluster-Mitgliedern relativ klein ist. Schließlich wird die Verwendung veralteter Zustandsinformationen minimiert, indem immer die aktuellsten Informationen zuerst ausgewertet werden. Die Nachteile liegen in dem notwendigen Informationsaustausch und im Aufwand zur Verwaltung der *Receiver*- und *Ok*-Listen, wobei das letztere durch eine entsprechend leistungsfähige Implementierung wenig ins Gewicht fallen sollte.

Da bei Verfahren mit verteilter Steuerung die Cluster-Mitglieder einzeln entscheiden, ob eine Anforderung lokal bearbeitet oder an ein anderes Cluster-Mitglied weitergeleitet wird, muss die Stabilität des Verfahrens sichergestellt werden, so dass eine Anforderung nicht ständig weitergeleitet wird, sondern schließlich von einem Cluster-Mitglied bearbeitet wird. Dies kann beispielsweise durch eine Begrenzung der Anzahl der Weiterleitungsvorgänge pro Anforderung erreicht werden. Da jeder Weiterleitungsvorgang darüber hinaus die Diensterbringung weiter verzögert, kann damit auch die Einhaltung der zulässigen Antwortverzögerungen unterstützt werden.

3.5.2.2 Interzonen-Lastverteilung

In diesem Abschnitt wird die Lastverteilung über Zonengrenzen hinweg vorgestellt. Dazu wird zunächst das Prinzip dieser *Interzonen*-Lastverteilung beschrieben, bevor auf die Granularität der Lastverteilung eingegangen wird. Anschließend erfolgen Betrachtungen zur Datenverwaltung und zum notwendigen Informationsaustausch zwischen den Zonen. Schließlich wird ein mögliches Verfahren für die Interzonen-Lastverteilung vorgestellt.

Prinzip

In einer H.323-basierten VoIP-Umgebung, die aus mehreren Zonen besteht, kann es vorkommen, dass der Gatekeeper bzw. der Gatekeeper-Cluster einer Zone überlastet ist, so dass nicht alle Anforderungen bedient werden können, während in benachbarten Zonen noch genügend freie Kapazitäten verfügbar wären. Gründe für eine derartige ungleichmäßige Belastung können beispielsweise in einer Veränderung der Randbedingungen, wie z. B. der Anzahl der Endpunkte einer Zone, oder in der Verfügbarkeit bestimmter spezieller Komponenten nur in einer einzelnen Zone liegen. In diesen Fällen kann eine Lastverteilung zwischen verschiedenen Zonen die Güte der Diensterbringung wesentlich erhöhen, so dass mehr Anforderungen erfolgreich bedient werden können.

Da durch das Konzept der Zonen eine Eingrenzung der verwalteten Daten einer VoIP-Umgebung realisiert wird, ist die Lastverteilung über Zonengrenzen hinweg deutlich aufwendiger als innerhalb einer Zone. Zum einen besitzt eine Zone in der Regel keine Konfigurationsdaten der Endpunkte anderer Zonen, die jedoch für die Kommunikationssteuerung notwendig sind. Zum anderen werden Zustandsdaten der anderen Zonen benötigt, um eine sinnvolle Lastverteilung zu ermöglichen.

Wie in Abschnitt 2.3.2 dargestellt, besteht nur eine logische Zuordnung eines Endpunkts zu einer Zone. Zur Realisierung der Interzonen-Lastverteilung kann diese Zuordnung zur Laufzeit der VoIP-Umgebung geändert werden, so dass eine gleichmäßigere Belastung der einzelnen Zonen erreicht wird. Eine derartige Änderung der Zuordnung entspricht im Prinzip einer Umstrukturierung der VoIP-Umgebung, wobei damit bereits Bereiche der Netzplanung berührt werden.

Bei der Verteilung der Last über Zonengrenzen hinweg besteht die Gefahr, dass eine lokal begrenzte Überlastsituation auf die anderen Zonen der VoIP-Umgebung ausgebreitet wird. Daher muss durch entsprechende Verfahren sicher gestellt werden, dass die Diensterbringung in den bisher wenig belasteten Zonen auch bei der Übernahme von Lastanteilen aus den überlasteten Zonen gesichert ist.

Die Interzonen-Lastverteilung kann sowohl zentral als auch verteilt gesteuert durchgeführt werden. Da jedoch keine dem Gatekeeper übergeordnete Steuerkomponenten für eine H.323-

basierte VoIP-Umgebung existiert, die diese zentrale Steuerung übernehmen könnte[1], scheint ein Verfahren mit verteilter Steuerung der Lastverteilung besser geeignet. Des Weiteren könnten bei einer zentralen Steuerung Skalierungsprobleme auftreten, da bei einer entsprechend großen VoIP-Umgebung die zu verwaltende Datenmenge sehr groß werden kann.

Granularität der Lastverteilung

Ebenso wie die Intrazonen-Lastverteilung kann die Interzonen-Lastverteilung mit unterschiedlichen Granularitäten erfolgen. Da die Durchführung der einzelnen Lastverteilungsverfahren relativ aufwendig ist, da z. B. Konfigurationsdaten von Endpunkten zwischen Zonen ausgetauscht werden müssen, darf diese Granularität nicht zu klein gewählt werden. Im Folgenden werden verschiedene Ebenen der Granularität für die Lastverteilung über Zonengrenzen hinweg vorgestellt:

- Verbindungs-Ebene
  Bei dieser Granularität wird die Steuerung einzelner VoIP-Verbindungen an andere Zonen weitergegeben, um die ursprünglich für die Bearbeitung zuständige Zone zu entlasten. Durch den Aufwand, der für die Weiterleitung der Verbindungsbearbeitung notwendig ist, erscheint jedoch eine Lastverteilung auf dieser Ebene wenig sinnvoll und wird daher nicht weiter betrachtet.

- Endpunkt-Ebene
  Eine größere Granularität stellt die Lastverteilung auf der Endpunkt-Ebene dar. Dabei werden, wie in Bild 3.9 dargestellt, ein oder mehrere Endpunkte einer anderen Zone zugeordnet, die weniger belastet ist als die ursprüngliche. Dies entspricht einer logischen Umstrukturierung der VoIP-Umgebung. Diese Umstrukturierung kann sowohl permanent als auch temporär erfolgen, so dass der oder die Endpunkte nach dem Ende der Überlastung wieder ihrer ursprünglichen Zone zugeordnet werden können.

- Gatekeeper-Ebene
  Bei entsprechend großen Lastunterschieden zwischen zwei Zonen können auch die lastaufnehmenden Komponenten verteilt werden, so dass ein Gatekeeper einer neuen Zone zugeordnet wird, wie es ebenfalls in Bild 3.9 dargestellt ist. Dies setzt jedoch voraus, dass in beiden betroffenen Zonen das Konzept des Gatekeeper-Clusters angewandt wird. Des Weiteren sollten die jeweiligen Gatekeeper-Cluster transparent für die Endpunkte realisiert sein, damit sie von dieser, zur Laufzeit stattfindenden Umstrukturierung der VoIP-Umgebung nicht betroffen sind. Ebenso wie bei der Lastverteilung auf Endpunkt-Ebene kann diese Form der Lastverteilung[2] permanent oder temporär durchgeführt werden. Der Aufwand für eine Lastverteilung auf der Gatekeeper-Ebene ist wesentlich höher als bei einer Verteilung

[1] Border Element und Clearing House bieten zwar übergeordnete Dienste an, wobei diese nur die Bestimmung der Zieladresse unterstützen und somit keine Steueraufgaben durchführen.

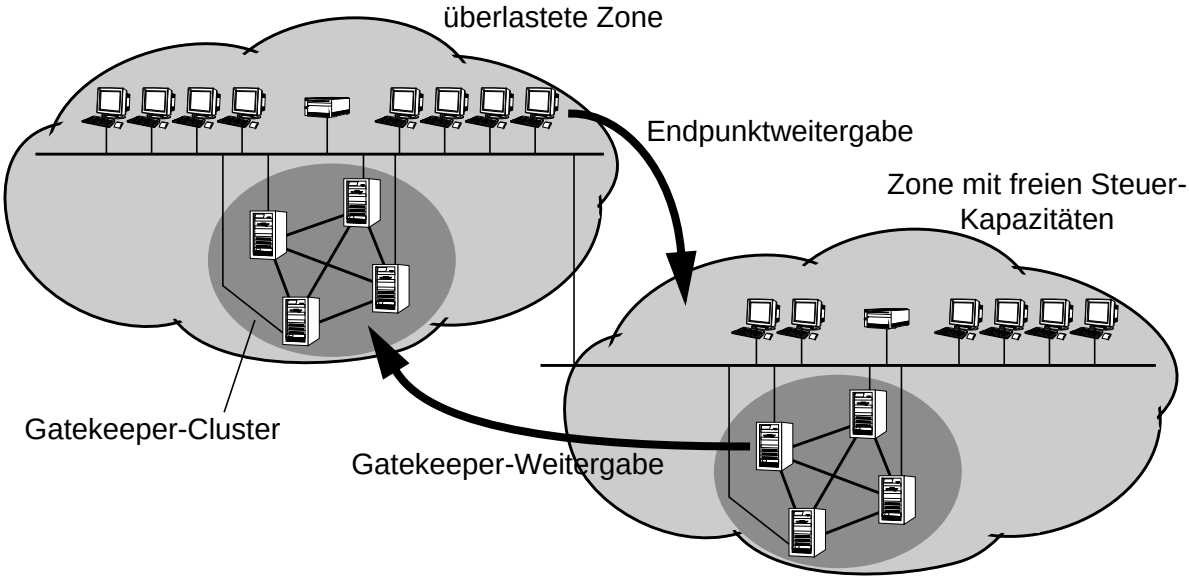


Bild 3.9: Interzonen-Lastverteilung mit unterschiedlichen Granularitäten

auf Endpunkt-Ebene, da beide betroffenen Gatekeeper-Cluster entsprechend umstrukturiert werden müssen. Dabei hängt der Aufwand auch von der Realisierung der Gatekeeper-Cluster selbst ab. Darüber hinaus muss der Gatekeeper für die neue Zone entsprechend umkonfiguriert werden, damit ihm die für die Diensterbringung notwendigen Endpunktdaten sowie die Zustandsdaten der Zone zur Verfügung stehen.

Die beiden letztgenannten Ebenen der Lastverteilung können auch gemeinsam innerhalb eines Lastverteilungsverfahrens verwendet werden. Dabei wird bei einem kleineren Unterschied der Belastung der Zonen die Lastverteilung auf Endpunkt-Ebene durchgeführt, während bei größeren Unterschieden die Verteilung auf der Gatekeeper-Ebene stattfindet. Des Weiteren kann die Lastverteilung auf Endpunkt-Ebene auch für Zonen angewandt werden, die nicht über einen Gatekeeper-Cluster verfügen, sondern nur durch einen einzelnen Gatekeeper verwaltet werden, oder wenn aus Gründen der Ausfallsicherheit eine Mindestanzahl von Gatekeepern in einem Cluster vorhanden sein muss.

Datenverwaltung

Bei der Interzonen-Lastverteilung wird das Zonen-Konzept, das eine Einschränkung der verwalteten Daten erlaubt, zwischenzeitlich außer Kraft gesetzt, um eine neue Zuordnung von Endpunkten und Gatekeepern zu Zonen durchzuführen. Dabei müssen die Daten, die bei einer derartigen Umstrukturierung aktualisiert werden müssen, berücksichtigt werden. Diese werden im Folgenden als Konfigurationsdaten bezeichnet. Des Weiteren benötigen die einzelnen

[2] Prinzipiell entspricht die Lastverteilung auf Gatekeeper-Ebene nicht einer Lastverteilung, sondern einer Verteilung der lastaufnehmenden Komponenten. Um jedoch die inhaltliche Verbindung zur Lastverteilung zu verdeutlichen, wird sie im weiteren Verlauf weiterhin unter dem Begriff Lastverteilung geführt.

Zonen Daten über die aktuelle Belastung der anderen Zonen, um mögliche Ziele der Lastverteilung zu bestimmen. Auf diese Zustandsdaten wird nach der folgenden Betrachtung der Konfigurationsdaten eingegangen.

Neben den bereits eingeführten Konfigurationsdaten eines Gatekeepers, wie z. B. Endpunktadressen, existieren für Gatekeeper-Cluster weitere, Cluster-spezifische Konfigurationsdaten, wie z. B. die Adressen der anderen Cluster-Mitglieder oder die Adresse des gemeinsamen Speicherbereichs für die Zustandsdaten. Diese Daten müssen bei der Lastverteilung über Zonengrenzen hinweg aktualisiert werden. Wenn ein Endpunkt einer anderen Zone zugeordnet wird, müssen dieser Zone alle Endpunkt-spezifischen Daten übergeben werden. Des Weiteren müssen die für die Adressauflösung zuständigen Komponenten darüber informiert werden, in welcher Zone dieser Endpunkt erreichbar ist, damit Verbindungen, die diesen Endpunkt als Ziel haben, direkt zu dieser Zone geleitet werden.

Wenn ein Gatekeeper einer anderen Zone zugeordnet wird, muss dieser zunächst aus seinem ursprünglichen Cluster entfernt werden. Dazu kann es, abhängig von der jeweiligen Cluster-Realisierung, notwendig sein, dass alle Cluster-Mitglieder über diese Umstrukturierung informiert werden, damit ihre Daten entsprechend angepasst werden. Um den Gatekeeper in den Gatekeeper-Cluster der Zielzone zu integrieren, müssen ebenfalls die entsprechenden Daten aktualisiert werden, so dass der wechselnde Gatekeeper an der Lastverteilung dieses Clusters teilnimmt. Darüber hinaus benötigt der Gatekeeper die zonenspezifischen Konfigurationsdaten, wie z. B. Endpunkttabellen, Gateway-Daten oder Adressen für den Zugriff auf die Zustandsdaten der Zone.

Ein weiterer Aspekt der Datenverwaltung bei der Interzonen-Lastverteilung stellen die Zustandsdaten der Zonen dar. Damit eine sinnvolle Lastverteilung über Zonengrenzen hinweg möglich ist, benötigen die einzelnen Zonen Informationen über die aktuelle Belastung bezüglich der Steuerung der anderen Zonen. Dazu kann, wie bereits in Abschnitt 3.5.1.1 beschrieben, ein gemeinsamer Lastzustand für einen Gatekeeper-Cluster bestimmt werden. Dieser Lastzustand wird z. B. mittels Austausch von Nachrichten an die anderen Zonen verteilt. Wenn detailliertere Informationen über die anderen Zonen benötigt werden, wie z. B. die Auslastung eines Gateways, wird dieser Informationsaustausch entsprechend aufwendiger.

Ein Verfahren für die Interzonen-Lastverteilung

Wie aus diesen Ausführungen ersichtlich wird, ist die Interzonen-Lastverteilung im Vergleich zur Intrazonen-Lastverteilung sehr aufwendig. Des Weiteren benötigen die einzelnen Maßnahmen relativ viel Zeit, da für die dabei notwendigen Umstrukturierungen die entsprechenden Konfigurations- und Zustandsdaten ausgetauscht werden müssen. Daher sollten diese Maßnahmen nur durchgeführt werden, wenn die entsprechende Überlastsituation lange genug anhält. Des Weiteren ist eine entsprechende Stabilität der Verteilung notwendig, um die VoIP-Umge-

bung nicht unnötig mit der Ausführung dieser Lastverteilung und dem damit verbundenen Umkonfigurieren von Zonen zu belasten. Im Folgenden wird ein mögliches Verfahren für die Interzonen-Lastverteilung vorgestellt, das diese Randbedingungen berücksichtigt.

Damit die Lastzustände ausreichend stabil bestimmt werden, d. h. dass Überlastungen nur angezeigt werden, wenn sie entsprechend lange andauern und nicht nur kurzzeitigen Impulsen entsprechen, werden die Lastindikatoren des Gatekeeper-Clusters geeignet geglättet. Dabei kann beispielsweise das in Abschnitt 3.2.1.2 vorgestellte Verfahren des gleitenden Mittelwerts mit einer großen Anzahl berücksichtigter Werte verwendet werden. Dabei werden die folgenden Lastzustände für einen Gatekeeper-Cluster unterschieden:

- *SenderGatekeeper*
  Dieser Lastzustand zeigt an, dass der Gatekeeper-Cluster einen Gatekeeper an eine andere Zone weitergeben könnte, so dass auch mit den übrigbleibenden Cluster-Mitgliedern die Diensterbringung erfolgreich durchgeführt werden könnte. Falls eine Zone nur über einen Gatekeeper verfügt oder bereits die Mindestanzahl von Gatekeepern in einem Cluster erreicht ist, wird dieser Zustand auch bei niedriger Belastung nicht eingenommen, sondern der folgende Zustand *ReceiverEndpoint*.

- *ReceiverEndpoint*
  Befindet sich ein Gatekeeper-Cluster bzw. ein in einer Zone alleinstehender Gatekeeper in diesem Lastzustand, verfügt er noch über genügend freie Kapazitäten, um einen oder mehrere weitere Endpunkte zu verwalten.

- *Ok*
  Dieser Lastzustand zeigt eine normale Belastung des Gatekeeper-Clusters an, d. h. dass weder Last an andere Cluster weiter gegeben werden sollte noch genügend freie Kapazitäten zur Übernahme weiterer Last zur Verfügung stehen.

- *SenderEndpoint*
  Wenn ein Gatekeeper-Cluster bzw. ein in einer Zone alleinstehender Gatekeeper sich in diesem Lastzustand befindet, ist er so überlastet, dass ein oder mehrere Endpunkte an eine andere Zone weitergegeben werden sollten.

- *ReceiverGatekeeper*
  Dieser Lastzustand zeigt eine starke Überlastung eines Gatekeeper-Clusters an, so dass ein weiterer Gatekeeper zur Lastbearbeitung übernommen werden könnte.

Damit alle Zonen in einer VoIP-Umgebung über die Lastzustände der anderen Zonen informiert sind, werden sie mittels Nachrichten zwischen den einzelnen Zonen ausgetauscht. Dabei erfolgt der Austausch nur zwischen jeweils einem ausgezeichneten Cluster-Mitglied der jeweiligen Gatekeeper-Cluster. Der Informationsaustausch kann entweder periodisch oder nach Änderung des Lastzustands erfolgen.

Die Lastverteilung selbst findet nur zwischen Zonen statt, deren Gatekeeper-Cluster bzw. Gatekeeper sich in geeigneten Lastzuständen befinden. In Tabelle 3.1 sind den Lastzuständen des überlasteten und des wenig belasteten Gatekeeper-Clusters bzw. Gatekeepers die entsprechenden Aktionen zur Lastverteilung zugeordnet.

Lastzustand des überlasteten Gatekeeper-Clusters bzw. Gatekeepers	**Lastzustand des wenig belasteten Gatekeeper-Clusters bzw. Gatekeepers**	**Aktionen zur Lastverteilung**
ReceiverGatekeeper	SenderGatekeeper	Gatekeeper-Weiterleitung
ReceiverGatekeeper	ReceiverEndpoint	Endpunkt-Weiterleitung
ReceiverGatekeeper	Ok	keine Aktion
SenderEndpoint	SenderGatekeeper	Endpunkt-Weiterleitung
SenderEndpoint	ReceiverEndpoint	Endpunkt-Weiterleitung
SenderEndpoint	Ok	keine Aktion

Tabelle 3.1: Zuordnung der Interzonen-Lastverteilungsaktionen zu den Lastzuständen der beteiligten Zonen

Wenn ein Endpunkt in eine andere Zone weitergeleitet werden soll, bestimmt die Ursprungszone einen Endpunkt, der gerade nicht aktiv ist. Dieser Endpunkt wird an die Zielzone weitergeleitet, indem die Konfigurationsdaten des Endpunkts an die Zielzone weitergegeben werden. Des Weiteren wird der Endpunkt über die Zugehörigkeit zu einer anderen Zone informiert. Schließlich wird den für die Adressauflösung zuständigen Komponenten der VoIP-Umgebung die neue Zonenzugehörigkeit des Endpunkts mitgeteilt.

Bei der Weiterleitung eines Gatekeepers wird von der Ursprungszone zunächst ein Gatekeeper ausgewählt, der nun keine neuen Anforderungen mehr erhält. Wenn alle Anforderungen, für die dieser Gatekeeper verantwortlich war, vollständig bearbeitet wurden, wird der Gatekeeper der Zielzone zugeordnet. Dazu wird der Ursprungs-Cluster umkonfiguriert und der Gatekeeper erhält die Zonen-spezifischen und Cluster-spezifischen Konfigurationsdaten der Zielzone. Schließlich wird er in den Gatekeeper-Cluster der Zielzone integriert, indem die anderen Cluster-Mitgliedern über den neuen Gatekeeper informiert werden, so dass die Konfiguration innerhalb des Gatekeeper-Clusters entsprechend angepasst werden kann.

Die Weiterleitung eines Gatekeepers sollte nur zwischen Zonen durchgeführt werden, die physikalisch nicht zu weit voneinander entfernt sind, da die Mitglieder eines Gatekeeper-Clusters gemeinsam auf Daten, wie z. B. die Zustandsdaten, zugreifen. Um die Konsistenz dieser Daten zu sichern, wird der Zugriff entsprechend gesteuert. Die dabei entstehenden Verzögerungen können durch räumlich weiter verteilte Cluster-Mitglieder vergrößert werden und damit zu einer Verschlechterung der Leistungsfähigkeit des gesamten Gatekeeper-Clusters führen.

Um eine Ausbreitung der Überlast auf die anderen Zonen zu verhindern, muss die Konfiguration der Lastzustände der Gatekeeper-Cluster entsprechend durchgeführt werden. Daher dürfen die Zustände *SenderGatekeeper* bzw. *ReceiverEndpoint* nur eingenommen werden, wenn genügend Kapazitäten zur Verfügung stehen, so dass die Diensterbringung in der betroffenen Zone auch nach der Durchführung der entsprechenden Lastverteilungsaktion sicher gestellt ist. Dies wird erreicht, indem durch den Lastzustand *Ok* ein Korridor zwischen den Zuständen für Über- und Niedriglast aufgespannt wird, der entsprechend breit gewählt werden muss.

3.5.3 Überlastabwehr

Wenn die Belastung bezüglich der Steuerung so groß ist, dass sie trotz der Maßnahmen der Lastverteilung nicht bewältigt werden kann, werden, wie bereits allgemein in Abschnitt 3.2.3 eingeführt, Überlastabwehrmaßnahmen angewendet. Im Folgenden werden Überlastabwehrmaßnahmen für Gatekeeper in einer H.323-basierten VoIP-Umgebung vorgestellt. Dabei wird in Abschnitt 3.5.3.1 das prinzipielle Vorgehen bei diesen Überlastabwehrmaßnahmen beschrieben, bevor in Abschnitt 3.5.3.2 die einzelnen Verfahren präsentiert werden.

3.5.3.1 Prinzipielles Vorgehen

Die Durchführung der Überlastabwehrmaßnahmen erfolgt in der Regel in den einzelnen Gatekeepern. Darüber hinaus können auch Cluster-bezogene Überlastabwehrmaßnahmen angewendet werden, wobei dies einen erhöhten Verwaltungs- und Steueraufwand für einen Gatekeeper-Cluster zur Folge hat. Dabei ergeben sich nur für einen zentral gesteuerten Cluster Vorteile, da in diesem Fall die Überlastabwehr komplett in die zentrale Komponente ausgelagert werden kann. Jedoch birgt dies wiederum die Gefahr, dass diese zentrale Komponente selbst überlastet wird und damit die Leistungsfähigkeit des Clusters unnötig einschränkt. Bei einer verteilten Steuerung des Clusters muss für eine Cluster-basierte Überlastabwehr weiterhin jedes Cluster-Mitglied über die entsprechenden Überlastabwehrmaßnahmen verfügen. In diesem Fall sollte jedoch sicher gestellt werden, dass Anforderungen, die wegen Überlastung nicht bearbeitet werden können, nicht unnötig an andere Cluster-Mitglieder weitergegeben werden. Bei der Anwendung von kooperierenden Lastverteilungsverfahren ist dies stets der Fall.

Um die Überlastabwehr für H.323-basierte VoIP-Umgebungen möglichst effektiv durchzuführen, so dass die Blindlast minimiert wird, sollte das Ablehnen von Verbindungen bereits nach der ersten Nachricht erfolgen. Daher wird in diesem Fall die RAS-Nachricht für die Verbindungszulassung ARQ mit der entsprechenden ablehnenden RAS-Nachricht ARJ beantwortet. Dabei kann dem rufenden Endgerät die Überlastung als Ursache für die Ablehnung mittels eines entsprechenden Elements der RAS-Nachricht angezeigt werden. Des Weiteren können Anfragen für zusätzliche Dienste sowie zur Änderung der Verbindungsparameter, um z. B.

eine höhere Übertragungskapazität zu erhalten, abgelehnt werden, um zumindest die Basisfunktionalität der Zone zu gewährleisten.

Die im folgenden Abschnitt vorgestellten Überlastabwehrmaßnahmen lehnen abhängig vom aktuell angezeigten Lastzustand des Gatekeepers Verbindungsanforderungen ab. Dies bedeutet, dass bei einem niedrig angezeigten Lastzustand weniger Verbindungsanforderungen abgelehnt werden als bei einem hohen. Damit soll eine stabile Diensterbringung des Gatekeepers in Überlastsituationen erreicht werden.

3.5.3.2 Überlastabwehrmaßnahmen

Im Folgenden werden Überlastabwehrmaßnahmen vorgestellt, die bereits in der Telekommunikation und zumindest auch teilweise in der Datenkommunikation Anwendung finden. Dabei werden Verfahren beschrieben, die für die Überlastabwehr für einen Gatekeeper geeignet erscheinen.

Prozentuale Drosselung

Bei der „Prozentualen Drosselung“ wird ein vorgegebener Prozentsatz der Verbindungsanforderungen abgelehnt. Die Auswahl der abzulehnenden Verbindungsanforderungen erfolgt dabei zufällig. Dazu wird für jede neue Verbindungsanforderung eine gleichverteilte Zufallszahl bestimmt und abhängig von ihrem Wert wird entschieden, ob die Anforderung abgelehnt oder bearbeitet wird. Zur Konfiguration der Überlastabwehrmaßnahme wird jeder Laststufe eine Ablehnungswahrscheinlichkeit zugeordnet.

Automatic Call Gapping

Das „Automatic Call Gapping“ führt die Überlastabwehr durch, indem neue Verbindungsanforderungen nur nach Ablauf eines vorgegebenen Intervalls angenommen werden. Neue Verbindungsanforderungen, die innerhalb dieses Intervalls ankommen, werden abgelehnt. Das Prinzip dieses Verfahrens ist in Bild 3.10 dargestellt. Die Länge des Intervalls ist dabei abhängig vom aktuellen Lastzustand des Gatekeepers, wobei gilt: Je größer die Belastung ist, desto größer müssen die entsprechenden Intervalllängen eingestellt werden.

Leaky Bucket

Beim „Leaky Bucket“ Verfahren werden alle ankommenden Verbindungsanforderungen zwischengespeichert und anschließend mit einer vorgegebenen Rate weitergegeben. Wenn die Anzahl der zwischengespeicherten Verbindungsanforderungen einen Schwellwert überschreitet, werden neu ankommende Verbindungsanforderungen abgelehnt. Die Rate, mit der Verbindungsanforderungen weitergegeben werden, hängt dabei vom jeweiligen Lastzustand des Gatekeepers ab. In Bild 3.11 ist die Funktionsweise des Verfahrens dargestellt.

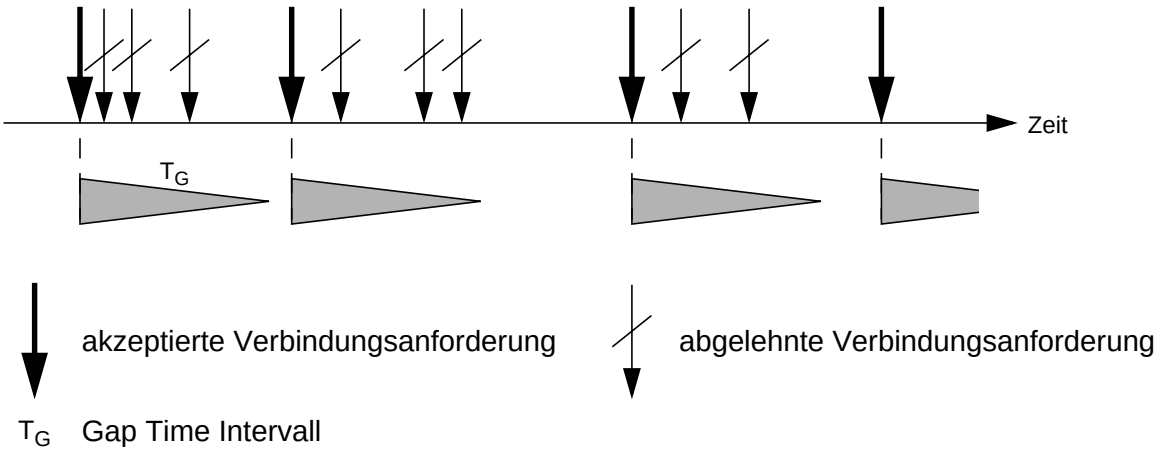


Bild 3.10: Funktionsweise des „Automatic Call Gapping“

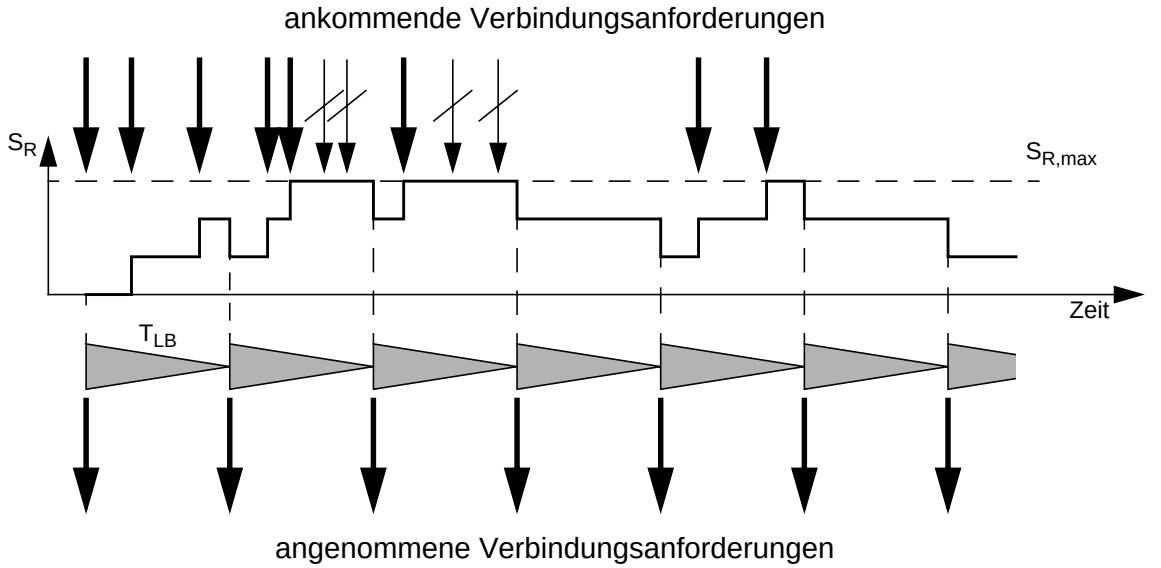
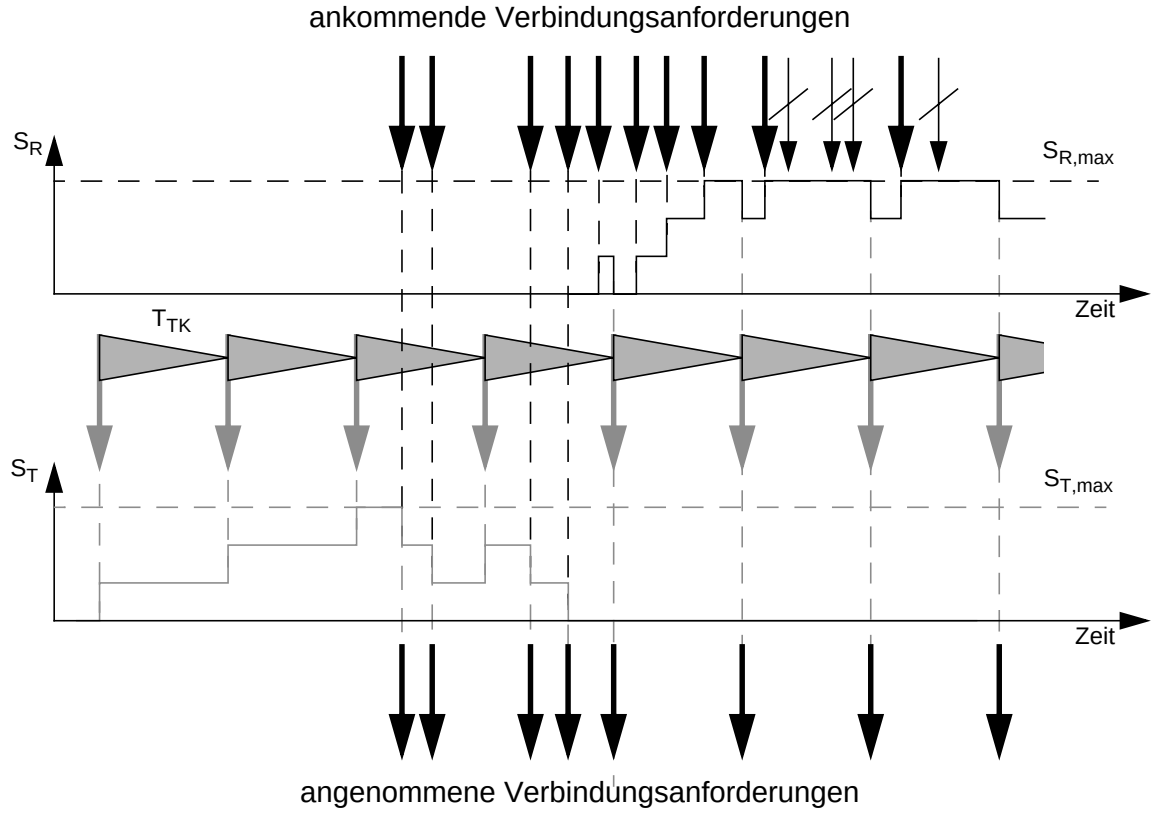


Bild 3.11: Funktionsweise des „Leaky Bucket“ Verfahrens

Token-Pool Leaky Bucket

Eine Erweiterung des „Leaky Bucket“ Verfahrens ist das in [106] vorgestellte Verfahren, das im weiteren Verlauf als „Token-Pool Leaky Bucket“ Verfahren bezeichnet wird. Die Funktionsweise des Verfahrens ist in Bild 3.12 dargestellt. Dabei werden mit einer vom Lastzustand abhängigen Rate Tokens erzeugt, die in einem sog. *Token-Pool* aufbewahrt werden, wobei die Anzahl von Tokens, die sich im Token-Pool befinden können, durch den Wert $S_{T,max}$ begrenzt

ist. Wenn eine Verbindungsanforderung beim Gatekeeper ankommt und es befindet sich ein Token im Token-Pool, wird ein Token aus dem Token-Pool entfernt und die Verbindungsanforderung wird bearbeitet. Ist bei einer ankommenden Verbindungsanforderung der Token-Pool leer, wird die Verbindungsanforderung zwischengespeichert. Wenn jedoch bereits der Maximalwert zwischengespeicherter Verbindungsanforderungen erreicht war, wird die Verbindungsanforderung abgelehnt. Wird ein neues Token erzeugt und es sind Verbindungsanforderungen zwischengespeichert, wird das Token sofort verbraucht, d. h. es wird die nächste der zwischengespeicherten Verbindungsanforderungen verarbeitet und der Token-Pool bleibt leer. Der Vorteil dieses Verfahrens gegenüber dem herkömmlichen „Leaky Bucket“ liegt darin, dass kurzzeitige, büschelförmige Ankünfte von Verbindungsanforderungen ohne weitere Verzögerung bearbeitet werden können, ohne dass langfristig die eingestellte Rate von angenommenen Verbindungsanforderungen überschritten wird.



S_R	Anzahl zwischengespeicherter Verbindungsanforderungen	$S_{R,max}$	Maximale Anzahl zwischengespeicherter Verbindungsanforderungen
↓	ankommendes Token	T_{TK}	Token-Erzeugungs-Zeitintervall
S_T	Token-Anzahl	$S_{T,max}$	Maximale Token-Anzahl

Bild 3.12: Funktionsweise des „Token-Pool Leaky Bucket“ Verfahrens

Fenster-Verfahren

Beim „Fenster“-Verfahren wird nur eine maximale Anzahl von Anforderungen, die noch nicht vollständig bearbeitet wurden, zugelassen. Dazu wird, wie in Bild 3.13 dargestellt, bei jeder ankommenden Anforderung überprüft, ob die aktuelle Fenstergröße größer als Null ist. Ist dies nicht der Fall, wird die Anforderung abgelehnt, ansonsten wird die Anforderung angenommen und die Fenstergröße wird entsprechend angepasst. Wenn eine Anforderung bearbeitet wurde, wird die Fenstergröße inkrementiert. Zur Anpassung an verschiedene Laststufen wird der Wert der maximalen Fenstergröße abhängig vom aktuellen Lastzustand des Gatekeepers eingestellt.

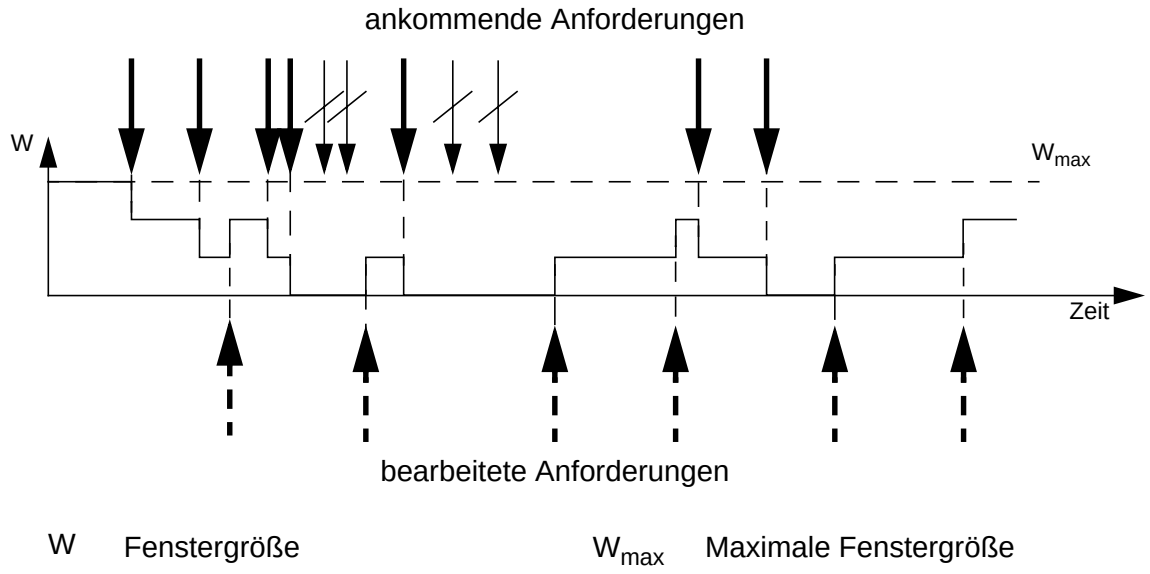


Bild 3.13: Funktionsweise des „Fenster“-Verfahrens

Dieses Verfahren benötigt zur Steuerung der Überlastabwehr die Anzeige, dass eine Anforderung vollständig bearbeitet wurde. Dabei kann eine Anforderung unterschiedlich definiert werden:

- Einzelne Signalisiertransaktion
  In diesem Fall werden einzelne Signalisiertransaktionen als Anforderung betrachtet. Beispielsweise entspricht die RAS-Nachricht ARQ einer Anforderung. Die entsprechende Nachricht, die die Bearbeitung der Anforderung anzeigt, wäre ACF bzw. ARJ. Bei der Signalisierung für die Verbindungssteuerung besteht eine Signalisiertransaktion z. B. aus dem Meldungspaar *Setup* - *Call Proceeding*. Der Nachteil bei dieser Definition ist, dass einzelne Transaktionen, die zusammen einen Verbindungsaufbau darstellen, getrennt betrachtet werden. Dies könnte zur Folge haben, dass z. B. die Zulassung für eine Verbindung erfolgreich abläuft, aber der Aufbau der Signalisierverbindung für die Verbindungssteuerung abgelehnt wird, da die aktuelle Fenstergröße bereits den Wert Null erreicht hat. Dies wäre jedoch nicht effektiv, da bereits Ressourcen für diese Verbindung verbraucht wurden.

- Vollständige Verbindung
  Bei dieser Möglichkeit wird eine vollständige Verbindung als eine einzelne Anforderung betrachtet. Dies bedeutet, dass die maximale Fenstergröße die maximale Anzahl aktiver Verbindungen, die gleichzeitig durch den Gatekeeper verwaltet werden können, festlegt. Ein wesentlicher Nachteil dieser Möglichkeit liegt darin, dass während des Auf- und des Abbaus einer Verbindung wesentlich mehr Ressourcen der Steuerung benötigt werden, als in der Nutzdatenaustauschphase, da diese in der Regel ohne die Beteiligung des Gatekeepers stattfindet. Dies führt entweder zu einer schlechten Ausnutzung der Kapazität des Gatekeepers, wenn bereits viele Signalisierverbindungen aufgebaut sind und somit neue Anforderungen abgelehnt werden, oder zu fehlschlagenden Verbindungsanforderungen, wenn bei wenig aufgebauten Verbindungen viele Verbindungsanforderungen in kurzer Zeit ankommen.

- Verbindungsphase
  Um die Nachteile der beiden oben genannten Möglichkeiten zu minimieren, werden die einzelnen Verbindungsphasen, wie z. B. Verbindungsauf- und -abbau (siehe auch Bild 3.4), jeweils als einzelne Anforderungen betrachtet. Damit werden keine Ressourcen für nur teilweise bearbeitete Anforderungen verschwendet und die Kapazität des Gatekeepers wird effektiver ausgenutzt, als bei der Betrachtung vollständiger Verbindungen. Da Verbindungsabbauanforderungen in der Regel nicht abgelehnt werden sollten, wäre eine sinnvolle Erweiterung, dass diese zwar bei der Bestimmung der aktuellen Fenstergröße miteinbezogen werden, aber nicht abgelehnt werden können. Dabei kann die aktuelle Fenstergröße kleiner als Null werden, so dass erst mehrere Anforderungen bearbeitet sein müssen, bevor eine neue Anforderung angenommen wird. Eine zusätzliche Erweiterung des Verfahrens wäre, dass unterschiedliche Verbindungsphasen unterschiedlich gewichtet werden. Damit würde der unterschiedliche Ressourcenbedarf des Gatekeepers in diesen Verbindungsphasen berücksichtigt werden. Dies wird auch als *Workload Model* bezeichnet, da damit bei der ersten Nachricht einer Anforderung bereits die Ressourcen für die vollständige Bearbeitung der Anforderung berücksichtigt werden.

3.5.4 Realisierungsaspekte

In diesem Abschnitt werden Aspekte vorgestellt, die die Realisierung der optimierten Steuerung im Gatekeeper betreffen. Dazu wird in Abschnitt 3.5.4.1 allgemein auf die Durchführung der Verfahren eingegangen und anschließend werden in Abschnitt 3.5.4.2 Einschränkungen bei der Kombination bestimmter Verfahren beschrieben.

3.5.4.1 Durchführung der Steuerungsoptimierung

Um die Steuerungsoptimierung in einem Gatekeeper durchzuführen, werden wie in Abschnitt 3.2 beschrieben Verfahren zur Lastzustandsermittlung, zur Lastverteilung und zur Überlastabwehr, wie sie in den vorigen Abschnitten vorgestellt wurden, angewendet. Dabei werden geeignete Kombinationen aus diesen Verfahren gebildet.

Damit eine effiziente Steuerungsoptimierung für einen Gatekeeper durchgeführt werden kann, muss zunächst sein aktueller Lastzustand mittels geeigneter Lastindikatoren bestimmt werden. Ist der Gatekeeper Mitglied eines Gatekeeper-Clusters und wird ein kooperierendes Lastverteilungsverfahren innerhalb des Clusters angewandt, werden die Lastzustände aller Cluster-Mitglieder zur Bestimmung der lastaufnehmenden Komponenten verwendet. Wenn eine Anforderung durch ein Lastverteilungsverfahren einem Gatekeeper zur Bearbeitung zugewiesen wurde, oder wenn ein Gatekeeper alleine für die Steuerung einer Zone zuständig ist, wendet der lastaufnehmende Gatekeeper eine entsprechende Überlastabwehrmaßnahme an, um festzustellen, ob die Anforderung abgelehnt werden muss oder bearbeitet werden kann. Wenn eine Anforderung einmal angenommen wurde, soll damit sichergestellt sein, dass sie vollständig und erfolgreich bearbeitet wird.

Bei den Interzonen-Verfahren wird aus den Lastindikatoren der einzelnen Cluster-Mitglieder eines Gatekeeper-Clusters und aus weiteren Messungen ein gemeinsamer Lastzustand für den ganzen Cluster bestimmt. Dieser wird zwischen den einzelnen Zonen einer VoIP-Umgebung ausgetauscht, damit die für eine kooperierende Interzonen-Lastverteilung notwendigen Informationen zur Verfügung stehen. Die Steuerung dieser Interzonen-Lastverteilung erfolgt jeweils durch ein ausgezeichnetes Mitglied eines Gatekeeper-Clusters.

Entscheidungsmethoden

Wenn eine Verbindungsanforderung in einem Cluster ankommt, muss entschieden werden, wie mit dieser Anforderung verfahren wird, d. h. ob und an welches Cluster-Mitglied die Anforderung weitergegeben wird. Bei einem Cluster mit zentral gesteuerter Lastverteilung gibt der Dispatcher die Anforderung an das entsprechende Cluster-Mitglied weiter, wobei bei einem kooperierenden Lastverteilungsverfahren der am geringsten belastete Gatekeeper ausgewählt wird.

Bei einem Cluster mit verteilter Steuerung der Lastverteilung muss jedes Cluster-Mitglied eigenständig entscheiden, ob die Anforderung lokal bearbeitet wird oder an ein anderes Cluster-Mitglied weitergeleitet wird. Da die Weiterleitung selbst Ressourcen benötigt, muss festgestellt werden, ob sich dieser Aufwand lohnt. Dies kann beispielsweise durch das Lastverteilungsverfahren selbst durchgeführt werden. So zeigt beim „Sender-Receiver“-Verfahren ein Gatekeeper durch seinen Zustand an, ob er eine weitere Anforderung bearbeiten kann.

Eine weitere Möglichkeit ist das Treffen der Entscheidung mittels der Ermittlung der dafür anfallenden Kosten. Dazu können beispielsweise die in den Gleichungen (3.10) und (3.11) definierten Kostenfunktionen angewandt werden.

$$C_{local} = (1 - p_{local,\,fail})C_{local,\,success} + p_{local,\,fail}C_{local,\,fail} \quad (3.10)$$

$$C_{intrazone} = (1 - p_{intrazone,\,fail})C_{intrazone,\,success} + p_{intrazone,\,fail}C_{intrazone,\,fail} \quad (3.11)$$

Dabei bezeichnen C_{local} und $C_{intrazone}$ die entstehenden Kosten, je nachdem ob die Verbindungsanforderung lokal bearbeitet oder zu einem anderen Gatekeeper weitergeleitet wird. Zur Bestimmung der Kosten wird für jede Aktion die Wahrscheinlichkeit, dass sie erfolgreich verläuft, ermittelt. Diese Wahrscheinlichkeiten sind dabei abhängig vom Lastzustand des lokalen Gatekeepers ($p_{local,\,fail}$) oder von den Lastzuständen der anderen Gatekeeper des Clusters ($p_{intrazone,\,fail}$). Des Weiteren werden die Kosten sowohl für das erfolgreiche Ablaufen[1] ($C_{local,\,success}$, $C_{intrazone,\,success}$) als auch für das Fehlschlagen ($C_{local,\,fail}$, $C_{intrazone,\,fail}$) der jeweiligen Aktion mit diesen Wahrscheinlichkeiten in Relation gesetzt. Schließlich wird die Aktion durchgeführt, die die niedrigsten Kosten verursacht. Dieses Verfahren kann auch für Anforderungen mit unterschiedlichem Ressourcenverbrauch bezüglich der Steuerressourcen verwendet werden, indem diese Anforderungen abhängig von ihrem Ressourcenverbrauch in Klassen unterteilt werden und die Kosten jeweils klassen-individuell festgelegt werden.

Reihenfolge der Ressourcenbewertung

Damit eine Verbindungsanforderung vollständig bearbeitet werden kann, werden verschiedene Ressourcen benötigt. Dies sind zum einen Steuerressourcen, um die notwendige Steuerung der Verbindung durchzuführen, und zum anderen Nutzdaten-Ressourcen, wie z. B. spezielle Komponenten oder Übertragungskapazitäten vom A- zum B-Teilnehmer.

Bei der Bewertung der Verfügbarkeit dieser Ressourcen wird in der Regel sequentiell vorgegangen, d. h. es wird für jede Ressource, die für die Verbindung benötigt wird, überprüft, ob sie zur Verfügung steht. Wenn eine Ressource nicht verfügbar ist, wird der Vorgang abgebrochen und die Verbindung wird abgelehnt, ohne dass weitere Ressourcen überprüft werden. Dabei existieren prinzipiell zwei Vorgehensweisen, die sich in der Reihenfolge der Ressourcenbewertung unterscheiden: Die „Steuerressourcen-optimierte“ Bewertung überprüft zunächst, ob genügend Steuerressourcen zur Verfügung stehen, bevor die Bewertung der Ressourcen für die Nutzdaten erfolgt. Insbesondere wird dabei die Überlastabwehr durchgeführt, bevor festgestellt, ob z. B. noch genügend Übertragungskapazitäten für diese Verbindung vorhanden wären. Bei der „Nutzdaten-Ressourcen-optimierten“ Bewertung werden dagegen die Ressourcen bezüglich der Nutzdaten vor denen der Steuerung überprüft.

[1] Ein Gewinn entspricht dabei negativen Kosten.

Der Vorteil der „Steuerressourcen-optimierten“ Bewertung liegt darin, dass die Verfahren, die u. a. auch den Gatekeeper vor Überlastung schützen und die Stabilität seiner Funktionsweise gewährleisten sollen, vor denen im Verhältnis aufwendigeren Verfahren zur Bewertung der Ressourcen für die Nutzdaten, z. B. zur Überprüfung der vorhandenen Übertragungskapazitäten, durchgeführt werden. Damit bleibt der Gatekeeper über einen größeren Lastbereich hinweg stabil. Jedoch können Steuerressourcen zur Verbindungsbearbeitung dadurch verschwendet werden, dass Verbindungen angenommen werden, für die nicht genügend Ressourcen bezüglich der Nutzdaten zur Verfügung stehen.

Wird die Bewertung der Ressourcen bezüglich der Nutzdaten optimiert, wird dagegen zuerst überprüft, ob die für die Nutzdaten benötigten Ressourcen zur Verfügung stehen. Anschließend wird nur für die Verbindungen, für die diese Ressourcen noch vorhanden sind, die Bewertung der Steuerressourcen durchgeführt. Damit werden im Prinzip nur Verbindungsanforderungen angenommen, die sowohl bezüglich der Steuer- als auch der Nutzdaten-Ressourcen erfolgreich ablaufen.

Die Entscheidung, welche der beiden Vorgehensweisen verwendet werden sollte, hängt von den Zielen der Realisierung ab: Wenn die Stabilität des Gatekeepers bis zu möglichst hoher Steuerlast gewährleistet sein soll, ist die „Steuerressourcen-optimierte“ Vorgehensweise besser geeignet. Ansonsten kann die „Nutzdaten-Ressourcen-optimierte“ Methode verwendet werden.

3.5.4.2 Einschränkungen

Je nach Granularität des Lastverteilungsverfahrens kann es vorkommen, dass einzelne Gatekeeper eines Gatekeeper-Clusters nur Teile einer Verbindung steuern. Da jedoch sowohl Lastindikatoren als auch Überlastabwehrmaßnahmen existieren, die darauf basieren, dass zumindest eine Verbindungsphase, wenn nicht sogar die ganze Verbindung, durch den gleichen Gatekeeper bearbeitet wird, bestehen Einschränkungen für die Verwendung dieser Verfahren bzw. für die Granularität der Lastverteilung.

So wird beispielsweise beim Lastindikator „Gewichtete Verbindungszustände“ der Lastzustand des Gatekeepers aus den Zuständen der Verbindungen, die über diesen Gatekeeper geführt werden, abgeleitet. Das Verfahren basiert auf der Annahme, dass aus diesen Verbindungszuständen der Ressourcenbedarf für die weitere Bearbeitung der Verbindungen geschätzt werden kann. Wenn jedoch die Nachrichten für diese Verbindungen von verschiedenen Gatekeepern behandelt werden, kann daraus nicht die Belastung eines einzelnen Gatekeepers ermittelt werden, da der Ressourcenbedarf für die Verbindungsbearbeitung zwischen den Gatekeepern aufgeteilt ist. Daher kann dieser Lastindikator nur verwendet werden, wenn die Granularität der Lastverteilung die Verbindungs-Ebene ist. Wenn den Verbindungszuständen der

Nutzdatenaustauschphase das Gewicht Null zugeordnet wird, kann er darüber hinaus auch bei einer Lastverteilung auf der Verbindungsphasen-Ebene angewandt werden.

In Tabelle 3.2 sind die einzelnen Lastindikatoren und ihre zulässigen Granularitäten der Lastverteilung enthalten.

Lastindikator	**zulässige Granularität der Lastverteilung**
Warteschlangenlänge	alle (Endpunkt-, Verbindungs-, Verbindungsphasen- und Nachrichten-Ebene)
Gradient der Warteschlangenlänge	alle
Rufankunftsrate	Endpunkt- und Verbindungs-Ebene, evtl. Verbindungsphasen-Ebene
Prozessorauslastung	alle
Anzahl offener Anfragen	Endpunkt-, Verbindungs- und Verbindungsphasen-Ebene
Gewichtete Verbindungszustände	Endpunkt- und Verbindungsebene, evtl. Verbindungsphasen-Ebene

Tabelle 3.2: Lastindikatoren und zulässige Granularitäten der Lastverteilung

Bei den Überlastabwehrmaßnahmen ist nur das „Fenster“-Verfahren von diesen Einschränkungen betroffen, während die anderen der vorgestellten Überlastabwehrmaßnahmen mit allen Granularitäten der Lastverteilung verwendet werden können. Da beim „Fenster“-Verfahren die Anzeige, dass eine Anforderung bearbeitet wurde, benötigt wird, um die aktuelle Fenstergröße anzupassen, muss zumindest diese Anforderung vollständig durch diesen Gatekeeper bearbeitet worden sein. Daher hängt die kleinste zulässige Granularität der Lastverteilung von der Definition einer Anforderung des „Fenster“-Verfahrens ab. So darf z. B. die Granularität der Lastverteilung die Verbindungsphasen-Ebene nicht unterschreiten, wenn eine Anforderung des „Fenster“-Verfahrens einer Verbindungsphase entspricht.

3.6 Steuerungsoptimierung für integriert verwaltetes Unternehmensnetz

Bisher wurde nur die optimierte Steuerung zur effizienten Verwendung der Ressourcen einer VoIP-Umgebung betrachtet. Wie aber in Abschnitt 2.1.3 beschrieben ist, soll durch die Konvergenz der Netze eine Vielzahl unterschiedlicher Dienste durch eine einheitliche Netzinfrastruktur unterstützt werden. Neben VoIP-Diensten sind dabei die folgenden Dienste zu nennen:

- Audio- und Video-Abrufdienste, wie z. B. *Video-on-Demand* (VoD)
- Interaktive Datenanwendungen, wie z. B. WWW und Netzspiele

- Datei-Transfer, wie z. B. E-Mail und Fax

Diese Dienste haben jeweils unterschiedliche Anforderungen bezüglich der Dienstgüte. So benötigen beispielsweise die Audio- und Video-Abrufdienste in der Regel eine Übertragungskapazität, deren Schwankungen nicht zu groß sein darf, während beim Datei-Transfer die Verzögerungen kaum Bedeutung haben, jedoch die Fehlerwahrscheinlichkeit äußerst gering sein muss.

Damit diese Dienste in einer integrierten Umgebung, d. h. gemeinsam mit VoIP-Diensten, unter Berücksichtigung der jeweils notwendigen Dienstgüte verwendet werden können, sollte eine gemeinsame, übergeordnete Steuerung der Ressourcen für diese Dienste durchgeführt werden. Dazu könnte in einer begrenzten Umgebung, wie es z. B. ein Unternehmensnetz darstellt, der Gatekeeper einer VoIP-Umgebung in seiner Funktionalität erweitert werden, so dass er auch die Ressourcenverwaltung dieser Dienste übernehmen kann. Damit der Gatekeeper diese Funktion wahrnehmen kann, müssen für die angeforderten Dienste jeweils die dafür benötigten Ressourcen beim Gatekeeper beantragt und nach der Diensterbringung wieder zurück gegeben werden, was z. B. mit der RAS-Signalisierung erfolgen kann. Dadurch erhält der Gatekeeper alle Informationen, die er für die Entscheidung, ob genügend Ressourcen für die Diensterbringung zur Verfügung stehen, benötigt.

Um die zur Verfügung stehenden Ressourcen für die einzelnen Dienste möglichst effektiv zu nutzen, kann der Gatekeeper die in Abschnitt 3.4 beschriebenen Verfahren für diese Dienste in adaptierter Form anwenden. Dazu muss zunächst mittels geeigneter Lastindikatoren die Belastung der einzelnen für die Diensterbringung notwendigen Komponenten ermittelt werden. Wenn mehrere Komponenten für die Diensterbringung zur Verfügung stehen, kann die Last entsprechend verteilt werden, so dass diese Komponenten gleichmäßig ausgelastet werden. Schließlich kann der Gatekeeper Überlastabwehrmaßnahmen für die einzelnen Komponenten durchführen, so dass ihre Ressourcen effizient ausgenutzt werden.

Durch diese zusätzlichen Aufgaben eines Gatekeepers, deren Bewältigung durch entsprechende Priorisierungen von Diensten und Benutzern noch komplexer werden kann, ist es um so mehr notwendig, dass er die Steuerung stabil und effektiv auch in Hoch- und Überlastsituationen durchführt. Daher nimmt die Bedeutung von Verfahren für seine optimierte Steuerung, wie sie in Abschnitt 3.5 vorgestellt wurden, weiter zu.

Kapitel 4

Untersuchungsmethoden

Für die Leistungsuntersuchung von Kommunikationssystemen existieren verschiedene Methoden, deren Anwendbarkeit von verschiedenen Faktoren abhängt. Nach [71] wird in die im Folgenden beschriebenen Methoden unterschieden.

Eine mögliche Untersuchungsmethode stellt die Messung an realen Systemen dar. Diese Methode setzt jedoch voraus, das ein derartiges System existiert und verfügbar ist.

Des Weiteren können Modelle der zu untersuchenden Systeme entwickelt werden. Die Betrachtung der Modelle erlaubt auch die Untersuchung von Systemen, die (noch) nicht existieren bzw. nicht verfügbar sind.

Bei den Modellen wird zwischen *physikalischen* und *mathematischen* Modellen unterschieden:

- Physikalische Modelle werden in einer realen Umgebung implementiert, wobei sich ihre Funktionalität auf die wesentlichen Merkmale, die untersucht werden sollen, beschränkt. Die Untersuchung eines physikalischen Modells erfolgt wie bei bestehenden Systemen mittels Messungen. Realisierungen physikalischer Modelle werden als *prototypische Implementierungen* bezeichnet, die Untersuchungen erfolgen in der Regel in einem *Testbett*. Häufig werden prototypische Implementierungen für den Beweis der Machbarkeit bestimmter Funktionen verwendet, wobei auch Untersuchungen zur Leistungsfähigkeit vorgenommen werden können.
- Mathematische Modelle der zu untersuchenden Kommunikationssysteme werden entweder mittels Simulationen oder mathematischer Verfahren analysiert. Bei der Modellierung wird ein Abbild des Systems entworfen, das nur über die relevanten Funktionen und Eigenschaften verfügt, die zur Ermittlung seiner Leistungsfähigkeit für den zu untersuchenden Bereich notwendig sind. Dabei hängt das Abstraktionsniveau des Modells u. a. vom Untersuchungsziel ab, da zur Erzeugung und Bewertung bestimmter Effekte ein entsprechender Detaillierungsgrad des Modells erforderlich sein kann.

In diesem Kapitel werden die Methoden für die Untersuchung der in Abschnitt 3.5 vorgestellten Verfahren zur Steuerungsoptimierung für Gatekeeper von H.323-basierten VoIP-Umgebungen beschrieben. Für die Untersuchung von Überlastabwehrmaßnahmen für existierende Gatekeeper-Realisierungen wird eine Form der prototypischen Implementierung verwendet, die in Abschnitt 4.1 vorgestellt wird. Die Untersuchung der Verfahren der Intrazonen- und Interzonen-Lastverteilung sowie ihr Zusammenwirken mit Überlastabwehrmaßnahmen erfolgt mittels Simulation eines entsprechenden Modells. Diese Methode wird in Abschnitt 4.2 vorgestellt.

Die Gründe für die Verwendung dieser Methoden liegen darin, dass zum einen kein reales System zur Verfügung steht, bei dem die Verfahren zur Steuerungsoptimierung entsprechend implementiert sind. Daher können keine Messungen an einem realen System durchgeführt werden, wobei bei dem in Abschnitt 4.1 vorgestellten Verfahren Überlastabwehrmaßnahmen gemeinsam mit realen Systemen untersucht werden können. Zum anderen wird für diese Untersuchungen ein hoher Detaillierungsgrad bezüglich des Ablaufs der Signalisierung benötigt, um beispielsweise die Problematik der Granularität der Lastverteilung aufzuzeigen. Darüber hinaus sollte auch das dynamische Verhalten der Verfahren betrachtet werden, so dass neben dem stationären auch der instationäre Fall untersucht wird.

4.1 Prototypische Implementierung und Messung

Um verschiedene Überlastabwehrmaßnahmen und ihre Eigenschaften zusammen mit existierenden Gatekeeper-Realisierungen zu untersuchen, erfolgt eine prototypische Implementierung dieser Verfahren. Das Prinzip für dieses Vorgehen wird in Abschnitt 4.1.1 beschrieben. Die Untersuchung dieser prototypischen Implementierung wird mittels entsprechender Messungen in einem geeigneten Testbett durchgeführt. Die Elemente des Testbetts werden in Abschnitt 4.1.2 vorgestellt.

4.1.1 Prinzip

Die Implementierung der Überlastabwehrmaßnahmen erfolgt auf dem sog. *PreServer*. Der PreServer befindet sich zwischen dem IP-Netz und dem Gatekeeper, so dass der komplette Signalisierverkehr vom und zum Gatekeeper über ihn geführt wird. Damit kann der PreServer die Überlastabwehr für den Gatekeeper durchführen, ohne dass Änderungen an der bestehenden Gatekeeper-Realisierung notwendig sind. Dieser Ansatz hat gegenüber der Implementierung der Überlastabwehr im Gatekeeper selbst den Vorteil, dass der PreServer unabhängig vom Gatekeeper realisiert ist. Somit kann der PreServer für die Untersuchung der Auswirkungen einer Überlastabwehr für unterschiedliche Gatekeeper-Realisierungen verwendet werden. Des Weiteren muss bei der Realisierung kein Wissen über die Implementierung des Gatekeepers vorhanden sein, was den Aufwand bei der PreServer-Realisierung wesentlich reduziert.

Zur Untersuchung der Überlastabwehrmaßnahmen muss eine entsprechend große Signalisierlast für den Gatekeeper erzeugt werden. Dabei kommen in der Regel Lastgeneratoren zum Einsatz, die eine Definition der Charakteristika der erzeugten Last erlauben. Zur Ermittlung der Eigenschaften, die sich aus dem Zusammenwirken der Überlastabwehrmaßnahmen und dem Gatekeeper ergeben, werden Messwerte aufgenommen und entsprechend ausgewertet. Diese Messungen werden in diesem Fall ebenfalls im PreServer durchgeführt.

4.1.2 Testbett

Die im vorigen Abschnitt erwähnten und in Bild 4.1 dargestellten Komponenten des verwendeten Testbetts werden in den folgenden Abschnitten vorgestellt.

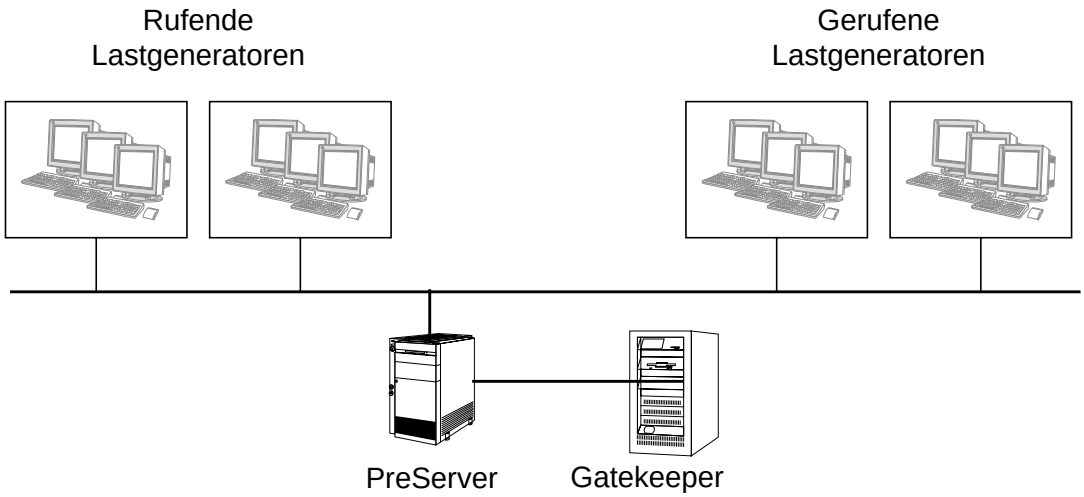


Bild 4.1: Komponenten des Testbetts

4.1.2.1 PreServer

Bevor im Folgenden der Funktionsumfang des PreServers vorgestellt wird, wird zunächst auf seine Realisierung eingegangen.

Realisierung

Der PreServer bearbeitet die Signalisiernachrichten in transparenter Form, d. h. er ist weder für die Endpunkte noch für den Gatekeeper sichtbar, da er die über ihn geführten Verbindungen nicht terminiert, sondern die einzelnen Signalisierpakete in geeigneter Form weiterleitet. Durch diese Funktionsweise kann eine hohe Leistungsfähigkeit des PreServers erreicht werden, da nur die für die Überlastabwehr und die Messungen notwendigen Daten bearbeitet werden. Jedoch sind dadurch die Eingriffsmöglichkeiten in H.323-Signalisierverbindungen für die Durchführung von Überlastabwehrmaßnahmen etwas eingeschränkt: Um eine Verbindung abzulehnen, muss der PreServer eine entsprechende ablehnende Signalisiernachricht an den Sender der Anfrage senden und anschließend die Anfrage verwerfen. Der Gatekeeper wird

dabei nicht miteinbezogen, d. h. er erhält weder die Anfrage noch erfährt er von der Ablehnung durch den PreServer. Da TCP Reihenfolgenummern zur gesicherten Übertragung der Datenpakete verwendet, müsste der PreServer, um eine ablehnende Signalisiernachricht innerhalb einer TCP-Verbindung zu erzeugen, die entsprechenden Reihenfolgenummern verwalten und diese gegebenenfalls bei weiterzuleitenden Datenpaketen und auch bei TCP-Bestätigungspaketen ersetzen. Daher wird innerhalb des PreServers darauf verzichtet, ablehnende Nachrichten in TCP-Verbindungen zu erzeugen. Da die initialen Verbindungsanfragen der RAS-Signalisierung mittels UDP übertragen werden, stellt das Verwerfen und Erzeugen von Nachrichten durch den PreServer jedoch kein Problem dar.

Der PreServer wird mittels eines handelsüblichen PCs mit zwei Schnittstellenkarten zum IP-Netz realisiert. Als Betriebssystem wird Linux verwendet und die Implementierung erfolgt in C++. Die Software-Struktur des PreServers besteht aus den folgenden, in Bild 4.2 dargestellten Modulen:

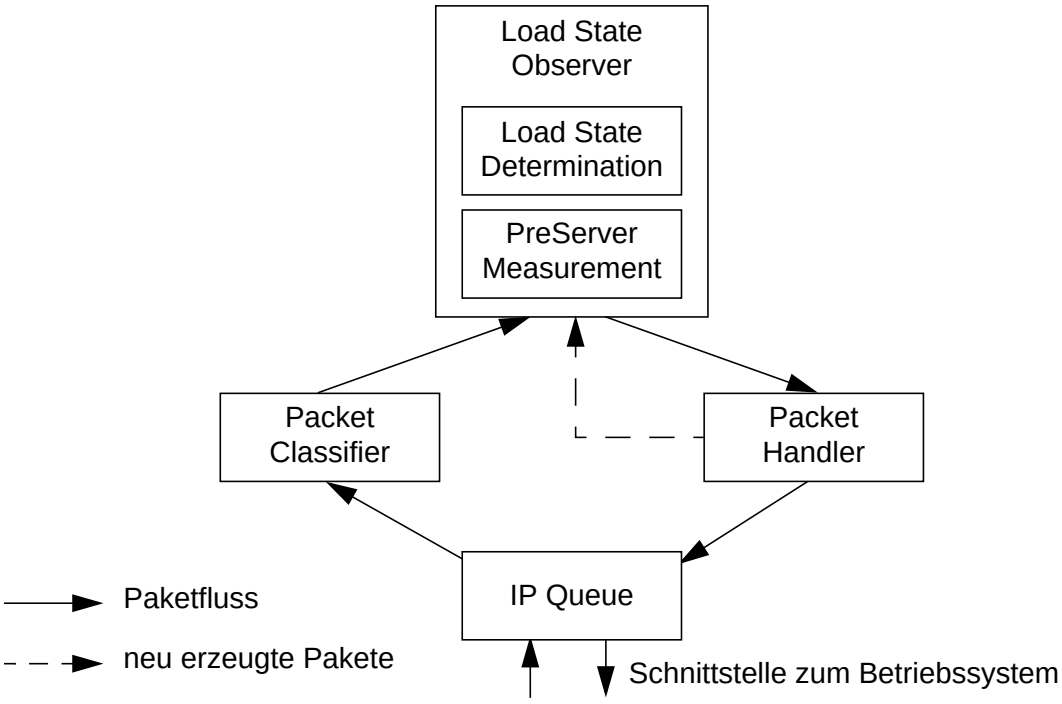


Bild 4.2: Software-Struktur des PreServer

- *IP Queue*
  Dieses Modul enthält die Schnittstelle zum Betriebssystem und liest die ankommenden Pakete aus der entsprechenden Eingangswarteschlange und gibt sie an das Modul *Packet Classifier* weiter. Des Weiteren werden vom Modul *Packet Handler* übergebene Pakete wieder dem Betriebssystem zur Weiterleitung weitergegeben.

- *Packet Classifier*
  Die erhaltenen Pakete werden in diesem Modul klassifiziert. Dabei wird die Decodierung

der H.323-Nachrichten durchgeführt, so dass diese entsprechend in den anderen Modulen bearbeitet werden können. Die klassifizierten Pakete werden anschließend an das Modul *Load State Observer* weitergegeben.

- *Load State Observer*
  Dieses Modul bestimmt aus den Lastindikatoren den aktuellen Lastzustands des Gatekeepers (Submodul *Load State Determination*) und führt die Messungen des PreServers durch (Submodul *PreServer Measurement*). Dazu müssen abhängig von den verwendeten Indikatoren unterschiedliche Informationen gehalten werden. Anschließend werden die Pakete und der aktuelle Lastzustand an das Modul *Packet Handler* weitergegeben.

- *Packet-Handler*
  Die Implementierung der Überlastabwehrmaßnahmen selbst erfolgt in diesem Modul. Abhängig vom Lastzustand werden die Pakete entsprechend der zu untersuchenden Überlastabwehrmaßnahme behandelt. Pakete, die an die entsprechenden Komponenten weitergegeben werden sollen, werden dem Modul *IP Queue* übergeben. Des Weiteren können in diesem Modul Pakete neu erzeugt werden, wenn z. B. eine Ablehnung einer Anfrage durchgeführt wird. Neu erzeugte Pakete werden zunächst dem Modul *Load State Observer* übergeben, so dass die Messwerte und die Lastindikatoren entsprechend aktualisiert werden. Anschließend werden sie ebenfalls dem Modul *IP Queue* zur Weiterleitung übergeben.

Funktionsumfang

Wie bereits erwähnt bearbeitet der PreServer alle RAS- und Verbindungssteuerungssignalisiernachrichten, die zwischen den Endpunkten und dem Server ausgetauscht werden. Diese Bearbeitung lässt sich in folgende Bereiche unterteilen:

- Bestimmung des Lastzustands des Gatekeepers
  Da der PreServer keinen Zugriff auf interne Zustandsdaten des Gatekeepers hat, werden Messungen innerhalb des PreServers durchgeführt, die eine Ableitung des Lastzustands des Gatekeepers erlauben. Dazu wird die Antwortverzögerung für die einzelnen Signalisiernachrichten bestimmt und anschließend mittels des exponentiellen Glättens gefiltert.

- Durchführung der Überlastabwehrmaßnahmen
  Zur Untersuchung der Wirksamkeit verschiedener Überlastabwehrmaßnahmen wurden „Prozentuale Drosselung“, „Automatic Call Gapping“, „Leaky Bucket“ sowie das „Fenster“-Verfahren, so wie sie jeweils in Abschnitt 3.5.3.2 beschrieben wurden, implementiert. Wenn eine Verbindungsanforderung abgelehnt werden soll, erzeugt der PreServer die entsprechende Nachricht. Der Gatekeeper selbst ist davon nicht betroffen. Ansonsten werden alle Nachrichten durch den PreServer weitergeleitet, so dass die Verbindungsbearbeitung weiterhin durch den Gatekeeper durchgeführt wird.

- Durchführung von Messungen
  Neben den Aufgaben für die Überlastabwehr führt der PreServer zusätzlich kontinuierlich Messungen durch und wertet die Messwerte statistisch aus. Dabei werden zum einen auftretende Ereignisse gezählt. Beispiele für diese Messwerte sind die Anzahl der Verbindungsanforderungen, die Anzahl der fehlgeschlagenen Verbindungsanforderungen oder die Anzahl der Wiederholungen von Verbindungsanforderungen. Zum anderen werden Zeitmessungen vorgenommen, um u. a. die Verzögerungen für die Nachrichtenbearbeitung durch den Gatekeeper zu bestimmen. Zu diesen Messwerten zählen die ARQ-Antwortverzögerung oder die Zeitdauer vom Empfang der *Setup*-Nachricht des Rufenden bis zum Senden der entsprechenden *Call Proceeding*-Nachricht.

4.1.2.2 Lastgenerator

Da zur Erzeugung einer genügend großen Last für die Untersuchung der Überlastabwehrmaßnahmen und des Verhaltens des Gatekeepers einzelne Endpunkte, die in einer Laborumgebung zur Verfügung stehen, in der Regel nicht ausreichen, werden spezialisierte Komponenten zur Lasterzeugung verwendet. Dabei existieren unterschiedliche Realisierungen von Lastgeneratoren.

Die einfachsten Realisierungen vereinen mehrere Endpunkte in einer Komponente, die gleichzeitig eine bestimmte Anzahl von Verbindungen aufbauen können, wobei sich die Konfiguration auf die Einstellung der Zeitdauer bis zur nächsten Verbindungsanforderung (*Idle-Duration*) und der Verbindungsdauer (*Connection-Duration*) beschränkt. Komplexere Realisierungen, wie sie z. B. in [88, 110] für eine ISDN-PBX vorgestellt werden, erlauben dagegen eine detaillierte Definition sowohl des funktionalen Verhaltens, um z. B. wiederholte Anrufversuche bei fehlschlagenden Verbindungsanforderungen (*Repeated Call Attempts*) zu realisieren, als auch des zeitlichen Verhaltens, so dass nicht nur Verbindungsdauern sondern auch z. B. die Zeitdauern, die auf Verbindungsannahmen gewartet werden, vorgegebenen statistischen Verteilungen gehorchen.

Der Lastgenerator für die in dieser Arbeit vorgestellten Untersuchungen ist der *H.323 Call Generator* der Firma *RADCOM*. Dieser erlaubt die Einstellung der Zeitdauer bis zur nächsten Verbindungsanforderung und der Verbindungsdauer. Da die Rate der Verbindungsanforderungen über die Anzahl der simulierten Verbindungen eingestellt wird, sinkt bei hoher Belastung des Gatekeepers die tatsächlich durch den Lastgenerator erzeugte Rate der Verbindungsanforderungen. Dies liegt daran, dass die Zeiten für den Verbindungsauf- und -abbau ansteigen und somit die gesamte Zeit, die für eine Verbindung benötigt wird, ebenfalls ansteigt. Bei fester Anzahl gleichzeitig aktiver Verbindungen ergibt sich somit eine geringere Rate von Verbindungsanforderungen. Da die Anzahl gleichzeitig simulierter Verbindungen begrenzt ist, ist auch die maximale Größe der generierten Last beschränkt.

4.1.2.3 Gatekeeper

Mit den bereits vorgestellten Komponenten des Testbetts kann jede H.323-konforme Gatekeeper-Realisierung bezüglich ihres Überlastverhaltens untersucht werden. Für die in dieser Arbeit vorgestellten Untersuchungen stand das System *SURPASS hiQ 20* der Firma *Siemens* zur Verfügung, das im wesentlichen die Aufgaben eines Gatekeepers in der *SURPASS*-Architektur übernimmt: Autorisierung und Registrierung von Benutzern und Adressauflösung. Des Weiteren kann es als sog. *H.323-Proxy* verwendet werden, d. h. dass es die Schnittstelle zwischen öffentlichen und dem privaten, entsprechend geschützten Teil eines IP-Netzes darstellt, so dass auch die Nutzdaten einer H.323-Verbindung über das System geführt werden. Das untersuchte System ist auf einer *Workstation* der Firma *SUN* vom Typ *Netra T1* realisiert.

4.2 Simulation

In diesem Abschnitt wird die Untersuchung der in Abschnitt 3.5 vorgestellten Verfahren mittels der Simulation eines entsprechenden Modells vorgestellt. Dazu wird in Abschnitt 4.2.1 die dabei angewandte Methode der zeitdiskreten, ereignisgesteuerten Simulation beschrieben. Anschließend wird in Abschnitt 4.2.2 das entwickelte Simulationsmodell vorgestellt. Schließlich wird in Abschnitt 4.2.3 auf das für die Untersuchungen verwendete Simulationswerkzeug eingegangen.

4.2.1 Zeitdiskrete, ereignisgesteuerte Simulation

Das in dieser Arbeit untersuchte System kann als *diskretes* System betrachtet werden, da es über eine endliche Anzahl von Zuständen verfügt und Zustandsänderungen jeweils durch Ereignisse, wie z. B. Nachrichten oder abgelaufene Timer, hervorgerufen werden. Für simulative Untersuchungen derartiger Systeme wird in der Regel die *zeitdiskrete*, *ereignisgesteuerte* Simulation angewandt.

Bei diesem Verfahren verfügt jedes Ereignis über einen Zeitstempel, der den Zeitpunkt seines Auftretens enthält. Die Ereignisse werden nach ihrem Zeitstempel geordnet in eine Liste (*Kalender*) eingetragen. Während der Simulation werden sie sukzessive aus dem Kalender ausgelesen und durch das simulierte System bearbeitet, wobei wieder neue Ereignisse entstehen können, die entsprechend in den Kalender eingetragen werden. Durch dieses Prinzip wird das System nur zu den Zeitpunkten betrachtet, an denen Ereignisse auftreten und sich somit sein Zustand ändert.

Ein Großteil der Zeitdauern, bis zum Auftreten eines bestimmten Ereignisses, wie z.B. der Zeitpunkt bis zum Ende der Bearbeitung einer Verbindungsanforderungsnachricht, ist nicht

konstant, sondern gehorcht entsprechenden statistischen Verteilungen. Zur Erzeugung von Werten nach diesen Verteilungen wird ein geeigneter Zufallszahlengenerator verwendet.

Um eine Simulation geeignet auswerten zu können, werden während der Simulation Messwerte aufgezeichnet und mit statistischen Verfahren behandelt. Messwerte können beispielsweise die Aufenthaltsdauer von Nachrichten in einer Komponente oder die Anzahl von bestimmten Ereignissen, wie z. B. Verbindungsanforderungen, sein. Da nur eine begrenzte Anzahl von Messungen für diese Messwerte vorgenommen werden können, handelt es sich bei diesen Werten stets um Schätzwerte, wobei mittels geeigneter Statistikmethoden eine Bewertung bezüglich ihrer Aussagesicherheit vorgenommen werden kann.

Eine detaillierte Beschreibung der Methode der zeitdiskreten, ereignisgesteuerten Simulation ist z. B. in [71] enthalten.

4.2.1.1 Stationäre Simulation

Bei der stationären Simulation wird ein eingeschwungenes System betrachtet, d. h. dass das beobachtete Systemverhalten unabhängig gegenüber Zeitverschiebungen ist. Dabei wird zunächst eine sog. Warmlaufphase durchgeführt, in der das System in den eingeschwungenen Zustand gebracht werden soll, bevor die Messungen zur Ermittlung seines stationären Verhaltens vorgenommen werden. Um die Aussagesicherheit der Messwerte bestimmen zu können, müssen die Simulationen mehrmals, unabhängig voneinander bei konstant bleibenden Systemparametern durchgeführt werden. Damit die Warmlaufphase aber nicht jedes Mal durchlaufen werden muss, werden in einem Simulationslauf nach der Warmlaufphase mehrere Teiltests durchgeführt. Für diese Teiltests werden die einzelnen Messwerte jeweils statistisch ausgewertet. Mit diesen Teiltestergebnissen wird dann die statistische Aussagesicherheit mittels Vertrauensintervallen bestimmt.

4.2.1.2 Instationäre Simulation

Im Gegensatz zur stationären Simulation wird bei der instationären Simulation das Verhalten des Systems bei transienten Vorgängen ermittelt. Dabei wird das Verhalten der einzelnen Messwerte über der Zeit bestimmt, so dass Aussagen über die Wirkgeschwindigkeit des Systems gemacht werden können. Diese Art der Simulation findet u. a. bei der Untersuchung von Überlastabwehrmaßnahmen Verwendung, da sie Aussagen über die Reaktionsfähigkeit und Stabilität der einzelnen Verfahren möglich macht [73, 99, 100, 128].

Bei der instationären Simulation wird der Zeitbereich in äquidistante Intervalle unterteilt. Für jeden Simulationslauf werden die entsprechenden Messwerte für diese Intervalle bestimmt und statistisch ausgewertet. Dabei müssen die einzelnen Simulationsläufe statistisch unabhängig voneinander sein, was z. B. durch unterschiedliche Startwerte für die Zufallszahlengeneratoren erreicht werden kann. Die Ergebnisse der einzelnen Simulationsläufe werden, ebenso wie bei

der stationären Simulation, statistisch ausgewertet, so dass mittels Vertrauensintervallen ihre statistische Aussagesicherheit ermittelt werden kann.

4.2.2 Simulationsmodell

Im Folgenden wird das Modell, das für die simulativen Untersuchungen verwendet wird, vorgestellt. Dazu wird in Abschnitt 4.2.2.1 zunächst eine Übersicht über das untersuchte Szenario gegeben. Anschließend wird in Abschnitt 4.2.2.2 die Modellierung der Verkehrserzeugung beschrieben. In Abschnitt 4.2.2.3 wird das Modell eines Gatekeepers vorgestellt und in Abschnitt 4.2.2.4 wird dieses Modell zu einem Gatekeeper-Cluster erweitert. Schließlich wird in Abschnitt 4.2.2.5 das Modell einer H.323-Zone beschrieben, das sich aus den eben genannten Modellen zusammensetzt.

4.2.2.1 Übersicht

Das untersuchte Szenario basiert, wie in Bild 4.3 dargestellt, auf einem Netz, das aus mehreren Zonen besteht, die gemeinsam verwaltet werden. Dies könnte z. B. ein Unternehmensnetz darstellen, das entsprechend strukturiert wurde. Die einzelnen Zonen werden durch einen Cluster von Gatekeepern oder einen einzelnen Gatekeeper gesteuert. Da der Gegenstand der Untersuchungen die Steuerung der VoIP-Umgebung ist, werden im weiteren Verlauf ausschließlich die Signalisiernachrichten und ihre Bearbeitung betrachtet. Weitere Komponenten sind Terminals, die VoIP-Rufe erzeugen bzw. entgegen nehmen. Andere Komponenten, wie Gateways oder MCUs, unterscheiden sich bezüglich ihrer Signalisiernachrichten im Vergleich zu einem Terminal kaum oder gar nicht und sind daher kein Bestandteil der weiteren Untersuchungen.

4.2.2.2 Verkehrserzeugung

Die Verkehrserzeugung erfolgt über Modelle von Endpunkten und dazugehörigen Teilnehmern. Dabei wird zwischen Endpunkten, die Rufe erzeugen, den A-Endpunkten, und Endpunkten, die Rufe entgegen nehmen, den B-Endpunkten, unterschieden. In Bild 4.4 ist das prinzipielle Simulationsmodell für beide Endpunkt-Kategorien dargestellt.

Dieses Modell verfügt über zwei Warteschlangen, wobei eine für interne, hochpriore Nachrichten verwendet wird (*High Priority Input Queue*), wie z. B. Timer-Nachrichten. Die zweite Warteschlange dient der Pufferung der von anderen Komponenten ankommenden Nachrichten (*Standard Input Queue*). In der Regel handelt es sich dabei um die simulierten Signalisiernachrichten. Der sog. *Priority-Multiplexer* sorgt dafür, dass zuerst alle hochprioren Nachrichten dem *Server* zur Bearbeitung übergeben werden, bevor eine Standard-Nachricht aus der entsprechenden Warteschlange abgeholt wird.

Der zentrale Bestandteil des Modells ist der Server, der die Bearbeitung der einzelnen Nachrichten durchführt. Zur Interpretation der Nachrichten wird im Server ein erweiterter, endlicher

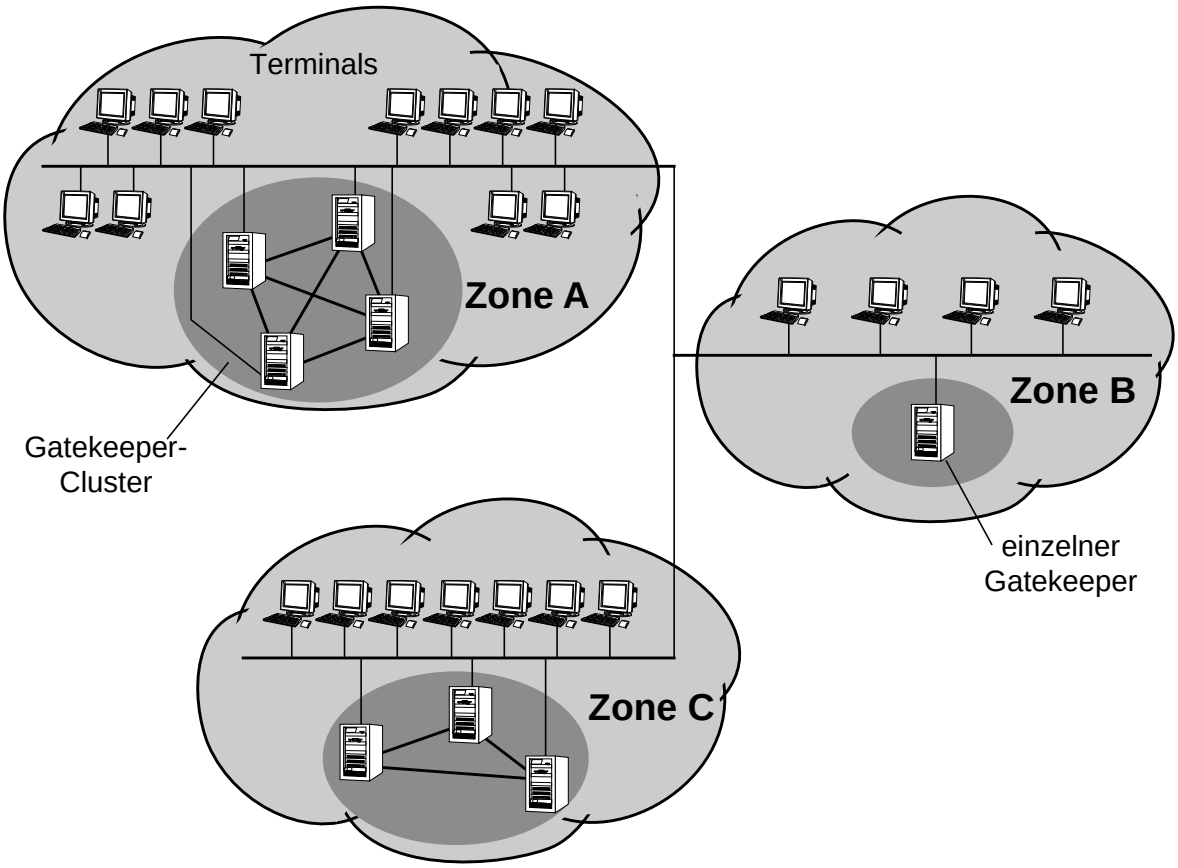


Bild 4.3: Untersuchtes Szenario

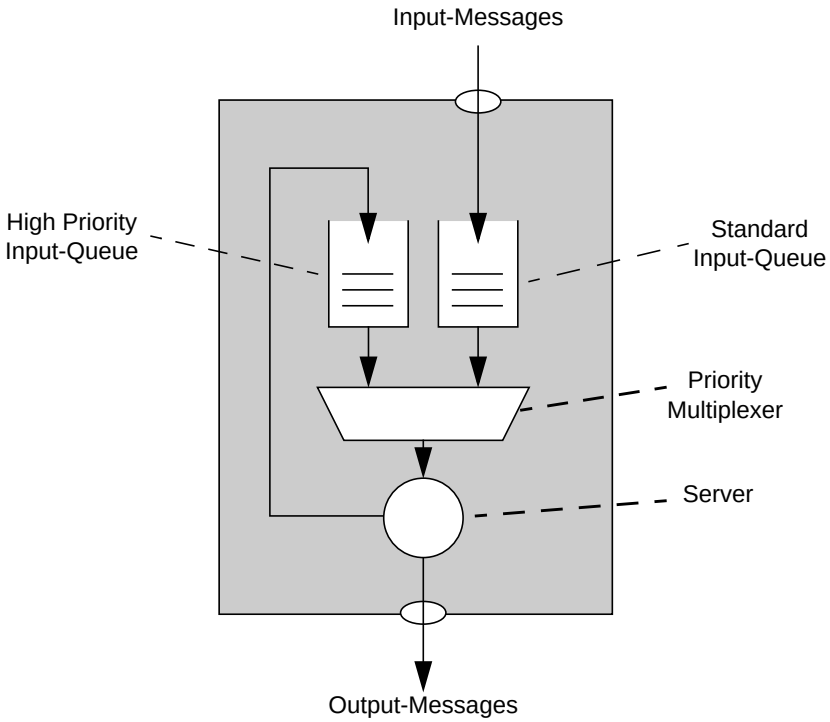


Bild 4.4: Modell eines Endpunkts

Zustandsautomat (EFSM - *Extended Finite State Machine*) ausgeführt, der wesentliche Teile des Signalisierprotokolls für H.323-basierte VoIP-Verbindungen nachbildet. Die Spezifikation der realisierten EFSM für einen A-Endpunkt ist in Anhang A.2 dargestellt. Für die Spezifikation wurde die von der ITU-T in [53] definierte Spezifikationssprache SDL (*Specification and Description Language*) verwendet. Dabei handelt es sich um die Prozeduren der RAS-Signalisierung und der Signalisierung für die Verbindungssteuerung. Die simulierte Bearbeitungszeit einer Nachricht wird durch den Zustand des Protokollautomaten und den Typ der bearbeiteten Nachricht festgelegt. Während der Nachrichtenbearbeitung können weitere Signalisiernachrichten für andere Komponenten entstehen. Des Weiteren werden beim Ablauf von Timern entsprechende hochpriore Nachrichten erzeugt, die intern der entsprechenden Warteschlange übergeben werden. Damit wird erreicht, dass durch den Timer-Ablauf die aktuelle Nachrichtenbearbeitung nicht unterbrochen wird (*non-preemptive priorities*).

Die Modelle der A- und der B-Endpunkte unterscheiden sich im wesentlichen nur in der im Server realisierten EFSM. Während ein modellierter B-Endpunkt auf ankommende Verbindungsanforderungen wartet und diese dann entsprechend bearbeitet, erzeugt ein modellierter A-Endpunkt neue Verbindungsanforderungen. Dazu wird er von einem Generator gesteuert, der für die Einhaltung der eingestellten Ankunftsrate von Verbindungsanforderungen zuständig ist. Der Generator initiiert Verbindungsanforderungen nach vorgegebenen statistischen Verteilungen, der weitere Verlauf der Verbindung wird durch die modellierten Endpunkte selbstständig gesteuert. Dabei werden neben der Haltedauer einer Verbindung auch Wartezeiten der Teilnehmer, wie z. B. die Zeitdauer bis eine Verbindung durch den gerufenen Teilnehmer angenommen wird, durch entsprechende statistische Verteilungen modelliert.

Wie oben beschrieben, erfolgt der Austausch der Signalisiernachrichten in weiten Teilen konform gegenüber der Empfehlung H.323. Darüber hinaus kann auch das Beantragen und Durchführen von zusätzlichen Dienstmerkmalen simuliert werden. Dazu wird der in Bild 4.5 dargestellte, generische Nachrichtenablauf nachgebildet. Die Initiierung der zusätzlichen Dienste erfolgt im Verbindungszustand nach statistischen Verteilungen, wobei die Bearbeitung mehrerer zusätzlicher Dienste im Laufe einer Verbindung simuliert werden kann.

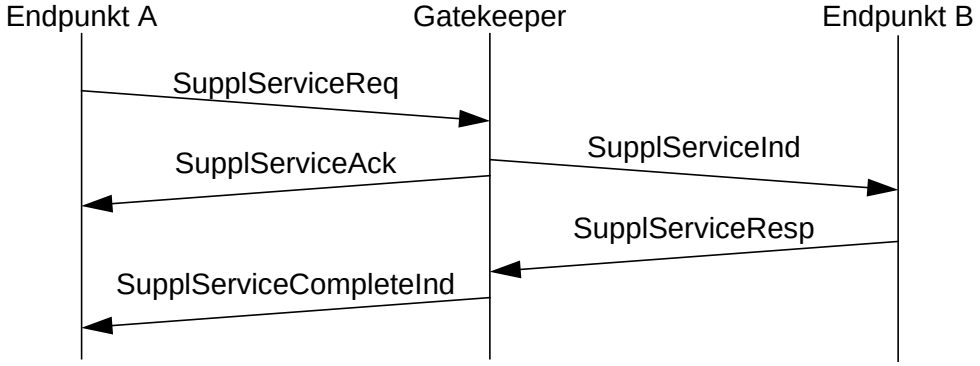


Bild 4.5: Generischer Nachrichtenablauf für zusätzliche Dienste

Die aus Zustandsübergängen der Endpunkte resultierenden Nachrichten werden jeweils an den zuständigen Gatekeeper bzw. Gatekeeper-Cluster gesendet. Dort werden sie verarbeitet und es werden entsprechend neue Nachrichten für andere Gatekeeper und für die an der Verbindung beteiligten Endpunkte erzeugt. Somit erfolgt kein direkter Nachrichtenaustausch zwischen den Endpunkten einer Verbindung. Bei der Modellierung des Nachrichtenaustauschs wird die Transferzeit zwischen Endpunkten und Gatekeepern vernachlässigt. Dies bedeutet, dass nur die Bearbeitungs- und Wartezeiten in den verschiedenen Komponenten simuliert werden. Dabei wird angenommen, dass durch entsprechenden Austausch von Signalisiernachrichten einzelne Komponenten einer VoIP-Umgebung in Über- oder Hochlast geraten, wobei aber das verwendete IP-Netz selbst noch normal belastet ist. Darüber hinaus ist diese Annahme auch für stärker belastete IP-Netze gerechtfertigt, wenn eine entsprechende Priorisierung der Signalisiernachrichten im IP-Netz gegeben ist.

4.2.2.3 Gatekeeper

Das Modell des Gatekeepers ist dem der Endpunkte sehr ähnlich, wobei zusätzlich hochpriore Nachrichten empfangen und gesendet werden. In Bild 4.6 sind die dafür verwendeten Komponenten enthalten. Diese hochprioren Nachrichten werden ausschließlich zwischen Gatekeepern verwendet, um Informationen zur Steuerung der Intrazonen- und Interzonen-Lastverteilungsverfahren auszutauschen.

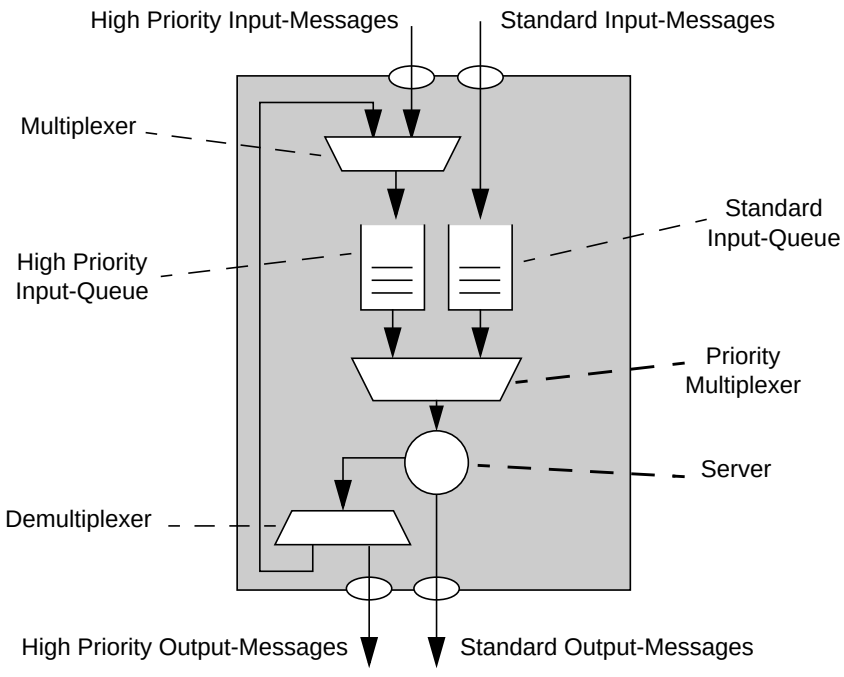


Bild 4.6: Modell eines Gatekeepers

Ein wesentlicher Unterschied zu den Modellen der Endpunkte besteht im Server und der darin enthaltenen EFSM zur Bearbeitung der Signalisiernachrichten, deren Spezifikation in Anhang A.1 dargestellt ist. Die Bearbeitungszeiten für die einzelnen Nachrichten werden individuell in

der EFSM festgelegt, wobei für die wesentlichen Bearbeitungszeiten, wie z. B. für Verbindungsanforderungen, Verteilungen festgelegt werden. Damit wird den Schwankungen bei den Bearbeitungszeiten, die sich z. B. aus dem Ziel einer Verbindung ergeben, Rechnung getragen. Zur Simulation verschieden leistungsfähiger Gatekeeper werden diese Bearbeitungszeiten jeweils mit einem Leistungsfaktor multipliziert, der in der Konfiguration für die Simulationen festgelegt wird.

Die Lastindikatorermittlung sowie die Durchführung der Überlastabwehrmaßnahmen sind, wie in Bild 4.7 dargestellt, innerhalb des Gatekeeper-Servers als eigenständige Module realisiert. Abhängig vom aktuellen Zustand der EFSM und von der angekommenen Nachricht, werden sie von der EFSM aufgerufen, um ihre Dienste zu erbringen.

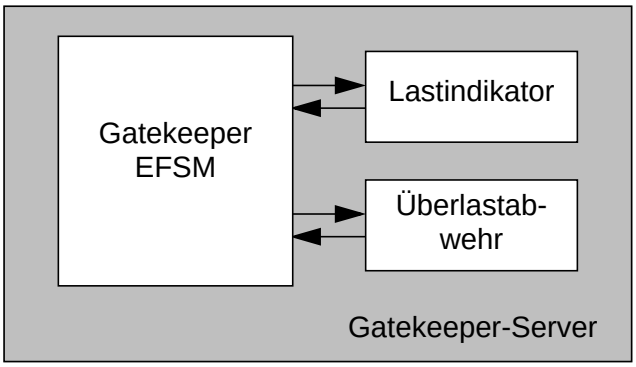


Bild 4.7: Integration von Lastindikatorermittlung und Überlastabwehrmaßnahmen im Gatekeeper-Modell

Die Interzonen-Lastverteilung, die ebenfalls Bestandteil des Gatekeeper-Servers ist, arbeitet nahezu unabhängig von der Gatekeeper-EFSM. Des Weiteren verfügt der Gatekeeper-Server über Daten zur Steuerung der einzelnen Verbindungen der Endpunkte, auf die hier aber nicht weiter eingegangen werden soll.

4.2.2.4 Gatekeeper-Cluster

Zur Modellierung eines Gatekeeper-Clusters wird das im vorigen Abschnitt vorgestellte Modell eines Gatekeepers verwendet und entsprechend erweitert.

Zur Untersuchung der Granularität der Lastverteilung ist es notwendig, den Zugriff auf die Zustandsdaten einer Verbindung zu simulieren. Dies erfolgt bei den Cluster-Modellen, indem zusätzliche Bearbeitungszeiten simuliert werden, sobald entweder Zustandsdaten von einer zentralen Instanz gelesen oder bei einer zentralen Instanz geschrieben werden müssen. So wird z. B. bei der Granularität auf Verbindungsebene bei der ersten Nachricht einer Verbindung die Zeit für das Lesen der Zustandsdaten zur normalen Bearbeitungszeit der Nachrichtenbearbeitung addiert. Bei Verbindungsende wird entsprechend die Zeit für das Schreiben der Zustandsdaten zur Bearbeitungszeit der entsprechenden Nachricht hinzugefügt.

Cluster mit zentraler Steuerung

Die Steuerung der Lastverteilung in dieser Cluster-Realisierung erfolgt durch einen Dispatcher. Dieser wird ebenso wie ein Gatekeeper modelliert (vgl. Bild 4.6), wobei er keine EFSM bezüglich der Bearbeitung eines Signalisierprotokolls zur Steuerung von VoIP-Verbindungen ausführt, dafür aber ein Modul zur Durchführung der Lastverteilung enthält. In Bild 4.8 ist das Modell des Clusters mit zentraler Steuerung dargestellt. Dabei wurden die Komponenten, die hochpriore Nachrichten betreffen, gestrichelt dargestellt. Bei diesem Modell können Nachrichten von den Gatekeepern des Clusters sowohl direkt an die Endpunkte gesendet werden als auch über den Dispatcher geführt werden, so dass dieser die Nachrichten schließlich an die Endpunkte weiterleitet.

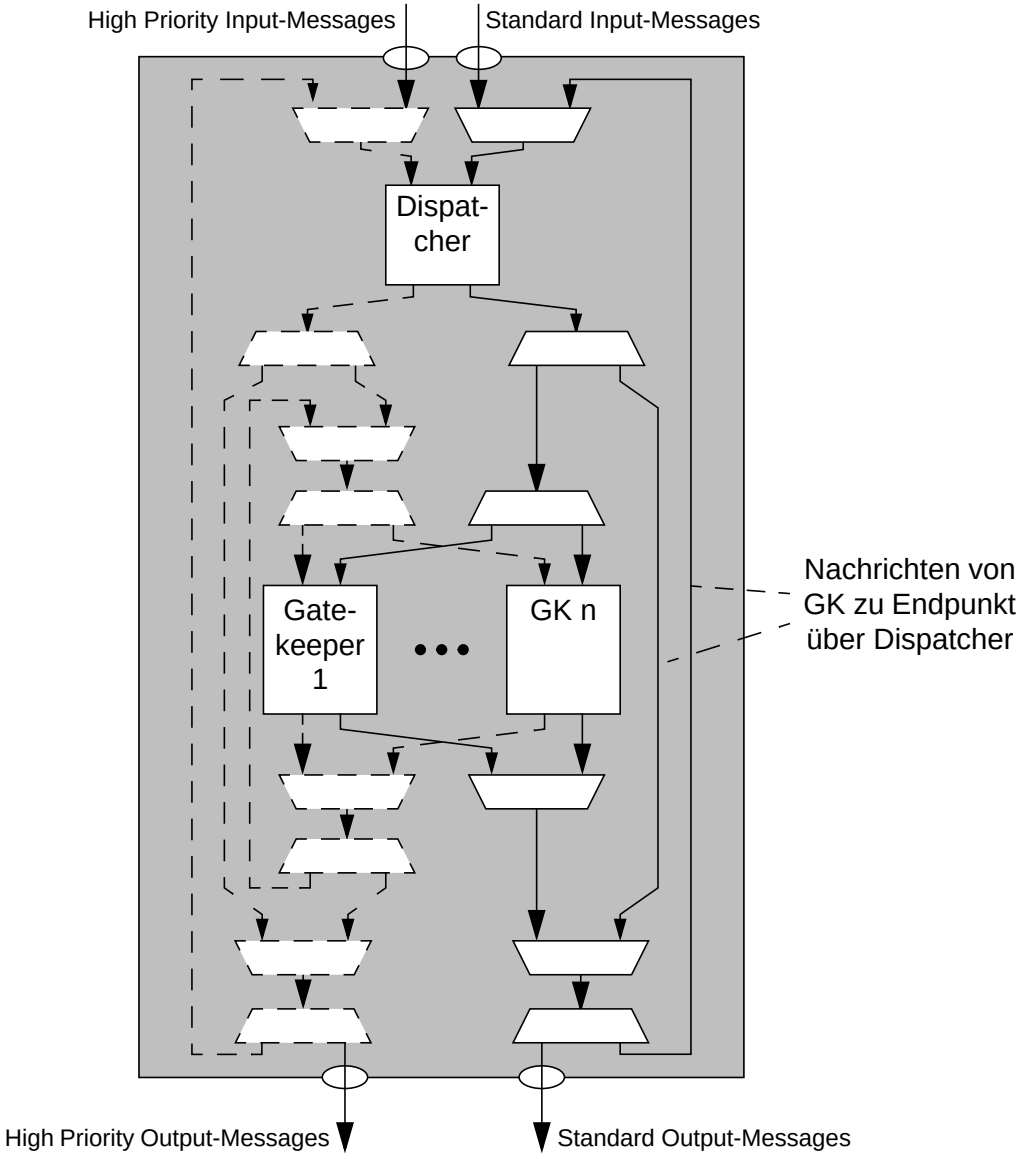


Bild 4.8: Modell eines zentral gesteuerten Gatekeeper-Clusters

Cluster mit verteilter Steuerung

Bei einem Gatekeeper-Cluster mit verteilter Steuerung führen die einzelnen Cluster-Mitglieder selbst die Lastverteilung durch. Daher verfügt der Gatekeeper-Server zusätzlich zu den in Bild 4.7 dargestellten Modulen noch über ein Modul zur Intrazonen-Lastverteilung. Dabei wird vor der Durchführung der Überlastabwermaßnahmen fest gestellt, ob die entsprechende Anfrage an ein anderes Cluster-Mitglied weiter geleitet werden soll.

Das Modell eines Gatekeeper-Clusters mit verteilter Steuerung ist in Bild 4.9 dargestellt, wobei auf die Komponenten, die hochpriore Nachrichten betreffen, verzichtet wurde, da sie gleich miteinander verbunden sind, wie die für die Standardnachrichten.

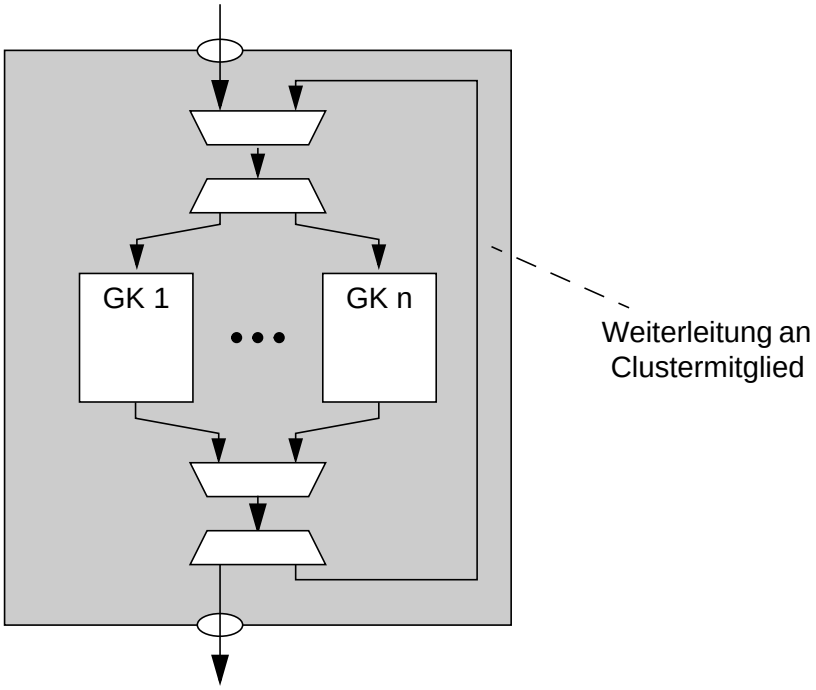


Bild 4.9: Modell eines Gatekeeper-Clusters mit verteilter Steuerung
(nur Komponenten für Standardnachrichten dargestellt)

4.2.2.5 Zone

In Bild 4.10 ist schließlich das Modell einer Zone dargestellt, wobei wiederum die Komponenten für den hochprioren Nachrichtenaustausch aus Übersichtlichkeitsgründen vernachlässigt wurden. Dieses Modell beinhaltet neben dem Modell des die Zone verwaltenden Gatekeeper bzw. Gatekeeper-Cluster die Modelle der A- und der B-Endpunkte der simulierten Zone. Des Weiteren werden die Verbindungsmöglichkeiten zwischen verschiedenen Zonen aufgezeigt: H.323-Signalisiernachrichten zwischen verschiedenen Zonen werden ausschließlich zwischen den Gatekeepern dieser Zonen ausgetauscht. Ein direkter Nachrichtenaustausch zwischen einem Gatekeeper und einem Endpunkt, der nicht Mitglied der Zone des Gatekeepers ist, ist nicht möglich.

Neben dem Modell der Zone ist in Bild 4.10 ein Generator zur Erzeugung vom Verbindungsanforderungen nach einer vorgegebenen statistischen Verteilung dargestellt. Die Initiierung von Verbindungsanforderungen erfolgt dabei ausschließlich bei Endpunkten, die sich im Ruhe-Zustand (*Idle*) befinden.

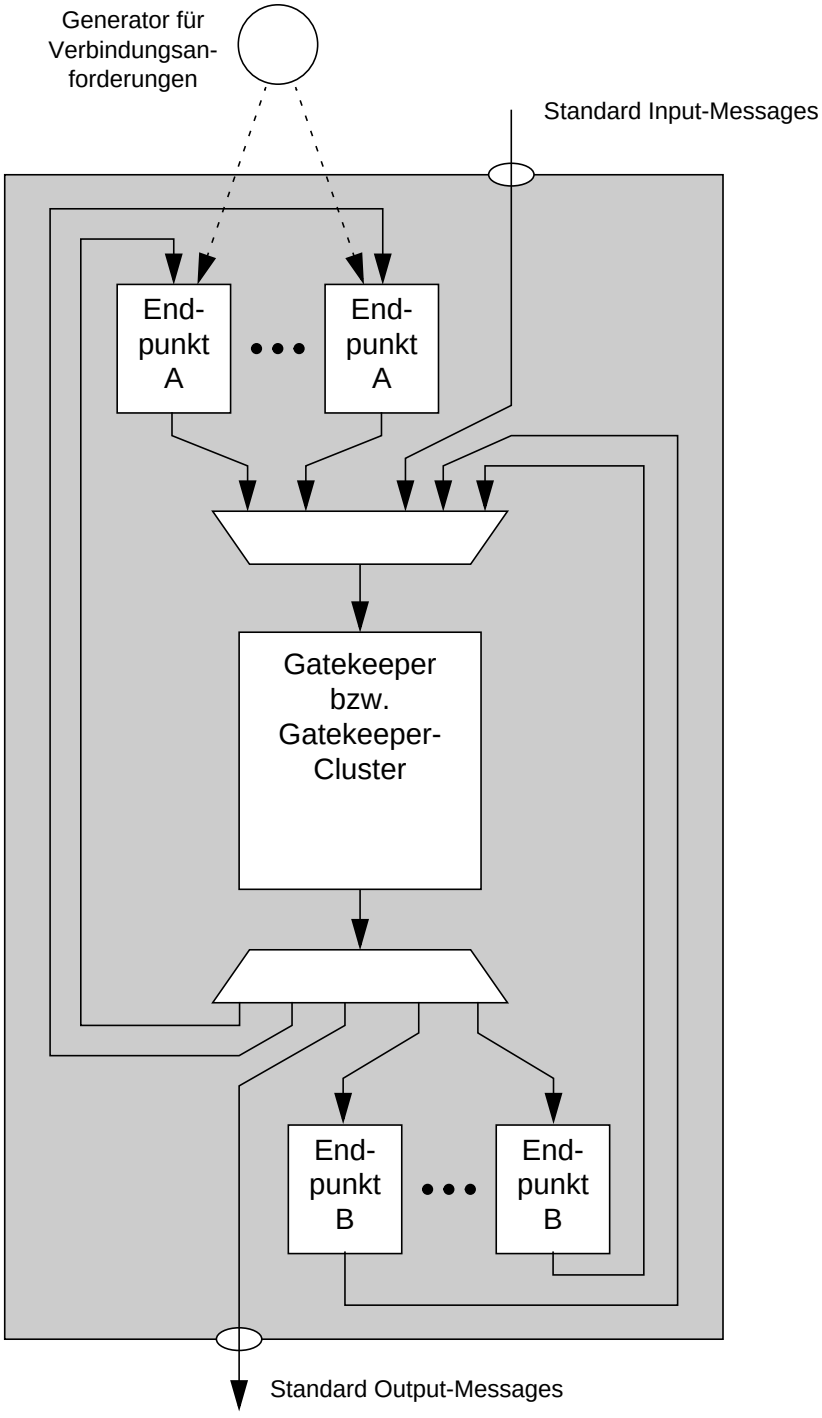


Bild 4.10: Modell einer Zone

4.2.3 Simulationswerkzeug

Für die Simulation der vorgestellten Modelle wird ein Programm eingesetzt, das mittels objektorientierter Methoden entworfen und in C++ implementiert wurde. Als Basis dient dabei die am Institut für Kommunikationsnetze und Rechnersysteme der Universität Stuttgart (IKR) entwickelte C++-Bibliothek *SimLib* [63, 64], die verschiedene Grundelemente für die Simulation von Kommunikationssystemen, wie z. B. Generatoren, Multiplexer, Messsysteme oder Module für statistische Auswertungen, zur Verfügung stellt.

Für die Simulation instationärer Vorgänge wurde während des Simulationslaufes ein sog. *Trace* aufgezeichnet, der die zwischen den einzelnen Simulationskomponenten ausgetauschten Nachrichten zusammen mit einem Zeitstempel enthält. Darüber hinaus werden bestimmte Ereignisse, wie z. B. das Fehlschlagen einer Verbindungsanforderung ebenfalls in diesen Trace geschrieben. Durch mehrmaliges Durchführen der Simulationen, jeweils mit unterschiedlichen Startwerten des Zufallszahlengenerators, können Traces von somit unabhängigen Simulationsläufen erzeugt werden, die dann zur Bestimmung des instationären Verhaltens der untersuchten Komponenten mittels entsprechender statistischer Methoden ausgewertet werden können.

Kapitel 5

Ergebnisse und Bewertung

In diesem Kapitel werden die Ergebnisse der Studien, die unter Anwendung der in Kapitel 4 beschriebenen Untersuchungsmethoden durchgeführt wurden, präsentiert. Des Weiteren erfolgt jeweils eine Bewertung dieser Ergebnisse.

Zunächst werden in Abschnitt 5.1 die Studien zur Steuerungsoptimierung für einen Gatekeeper vorgestellt und bewertet. Anschließend werden in Abschnitt 5.2 die Untersuchungen zur Steuerungsoptimierung für einen Gatekeeper-Cluster präsentiert. Dabei wird u. a. eine Bewertung der untersuchten Lastverteilungsverfahren, sowie möglicher Ebenen der Granularität der Lastverteilung durchgeführt. Schließlich erfolgen in Abschnitt 5.3 die Untersuchungen zur Steuerungsoptimierung über Zonengrenzen hinweg. Dies beinhaltet die Untersuchung und Bewertung des in Abschnitt 3.5.2.2 eingeführten Interzonen-Lastverteilungsverfahrens.

5.1 Steuerungsoptimierung für einen Gatekeeper

Die Steuerungsoptimierung für einen Gatekeeper beinhaltet, wie in Abschnitt 3.5 beschrieben, die Bestimmung des aktuellen Lastzustands mittels geeigneter Lastindikatoren und die Durchführung von Überlastabwehrmaßnahmen. Da zunächst die Steuerungsoptimierung für einen einzelnen Gatekeeper untersucht wird, werden die Verfahren zur Lastverteilung in diesem Abschnitt nicht betrachtet.

In Abschnitt 5.1.1 werden die Ergebnisse der Untersuchungen an einer prototypischen Implementierung der Verfahren vorgestellt. Anschließend werden in Abschnitt 5.1.2 die Ergebnisse der Simulationsstudien präsentiert. Schließlich werden diese Ergebnisse in Abschnitt 5.1.3 bewertet.

5.1.1 Untersuchungen an prototypischer Implementierung

Die in diesem Abschnitt vorgestellten Ergebnisse wurden mit dem in Abschnitt 4.1 vorgestellten PreServer gewonnen. Der PreServer verwendet als Lastindikator für den Gatekeeper die Antwortverzögerung für die einzelnen Signalisiernachrichten, wobei fünf Lastzustände unterschieden werden. Des Weiteren sind die Überlastabwehrmaßnahmen „Prozentuale Drosselung“ (*Percentage Throttling* - PT), „Automatic Call Gapping“ (ACG), „Leaky Bucket“ (LB) und das „Fenster“-Verfahren (*Window-Method* - WIN) prototypisch implementiert.

Bei den im Folgenden präsentierten Ergebnissen werden die einzelnen Messwerte in Abhängigkeit des Angebots dargestellt. Das Angebot stellt dabei die Rate der Verbindungsanforderungen im Verhältnis zum maximalen Durchsatz, d. h. der maximalen Rate an erfolgreichen Verbindungsanforderungen, dar. Wie in Abschnitt 4.1.2.2 beschrieben, hängt das durch die Lastgeneratoren erzeugte Angebot selbst vom Ablauf der Messungen ab, so dass es ebenfalls während der Tests gemessen wurde.

Die einzelnen Punkte der folgenden Kurven (Bilder 5.1 bis 5.4) wurden jeweils während eines Tests nach dem Durchlaufen einer Warmlaufphase bestimmt. Es handelt sich somit um Untersuchungen im stationären Fall, wobei jeweils der Mittelwert innerhalb der stationären Phasen bestimmt wurde. Die Aussagesicherheit dieser Ergebnisse ist beschränkt, jedoch erlauben sie eine Einschätzung bezüglich der Realitätsnähe des entwickelten Simulationsmodells der weiteren Untersuchungen.

In Bild 5.1 ist der mittlere Durchsatz über dem mittleren Angebot für die verschiedenen Überlastabwehrmaßnahmen und ohne Anwendung einer Überlastabwehrmaßnahme (*No Overload Protection* - No OvP) dargestellt. Der Durchsatz wird dabei im Verhältnis zum Maximaldurchsatz des Gatekeepers angegeben. Des Weiteren ist der ideale Verlauf der Durchsatzkurve enthalten. Bis zu einem mittleren Durchsatz von ca. 0.8 Erlang befinden sich alle Messpunkte an bzw. auf der Ideallinie. Bei einem höheren mittleren Angebot wird bei der Anwendung von Überlastabwehrmaßnahmen der mittlere Durchsatz geringer als das Angebot, d. h. dass nicht alle Verbindungsanforderungen bearbeitet werden. Wenn keine Überlastabwehrmaßnahmen angewandt werden, wird die Ideallinie erst bei einem höheren mittleren Angebot verlassen, jedoch fällt der mittlere Durchsatz anschließend sehr stark ab, so dass ab einem mittleren Angebot von knapp unter 1.0 Erlang der mittlere Durchsatz bei Anwendung einer der Überlastabwehrmaßnahmen höher ist als im Fall ohne Überlastabwehr.

In Bild 5.2 ist die mittlere Antwortverzögerung für verbindungsbezogene RAS-Anfragen (ARQ, DRQ) über dem Angebot dargestellt. Während für die Fälle mit Überlastabwehrmaßnahmen die mittlere Antwortverzögerung auch im Überlastbereich unter 900 ms bleibt, steigt sie für den Fall ohne Überlastabwehrmaßnahme ab einem mittleren Angebot von ca. 0.95 Erlang auf Werte von über 1600 ms an.

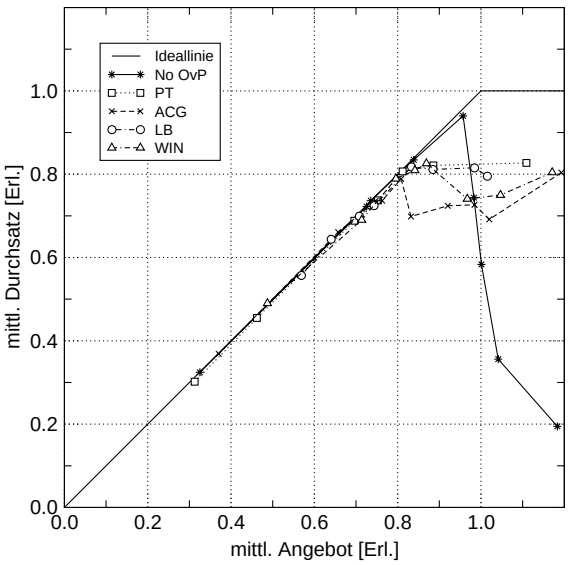


Bild 5.1: Vergleich des Durchsatzes

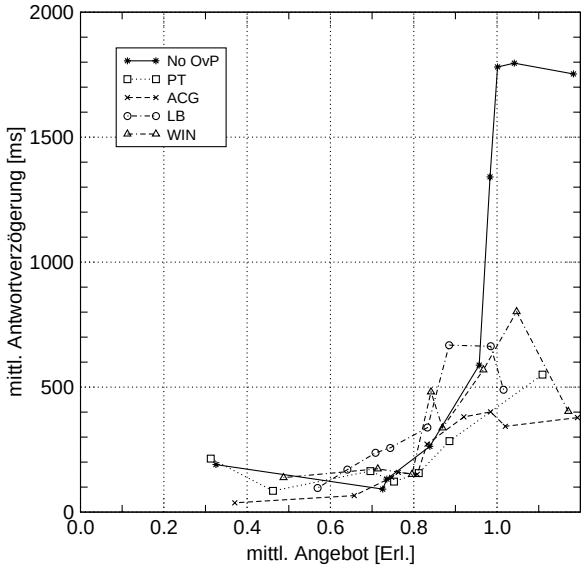


Bild 5.2: Vergleich der Antwortverzögerungen

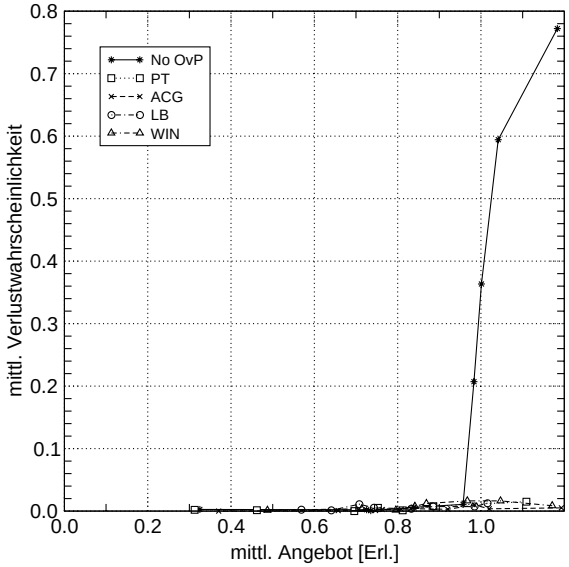


Bild 5.3: Vergleich der Verlustwahrscheinlichkeiten

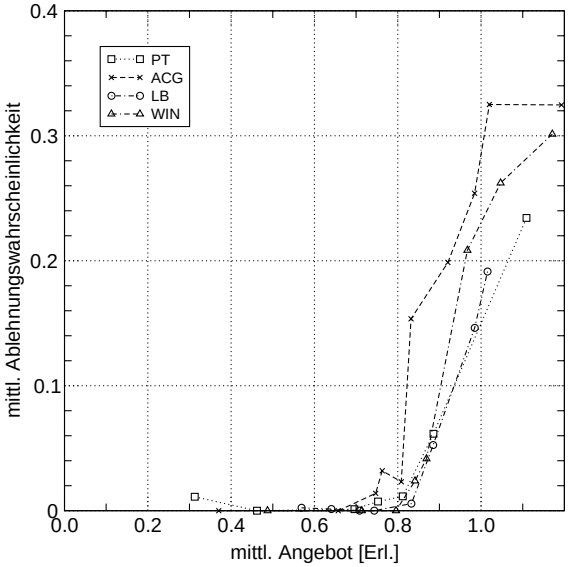


Bild 5.4: Vergleich der Ablehnungswahrscheinlichkeiten

Die in Bild 5.3 abgebildete mittlere Verlustwahrscheinlichkeit stellt den fehlschlagenden Anteil der Verbindungsanforderungen dar. Dabei sind durch Überlastabwehrmaßnahmen abgelehnte Verbindungsanforderungen nicht enthalten. Während bei der Anwendung von Überlastabwehrmaßnahmen diese Wahrscheinlichkeit auch im Überlastbereich sehr gering ist, steigt sie für den Fall ohne Überlastabwehrmaßnahmen ab einem mittleren Angebot von ca. 0.95 Erlang sehr stark an.

In Bild 5.4 ist schließlich die mittlere Ablehnungswahrscheinlichkeit für Verbindungsanforderungen über dem Angebot dargestellt. Dabei wird deutlich, dass bereits ab einem mittleren Angebot von ca. 0.8 Erlang Verbindungsanforderungen durch die Überlastabwehrmaßnahmen abgelehnt werden, um die Diensterbringung durch den Gatekeeper sicher zu stellen.

Als Fazit dieser Untersuchungen kann festgestellt werden, dass die Anwendung von Überlastabwehrmaßnahmen auf jeden Fall sehr sinnvoll ist, da sie zum einen den Durchsatz auch im Hoch- und Überlastbereich gegenüber dem Fall ohne Anwendung von Überlastabwehrmaßnahmen deutlich erhöhen und zum anderen die Antwortzeiten begrenzen. Darüber hinaus ist die Wahrscheinlichkeit für das Fehlschlagen von Verbindungsanforderungen sehr gering. Jedoch wird bei der Anwendung der Überlastabwehrmaßnahmen ein geringerer maximaler Durchsatz erreicht, als ohne Anwendung von Überlastabwehrmaßnahmen. Dies ist zum einen durch den Ressourcenverbrauch der Überlastabwehrmaßnahmen selbst bedingt. Zum anderen wurde eine konservative Ablehnungsstrategie angewandt, um das Fehlschlagen von Verbindungsanforderungen zu verhindern.

5.1.2 Simulative Untersuchungen

Die folgenden Ergebnisse wurden mittels der in Abschnitt 4.2 vorgestellten Methode der Simulation ermittelt. Dazu wurde das Modell einer einzelnen Zone, die durch einen Gatekeeper verwaltet wird, untersucht.

Das Ziel der Steuerungsoptimierung dieser Untersuchungen ist die Maximierung des Durchsatzes unter Einhaltung der folgenden Randbedingungen:

- Einhaltung einer maximalen ARQ-ACF-Verzögerung von 1000 ms bis zu einer Überlast von 100%, was einem Angebot von 2.0 Erlang entspricht.
- Keine fehlschlagenden Verbindungsanforderungen bis zu einer Überlast von 100%.

Die Generatoren zur Verkehrserzeugung initiieren Verbindungsanforderungen der A-Teilnehmer mit negativ-exponentiell verteilten Zwischenankunftsabständen (*interarrival times)*. Die Haltedauern der Verbindungen sind ebenfalls negativ-exponentiell verteilt. Wie bei den Untersuchungen mittels der prototypischen Implementierung werden fünf Lastzustände unterschieden, denen jeweils entsprechende Parameter der Überlastabwehrmaßnahmen zugeordnet sind.

In Abschnitt 5.1.2.1 werden Ergebnisse von Untersuchungen im stationären und instationären Fall für verschiedene Lastindikatoren vorgestellt. Anschließend werden in Abschnitt 5.1.2.2 entsprechende Ergebnisse für verschiedene Überlastabwehrmaßnahmen präsentiert. Schließlich werden in Abschnitt 5.1.2.3 Auswirkungen zusätzlicher Dienste und damit verbundener veränderter Verkehrscharakteristika untersucht.

5.1.2.1 Untersuchung von Lastindikatoren

Im Folgenden werden Ergebnisse von Untersuchungen verschiedener Lastindikatoren vorgestellt. Als Überlastabwehrmaßnahme wurde dabei das „Leaky Bucket“ Verfahren angewandt, das für alle Lastindikatoren gleich konfiguriert war. Es wurde das Verhalten der Lastindikato-

ren „Anzahl offener Anfragen“ (*Number of Open Requests* - NOR), „Warteschlangenlänge“ (*Queue Length* - QL) und „Gewichtete Verbindungszustände“ (*Weighted Connection States* - WCS) untersucht. Bei der Verkehrserzeugung wurden keine Anfragen für zusätzliche Dienste erzeugt.

Untersuchung des stationären Verhaltens

Die folgenden Ergebnisse werden jeweils in Abhängigkeit des Angebots betrachtet, wobei das Angebot, wie in Abschnitt 5.1.1, der Ankunftsrate von Verbindungsanforderungen im Verhältnis zum maximalen Durchsatz des simulierten Gatekeepers entspricht. Die Ergebnisse werden bis zu einer Überlast von 150% dargestellt, was bei einem einzelnen Gatekeeper ein Angebot von 2.5 Erlang ergibt.

Aus Übersichtlichkeitsgründen wurde bei den folgenden Diagrammen meist auf die Darstellung der Vertrauensintervalle der Ergebnisse verzichtet. Die Durchführung der Untersuchungen wurde so gestaltet, dass die Vertrauensintervalle in der Regel hinreichend klein sind, so dass die Aussagesicherheit der Ergebnisse gegeben ist.

In den Bildern 5.5 und 5.6 ist der mittlere Durchsatz über dem Angebot für die unterschiedliche Lastindikatoren, die gemeinsam mit der Überlastabwehrmaßnahme „Leaky Bucket“ angewendet werden, sowie für den Fall ohne Überlastabwehr (No OvP) dargestellt. Die Ideallinie zeigt den idealen Verlauf des Durchsatzes an. Aus Bild 5.5 wird deutlich, wie der Durchsatz ohne Durchführung einer Überlastabwehr abfällt, wenn das Angebot den Wert 1.0 Erlang übersteigt. Des Weiteren fällt auch der Durchsatz bei Anwendung einer Überlastabwehrmaßnahme ab, jedoch deutlich weniger steil als im Falle ohne Überlastabwehr. Dieses Abfallen des Durchsatzes ist durch den Ressourcenverbrauch für das Ablehnen von Verbindungsanforderungen bedingt. In weiteren Untersuchungen, die hier nicht aufgeführt werden, konnte eine entsprechende Änderung der Steigung durch Variieren der Bearbeitungszeiten für das Ablehnen von Verbindungsanforderungen beobachtet werden.

Bild 5.6 enthält eine Detailansicht für den Durchsatz über dem Angebot, um die einzelnen Lastindikatoren zu vergleichen. In diesem Diagramm sind die Vertrauensintervalle enthalten, damit die Signifikanz der Unterschiede zwischen den Verfahren bewertet werden kann. Bei einem Angebot von 1.0 Erlang wird mit dem Lastindikator „Warteschlangenlänge“ ein etwas höherer mittlerer Durchsatz als mit den Lastindikatoren „Anzahl offener Anfragen“ oder „Gewichtete Verbindungszustände“ erzielt. Da die Unterschiede aber gering sind und sich die Vertrauensintervalle jeweils überschneiden, kann bezüglich des Durchsatzes im stationären Fall keine Überlegenheit eines Lastindikators gegenüber einem anderen gesichert festgestellt werden.

In Bild 5.7 ist die mittlere ARQ-ACF-Verzögerung über dem Angebot dargestellt. Ab einem Angebot von knapp unter 1.0 Erlang steigt diese Verzögerung für alle untersuchten Fälle stark

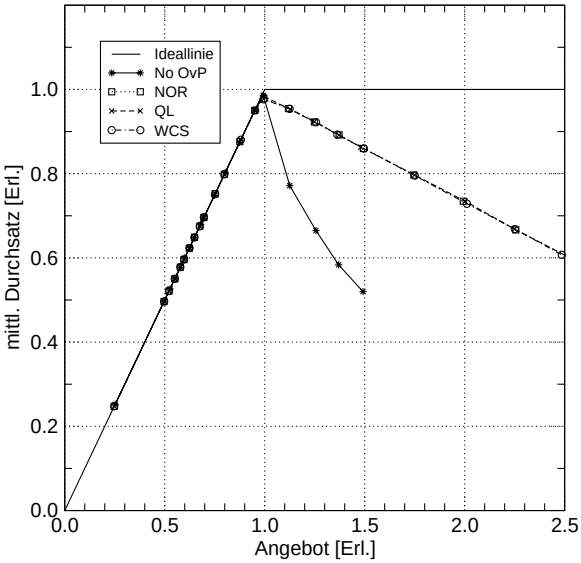


Bild 5.5: Vergleich des Durchsatzes

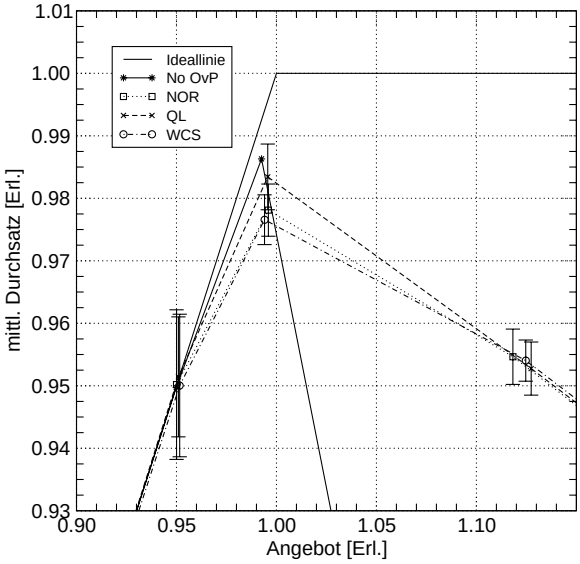


Bild 5.6: Vergleich des Durchsatzes - Detailansicht

an, wobei sie für den Fall ohne Überlastabwehr nahezu senkrecht ansteigt und dabei einen Wert von knapp 2000 ms erreicht. Das große Vertrauensintervall für die Antwortverzögerung beim Angebot von 1.0 Erlang beim Fall ohne Überlastabwehr lässt sich dadurch erklären, dass bei diesem Angebot die Grenze der Stabilität des Systems erreicht ist, so dass minimalste Schwankungen des Angebots bereits zu relativ großen Auswirkungen auf das Verhalten des Systems führen. Des Weiteren wird die angegebene Forderung, dass bis zu einer Überlast von 100% die Antwortverzögerung 1000 ms nicht überschreiten darf, durch die untersuchten Lastindikatoren zusammen mit der Überlastabwehrmaßnahme „Leaky Bucket“ erfüllt. Dabei steigt die Antwortverzögerung beim Lastindikator „Warteschlangenlänge“ zunächst schneller mit dem Angebot an als bei den Lastindikatoren „Anzahl offener Anfragen“ und „Gewichtete Verbindungszustände“, jedoch wird dieser Anstieg ab einem Angebot von ca. 1.1 Erlang deutlich schwächer, so dass die beobachtete Antwortverzögerung ab einem Angebot von ca. 2.1 Erlang geringer als für die anderen Lastindikatoren ist.

Die Wahrscheinlichkeit für das Fehlschlagen von Verbindungsanforderungen ist in Bild 5.8 abgebildet. Dabei zeigt sich, dass bei der Anwendung der Lastindikatoren gemeinsam mit einer Überlastabwehrmaßnahme bis zumindest zu einem Angebot von 2.5 Erlang keine Verbindungsanforderungen fehlschlagen. Wenn jedoch keine Überlastabwehr durchgeführt wird, steigt diese Wahrscheinlichkeit ab einem Angebot von knapp unter 1.0 Erlang sehr schnell an. Schließlich werden in Bild 5.9 die Wahrscheinlichkeiten für die Ablehnung von Verbindungen dargestellt, wobei sich keine nennenswerten Unterschiede zwischen den einzelnen Lastindikatoren ergeben.

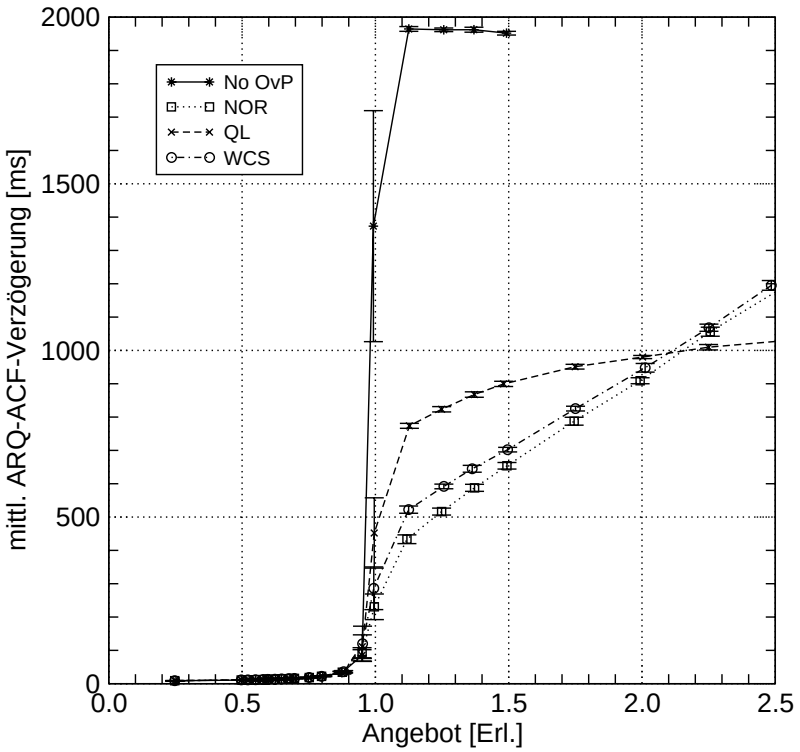


Bild 5.7: Vergleich der ARQ-ACF-Verzögerung

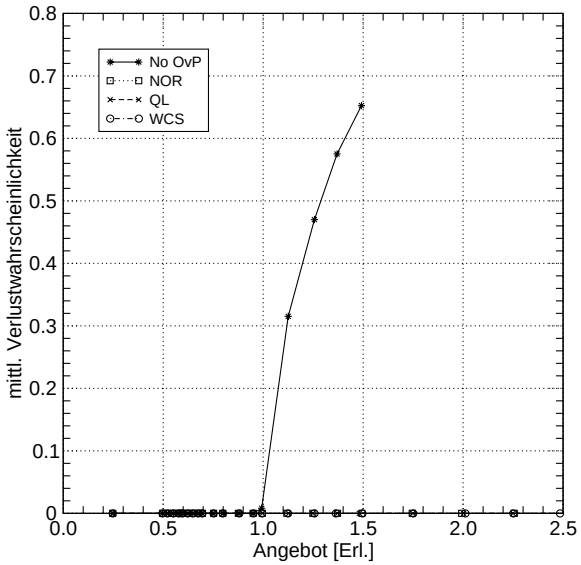


Bild 5.8: Vergleich der Verlustwahrscheinlichkeiten

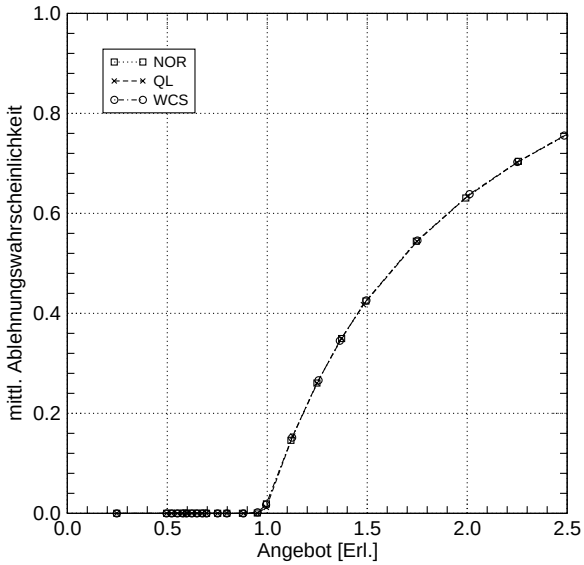


Bild 5.9: Vergleich der Ablehnungswahrscheinlichkeiten

Untersuchung des instationären Verhaltens

Bei den im Folgenden vorgestellten Ergebnisse wurde der zeitliche Verlauf der Messwerte ermittelt. Dazu wurden die Simulationen 50 mal mit jeweils unterschiedlichen Startwerten der Zufallszahlengeneratoren durchgeführt. Anschließend wurden für die jeweiligen Intervalle Mittelwert und Vertrauensintervall für den Messwert bestimmt. Dabei beträgt die Intervalllänge eine Sekunde. Der Wert 50 für die Anzahl der durchgeführten Simulationsläufe stellt

einen Kompromiss zwischen der Genauigkeit der ermittelten Mittelwerte und dem Aufwand für die durchgeführten Simulationen dar.

Um das Verhalten der einzelnen Verfahren bei einem sprunghaften Anstieg und späterem Abfallen der Last zu ermitteln, wurde das in Bild 5.10 dargestellte Lastprofil bezüglich der Verbindungsanforderungen für die Simulationen verwendet. Bei diesem Lastprofil wird zunächst eine Basislast angelegt, die den Gatekeeper zu ca. 80% auslastet. Nach 40 Sekunden erfolgt ein sprunghafter Anstieg der Last auf ca. 50% Überlast, die nach weiteren 60 Sekunden wieder auf die Basislast reduziert wird. Im weiteren Verlauf wird dieses Lastprofil als Rechteckimpuls bezeichnet. In Bild 5.10 sind die Mittelwerte des gemessenen Angebots sowie die Vertrauensintervalle für die Messintervalle enthalten. Die ermittelten Vertrauensintervalle zeigen erhebliche Schwankungen der Messwerte an, was durch die geringe Anzahl von Messwerten pro Intervall erklärt werden kann. Jedoch lassen diese Ergebnisse das Bestimmen des grundsätzlichen Verlaufs der Messwerte zu.

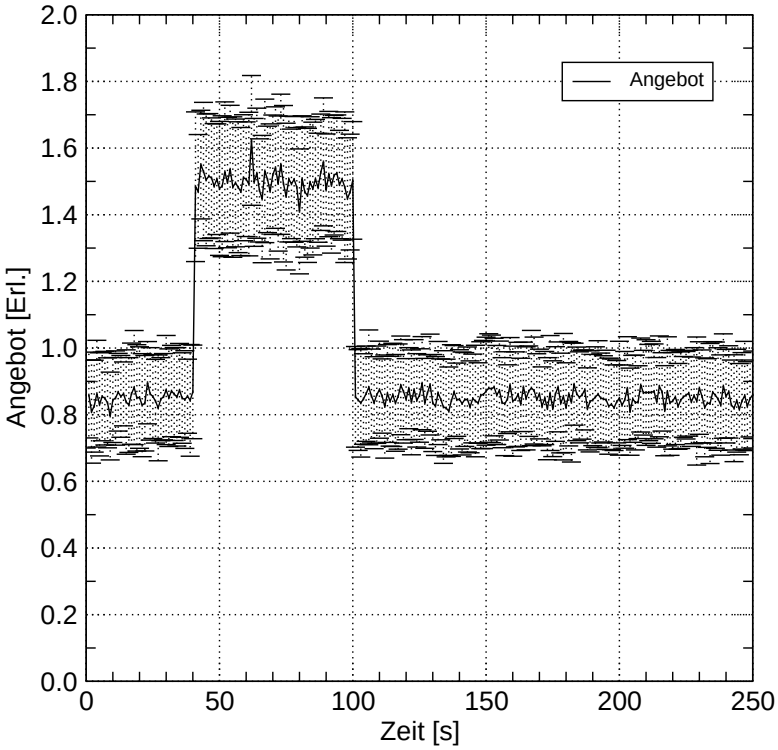


Bild 5.10: Lastprofil zur Ermittlung des instationären Verhaltens

Bild 5.11 enthält den Durchsatz über der Zeit, wie er sich durch den Rechteckimpuls ergibt, wenn keine Überlastabwehr durchgeführt wird. Dabei ergibt sich beim Wechsel zur Überlastphase, der nach 40 Sekunden stattfindet, ein kurzes Abfallen des Durchsatzes. Anschließend ist der Durchsatz bis ca. 60 Sekunden etwas höher als vor dem Lastsprung. Erst ab diesem Zeitpunkt fällt der Durchsatz deutlich ab. Wenn nach 100 Sekunden das Angebot wieder zur Basisbelastung zurückkehrt, steigt auch wieder der Durchsatz an, wobei erst nach ca. 120 Sekunden

ein relativ stabiler Durchsatz erzielt wird, der jedoch höher liegt, als bei der Basislast vor dem Lastsprung. Erst nach ca. 190 Sekunden wird wieder der ursprüngliche Durchsatz, wie er vor dem Lastsprung war, erreicht. Dieser höhere Durchsatz und das stark verzögerte Erreichen des ursprünglichen Durchsatzes lässt sich wie folgt erklären: Zur Bestimmung des Angebots werden die initialen Verbindungsanforderungsnachrichten verwendet. Falls im Verlauf des Verbindungsaufbaus Nachrichten nicht rechtzeitig beantwortet werden, erfolgt in der Regel eine Wiederholung der entsprechenden Nachricht. Da diese Wiederholungen für mehrere Nachrichten im Laufe eines Verbindungsaufbaus vorkommen können, ergibt sich eine Stauung der noch nicht vollständig bearbeiteten Verbindungsanforderungen. Bis diese Stauung durch den Gatekeeper beseitigt wurde, wird somit ein höherer Durchsatz erzielt, der aber nahezu der Maximalbelastung des Gatekeepers entspricht. Dies bedeutet, dass die Überlastsituation beim Gatekeeper erst nach 190 Sekunden beendet ist.

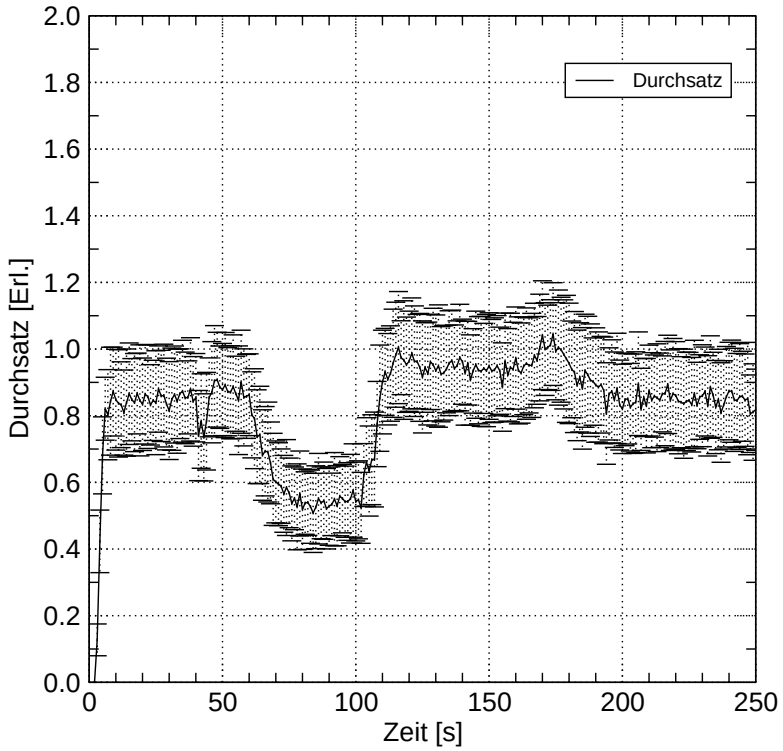


Bild 5.11: Durchsatz bei Rechteckimpuls ohne Überlastabwehr

In Bild 5.11 sind neben der Mittelwerte für den Durchsatz auch die entsprechenden Vertrauensintervalle abgebildet. Diese bewegen sich in der gleichen Größenordnung wie die des Angebots. Da diese Größenordnung auch für die weiteren Abbildungen gültig sind, werden die Vertrauensintervalle im Folgenden aus Übersichtlichkeitsgründen nicht dargestellt.

Um das instationäre Verhalten für die verschiedenen Lastindikatoren zu vergleichen, sind in den Bildern 5.12 bis 5.15 jeweils der zeitliche Verlauf des Durchsatzes und der ARQ-ACF-Verzögerung gemeinsam mit dem Angebot des Rechteckimpulses dargestellt. Dabei wird die

linke Ordinatenachse jeweils für das Angebot und den Durchsatz verwendet, während die rechte die Skalierung für die ARQ-ACF-Verzögerung anzeigt.

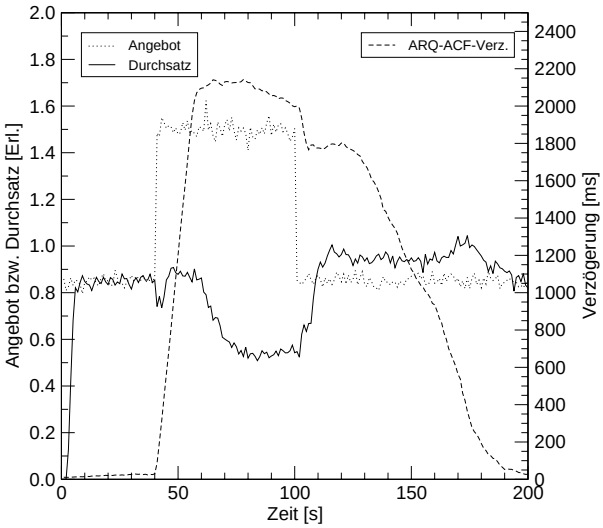


Bild 5.12: Durchsatz und Antwortverzögerung ohne Überlastabwehr

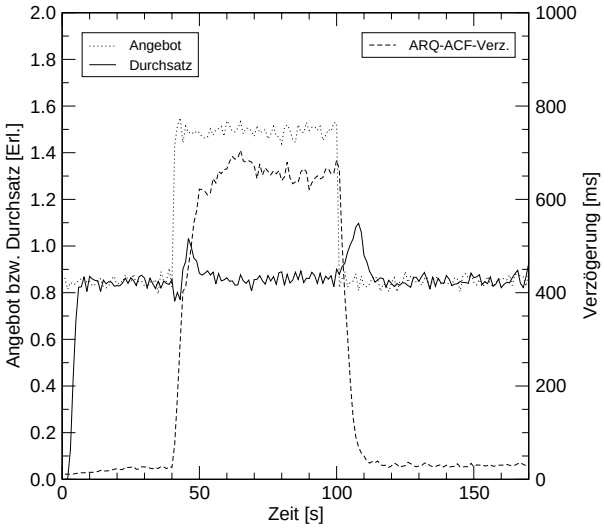


Bild 5.13: Durchsatz und Antwortverzögerung für Lastindikator NOR und Überlastabwehrmaßnahme „Leaky Bucket“

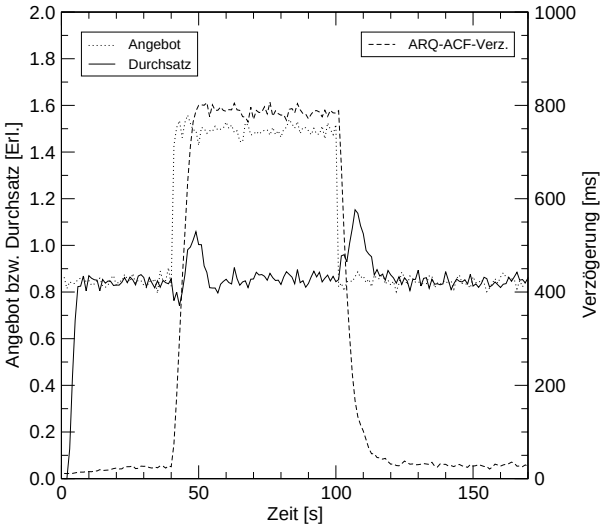


Bild 5.14: Durchsatz und Antwortverzögerung für Lastindikator QL und Überlastabwehrmaßnahme „Leaky Bucket“

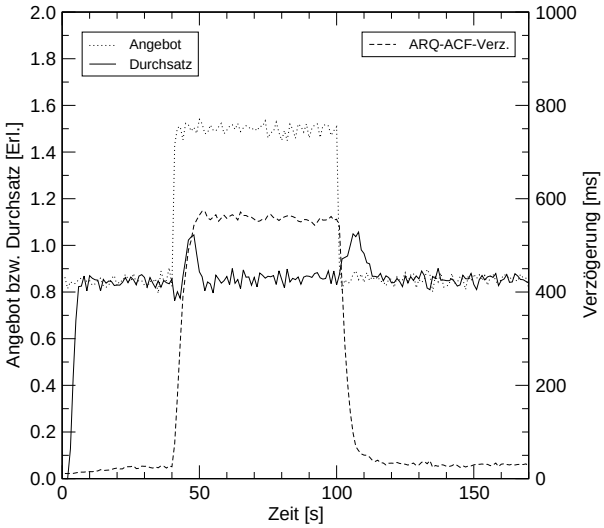


Bild 5.15: Durchsatz und Antwortverzögerung für Lastindikator WCS und Überlastabwehrmaßnahme „Leaky Bucket“

Bild 5.12 stellt die entsprechenden Werte für den Fall, dass keine Überlastabwehr durchgeführt wird, dar. Dabei sind die im Vergleich zu den Bildern 5.13 bis 5.15 veränderten Skalierungen der Abszissen- sowie der Ordinatenachse für die Antwortverzögerung zu beachten. Das langsame Abfallen der Antwortverzögerung zeigt wiederum das schon erwähnte Verhalten des Gatekeepers, vergleichsweise langsam den Normalzustand zu erreichen.

Das Verhalten bei der Verwendung der verschiedenen Lastindikatoren, die gemeinsam mit der Überlastabwehrmaßnahme „Leaky Bucket“ verwendet werden, ist in den Bildern 5.13 bis 5.15

dargestellt. Dabei ist beim Vergleich der Durchsatzverläufe kaum ein Unterschied festzustellen. Als Reaktion auf den Lastsprung nach 40 Sekunden fällt der Durchsatz ein wenig ab, steigt dann aber über den entsprechenden Wert der Basislast an und erreicht bei ca. 60 Sekunden dem diesen Angebot entsprechenden stationären Zustand. Nach Beendigung des Überlastimpulses nach 100 Sekunden steigt der Durchsatz wiederum kurz an, so dass die Verbindungsanforderungen, die sich noch in der Eingangswarteschlange des Gatekeepers befinden, abgearbeitet werden. Der Gatekeeper erreicht schon nach ca. 120 Sekunden seinen Normallastzustand im Vergleich zu den 190 Sekunden ohne Überlastabwehr.

Bei der Untersuchung der Antwortverzögerung für die drei Lastindikatoren ergeben sich während der Überlastphase Unterschiede. Dabei wird mit dem Lastindikator „Gewichtete Verbindungszustände“ die geringste Antwortverzögerung erzielt. Die für die einzelnen Lastindikatoren beobachteten Antwortverzögerungen entsprechen denen, die bei den Untersuchungen des stationären Verhaltens für ein Angebot von 1.5 Erlang (vgl. Bild 5.7) ermittelt wurden. Darüber hinaus ist noch anzumerken, dass es bei der Anwendung von Überlastabwehrmaßnahmen zu keinen fehlschlagenden Verbindungsanforderungen während des Rechteckimpulses kommt.

5.1.2.2 Untersuchung von Überlastabwehrmaßnahmen

Im Folgenden werden die Überlastabwehrmaßnahmen „Automatic Call Gapping“ (ACG), „Leaky Bucket“ (LB), „Prozentuale Drosselung“ (PT), „Token-Pool Leaky Bucket“ (TB) und „Fenster“-Verfahren (WIN) sowohl für den stationären als auch für den instationären Fall untersucht. Als Lastindikator wird dabei jeweils die „Warteschlangenlänge“ bei gleicher Konfiguration verwendet. Ebenso wie in Abschnitt 5.1.2.1 werden keine Anfragen für zusätzliche Dienste durchgeführt.

Untersuchung des stationären Verhaltens

Die wesentlichen Ergebnisse der Untersuchungen für den stationären Fall sind in den Bildern 5.16 bis Bild 5.18 dargestellt. Wie aus den Bildern 5.16 und 5.17 deutlich wird, sind die Unterschiede bezüglich des mittleren Durchsatzes zwischen den verschiedenen Überlastabwehrmaßnahmen so gering, dass keine eindeutige Aussage getroffen werden kann, welches der Verfahren am besten diesbezüglich geeignet wäre. Jedoch erzielen alle der untersuchten Verfahren einen deutlich höheren mittleren Durchsatz als im Fall ohne Überlastabwehr. Ansonsten gelten die in Abschnitt 5.1.2.1 gemachten Bemerkungen für den Durchsatz im stationären Fall.

Bild 5.18 enthält die mittlere ARQ-ACF-Verzögerung über dem Angebot für die verschiedenen Überlastabwehrmaßnahmen. Bei allen Maßnahmen wird die maximale Verzögerung von 1000 ms bei 100% Überlast eingehalten. Dabei wird mit dem ACG-Verfahren die geringste Verzögerung erzielt, während bei „Leaky Bucket“ und „Token-Pool Leaky Bucket“ die Verzögerungen am größten sind.

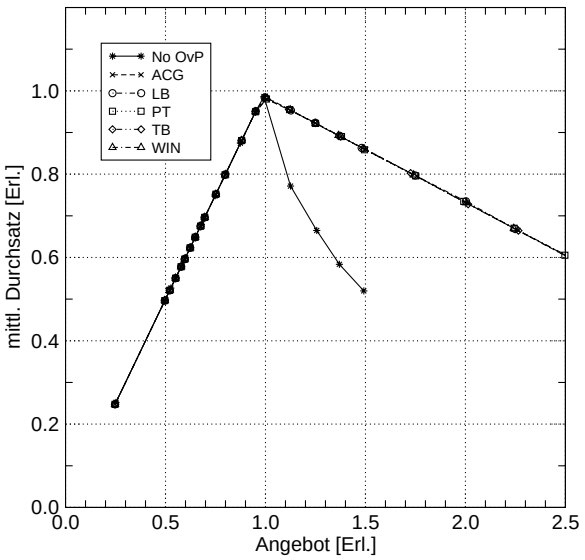


Bild 5.16: Vergleich des Durchsatzes

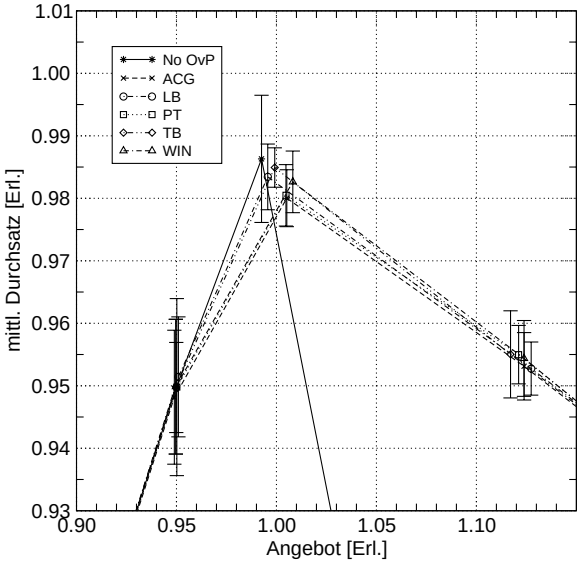


Bild 5.17: Vergleich des Durchsatzes - Detailansicht

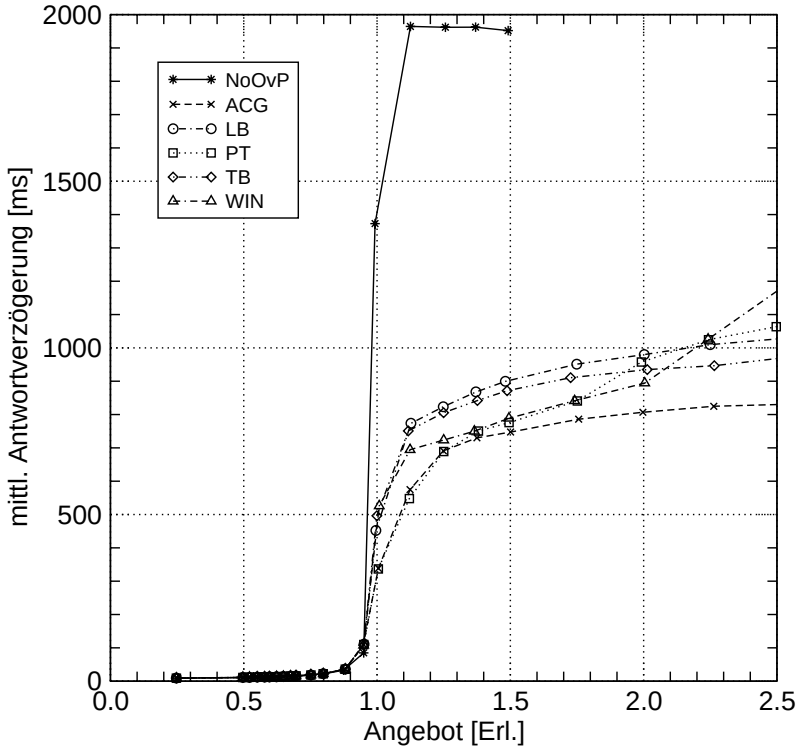


Bild 5.18: ARQ-ACF-Verzögerungen für verschiedene Überlastabwehrmaßnahmen

Des Weiteren wurden die Wahrscheinlichkeiten für das Fehlschlagen von Verbindungsanforderungen sowie die Ablehnungswahrscheinlichkeit zwischen den einzelnen Verfahren verglichen, wobei sich keine nennenswerten Unterschiede ergaben. Bei allen untersuchten Überlastabwehrmaßnahmen wurde die Forderung eingehalten, dass bis zu einer Überlast von 100% keine Verbindungsanforderungen fehlschlagen dürfen.

Untersuchung des instationären Verhaltens

Ebenso wie für die Untersuchungen der Lastindikatoren wurde für die folgenden Studien der in Abschnitt 5.1.2.1 beschriebene Rechteckimpuls als Lastprofil angelegt. Bei den Bildern 5.19 bis 5.23 wird jeweils der zeitliche Verlauf des Angebots und des Durchsatzes sowie der ARQ-ACF-Verzögerung für die verschiedenen Überlastabwehrmaßnahmen dargestellt.

Bei der Betrachtung des Verlaufs des Durchsatzes ist das Verhalten für die einzelnen Überlastabwehrmaßnahmen nahezu gleich. Die dabei auftretenden Effekte wurden bereits bei der entsprechenden Vorstellung der Ergebnisse für verschiedene Lastindikatoren beschrieben. Es ist darüber hinaus festzustellen, dass durch die Anwendung der untersuchten Überlastabwehrmaßnahmen jeweils entsprechend schnell auf Laständerungen reagiert wird. So wird ca. 20 Sekunden nach der jeweiligen Laständerung wieder ein stabiler Zustand erreicht.

Der Vergleich der ARQ-ACF-Verzögerungen zeigt kleinere Unterschiede zwischen den Verfahren auf, wobei insbesondere beim „Token-Pool Leaky Bucket“ Verfahren die Antwortzeiten während der Überlastphase etwas höher als bei den anderen untersuchten Überlastabwehrmaßnahmen sind. Bei ACG werden wiederum wie bei den stationären Betrachtungen die niedrigsten Antwortverzögerungen beobachtet.

Bei den durchgeführten Studien für den Rechteckimpuls werden bei Anwendung einer Überlastabwehrmaßnahme keine fehlgeschlagenen Verbindungsanforderungen beobachtet.

5.1.2.3 Auswirkungen zusätzlicher Dienste

In den voran gegangenen Abschnitten bestand der erzeugte Signalisierverkehr ausschließlich aus Nachrichten für den Verbindungsauf- und -abbau. Im Folgenden werden die Auswirkungen durch zusätzliche Dienste untersucht, die eine Änderung der Verkehrscharakteristika bewirken. Dabei wird der in Bild 4.5 dargestellte generische Nachrichtenablauf für die Bearbeitung zusätzlicher Dienste nachgebildet.

Für diese zusätzlichen Signalisierprozeduren werden folgende Eigenschaften festgelegt:

- Zusätzliche Dienste werden durch A-Endpunkte initiiert. Die Anzahl beantragter zusätzlicher Dienste (*Supplementary Service* – SS) innerhalb einer Verbindung folgt einer geometrischen Verteilung mit dem Mittelwert $\overline{SS_{num}}$.
- Die jeweilige Zeit vom Eintreten in den Verbindungszustand bis zur Anforderung eines zusätzlichen Dienstes durch einen A-Endpunkt ist negativ-exponentiell verteilt.
- Die vollständige Bearbeitungszeit für einen zusätzlichen Dienst im Gatekeeper entspricht der mittleren Bearbeitungszeit für einen vollständigen Verbindungsaufbau.

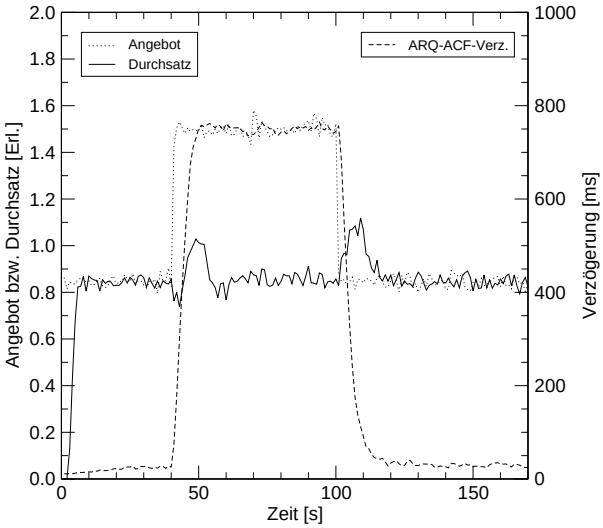


Bild 5.19: Durchsatz und Antwortverzögerung für Überlastabwehrmaßnahme ACG

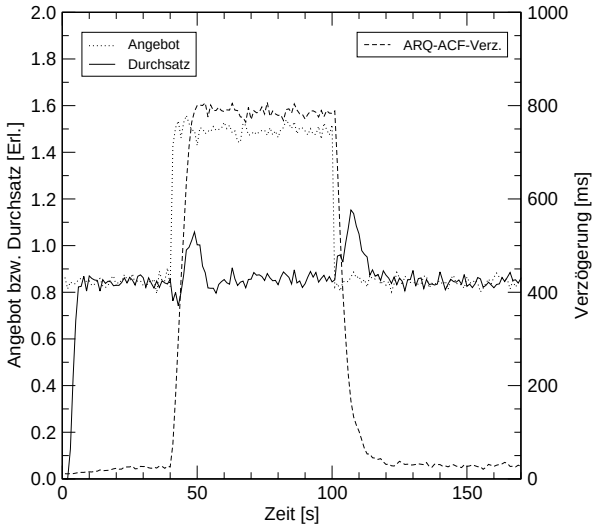


Bild 5.20: Durchsatz und Antwortverzögerung für Überlastabwehrmaßnahme LB

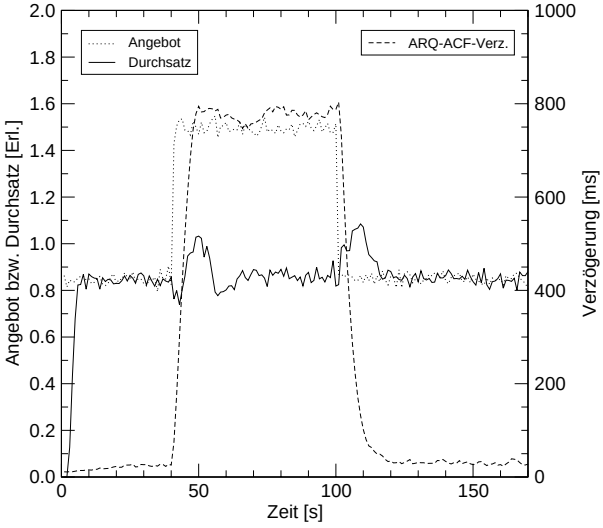


Bild 5.21: Durchsatz und Antwortverzögerung für Überlastabwehrmaßnahme PT

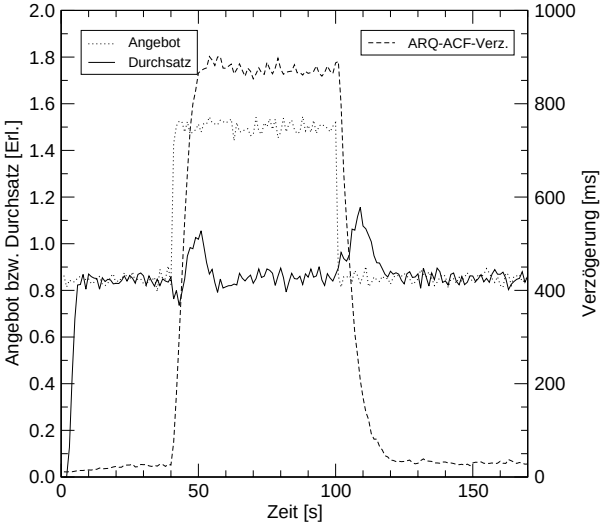


Bild 5.22: Durchsatz und Antwortverzögerung für Überlastabwehrmaßnahme TB

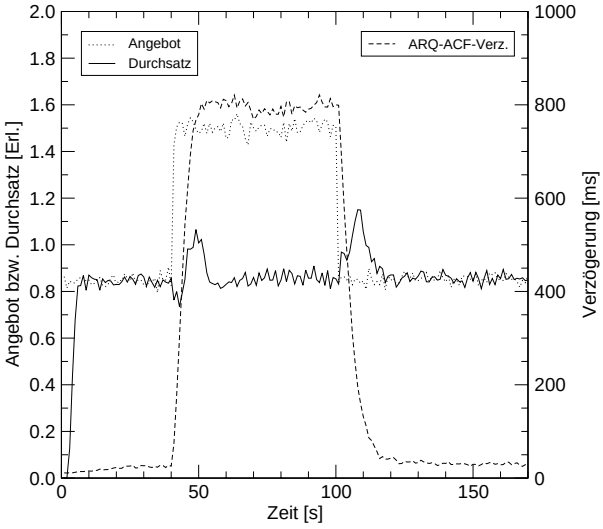


Bild 5.23: Durchsatz und Antwortverzögerung für Überlastabwehrmaßnahme WIN

Durch diese zusätzlichen Anfragen innerhalb aufgebauter Verbindungen, steigt der Ressourcenverbrauch pro Verbindung an, was beispielsweise zu einem niedrigeren maximalen Durchsatz eines Gatekeepers führt. In den folgenden Abbildungen wird jedoch weiterhin das Angebot und der mittlere Durchsatz bezüglich des Falls ohne zusätzliche Dienstanfragen dargestellt.

In Bild 5.24 ist jeweils der mittlere Durchsatz für verschiedene $\overline{SS_{num}}$-Werte über dem Angebot dargestellt, wobei jeweils eine Kurve ohne Durchführung einer Überlastabwehrmaßnahme (No OvP) und eine mit Durchführung einer Überlastabwehrmaßnahme (OvP) ermittelt wurde. Dabei wird deutlich, dass der maximale Durchsatz bei höheren $\overline{SS_{num}}$-Werten und somit einer höheren Anzahl von Anfragen für zusätzliche Dienste geringer wird. Des Weiteren wird die Wirksamkeit von Überlastabwehrmaßnahmen gezeigt, durch die zwar jeweils kein höherer maximaler Durchsatz erzielt wird, jedoch wird der Durchsatz auch in den höheren Lastbereichen jeweils entsprechend stabil gehalten. Darüber hinaus schlägt bei Anwendung von Überlastabwehrmaßnahmen keine der Verbindungsanforderungen fehl.

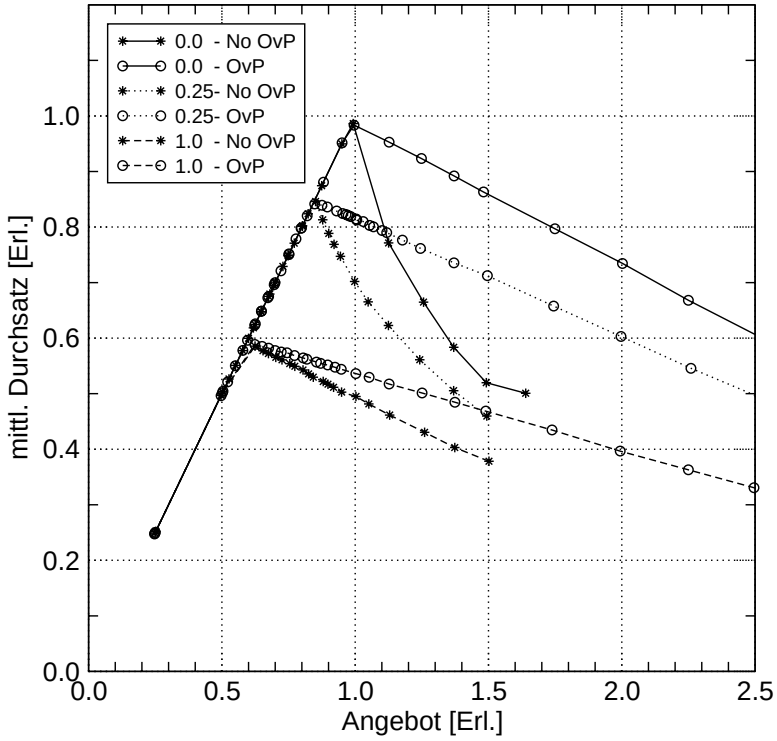


Bild 5.24: Durchsatz bei unterschiedlichen $\overline{SS_{num}}$-Werten über dem Angebot

Wie in den Bildern 5.25 und 5.26 dargestellt, ergeben sich für höhere $\overline{SS_{num}}$-Werte deutlich höhere mittlere ARQ-ACF-Verzögerungen. In Bild 5.25 ist dabei die ARQ-ACF-Verzögerung bei der Anwendung des Lastindikators „Warteschlangenlänge“ und in Bild 5.26 beim Lastindikator „Gewichtete Verbindungszustände“ dargestellt. Für beide Lastindikatoren gilt, dass bereits bei einer mittleren Anzahl von 0.25 zusätzlichen Dienstanfragen pro Verbindung der Wert von 1000 ms bei einem Angebot von 2.0 Erlang überschritten ist. Des Weiteren ist kein wesentlicher Unterschied im Verhalten der beiden Lastindikatoren bezüglich der $\overline{SS_{num}}$-Werte festzustellen.

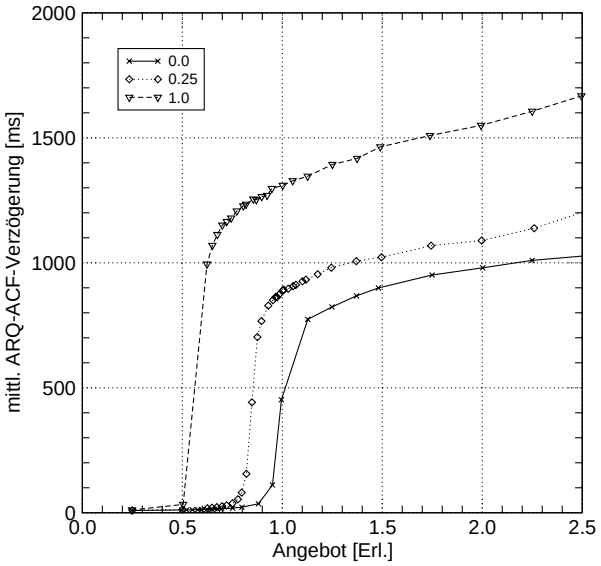


Bild 5.25: Antwortverzögerung für unterschiedliche $\overline{SS_{num}}$-Werte bei Lastindikator QL

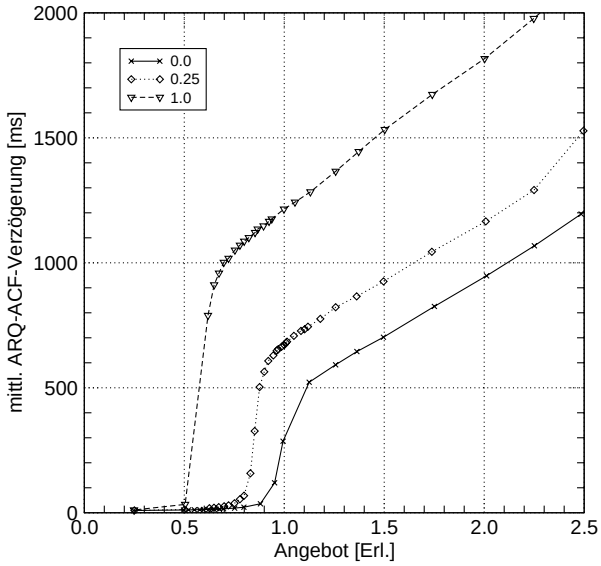


Bild 5.26: Antwortverzögerung für unterschiedliche $\overline{SS_{num}}$-Werte bei Lastindikator WCS

Die Ursache für die höheren Antwortzeiten liegt an dem gestiegenen Ressourcenverbrauch für die einzelnen Verbindungen. Dadurch sind die Voraussetzungen für die Abstimmung zwischen Lastindikatoren und Überlastabwehrmaßnahme nicht mehr gegeben. Um jedoch die Antwortzeiten auch für diese Fälle möglichst einzuhalten, kann ein Anpassungsfaktor f_a bestimmt werden, indem der Basisressourcenverbrauch R_{base} mit dem tatsächlichen mittleren Ressourcenverbrauch pro Verbindung $R_{effective}$, der fortlaufend im Gatekeeper bestimmt werden kann, ins Verhältnis gesetzt wird:

$$f_a = \frac{R_{base}}{R_{effective}} \tag{5.1}$$

Bei der Bestimmung der Lastzustände des Gatekeepers werden dann anstatt der Konfigurationswerte, die für den Fall ohne zusätzliche Dienstanfragen vorgesehen sind, die angepassten Werte verwendet, die jeweils durch die Multiplikation mit dem Anpassungsfaktor ermittelt werden. Durch fortlaufende Anpassung von f_a kann sich der Gatekeeper somit auf Änderungen der Verkehrscharakteristika, wie sie durch zusätzliche Dienste entstehen können, einstellen. In den Bildern 5.27 und 5.28 sind jeweils die ARQ-ACF-Verzögerungen über dem Angebot bei der Anwendung der Lastindikatoren „Warteschlangenlänge“ und „Gewichtete Verbindungsanfragen“ mit und ohne Anwendung des Anpassungsfaktors f_a enthalten. Dabei wird deutlich, dass die Antwortverzögerungen für beide Lastindikatoren deutlich niedriger als ohne Anwendung des Anpassungsfaktors ausfallen. Jedoch sind sie im Überlastbereich etwas höher als im Fall ohne zusätzliche Dienstanfragen.

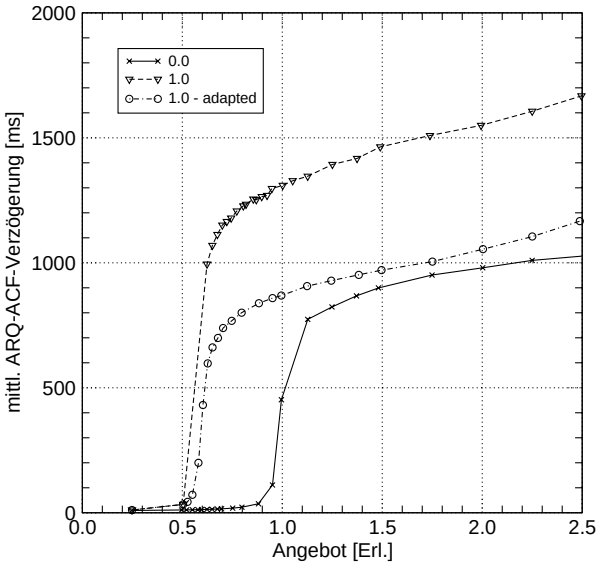


Bild 5.27: Antwortverzögerung für unterschiedliche $\overline{SS_{num}}$-Werte bei angepasstem Lastindikator QL

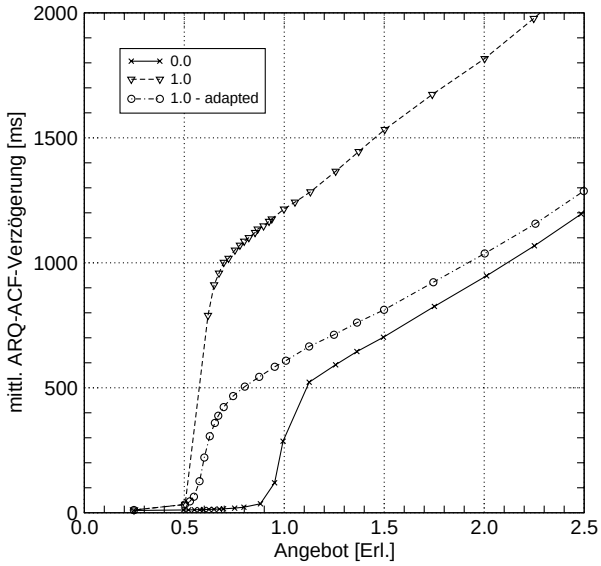


Bild 5.28: Antwortverzögerung für unterschiedliche $\overline{SS_{num}}$-Werte bei angepasstem Lastindikator WCS

5.1.3 Bewertung

Die Untersuchungen, die mit einer prototypischen Implementierung von Überlastabwehrmaßnahmen durchgeführt wurden, sowie die stationären und instationären Simulationsstudien zeigen die Wirksamkeit der untersuchten Überlastabwehrmaßnahmen. Dabei sind insbesondere zu nennen:

- Erhöhung des Durchsatzes im Überlastfall,
- Begrenzung der Antwortzeiten,
- Verhinderung des Fehlschlagens von Verbindungsanforderungen.

Damit kann eine hohe Stabilität des Gatekeepers erzielt werden, so dass auch in Hoch- und Überlastsituationen seine Diensterbringung sichergestellt ist. Des Weiteren kann festgestellt werden, dass durch die untersuchten Lastindikatoren und Überlastabwehrmaßnahmen eine ausreichende Wirkgeschwindigkeit erzielt wird, so dass zum einen Überlastsituationen schnell erkannt werden und somit entsprechend reagiert werden kann. Zum anderen halten Überlastsituationen nicht länger als notwendig an, so dass nach dem Rückgang des Angebots auf Normallast die Überlastung des Gatekeepers schnell zurück geht und er somit nach relativ kurzer Zeit wieder den Normallastzustand einnimmt. Somit sind die Ergebnisse dieser Studien vergleichbar mit denen, die für die klassische Telekommunikation durchgeführt wurden.

Bei der Betrachtung der Ergebnisse für die untersuchten Lastindikatoren „Anzahl offener Anfragen“, „Warteschlangenlänge“ und „Gewichtete Verbindungszustände“ ergeben sich kaum Unterschiede. Der erzielte Durchsatz ist nahezu gleich und auch die Forderung für die

Antwortverzögerung sowie für fehlschlagende Verbindungsanforderungen bis zur einer Überlast von 100% werden jeweils eingehalten. Auch die Untersuchungen des instationären Verhaltens sowie die Studien, die die Auswirkungen durch zusätzliche Dienste ermitteln, zeigen keine gravierenden Unterschiede zwischen den einzelnen Lastindikatoren. Wenn man jedoch den Aufwand während der Durchführung der Lastzustandsbestimmung betrachtet, ergeben sich für den Lastindikator „Warteschlangenlänge“ Vorteile, da er sehr einfach zu bestimmen ist. Des Weiteren werden keine weiteren Informationen benötigt, wie z. B. der aktuelle Verbindungszustand, um festzustellen, dass eine neue Verbindungsphase begonnen hat. Schließlich ist der Aufwand für die Parametrisierung des Lastindikators „Warteschlangenlänge“ geringer, da keine zusätzlichen Parameter, wie z. B. Verbindungszustandsgewichte, entsprechend konfiguriert werden müssen.

Ebenso wie bei den Lastindikatoren sind die Unterschiede zwischen den untersuchten Überlastabwehrmaßnahmen sehr gering, wobei die gestellten Forderungen bezüglich der Antwortzeiten und der fehlschlagenden Verbindungsanforderungen bis zu einer Überlast von 100% jeweils eingehalten werden. Auch bezüglich des instationären Verhaltens ergeben sich für die einzelnen Maßnahmen wenig Unterschiede, so dass alle untersuchten Verfahren zur Überlastabwehr eingesetzt werden können. Bei der Betrachtung des Aufwands für die Durchführung der einzelnen Maßnahmen ergeben sich leichte Nachteile für das „Leaky Bucket“ und das „Token-Pool Leaky Bucket“ Verfahren, da dort jeweils eine Zwischenspeicherung der Verbindungsanforderungsnachrichten durchgeführt wird. Beim „Fenster“-Verfahren muss dagegen das Ende einer Anfrage erkannt werden, was sowohl durch vollständiges Bearbeiten oder Ablehnung als auch durch Fehlschlagen der Anfrage erfolgen kann. Bei ACG und der „Prozentualen Drosselung“ ist der Aufwand während der Durchführung der Maßnahmen am geringsten. Des Weiteren ist die Parametrisierung dieser beiden Verfahren sowie für das „Fenster“-Verfahren weniger aufwendig, da nur wenige Parameter eingestellt werden müssen.

Bei der Untersuchung der Auswirkungen durch die Bearbeitung zusätzlicher Dienste zeigt sich, dass die Überlastabwehr auch in diesen Fällen bezüglich des Durchsatzes wirksam ist. Jedoch werden die Antwortzeiten beträchtlich größer, wenn die mittlere Anzahl von zusätzlichen Dienstanfragen entsprechend höher ist. Daher wird ein Anpassungsfaktor für die Bestimmung des aktuellen Lastzustands vorgeschlagen, der den Ressourcenverbrauch pro Verbindung des Gatekeepers für die konfigurierte Last mit dem tatsächlichen Ressourcenverbrauch ins Verhältnis setzt. Der tatsächliche Ressourcenverbrauch könnte dabei durch fortlaufende Messungen im Gatekeeper ermittelt werden. Damit kann der Anpassungsfaktor fortlaufend aktualisiert werden, so dass auch auf sich dynamisch ändernde Verkehrscharakteristika entsprechend reagiert werden kann. Die Anwendung dieses Anspassungsfaktors zeigt eine deutliche Verbesserung der Antwortverzögerungen, sowohl beim Lastindikator „Warteschlangenlänge“ als auch beim Lastindikator „Gewichtete Verbindungszustände“.

5.2 Steuerungsoptimierung eines Gatekeeper-Clusters

In diesem Abschnitt werden die Ergebnisse der Studien für die Steuerungsoptimierung eines Gatekeeper-Clusters, wie er in Abschnitt 3.5.2 eingeführt wurde, vorgestellt. Insbesondere werden dabei die Intrazonen-Lastverteilungsverfahren eines Gatekeeper-Clusters untersucht. Die Studien werden mittels stationärer und instationärer Simulationen durchgeführt.

Die Untersuchungen bezüglich der Granularität der Lastverteilung werden im folgenden Abschnitt 5.2.1 durchgeführt. Anschließend werden in Abschnitt 5.2.2 Studien für verschiedene Lastverteilungsverfahren ohne Anwendung von Überlastabwehrmaßnahmen in den Gatekeepern präsentiert. Die Untersuchungen für die gemeinsame Anwendung von Lastverteilungs- und Überlastabwehrverfahren werden in Abschnitt 5.2.3 vorgestellt. In Abschnitt 5.2.4 erfolgt schließlich die Bewertung der Ergebnisse dieser Studien.

5.2.1 Granularität der Lastverteilung

Zur Untersuchung der Granularität der Intrazonen-Lastverteilung wird ein Gatekeeper-Cluster betrachtet, der aus einem Dispatcher, der die Steuerung der Lastverteilung durchführt, und drei Gatekeepern besteht. Die Gatekeeper verfügen jeweils über die gleiche Leistungsfähigkeit und es wird keine Überlastabwehr durchgeführt. Als Lastverteilungsverfahren wird das „Round-Robin“-Verfahren angewandt, wobei Nachrichten von den Gatekeepern direkt an die Endpunkte gesendet werden, so dass diese Nachrichten nicht über den Dispatcher geführt werden.

Die Modellierung der Verwaltung der Zustandsdaten erfolgt, indem für das Lesen und Schreiben der Zustandsdaten jeweils zusätzliche Bearbeitungszeiten simuliert werden. Je nach Granularität der Lastverteilung fallen diese zusätzlichen Bearbeitungszeiten für jede Nachricht, jede Verbindungsphase oder einmal pro Verbindung an. Diese Bearbeitungszeiten werden im Verhältnis zur Bearbeitungszeit für einen vollständigen Verbindungsaufbau betrachtet. Dieses Verhältnis wird im weiteren Verlauf als *Zustandsdaten-Aufwand* bezeichnet. So bedeutet ein Zustandsdaten-Aufwand von 10%, dass das Lesen und Schreiben der Zustandsdaten 10% der Bearbeitungszeit eines vollständigen Verbindungsaufbaus benötigt.

In Bild 5.29 ist der mittlere Durchsatz über dem Angebot für verschiedene Granularitäten der Lastverteilung inklusive der Vertrauensintervalle dargestellt. Diese Ergebnisse wurden bei einem Zustandsdaten-Aufwand von 10% und ohne zusätzliche Dienstanfragen gewonnen. Aus diesen Ergebnissen wird deutlich, dass mit der Verteilung auf Verbindungsebene der höchste maximale Durchsatz erzielt wird, während er bei der Verteilung auf Verbindungsphasenebene etwas und auf Nachrichtenebene deutlich geringer ist. Die Ursache liegt im Aufwand für die Verwaltung der Zustandsdaten: Bei der Lastverteilung auf Verbindungsebene müssen die Zustandsdaten einmal pro Verbindung gelesen und geschrieben werden, während sie bei der

Verteilung auf Nachrichtenebene für jede ankommende Signalisiernachricht gelesen und geschrieben werden. Im Laufe einer Verbindung kommen ca. 9 Signalisiernachrichten beim Gatekeeper an, so dass bei einem Zustandsdaten-Aufwand von 10% nahezu der Aufwand für einen vollständigen Verbindungsaufbau (ca. 90%) für die Verwaltung der Zustandsdaten bei der Lastverteilung auf Nachrichtenebene zusätzlich benötigt wird. Bei der Lastverteilung auf Verbindungsphasenebene werden die Zustandsdaten für jede Verbindungsphase gelesen und geschrieben, so dass der Aufwand zwar größer als bei der Verteilung auf Verbindungsebene ist, da eine Verbindung zumindest aus einer Verbindungsaufbau- und einer Verbindungsabbauphase besteht, jedoch ist er deutlich geringer als bei der Verteilung auf Nachrichtenebene.

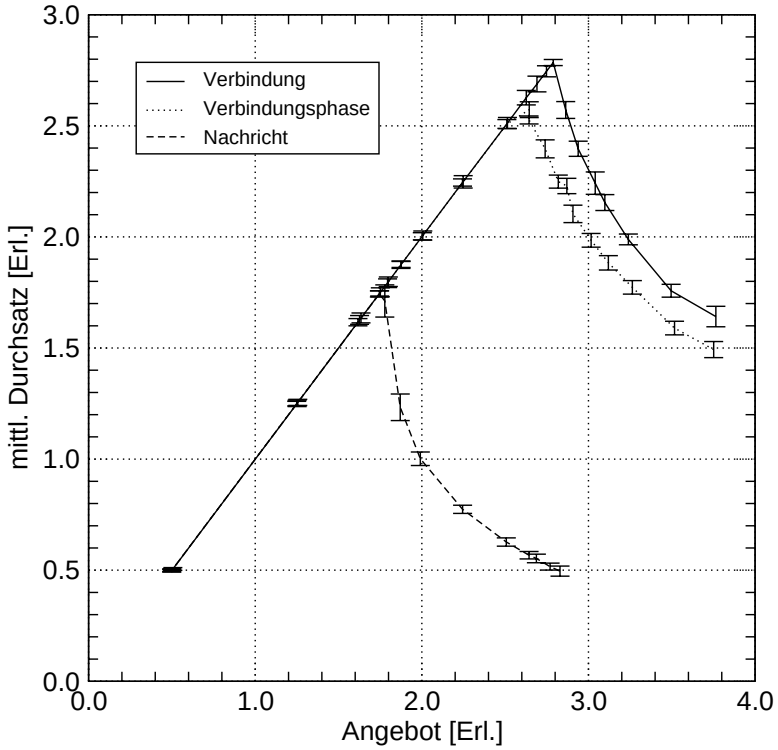


Bild 5.29: Durchsatz bei verschiedenen Verteilungsgranularitäten (Zustandsdaten-Aufwand 10%, $\overline{SS_{num}}$ = 0.0)

Bild 5.30 enthält die zum vorigen Abschnitt entsprechenden Ergebnisse für einen reduzierten Zustandsdaten-Aufwand von 1%. Dabei wird deutlich, dass für alle Lastverteilungsgranularitäten jeweils ein höherer maximaler Durchsatz erzielt wird. Darüber hinaus werden die Unterschiede beim Durchsatz zwischen den Granularitäten deutlich geringer, was sich durch den niedrigeren Aufwand für die Zustandsdatenverwaltung ergibt. In Bild 5.31 ist eine Detailansicht von Bild 5.30 um den Wert des maximalen Durchsatzes enthalten. Dabei kann erkannt werden, dass der Unterschied zwischen der Lastverteilung auf Verbindungsebene und der auf Verbindungsphasenebene sehr gering bzw. teilweise gar nicht mehr vorhanden ist. Das Verschwinden der Unterschiede kann durch eine etwas gleichmäßigere Auslastung der einzelnen Cluster-Mitglieder bei der Verbindungsphasengranularität begründet werden.

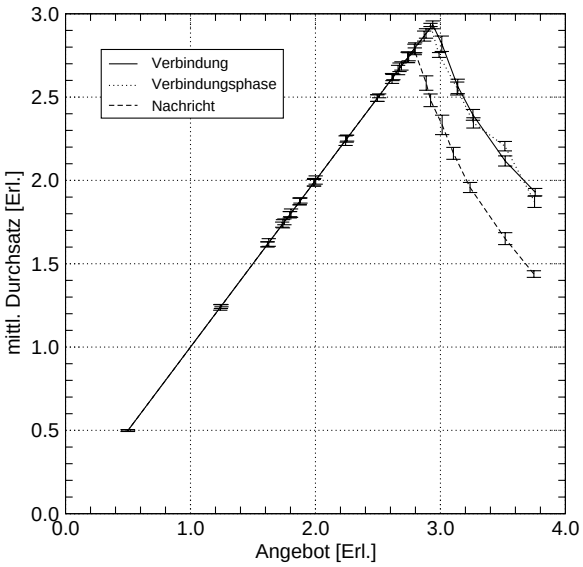


Bild 5.30: Durchsatz bei verschiedenen Verteilungsgranularitäten (Zustandsdaten-Aufwand 1%, $\overline{SS_{num}}$ = 0.0)

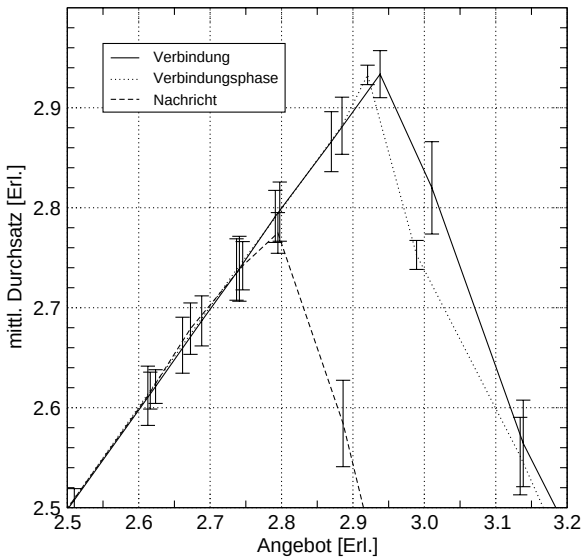


Bild 5.31: Durchsatz bei verschiedenen Verteilungsgranularitäten - Detailansicht (Zustandsdaten-Aufwand 1%, $\overline{SS_{num}}$ = 0.0)

Um die Auswirkungen der Bearbeitung zusätzlicher Dienste festzustellen, ist in Bild 5.32 der mittlere Durchsatz über dem Angebot für verschiedene Verteilungsgranularitäten für eine mittlere Anzahl von zusätzlichen Dienstanfragen pro Verbindung von 1.0 dargestellt. Der Aufwand für die Zustandsdatenverwaltung beträgt dabei 1%. Neben dem jeweils deutlich geringeren maximalen Durchsatz im Vergleich zum Fall ohne zusätzliche Dienstanfragen kann erkannt werden, dass die Unterschiede zwischen der Lastverteilung auf Nachrichtenebene und der auf Verbindungsebene im Vergleich zum Szenario ohne zusätzliche Dienstanfragen geringer sind. Zur Bewertung der Unterschiede ist in Bild 5.33 ein Ausschnitt von Bild 5.32 enthalten. Dabei wird wiederum deutlich, dass die Unterschiede zwischen der Lastverteilung auf Verbindungsebene und der auf Verbindungsphasenebene gering sind, wobei sich in diesem Fall die Vertrauensintervalle jeweils kaum überschneiden, so dass mit der Verteilung auf Verbindungsebene mit großer Wahrscheinlichkeit ein höherer Durchsatz erzielt wird, als mit der Verteilung auf Verbindungsphasenebene.

Für einen Vergleich der Auslastung der Gatekeeper für die verschiedenen Verteilungsgranularitäten wird in den Bildern 5.34 bis 5.39 jeweils ein Ausschnitt des zeitlichen Verlaufs der ARQ-ACF-Verzögerungen eines Simulationslaufs sowohl für den gesamten Cluster als auch für die einzelnen Cluster-Mitglieder (GK 1, GK 2 und GK 3) dargestellt. Bei diesen Untersuchungen wird keine Überlastabwehr durchgeführt und der Mittelwert für die Anzahl zusätzlicher Dienstanfragen pro Verbindung beträgt 1.0. In der linken Spalte ist der zeitliche Verlauf der Antwortverzögerungen für eine 80%-Belastung des Clusters dargestellt, während in der rechten Spalte eine Belastung von ca. 120%, was somit einer Überlastung entspricht, vorliegt. Die obere Reihe enthält die Ergebnisse für die Lastverteilung auf Verbindungsebene, die mitt-

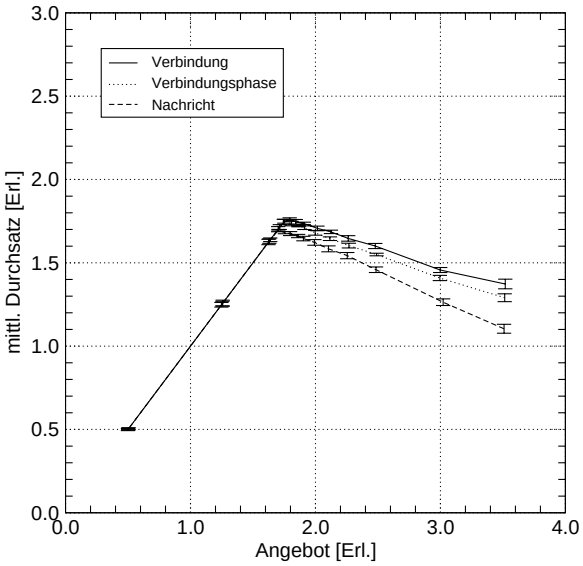


Bild 5.32: Durchsatz bei verschiedenen Verteilungsgranularitäten (Zustandsdaten-Aufwand 1%, $\overline{SS_{num}}$ = 1.0)

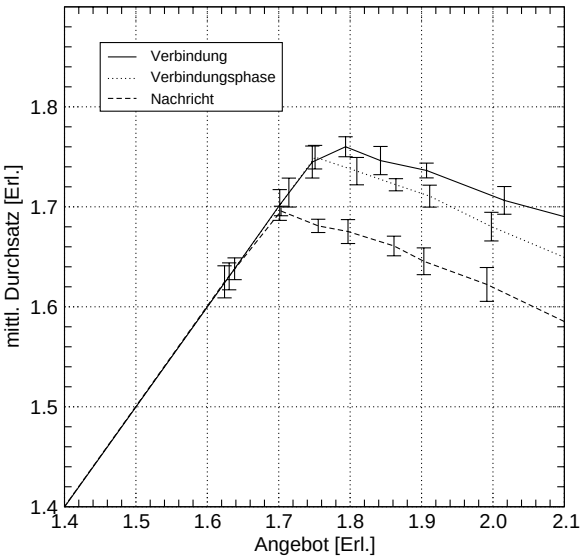


Bild 5.33: Durchsatz bei verschiedenen Verteilungsgranularitäten - Detailansicht (Zustandsdaten-Aufwand 1%, $\overline{SS_{num}}$ = 1.0)

lere für die Verbindungsphasenebene und die untere Reihe für die Verteilung auf Nachrichtenebene.

Beim Vergleich des zeitlichen Verlaufs der Antwortverzögerungen für Normallast (Bilder 5.34, 5.36 und 5.38) lässt sich feststellen, dass bei der Lastverteilung auf Verbindungsebene größere Ausschläge der Antwortverzögerung auftreten als bei der Verteilung auf Verbindungsphasen- und auf Nachrichtenebene. Insbesondere bei der Verteilung auf Verbindungsphasenebene scheinen die Antwortverzögerungen der einzelnen Gatekeeper relativ ähnlich zu sein. Bei der Betrachtung des zeitlichen Verlaufs der Antwortverzögerungen im Überlastfall (Bilder 5.35, 5.37 und 5.39) können ebenfalls Unterschiede festgestellt werden. Dabei ergeben sich wiederum bei der Lastverteilung auf Verbindungsebene größere Schwankungen der Antwortverzögerungen als bei der Verteilung auf Verbindungsphasen- und Nachrichtenebene. Jedoch sind in diesem Fall kaum Unterschiede zwischen der Verbindungsphasen- und der Nachrichtenebene zu erkennen.

Aus diesen zeitlichen Verläufen der Antwortverzögerungen lässt sich schließen, dass bei der Lastverteilung auf Verbindungsebene eine weniger gleichmäßige Belastung der einzelnen Cluster-Mitglieder erreicht wird als bei der Verteilung auf Verbindungsphasen- oder Nachrichtenebene. Jedoch sind die absoluten Werte der auftretenden Schwankungen so klein, dass die Unterschiede der Belastungen nur geringfügig sind und somit keine großen Auswirkungen auf das Gesamtverhalten des Clusters haben.

Damit auch bei hohem Zustandsdaten-Aufwand jeweils der korrekte Lastzustand durch die Gatekeeper angezeigt wird, muss dem gestiegenen Ressourcenverbrauch pro Verbindung Rechnung getragen werden. Dazu kann wiederum der in Abschnitt 5.1.2.3 eingeführte Anpas-

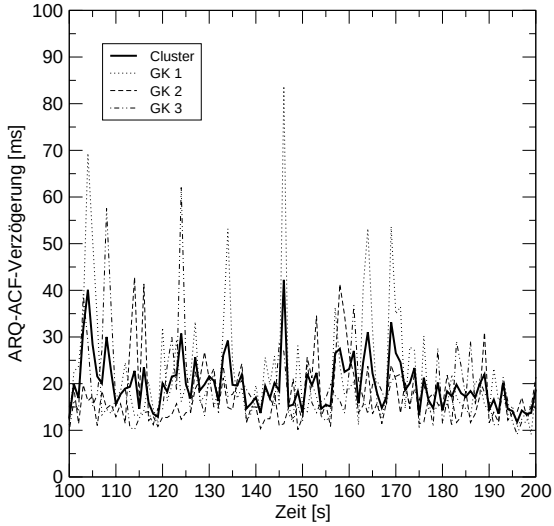


Bild 5.34: Antwortverzögerungsverlauf für Verbindungsgranularität - 80%-Last

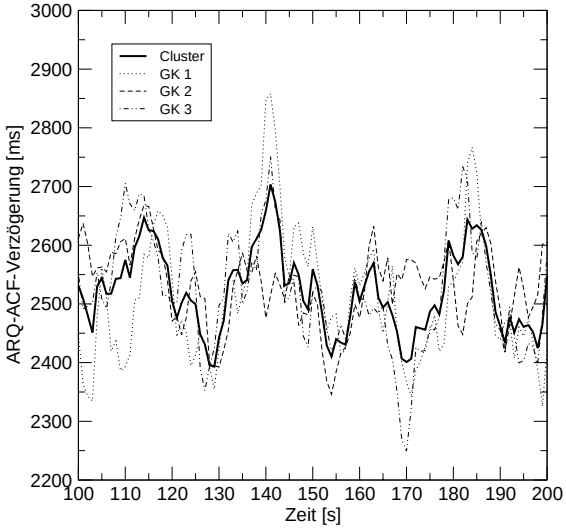


Bild 5.35: Antwortverzögerungsverlauf für Verbindungsgranularität - 120%-Last

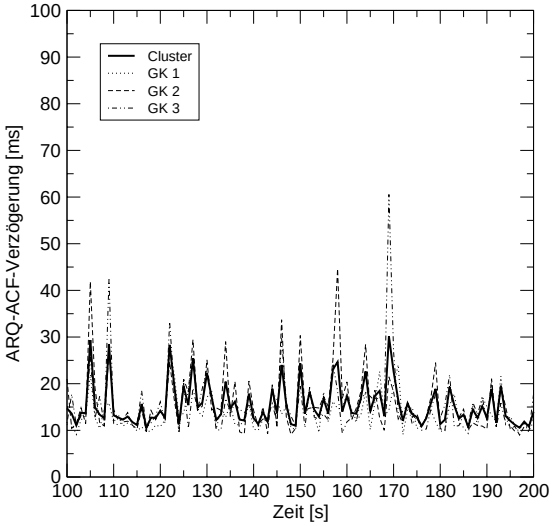


Bild 5.36: Antwortverzögerungsverlauf für Verbindungsphasengranularität - 80%-Last

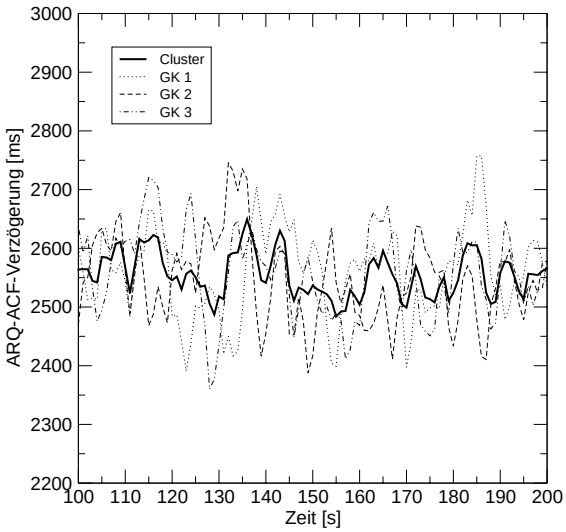


Bild 5.37: Antwortverzögerungsverlauf für Verbindungsphasengranularität - 120%-Last

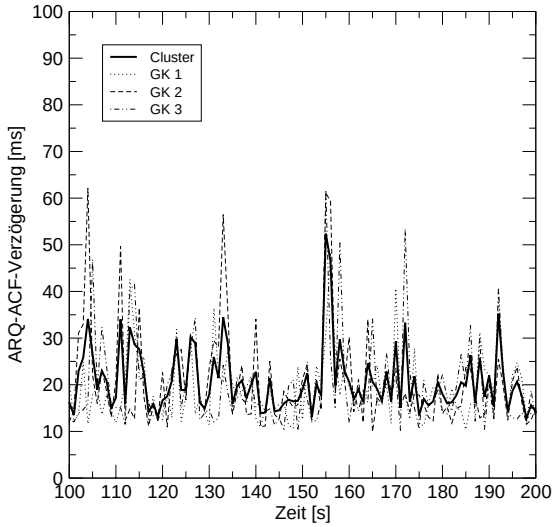


Bild 5.38: Antwortverzögerungsverlauf für Nachrichtengranularität - 80%-Last

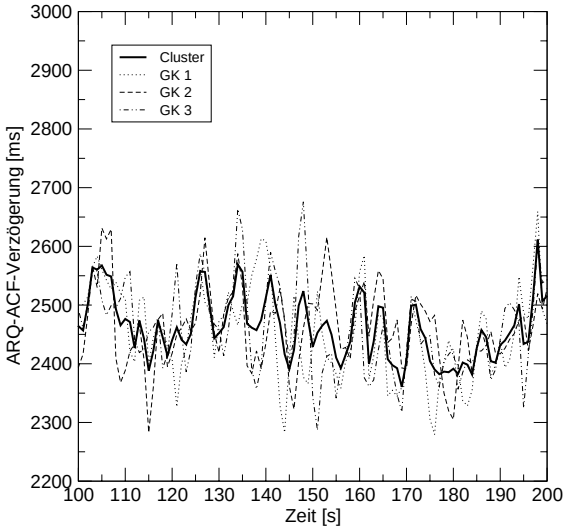


Bild 5.39: Antwortverzögerungsverlauf für Nachrichtengranularität - 120%-Last

sungsfaktor f_a verwendet werden. Dabei werden für die Bestimmung der Lastzustände anstatt der Konfigurationswerte, die für den Fall ohne Zustandsdaten-Aufwand gelten, angepasste Konfigurationswerte verwendet, die durch die Multiplikation des Anpassungsfaktor mit den ursprünglichen Konfigurationswerten bestimmt werden. Da der Anpassungsfaktor nur den Unterschied zwischen tatsächlichem Ressourcenverbrauch für die Verbindungsbearbeitung und dem Ressourcenverbrauch darstellt, der für die Einstellung der Lastzustände eines Gatekeepers verwendet wurde, kann er für die Anpassung sowohl an den Zustandsdaten-Aufwand als auch an die Bearbeitung zusätzlicher Dienste sowie an beides gemeinsam verwendet werden.

5.2.2 Lastverteilung ohne Überlastabwehrmaßnahmen

In diesem Abschnitt werden verschiedene Intrazonen-Lastverteilungsverfahren untersucht, die in Abschnitt 3.5.2.1 beschrieben wurden, wobei zunächst keine Überlastabwehrmaßnahmen in den Gatekeepern des Clusters angewendet werden. Um für die kooperierenden Lastverteilungsverfahren eine Differenzierung der Lastzustände im Niedriglastbereich zu erreichen, wurde dieser Lastbereich weiter unterteilt, so dass für diese Verfahren acht Lastzustände der Gatekeeper unterschieden werden.

- Statische Verteilung der Endpunkte auf die Gatekeeper - *Static*
  Bei diesem Verfahren wird jedem Endpunkt ein Gatekeeper fest zugeordnet. Diese Zuordnung bleibt auch in Hoch- und Überlastsituationen bestehen.

- *Round-Robin*
  Dieses zentral gesteuerte Verfahren verwendet einen Dispatcher zur Lastverteilung. Dabei erfolgt eine zyklische Zuteilung der Anfragen an die Gatekeeper des Clusters.

- Lastzustandsabhängige Verteilung - *Least-Loaded*
  Dieses Verfahren ist zentral gesteuert realisiert. Dabei zeigen die einzelnen Gatekeeper des Clusters ihren aktuellen Lastzustand dem Dispatcher mittels entsprechender Nachrichten an, wobei die Aktualisierung der Lastzustände in Zeitintervallen von einer Sekunde erfolgt. Wenn sich mehrere Gatekeeper im gleichen Lastzustand befinden, wählt der Dispatcher diese jeweils abwechselnd aus, um ungleichmäßigen Belastungen, wie sie durch die Granularität der Lastzustände und durch die Intervalldauern entstehen können, vorzubeugen.

- *Sender-Receiver*-Verfahren
  Bei diesem Verfahren mit verteilter Steuerung erfolgt die Anzeige des aktuellen Lastzustands für die Intrazonen-Lastverteilung (*Sender, Ok, Receiver*) jeweils durch entsprechende Nachrichten nach einer Änderung dieses Lastzustands. Der Zustand *Sender* wird dabei eingenommen, bevor einzelne Verbindungsanforderungen fehlschlagen bzw., bei

Durchführung von Überlastabwehrmaßnahmen (in Abschnitt 5.2.3), bevor Verbindungsanforderungen abgelehnt werden.

Bei den im Folgenden vorgestellten Untersuchungen besteht ein Gatekeeper-Cluster jeweils aus drei Gatekeepern, die über die gleiche Leistungsfähigkeit verfügen und als Lastindikator die „Warteschlangenlänge“ verwenden, und ggf. aus einem Dispatcher. Die Lastverteilung erfolgt auf Verbindungsebene, wobei der Zustandsdaten-Aufwand 1% beträgt. Bei den Lastverteilungsverfahren mit verteilter Steuerung wird jeweils zu Beginn eines Simulationslaufes jedem Endpunkt ein Gatekeeper zugeordnet. Dabei wird jede neue Verbindungsanforderung zunächst an diesen ursprünglichen Gatekeeper gesendet, auch wenn die vorhergehende Anforderung durch einen anderen Gatekeeper bearbeitet wurde. Um die Wirksamkeit der Lastverteilungsverfahren zu bestimmen, erfolgt eine ungleichmäßige Verteilung der Endpunkte auf die Gatekeeper, wobei ein Verhältnis der Anzahl der zugeordneten Endpunkte von 1:2:4 eingestellt wird, so dass Gatekeeper 3 die vierfache Anzahl von Endpunkten zugeordnet ist als Gatekeeper 1. Des Weiteren wurde als Referenz die Verwaltung einer Zone durch einen Gatekeeper, der über die dreifache Leistungsfähigkeit der einzelnen Gatekeeper eines Clusters verfügt, untersucht (*3fach-GK*).

In Bild 5.40 ist der mittlere Durchsatz über dem Angebot für die verschiedenen Intrazonen-Lastverteilungsverfahren dargestellt. Wie zu erwarten war, erzielt der 3-fach-Gatekeeper den höchsten maximalen Durchsatz. Des Weiteren ist der 3-fach-Gatekeeper den zentral gesteuerten Lastverteilungsverfahren im Überlastbereich überlegen. Bei der statischen Lastverteilung fällt der Durchsatz gegenüber den anderen Verfahren bereits bei einem Angebot von ca. 1.7 Erlang ab. Dies ist in der ungleichmäßigen Belastung der Gatekeeper des Clusters begründet, wie aus Bild 5.41 ersichtlich wird, in dem die mittleren Lastzustände der drei Gatekeeper über dem Angebot dargestellt sind. Bei weiter anwachsendem Angebot nähert sich der Durchsatz bei der statischen Lastverteilung dem der zentral gesteuerten Verfahren und dem des 3-fach-Gatekeepers an, fällt aber ab einem Angebot von ca. 3.5 Erlang wieder etwas ab. Die Ursache für dieses Verhalten liegt in der strikten Trennung der Zuordnung der Endpunkte zu den Gatekeepern des Clusters. GK 3 ist ab ca. einem Gesamtangebot von 1.7 Erlang überlastet und da keine Anforderungen an andere Cluster-Mitglieder weitergegeben werden können, schlagen ab diesem Angebot Verbindungsanforderungen für diesen Gatekeeper fehl. Bei GK 2 tritt die Überlastsituation erst ab einem Gesamtangebot von ca. 3.5 Erlang auf, so dass ab diesem Angebot weitere Verbindungsanforderungen fehlschlagen. GK 1 befindet sich jedoch im gesamten untersuchten Angebotsbereichs im Normallastzustand, da er weniger Endpunkte steuert, so dass er deren Verbindungsanforderungen erfolgreich bearbeiten kann.

Ähnlich verhält es sich mit dem „Sender-Receiver“-Verfahren, bei dem der Durchsatz ab einem Angebot von ca. 3.1 Erlang bzw. 3.3 Erlang den Durchsatz bei Anwendung von zentral gesteuerten Lastverteilungsverfahren bzw. des 3-fach-Gatekeepers übersteigt. Die Begründung

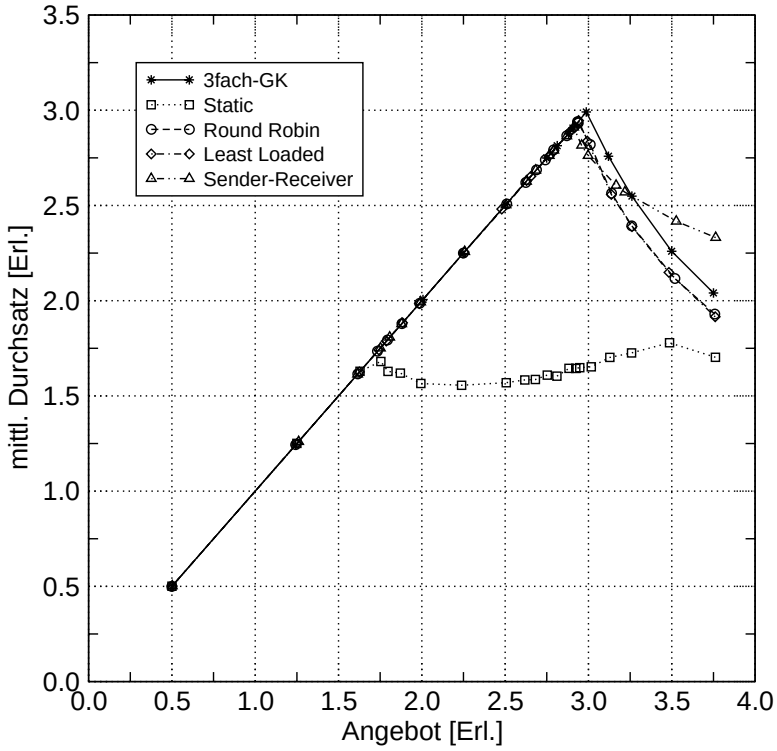


Bild 5.40: Mittlerer Durchsatz für verschiedene Intrazonen-Lastverteilungsverfahren und einen 3-fach-Gatekeeper (ohne Überlastabwehr)

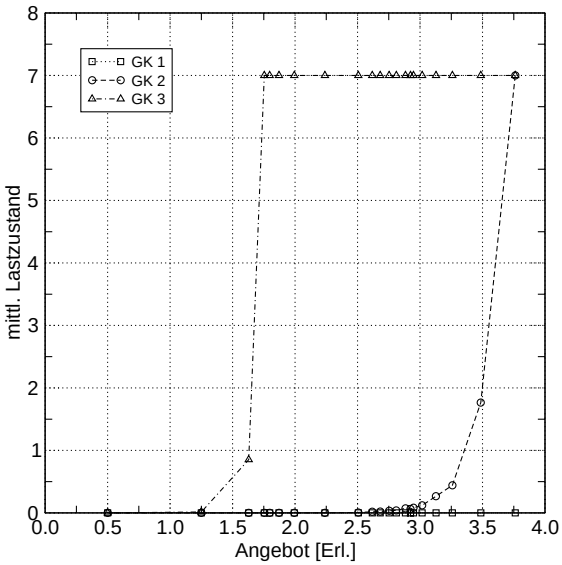


Bild 5.41: Lastzustand über dem Angebot der einzelnen Cluster-Mitglieder bei statischer Lastverteilung (ohne Überlastabwehr)

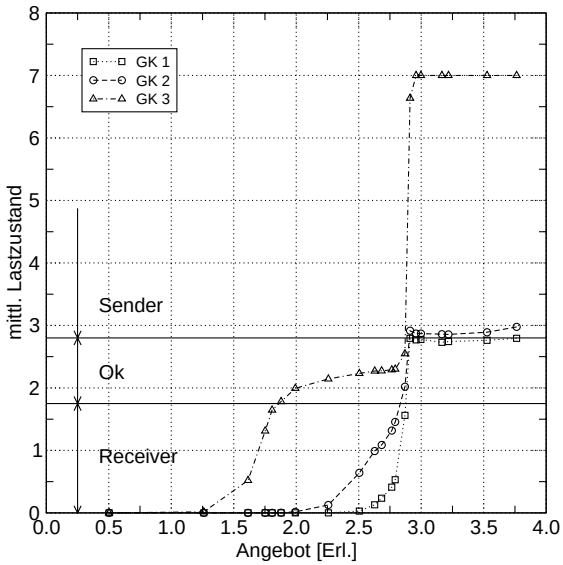


Bild 5.42: Lastzustand über dem Angebot der Cluster-Mitglieder beim „Sender-Receiver“-Verfahren (ohne Überlastabwehr)

dafür ist, dass bei diesem Verfahren Lastverteilung nur durchgeführt wird, wenn lastbearbeitende Komponenten vorhanden sind, deren Lastzustand anzeigt, dass sie noch weitere Anfragen bearbeiten können. Für das untersuchte Szenario bedeutet dies, dass GK 2 und GK 3 nur Verbindungsanfragen an GK 1 weiterleiten, wenn dieser anzeigt, dass er sich im Lastzustand für die Intrazonen-Lastverteilung *Receiver* oder *Ok* befindet. Wenn alle Mitglieder des Clusters

den Zustand *Sender* eingenommen haben, wird keine Lastverteilung durchgeführt, d. h. alle Anfragen werden vom derzeit zuständigen Gatekeeper bearbeitet. Wie in Bild 5.42 dargestellt befinden sich ab einem Angebot von ca. 2.9 Erlang alle Gatekeeper im Mittel im Zustand *Sender*, wobei GK 1 und GK 2 nicht überlastet sind, da ihr Lastzustand unter dem Wert 4 liegt, der eine geringe Überlastung anzeigt. Daher bearbeiten GK 1 und GK 2 trotz der großen Überlastung von GK 3 ankommende Anfragen erfolgreich, so dass der Durchsatz ab dem genannten Angebot höher als bei den anderen Intrazonen-Lastverteilungsverfahren ist. Zwar liegt der Mittelwert des Intrazonen-Lastzustands von GK 1 teilweise knapp unter der Schwelle zum Zustand *Sender*, jedoch bedeutet dies, dass GK 1 einen Großteil der Zeit den Zustand *Sender* einnimmt, aber auch für kurze Phasen im Zustand *Ok* ist. In diesen kurzen Phasen kann er eine geringe Zahl von Anfragen, die von GK 2 und GK 3 weitergeleitet wurden, bearbeiten. Diese fallen jedoch kaum ins Gewicht, da die Überlastung von GK 3 bereits sehr groß ist und GK 2 noch nicht überlastet ist.

Zur Verdeutlichung der Unterschiede der Intrazonen-Lastverteilungsverfahren im Bereich des Maximaldurchsatzes ist in Bild 5.43 ein entsprechender Ausschnitt von Bild 5.40 enthalten, der den mittleren Durchsatz der Lastverteilungsverfahren und des 3-fach-Gatekeepers (inklusive der Vertrauensintervalle) über dem Angebot darstellt. Dabei können kaum Unterschiede zwischen den beiden zentral gesteuerten Lastverteilungsverfahren „Round-Robin“ und „Least-Loaded“ festgestellt werden. Des Weiteren wird mit dem „Sender-Receiver“-Verfahren im Bereich zwischen ca 2.9 und knapp unter 3.1 Erlang ein geringfügig geringerer mittlerer Durchsatz erzielt als mit den beiden anderen Lastverteilungsverfahren. Wie bereits erwähnt, ist jedoch ab einem Angebot von ca. 3.1 Erlang der mittlere Durchsatz bei Anwendung des „Sender-Receiver“-Verfahrens höher als bei den anderen Verfahren.

5.2.3 Lastverteilung mit Überlastabwehrmaßnahmen

In diesem Abschnitt wird die gemeinsame Anwendung von Intrazonen-Lastverteilungsverfahren und Überlastabwehrmaßnahmen untersucht. Dazu werden die Lastverteilungsverfahren aus dem vorangehenden Abschnitt, „Static“, „Round-Robin“, „Least-Loaded“ und „Sender-Receiver“ sowie als Referenz der 3-fach-Gatekeeper, jeweils gemeinsam mit der Überlastabwehrmaßnahme „Leaky Bucket“ angewendet. In Abschnitt 5.2.3.1 werden Untersuchungen des stationären Verhaltens vorgestellt, während in Abschnitt 5.2.3.2 Untersuchungen für den instationären Fall, bei denen das Verhalten bei einem Rechteckimpuls ermittelt wird, beschrieben werden.

Die Zusammensetzung der Gatekeeper-Cluster und die Konfiguration der Lastverteilungsverfahren entsprechen jeweils den in Abschnitt 5.2.2 beschriebenen. Insbesondere werden wiederum acht Lastzustände für die Gatekeeper verwendet, um die aktuelle Belastung auch im Niedriglastbereich differenziert bestimmen zu können. Die Konfiguration der Überlastabwehr-

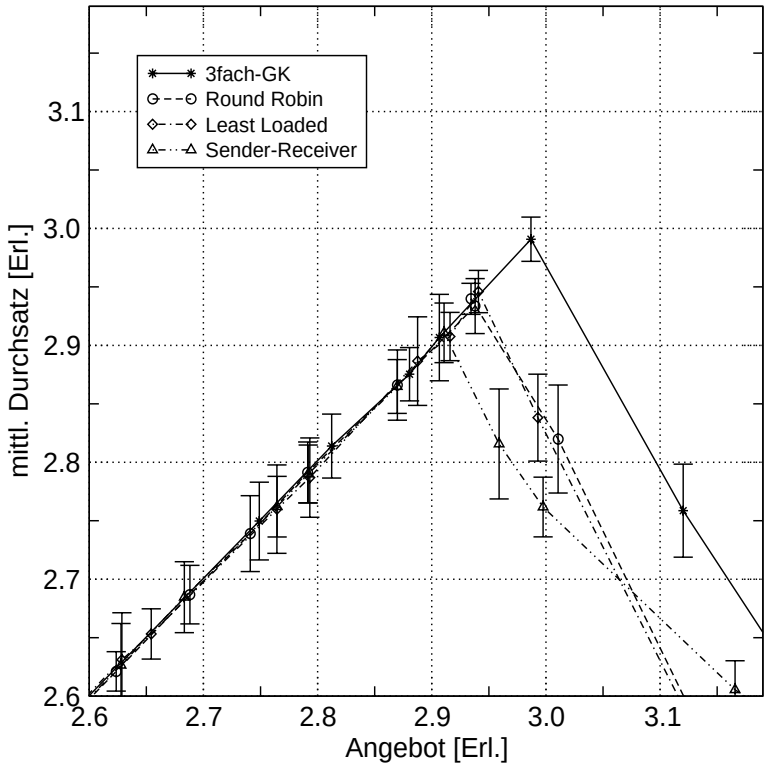


Bild 5.43: Mittlerer Durchsatz für verschiedene Lastverteilungsverfahren und einen 3-fach-Gatekeeper (ohne Überlastabwehr) - Detailansicht

maßnahme „Leaky Bucket“ entspricht der in Abschnitt 5.1.2.2 angewendeten, wobei eine Anpassung an die zusätzlichen Lastzustände für den Niedriglastbereich vorgenommen wird: Der ursprüngliche Lastzustand 0, der eine niedrige Belastung anzeigt, wird in die 4 Lastzustände 0 bis 3 unterteilt. In diesen Lastzuständen werden keine Anfragen abgelehnt. Die Konfiguration der anderen Lastzustände wurde nicht verändert, so dass die ursprünglichen Lastzustände 1 bis 4 nun den Lastzuständen 4 bis 7 entsprechen. Da die Lastverteilung auf Verbindungsebene durchgeführt wird und der Zustandsdaten-Aufwand 1% beträgt, ist der dadurch gestiegene Ressourcenverbrauch vernachlässigbar, so dass kein Anpassungsfaktor bei den im Folgenden vorgestellten Untersuchungen verwendet wird.

5.2.3.1 Untersuchung des stationären Verhaltens

In Bild 5.44 ist der mittlere Durchsatz über dem Angebot für die verschiedenen Intrazonen-Lastverteilungsverfahren sowie für den 3-fach-Gatekeeper dargestellt. Durch die Anwendung der Überlastabwehrmaßnahme wird für jedes Lastverteilungsverfahren ein höherer Durchsatz als ohne Anwendung einer Überlastabwehrmaßnahme erzielt (vgl. dazu Bild 5.40).

Wie im Fall ohne Überlastabwehr fällt der Durchsatz bei Anwendung der statischen Lastverteilung bereits ab einem mittleren Angebot von ca. 1.6 Erlang gegenüber den anderen Lastverteilungsverfahren deutlich ab. Die Begründung dazu entspricht der in Abschnitt 5.2.2 gegebenen.

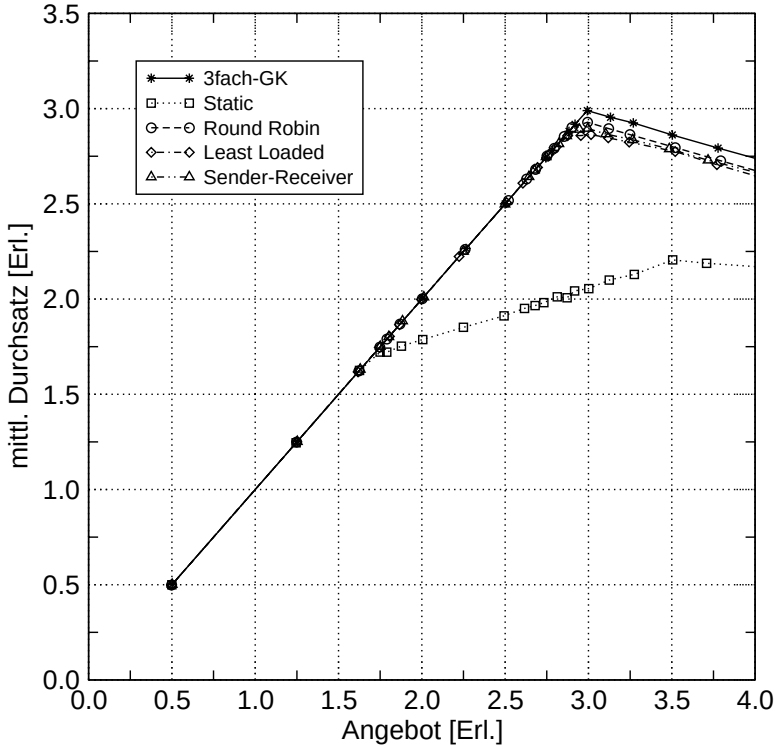


Bild 5.44: Mittlerer Durchsatz für verschiedene Lastverteilungsverfahren und einen 3-fach-Gatekeeper (jeweils mit Anwendung von Überlastabwehrmaßnahmen)

Wie erwartet erzielt der 3-fach-Gatekeeper in den untersuchten Lastbereichen jeweils den höchsten Durchsatz. Im Gegensatz zum Fall ohne Überlastabwehr ergibt sich beim „Sender-Receiver“-Verfahren mit Überlastabwehr kein höherer Durchsatz als bei den zentral gesteuerten Lastverteilungsverfahren bzw. dem 3-fach-Gatekeeper, wenn diese ebenfalls jeweils eine Überlastabwehrmaßnahme anwenden. In diesem Fall hat somit die Beschränkung der Lastverteilung ausschließlich auf wenig belastete Gatekeeper einen geringeren Einfluss.

Zum Vergleich des Durchsatzes der verschiedenen Intrazonen-Lastverteilungsverfahren ist in Bild 5.45 ein Ausschnitt aus Bild 5.44 im Bereich des Maximaldurchsatzes dargestellt, wobei die Vertrauensintervalle ebenfalls enthalten sind. Dabei wird mit dem „Round-Robin“-Verfahren ein geringfügig höherer Durchsatz erzielt als mit dem „Sender-Receiver“-Verfahren, das wiederum etwas besser abschneidet als das kooperierende „Least-Loaded“ Verfahren. Jedoch sind die Unterschiede zwischen diesen Verfahren jeweils im Bereich der Vertrauensintervalle, so dass bezüglich des Durchsatzes keine deutliche Überlegenheit eines der Verfahren festgestellt werden kann, wenn es gemeinsam mit der Überlastabwehrmaßnahme „Leaky Bucket“ angewendet wird.

In den Bildern 5.46 und 5.47 sind die mittleren ARQ-ACF-Verzögerungen der einzelnen Cluster-Mitglieder für das „Round-Robin“- und das „Sender-Receiver“-Verfahren über dem Angebot aufgetragen. Dabei kann zunächst erkannt werden, dass die Antwortzeiten im dargestellten Bereich bis 4.0 Erlang unter 1000 ms bleiben. Des Weiteren kann aus Bild 5.46 entnommen

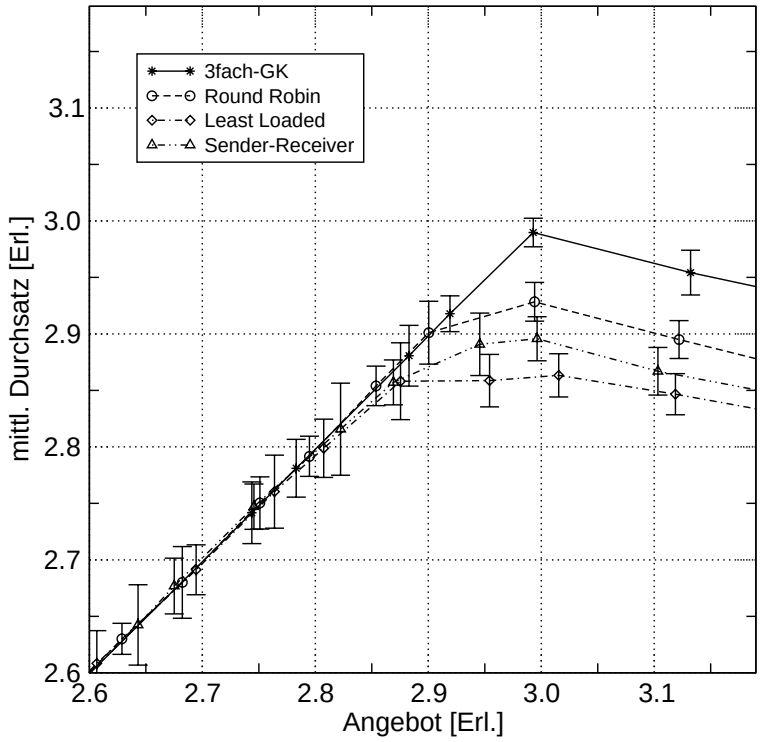


Bild 5.45: Detailansicht des mittlerern Durchsatz für verschiedene Lastverteilungsverfahren und einen 3-fach-Gatekeeper (jeweils mit Anwendung von Überlastabwehrmaßnahmen)

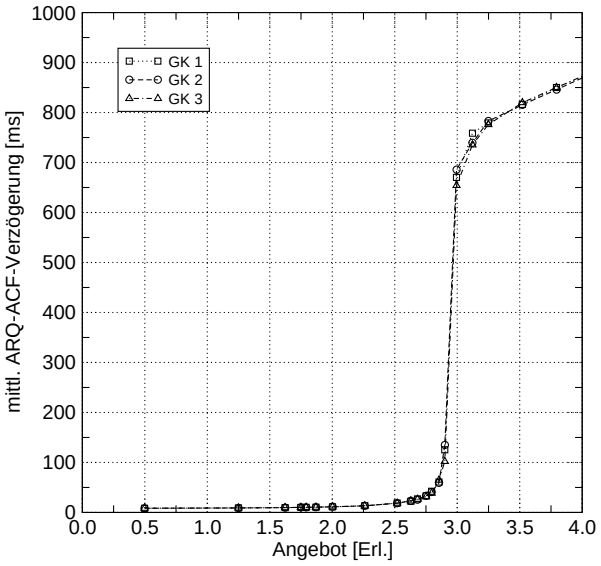


Bild 5.46: Mittl. ARQ-ACF-Verzögerung der Cluster-Mitglieder beim „Round-Robin“-Verfahren (mit Überlastabwehr)

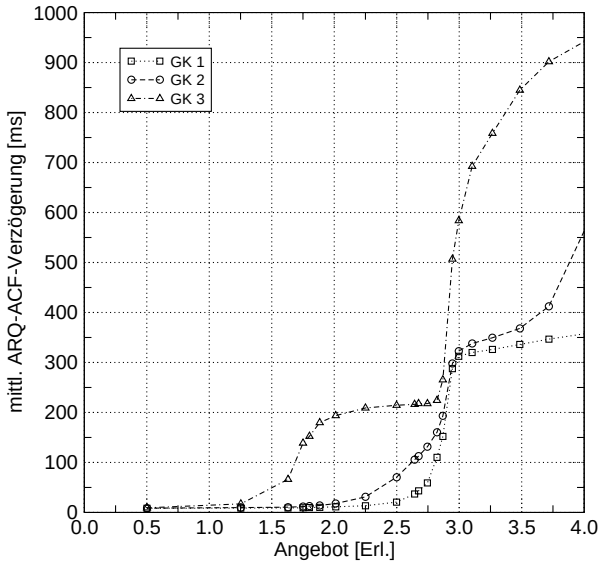


Bild 5.47: Mittl. ARQ-ACF-Verzögerung der Cluster-Mitglieder beim „Sender-Receiver“-Verfahren (mit Überlastabwehr)

werden, dass beim „Round-Robin“-Verfahren eine sehr gleichmäßige Belastung der Cluster-Mitglieder erreicht wird, da alle Gatekeeper des Clusters die gleiche mittlere Antwortzeit beim jeweiligen Angebot besitzen. Beim „Sender-Receiver“-Verfahren ergeben sich unterschiedlich starke Belastungen der Cluster-Mitglieder. Da die Endpunkte ungleichmäßig auf die Gatekeeper verteilt sind, gerät GK 3 beim niedrigsten Gesamtangebot in Hoch- und Überlast. Zwar

gibt er Verbindungsanforderungen an die anderen Cluster-Mitglieder weiter, jedoch bleibt seine Antwortzeit im Vergleich zu den anderen Gatekeepern hoch. Da GK 2 mehr Endpunkte als GK 1 verwaltet, erreicht er bei einem niedrigeren Gesamtangebot die Hoch- und Überlastzustände, was sich in höheren Antwortzeiten niederschlägt. Da ab einem Angebot von ca. 2.9 Erlang alle Gatekeeper des Clusters im Lastzustand *Sender* sind, erfolgt ab diesem Angebot keine Lastverteilung mehr, was zu den unterschiedlichen Antwortzeiten führt.

5.2.3.2 Untersuchung des instationären Verhaltens

Im Folgenden werden Ergebnisse zur Untersuchung des instationären Verhaltens der Intrazonen-Lastverteilungsverfahren „Round-Robin“ und „Sender-Receiver“, die jeweils gemeinsam mit der Überlastabwehrmaßnahme „Leaky Bucket“ angewendet werden, vorgestellt. Dabei erfolgt die Konfiguration der Verfahren ebenso wie bei den Untersuchungen für den stationären Fall. Um ihr Verhalten bei einem Lastsprung zu ermitteln, wird ein Rechteck-Lastprofil verwendet, wobei der Cluster zunächst mit ca. 80% belastet wird, anschließend erfolgt ein Lastsprung auf knapp 150% und nach ca. 60 s fällt die Belastung wieder auf ca. 80%. Ebenso wie bei den Untersuchungen für den stationären Fall sind die Endpunkte beim „Sender-Receiver“-Verfahren ungleichmäßig im Verhältnis 1:2:4 auf die Gatekeeper verteilt. Die folgenden Abbildungen zeigen die Mittelwerte der Untersuchungen über 50 Simulationsläufe. Auf die Darstellung der Vertrauensintervalle wurde aus Übersichtlichkeitsgründen verzichtet, wobei auf die in Abschnitt 5.1.2.1 gegebenen Bemerkungen zur Untersuchung des instationären Verhaltens hingewiesen wird.

Die Bilder 5.48 und 5.49 enthalten den zeitlichen Verlauf des Angebots und des Durchsatzes (linke Ordinatenachse) sowie der ARQ-ACF-Verzögerung (rechte Ordinatenachse) für das „Round-Robin“- und das „Sender-Receiver“-Verfahren. Dabei zeigen beide Verfahren bezüglich des Durchsatzes ein ähnliches Verhalten, wobei beim „Sender-Receiver“-Verfahren die Spitze des Durchsatzes am Ende der Überlastphase weniger hoch ausfällt als bei „Round-Robin“. Dies kann damit begründet werden, dass beim „Sender-Receiver“-Verfahren die Gatekeeper unterschiedlich belastet sind und sich somit eine geringere Anzahl von Verbindungsanforderungen in der Warteschlange der weniger belasteten Gatekeeper befinden. Dies führt dazu, dass bei Beendigung der Überlast die Anzahl der wartenden Verbindungsanforderungen, die nun erfolgreich bearbeitet werden können, etwas kleiner ist.

Bei den über alle Cluster-Mitglieder gemittelten ARQ-ACF-Verzögerungen in den Bildern 5.48 und 5.49 ergeben sich jedoch Unterschiede zwischen den Verfahren. Während in der Überlastphase beim „Round-Robin“-Verfahren die Antwortzeit nahezu konstant bei ca. 950 ms liegt, steigt sie beim „Sender-Receiver“-Verfahren von ca. 650 ms bei Beginn der Überlast bis auf ca. 750 ms nach etwa der Hälfte der Überlast. Die restliche Zeit der Überlastphase bleibt sie dann ungefähr konstant auf diesem Wert. Für eine genauere Betrachtung der Antwortzeiten ist in den Bildern 5.50 und 5.51 der zeitliche Verlauf der entsprechenden Werte der einzelnen

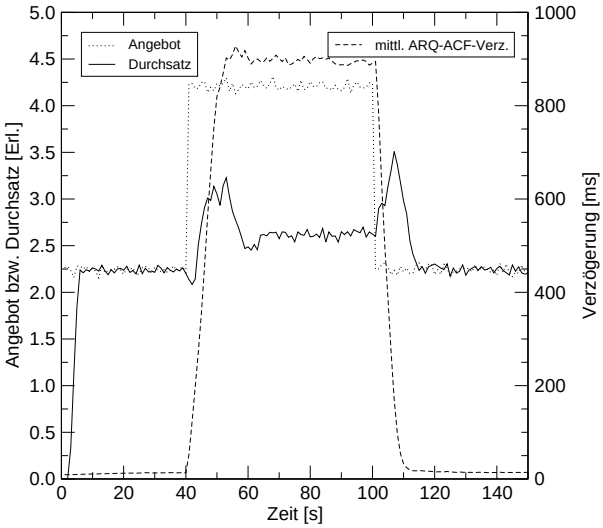


Bild 5.48: Durchsatz und Antwortverzögerung des Clusters beim „Round-Robin“-Verfahren (mit Überlastabwehr)

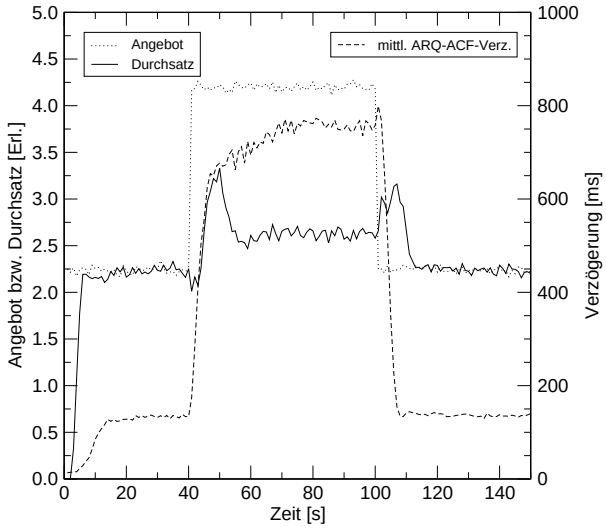


Bild 5.49: Durchsatz und Antwortverzögerung des Clusters beim „Sender-Receiver“-Verfahren (mit Überlastabwehr)

Cluster-Mitglieder dargestellt. Dabei ergeben sich beim „Round-Robin“-Verfahren nahezu identische Werte für die verschiedenen Gatekeeper, was sich auch in den Lastzuständen der Gatekeeper, die in Bild 5.52 enthalten sind, zeigt. Beim „Sender-Receiver“-Verfahren sind die Antwortverzögerungen für die einzelnen Gatekeeper sehr unterschiedlich. Während sie bei GK 1 in der Überlastphase bei unter 400 ms liegen, befinden sie sich bei GK 3 im Bereich von 1000 ms. Darüber hinaus liegen sie bei GK 3 bei der 80%-Belastung bereits bei ca. 200 ms, was auf eine nennenswerte Belastung schließen lässt. Der Anstieg der mittleren Antwortverzögerung des Clusters ist allein auf das Verhalten von GK 2 zurückzuführen, da seine Antwortverzögerung im Verlauf der Überlastsituation zunächst von ca. 400 ms auf ca. 700 ms ansteigt und anschließend bei diesem Wert bleibt. Bei der Betrachtung der Lastzustände in Bild 5.53 ergibt sich ein ähnliches Bild: Während sich GK 3 während der Überlastphase des Clusters in einem Überlastzustand befindet, ist GK 1 knapp im Intrazonen-Lastzustand *Sender*, so dass keine Anfragen anderer Cluster-Mitglieder bei ihm ankommen, er aber auch keine Verbindungsanforderungen ablehnt (siehe auch Bild 5.54). Bei GK 2 steigt der Lastzustand bis zur Hälfte der Überlastphase von 3 auf 4 an und bleibt anschließend für den Rest der Überlastphase bei diesem Wert. Dieser langsame Anstieg lässt sich dadurch erklären, dass die Belastung, die GK 2 erfährt, knapp über seinem Maximaldurchsatz liegt, so dass die Anzahl wartender Nachrichten in seiner Eingangswarteschlange langsam ansteigt, so dass zum einen die Antwortverzögerung größer wird und zum anderen sein Lastzustand eine wachsende Belastung anzeigt. Der Lastzustand bleibt schließlich im Mittel auf dem Wert 4, obwohl in diesem Zustand keine Verbindungsanforderungen abgelehnt werden. Dies lässt sich durch sehr kurze Phasen im Zustand 5 erklären, in dem wenige Anforderungen abgelehnt werden und somit wieder ein Zurückgehen auf Zustand 4 bewirken. Dies wird auch durch die geringe Ablehnungswahrscheinlichkeit von GK 2, die in Bild 5.54 dargestellt ist, bestätigt.

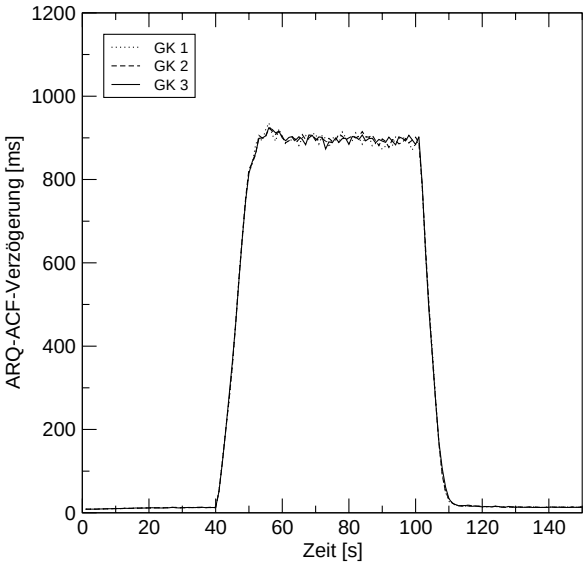


Bild 5.50: Antwortverzögerung der Cluster-Mitglieder beim „Round-Robin“-Verfahren (mit Überlastabwehr)

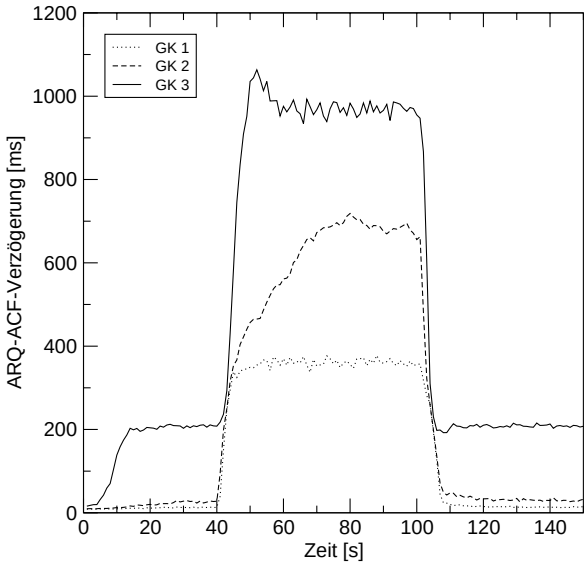


Bild 5.51: Antwortverzögerung der Cluster-Mitglieder beim „Sender-Receiver“-Verfahren (mit Überlastabwehr)

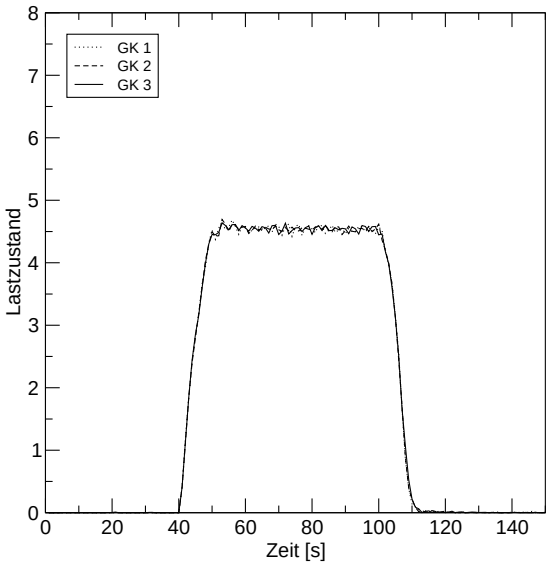


Bild 5.52: Lastzustände der Cluster-Mitglieder beim „Round-Robin“-Verfahren (mit Überlastabwehr)

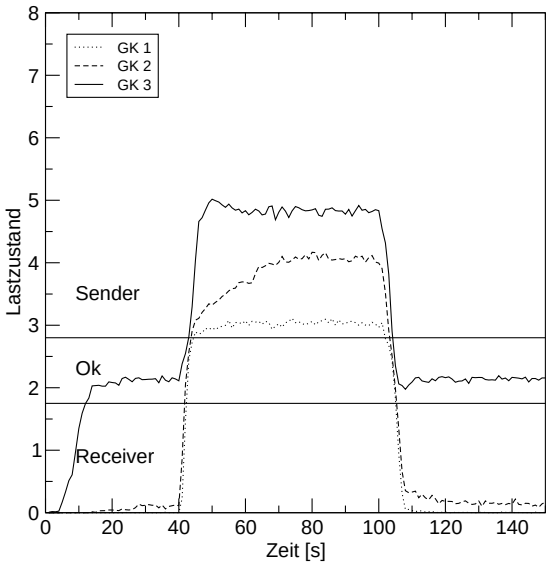


Bild 5.53: Lastzustände der Cluster-Mitglieder beim „Sender-Receiver“-Verfahren (mit Überlastabwehr)

5.2.4 Bewertung

Die in den Abschnitten 5.2.1 bis 5.2.3 vorgestellten Ergebnisse zeigen, dass sich durch die Bildung eines Gatekeeper-Clusters einige Vorteile gegenüber alleinstehenden Gatekeepern ergeben, insbesondere wenn entsprechende Lastverteilungsverfahren innerhalb des Clusters angewendet werden. Neben der Redundanz und der damit verbundenen hohen Verfügbarkeit kann ein hoher Durchsatz erzielt werden. Dabei können durch Anwendung entsprechender Über-

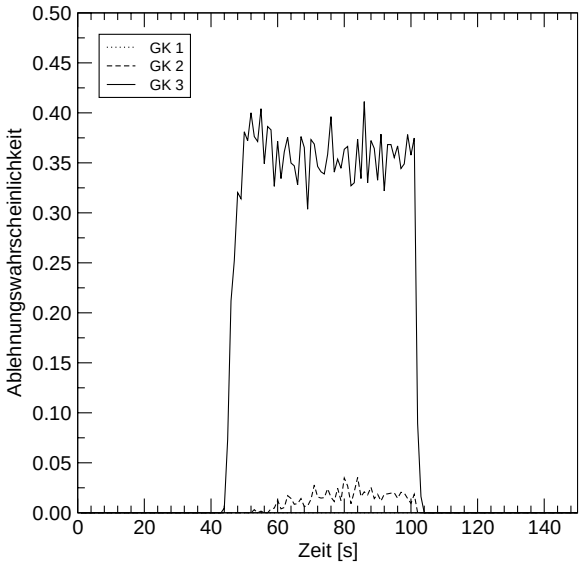


Bild 5.54: Ablehnungswahrscheinlichkeit der Cluster-Mitglieder beim „Sender-Receiver“-Verfahren

lastabwehrmaßnahmen zum einen die Antwortzeiten beschränkt und zum anderen das Fehlschlagen von Anfragen in weiten Lastbereichen verhindert werden.

Der Nachteil eines Clusters ergibt sich aus dem Aufwand für den Zugriff auf die für die Verbindungsbearbeitung notwendigen Daten. Bei den diesbezüglich durchgeführten Untersuchungen wurde festgestellt, dass dieser Aufwand einen erheblichen Einfluss auf den maximalen Durchsatz des Clusters haben kann. Um diesen Einfluss zu verringern, sollte die Lastverteilung mit einer angemessen großen Granularität erfolgen. Aus den Ergebnissen aus Abschnitt 5.2.1 kann abgeleitet werden, dass die Verteilung auf Verbindungsebene den kleinsten Einfluss auf die Leistungsfähigkeit des Clusters hat. Dabei kann kaum eine nennenswerte ungleichmäßige Belastung der einzelnen Cluster-Mitglieder durch diese Granularität im Vergleich zur Verteilung auf Verbindungsphasen- oder Nachrichtenebene festgestellt werden.

Beim Vergleich der Lastverteilungsverfahren zeigt sich, dass die statische Lastverteilung, die im Prinzip einer Aufteilung einer Zone in Unterzonen entspricht, bei inhomogener Belastung wesentlich schlechter als die anderen Verfahren abschneidet, da sie keine dynamische Verteilung der Verbindungsanforderungen durchführt. Im Fall ohne Anwendung von Überlastabwehrmaßnahmen erzielt das „Sender-Receiver“-Verfahren, das verteilt gesteuert ist, ab einem bestimmten Überlastbereich ein höheren Durchsatz als die anderen Verfahren und sogar als der 3-fach-Gatekeeper. Dies wird dadurch erreicht, dass Lastverteilung nur durchgeführt wird, wenn es einen Abnehmer für die Anforderungen gibt, der selbst nicht in Hoch- oder Überlast ist. Damit wird die Verteilung einer Überlast verhindert. Im Fall mit Überlastabwehr ist diese Eigenschaft kaum von Bedeutung, da durch die Überlastabwehrmaßnahmen eine effiziente Nutzung der Ressourcen aller Gatekeeper erreicht wird. Daher schneidet das „Sender-Receiver“-Verfahren mit Anwendung von Überlastabwehrmaßnahmen ähnlich wie die anderen Last-

verteilungsverfahren ab. Zwischen den beiden zentral gesteuerten Lastverteilungsverfahren „Round-Robin“ und „Least-Loaded“ ergeben sich bezüglich des Durchsatzes im Fall ohne Überlastabwehr keine und mit Überlastabwehr nur geringe Unterschiede, wobei dort „Round-Robin“ ein wenig besser abschneidet.

Im weiteren Verlauf werden Untersuchungen mit jeweils einem Vertreter der Lastverteilungsverfahren mit zentraler Steuerung, „Round-Robin“, und mit verteilter Steuerung, „Sender-Receiver“, vorgestellt. Dabei wird festgestellt, dass die einzelnen Cluster-Mitglieder beim „Round-Robin“-Verfahren sowohl mit als auch ohne Überlastabwehr sehr gleichmäßig belastet werden. Beim „Sender-Receiver“-Verfahren ergeben sich deutlich ungleichmäßigere Belastungen, da dort erst Lastverteilung durchgeführt wird, wenn die Gefahr der Überlastung eines Gatekeepers besteht. Dies ist jedoch sinnvoll, da bei den Lastverteilungsverfahren mit verteilter Steuerung die Verteilung selbst Ressourcen der Gatekeeper verbraucht und daher nur durchgeführt werden sollte, wenn es notwendig ist. Die Verteilung der Lastverteilungszustände fällt dagegen kaum ins Gewicht, da sie nur bei Änderungen durchgeführt werden. Im Gegensatz dazu muss beim „Round-Robin“-Verfahren der Dispatcher jede Nachricht an einen Gatekeeper weiterleiten, so dass die Ressourcen für das Weiterleiten der Nachrichten selbst auf jeden Fall verbraucht werden.

Des Weiteren können durch die Anwendung von Überlastabwehrmaßnahmen die Antwortverzögerungen eines Clusters beschränkt werden, so dass sie zumindest bis zu einer Last von über 130% unter einer Sekunde liegen. Die Untersuchung des instationären Verhaltens zeigt, dass die Kombination aus Lastverteilung und Überlastabwehr auch bei einem Lastsprung genügend schnell reagiert und damit die Diensterbringung sicherstellt.

Bei der Bewertung der Verfahren muss der Aufwand für die Lastverteilung beachtet werden. Insbesondere benötigen die Verfahren mit zentraler Steuerung mit dem Dispatcher eine zusätzliche Komponente, die die ankommenden Anforderungen verteilt. Da der Dispatcher jedoch keine Verbindungsbearbeitung durchführt, kann er sehr einfach realisiert werden. Dabei muss aber u. a. die Granularität der Lastverteilung beachtet werden. Wenn die Verteilung z. B. auf Verbindungsphasen-Ebene erfolgt, muss der Dispatcher erkennen, wann eine Verbindungsphase beendet ist bzw. eine neue beginnt. Dies kann entweder intern durch eine relativ einfache Bearbeitung der Signalisiernachrichten durch den Dispatcher erfolgen oder durch spezielle Nachrichten, die zwischen Dispatcher und Gatekeeper ausgetauscht werden. Der Vorteil der zentral gesteuerten Lastverteilungsverfahren ist, dass die Gatekeeper selbst in der Regel nicht an der Durchführung der Lastverteilung beteiligt sind und damit über keine zusätzliche Funktionalität verfügen müssen. Im Gegensatz dazu müssen bei den Lastverteilungsverfahren mit verteilter Steuerung alle Cluster-Mitglieder über die entsprechenden Funktionen zur Lastverteilung verfügen. Dies bedeutet, dass sie in Hochlastphasen zusätzlich durch die Durchführung der Lastverteilung belastet werden. Daher ist es sinnvoll, die Lastverteilung nur dann durchzu-

führen, wenn es notwendig ist, d. h. dass ein Gatekeeper ansonsten in Überlast gerät, und wenn sie möglich ist, d. h. dass ein Gatekeeper vorhanden ist, der noch genügend Ressourcen für die Bearbeitung der weitergeleiteten Last besitzt. Dies ist beispielsweise beim „Sender-Receiver“-Verfahren entsprechend realisiert. Ein wesentlicher Vorteil der Verfahren mit verteilter Steuerung ist, dass sie keine zusätzliche, zentrale Komponente benötigen, da damit zum einen weitere Kosten und Administrationsaufwand entfallen und zum anderen keine Abhängigkeit von einer zentralen Komponente entsteht.

Die Gatekeeper-Cluster dieser Untersuchungen bestehen jeweils aus Gatekeepern mit gleicher Leistungsfähigkeit. Für inhomogene Cluster, bei denen Gatekeeper mit unterschiedlicher Leistungsfähigkeit eine Zone steuern, müssen die Lastverteilungsverfahren gegebenenfalls angepasst werden. Z. B. sollte anstatt dem „Round-Robin“-Verfahren das erweiterte „Weighted-Round-Robin“-Verfahren angewandt werden, bei dem ein Gewichtungsfaktor die Weiterleitung von Anforderungen entsprechend der Leistungsfähigkeiten der Gatekeeper gewichtet. Für das „Sender-Receiver“-Verfahren ist keine Erweiterung notwendig, da die Weiterleitung der Anforderungen von Informationen über den Lastzustand der einzelnen Gatekeeper abhängt.

Zusammenfassend betrachtet scheint das „Sender-Receiver“-Verfahren das flexiblere Intrazonen-Lastverteilungsverfahren darzustellen, wobei der zusätzliche Aufwand für die Integration der Lastverteilungsverfahren in den Gatekeepern berücksichtigt werden muss. Dagegen benötigt das „Round-Robin“-Verfahren mit dem Dispatcher eine zusätzliche Komponente. Die Realisierung der Lastverteilung ist jedoch sehr einfach und damit wenig fehleranfällig. Des Weiteren können Standard-Gatekeeper ohne Lastverteilungsfunktionalität im Cluster angewendet werden.

5.3 Steuerungsoptimierung über Zonengrenzen hinweg

Bei einer entsprechenden Überlastung einer Zone kann es sinnvoll sein, Last über Zonengrenzen hinweg zu verteilen. Dazu wurde in Abschnitt 3.5.2.2 ein Interzonen-Lastverteilungsverfahren vorgestellt, das im Folgenden untersucht und bewertet wird. Dabei wird nur das instationäre Verhalten ermittelt, da vor allem das dynamische Verhalten während der Lastverteilung von Interesse ist. Das stationäre Verhalten kann dagegen relativ einfach durch entsprechende Einzeluntersuchungen von Zonen mit unterschiedlichen Endpunkt- und Gatekeeper-Konfigurationen abgeleitet werden. In Abschnitt 5.3.1 werden die Ergebnisse dieser Untersuchungen vorgestellt, während in Abschnitt 5.3.2 eine Bewertung dieser Ergebnisse erfolgt.

5.3.1 Untersuchung des instationären Verhaltens eines Interzonen-Lastverteilungsverfahrens

Die im Folgenden vorgestellten Ergebnisse enthalten den Verlauf interessierender Messwerte jeweils während eines Simulationslaufes. Dies bedeutet, dass die statistische Aussagesicherheit dieser Werte beschränkt ist, jedoch erlauben sie die Ermittlung des grundsätzlichen Verhaltens des untersuchten Interzonen-Lastverteilungsverfahrens.

Für die Untersuchung der Interzonen-Lastverteilung wird eine aus zwei Zonen bestehende Konfiguration betrachtet, wobei das Angebot an die erste Zone (Zone 1) eine Überlastung der Zone zur Folge hat, wobei die entsprechende Last durch mehrere tausend Endpunkte erzeugt wird, während das Angebot an die zweite Zone (Zone 2) diese kaum belastet. In beiden Zonen wird ein Gatekeeper-Cluster verwendet, der aus jeweils 3 Gatekeepern mit der gleichen Leistungsfähigkeit besteht. Die Belastung der Gatekeeper wird mittels des Lastindikators „Warteschlangenlänge“ ermittelt und es wird die Überlastabwehrmaßnahme „Leaky Bucket“ angewandt. Die Lastverteilung innerhalb einer Zone zwischen den Mitgliedern eines Gatekeeper-Clusters erfolgt jeweils durch das „Sender-Receiver“-Verfahren, wobei die Verteilung auf Verbindungsebene stattfindet und der Aufwand für die Zustandsdaten 1% beträgt. Die lasterzeugenden Endpunkte beantragen keine zusätzlichen Dienste, so dass kein Anpassungsfaktor für erhöhten Ressourcenaufwand für die Verbindungsbearbeitung in den einzelnen Gatekeepern verwendet wird.

Zur Simulation des Interzonen-Lastverteilungsverfahrens wird der Aufwand für die Weiterleitung eines Endpunkts bzw. eines Gatekeepers von einer Zone zu einer anderen im Verhältnis zum Aufwand für einen vollständigen Verbindungsaufbau festgelegt. So bedeutet ein Weiterleitungsaufwand für einen Endpunkt mit dem Wert 10, dass dafür der 10-fache Ressourcenbedarf wie für einen vollständigen Verbindungsaufbau benötigt wird. Dieser Aufwand für die Weiterleitung fällt sowohl in der Ursprungs- als auch in der Zielzone der weitergeleiteten Einheit an. Des Weiteren wird festgelegt, welcher Anteil der Ressourcen des Gatekeepers, der für die Steuerung der Interzonen-Lastverteilung in einer Zone zuständig ist, maximal für die Durchführung der Weiterleitung zur Verfügung steht. Der restliche Anteil der Ressourcen wird für die Verbindungsbearbeitung verwendet. Beispielsweise bedeutet ein Verhältnis von 1:10, dass im Vergleich zur Verbindungssteuerung ein Zehntel der Ressourcen für die Weiterleitung eines Endpunkts oder eines Gatekeepers zur Verfügung stehen. Da dieser Ressourcenanteil nur während der Durchführung der Lastverteilung benötigt wird, kann er ansonsten für die Verbindungsbearbeitung verwendet werden.

Wie in Abschnitt 3.5.2.2 beschrieben, ist es bei der Interzonen-Lastverteilung notwendig, dass der Lastzustand einer Zone entsprechend stabil bestimmt wird, damit die relativ aufwendigen Maßnahmen nicht unnötig durchgeführt werden. Um die Auswirkungen der Fenstergröße bei

der Anwendung eines gleitenden Mittelwerts[1] bei der Lastzustandsbestimmung zu ermitteln, werden Simulationsläufe mit verschiedenen Fenstergrößen durchgeführt. Bei diesen Simulationen wird eine Last angelegt, die dem Zustand *ReceiverGatekeeper* entspricht. Die Zonen-Lastzustandsdaten werden alle 5 s aus den Lastzuständen des Gatekeeper-Clusters und den nicht erfolgreichen Verbindungsanforderungen der Zone aktualisiert (siehe dazu auch die Ausführungen für einen Lastindikator für einen Gatekeeper-Cluster in Abschnitt 3.5.1.2). In Tabelle 5.1 ist die Zeitdauer, bis der Lastzustand *ReceiverGatekeeper* angezeigt wird, für die verschiedenen Fenstergrößen enthalten. Beispielsweise muss bei einer Fenstergröße von 1000 diese Last über eine Stunde (1 Stunde und 13 Minuten) anliegen, bis der Lastzustand angezeigt wird und entsprechende Maßnahmen eingeleitet werden. Somit kann über die Fenstergröße eingestellt werden, nach welcher Zeitdauer, bei der eine Überlastung vorliegt, die Maßnahmen der Interzonen-Lastverteilung eingeleitet werden.

Fenstergröße	**Zeit bis Zustand *ReceiverGatekeeper* [s]**
10	61
100	456
1000	4406

Tabelle 5.1: Zeit bis zum Anzeigen des Zonen-Lastzustands *ReceiverGatekeeper*

Im weiteren Verlauf der Untersuchungen wird die Fenstergröße auf den Wert 100 und das Aktualisierungsintervall der Zonen-Lastzustände auf 1 s eingestellt. Dieses Intervall ist für ein reales System etwas zu kurz gewählt, da damit langfristige Änderungen der Lastzustände erkannt werden sollen und somit ein so kurzes Aktualisierungsintervall das System unnötig belastet. Damit wird jedoch der Aufwand für die Simulationen eingeschränkt, da die einzelnen Maßnahmen nach kürzerer Überlastdauer initiiert werden.

Die folgenden Ergebnisse zeigen nur den Verlauf der Messwerte beim Weiterleiten eines Gatekeepers. Für das Weiterleiten von Endpunkten werden keine Ergebnisse dargestellt, da die Auswirkungen durch diese Maßnahmen im untersuchten Zeitbereich äußerst gering waren. Die Ursache dafür liegt an dem Lastanteil, der durch einen einzelnen Endpunkt erzeugt wird. Da dieser so klein ist, hat auch das Weiterleiten von mehreren Endpunkten kaum Einfluss auf die gesamte Last des überlasteten Gatekeeper-Clusters. Die Schlussfolgerungen daraus werden in Abschnitt 5.3.2 gegeben.

Zur Untersuchung der Auswirkungen des für die Interzonen-Lastverteilung zur Verfügung stehenden Ressourcenanteils werden zwei Simulationsläufe durchgeführt, bei denen jeweils ein

[1] Die Fenstergröße bezeichnet dabei die Anzahl der Werte, die für die Mittelwertbildung verwendet werden.

Gatekeeper von einer wenig belasteten in eine überlastete Zone weitergeleitet wird. Der Aufwand für die Weiterleitung eines Gatekeepers beträgt dabei das 100-fache des Aufwands für einen vollständigen Verbindungsaufbau. In Tabelle 5.2 wird die jeweilige Dauer für die Weiterleitung sowie die durchschnittliche ARQ-ACF-Verzögerung beim steuernden Gatekeeper der überlasteten Zone während des Weiterleitens für unterschiedliche verfügbare Ressourcenanteile dargestellt.

Ressourcenverhältnis Interzonen-Lastverteilung : Verbindungssteuerung	**Zeitdauer für Gatekeeper-Weiterleitung [s]**	**Durchschnittliche ARQ-ACF-Verzögerung des steuernden Gatekeepers [ms]**
1:1	3.7	1542.7
1:10	19	703.3

Tabelle 5.2: Auswirkungen des Ressourcenverhältnisses bei der Interzonen-Lastverteilung

Bei einem Ressourcenverhältnis von 1:1 steht für die Durchführung der Interzonen-Lastverteilung der 5.5-fache Ressourcenanteil im Vergleich zum Verhältnis 1:10 zur Verfügung. Daraus ergibt sich die um den Faktor von ca. 5.5 kürzere Dauer für die Gatekeeper-Weiterleitung. Jedoch sind dadurch weniger Ressourcen für die Verbindungsbearbeitung verfügbar, was sich in der deutlich höheren ARQ-ACF-Verzögerung des steuernden Gatekeepers während der Gatekeeper-Weiterleitung niederschlägt. Diese höheren Verzögerungen treten nur beim steuernden Gatekeeper auf, da das Intrazonen-Lastverteilungsverfahren „Sender-Receiver“ angewandt wird, das bei hoher Belastung aller Cluster-Mitglieder keine Lastverteilung innerhalb des Clusters durchführt (siehe dazu auch die Ausführungen in Abschnitt 5.2.2).

In den Bildern 5.55 und 5.56 ist der Verlauf des Durchsatzes sowie der Verlauf der durchschnittlichen ARQ-ACF-Verzögerung in den beiden Zonen dargestellt. Dabei werden jeweils die Mittelwerte von 5s-Intervallen aufgetragen. Die Konfiguration entspricht der der vorigen Untersuchungen, wobei das Ressourcenverhältnis zwischen Interzonen-Lastverteilung und Verbindungssteuerung auf 1:10 eingestellt wird.

Wie aus Bild 5.55 ersichtlich ist, wird in der überlasteten Zone 1 zu Beginn der Belastung zunächst ein höherer Durchsatz erzielt. Dies lässt sich durch die zu Beginn leeren Warteschlangen der beteiligten Gatekeeper erklären. Anschließend fällt der Durchsatz etwas, wobei er anschließend relativ stabil bis zum Beginn der Gatekeeper-Weiterleitung von der kaum belasteten Zone 2 zu Zone 1 bleibt. Nach der Gatekeeper-Weiterleitung steigt der Durchsatz stark an und hält sich anschließend auf dem deutlich höheren Durchsatzwert relativ stabil. Während der Gatekeeper-Weiterleitung verändert sich der Durchsatz zunächst nicht wesentlich, was durch bereits weit vorangeschrittene Verbindungsaufbauvorgänge erklärt werden kann, bevor er anschließend etwas abfällt. Dabei wird aber nicht deutlich, ob dieses Abfallen ausschließlich

durch die Gatekeeper-Weiterleitung verursacht wird, oder durch statistische Schwankungen unterstützt wird. In Zone 2 sind während des gesamten Verlaufs keine Änderungen des Durchsatzes erkennbar, weil die Belastung in dieser Zone so gering ist, dass weder die Gatekeeper-Weiterleitung noch die Verbindungsbearbeitung durch einen um einen Gatekeeper reduzierten Cluster die Verbindungsbearbeitung beeinflussen.

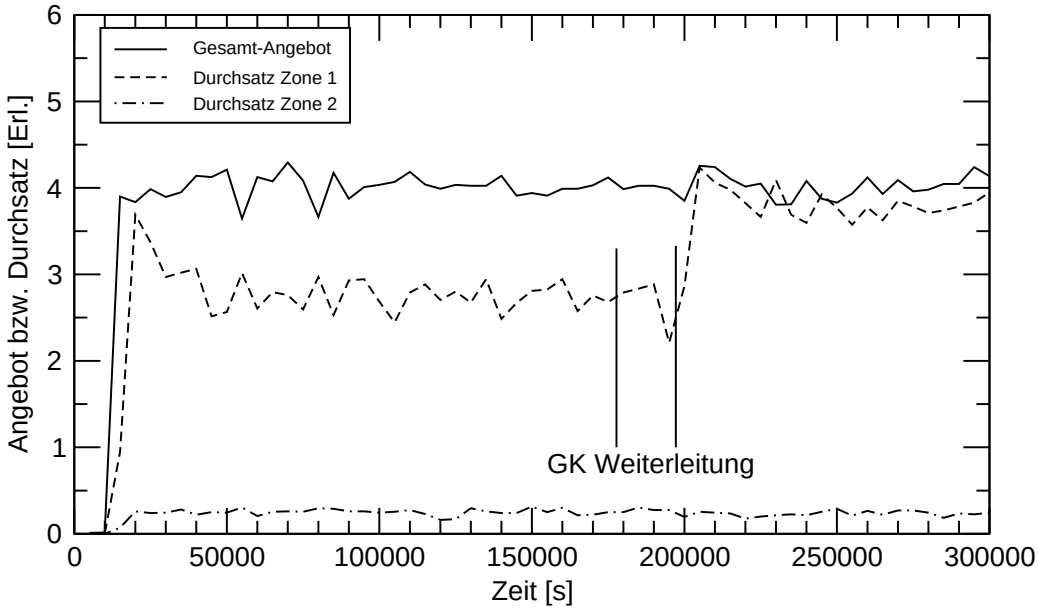


Bild 5.55: Verlauf des Durchsatzes während der Interzonen-Lastverteilung bei einem Ressourcenverhältnis von 1:10

Der in Bild 5.56 aufgetragene Verlauf der durchschnittlichen ARQ-ACF-Verzögerung zeigt, dass in Zone 1 zunächst relativ hohe Verzögerungen vorliegen. Dabei ist ein leichtes Abfallen feststellbar, was sich dadurch erklären lässt, dass der stationäre Zustand noch nicht erreicht ist. In weiteren Simulationen, bei denen die Gatekeeper-Weiterleitung später initiiert wurde, konnte ein stationärer Zustand festgestellt werden, so dass die Verzögerungen stabil blieben. Diese Untersuchungen werden hier nicht aufgeführt, da sie für die Interzonen-Lastverteilung selbst keine Bedeutung besitzen. Während der Durchführung der Gatekeeper-Weiterleitung ist in Zone 1 ein leichter Anstieg der ARQ-ACF-Verzögerung zu beobachten, wobei die Unterschiede zu den statistischen Schwankungen gering sind. Nachdem der zusätzliche Gatekeeper aus Zone 2 in Zone 1 integriert ist, fällt die ARQ-ACF-Verzögerung stark ab. Das folgende kurzzeitige Ansteigen der Verzögerung zwischen 210000 und 230000 s lässt sich durch die Intrazonen-Lastverteilung innerhalb des Clusters erklären: Beim dezentral gesteuerten „Sender-Receiver“-Verfahren durchläuft eine Anforderung zunächst die Eingangswarteschlange des ersten Gatekeepers. Wenn dieser stark belastet ist und im Cluster ein wenig belasteter Gatekeeper verfügbar ist, wird diese Anforderung an den wenig belasteten Gatekeeper weitergeleitet. Dort durchläuft die Anforderung wiederum die Eingangswarteschlange bevor sie bearbeitet wird. Wenn sich der weniger belastete Gatekeeper bereits einer relativ starken Belastung annähert, erfährt die Anforderung dort wiederum eine signifikante Verzögerung, so

dass aus der Summe der beiden Verzögerungen der dargestellte Effekt entsteht. Wenn die Gatekeeper innerhalb des Clusters in etwa gleich belastet sind, entfällt dieser Effekt, da eine Weiterleitung von Anforderungen selten auftritt. Daher erreichen die Antwortverzögerungen ab dem Zeitpunkt von ca. 240000 s einen relativ stabilen Wert von unter 150 ms und zeigen somit eine deutliche Entlastung des Clusters der Zone 1 an. Wie beim Durchsatz sind bei der ARQ-ACF-Verzögerung von Zone 2 keine nennenswerten Einflüsse durch die Gatekeeper-Weiterleitung und die Verbindungsbearbeitung ohne den weitergeleiteten Gatekeeper bemerkbar.

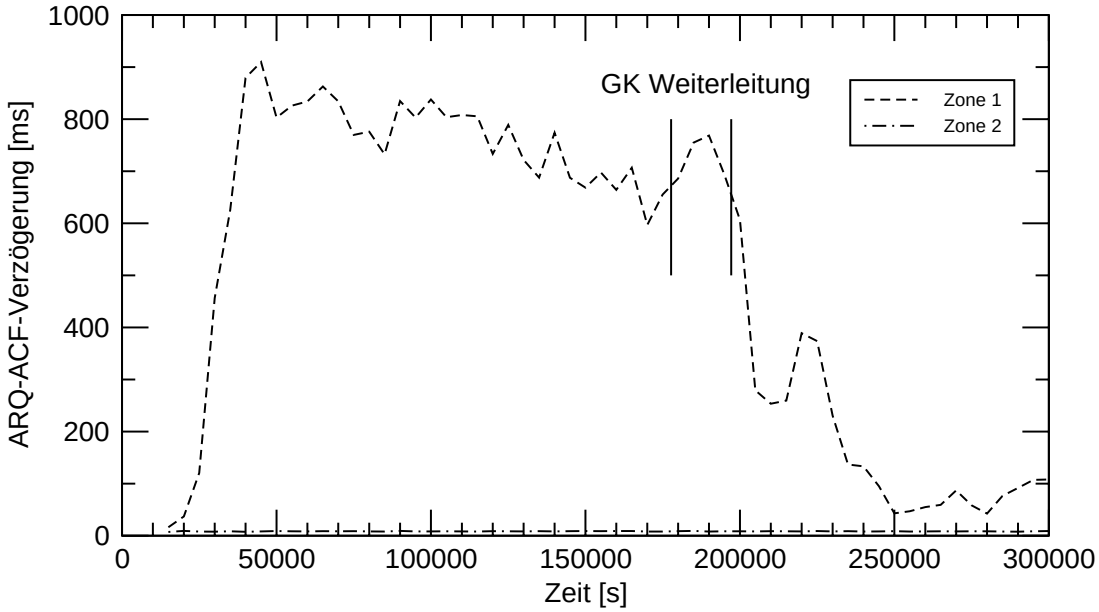


Bild 5.56: Verlauf der ARQ-ACF-Verzögerung während der Interzonen-Lastverteilung bei einem Ressourcenverhältnis von 1:10

5.3.2 Bewertung

In den beschriebenen Untersuchungen wurde zunächst der Einfluss der Fenstergröße auf die Lastindikatorbestimmung für einen Gatekeeper-Cluster ermittelt. Dabei konnte festgestellt werden, dass durch eine entsprechende Konfiguration dieser Fenstergröße zusammen mit dem Aktualisierungsintervall eine ausreichende Stabilität erreicht werden kann. Des Weiteren kann damit die Zeitspanne bis zur Einleitung entsprechender Maßnahmen für die Interzonen-Lastverteilung eingestellt werden.

Für die Durchführung der Interzonen-Lastverteilung werden für die Weitergabe von Komponenten und ihrer anschließenden Integration in der Zielzone Ressourcen der steuernden Komponenten verbraucht. Da dies zusätzlich zur Verbindungsbearbeitung erfolgt, hat der für die Interzonen-Lastverteilung zur Verfügung stehende Ressourcenanteil sowohl Einfluss auf die Lastverteilung selbst als auch auf die Verbindungsbearbeitung. Dabei gilt, dass je höher der Anteil für die Interzonen-Lastverteilung ist, desto schneller ist die Weitergabe einer Komponente vollzogen. Jedoch ergeben sich während der Durchführung der Weitergabe auch entspre-

chend höhere Antwortzeiten bei der steuernden Komponente in der überlasteten Zone, so dass sich dies auf die Dienstgüte, die ein Teilnehmer erfährt, niederschlägt. Daher muss bei der Konfiguration der Interzonen-Lastverteilung ein Kompromiss zwischen schneller Durchführung der Lastverteilung und resultierender Dienstgüte während der Durchführung der Maßnahmen gefunden werden.

Bei der Interzonen-Lastverteilung durch Weitergabe von einzelnen Endpunkten von einer überlasteten in eine wenig belastete Zone konnte keine nennenswerte Entlastung des überlasteten Gatekeeper-Clusters erzielt werden. Wie bereits erwähnt, ist der Lastanteil, der durch einzelne Endpunkte erzeugt wird, im Vergleich zur Gesamtlast relativ gering, so dass die daraus resultierende Reduzierung der Last kaum Einfluss auf das Verhalten des Clusters hat. Eine sinnvolle Erweiterung wäre es daher, anstatt einzelner Endpunkte Endpunktgruppen weiterzugeben. Dies könnte auch bezüglich der Weitergabe der Konfigurationsdaten Vorteile bringen, da diese für eine Gruppe weitgehend übereinstimmen. Jedoch muss bei der Durchführung der Weitergabe selbst beachtet werden, dass die einzelnen Endpunkte nie bzw. nur für sehr kurze Zeit nicht verfügbar sind, damit der Teilnehmer dies nicht als Störung des Systems interpretiert. Des Weiteren könnte die Weitergabe spezieller zentraler Endpunkte, wie z. B. Gateways oder MCUs, größeren Einfluss auf die Belastung eines Clusters haben, da über sie in der Regel viele Verbindungen geführt werden. Im Gegensatz zur Weiterleitung von teilnehmerbezogenen Endpunkten muss in diesem Fall nur die Verfügbarkeit des betroffenen Dienstes sichergestellt sein, wobei dieser während der Weiterleitung auch durch andere Komponenten erbracht werden kann.

Die Weiterleitung eines Gatekeepers zu einer überlasteten Zone zeigt die gewünschten Effekte. So kann zum einen der Durchsatz erhöht werden und zum anderen werden die Antwortzeiten reduziert. Bei dieser Form der Lastverteilung muss jedoch eine dauerhafte Überlastung der Zone vorliegen, da die Durchführung relativ aufwendig ist. Dies muss entsprechend durch die Konfiguration der Lastindikatoren der Gatekeeper-Cluster eingestellt werden. Des Weiteren muss durch die Konfiguration dieser Lastindikatoren sichergestellt sein, dass die weniger belastete Zone, die einen Gatekeeper an eine überlastete Zone weitergibt, mit der reduzierten Anzahl von Cluster-Mitgliedern alle ankommenden Verbindungsanforderungen erfolgreich bearbeiten kann, da sich ansonsten eine Ausbreitung der Überlastung auf weitere Zonen ergeben kann. Um die Diensterbringung während der Weiterleitung auch in der überlasteten Zone sicherzustellen, muss der Ressourcenanteil, der für die Interzonen-Lastverteilung zur Verfügung steht, wie oben bereits erwähnt, entsprechend eingestellt werden.

In den durchgeführten Untersuchungen wurde als Intrazonen-Lastverteilungsverfahren das dezentral gesteuerte „Sender-Receiver“-Verfahren angewandt. Wenn dagegen ein zentral gesteuertes Verfahren, wie z. B. „Round-Robin“, verwendet wird, könnte die Steuerung der Interzonen-Lastverteilung in der zentralen Komponente, dem Dispatcher, durchgeführt wer-

den. Damit wären die Gatekeeper des Clusters weder von der Intrazonen- noch von der Interzonen-Lastverteilung betroffen, so dass Standard-Realisierungen dafür verwendet werden können. Dabei muss jedoch der zusätzliche Ressourcenbedarf beim Dispatcher beachtet werden, damit dieser durch die zusätzlichen Aufgaben nicht selbst in Überlast gerät und damit die Leistungsfähigkeit der gesamten Zone einschränkt.

Bei den vorgestellten Untersuchungen wurden bezüglich der Ressourcen, die für die Weiterleitung einer Komponente benötigt werden, Annahmen getroffen, die derzeit nicht bestätigt werden können, da keine Untersuchungen oder gar Implementierungen existieren, deren Gegenstand eine derartige Lastverteilung bzw. Weiterleitung von Komponenten ist. Um die Allgemeingültigkeit der Ergebnisse abzusichern, wurden daher auch Untersuchungen mit wesentlich höherem Ressourcenbedarf für die Weiterleitung durchgeführt. Dabei konnte festgestellt werden, dass das grundsätzliche Verhalten gleich bleibt, jedoch sich Verschiebungen bezüglich der Zeitdauern, wie z. B. des Zeitbedarfs für die Weiterleitung eines Gatekeepers, ergeben.

Die Grundlage für die Interzonen-Lastverteilung ist, dass zwischen Endpunkten und Gatekeepern bzw. Gatekeeper-Clustern eine logische Zuordnung existiert, die dynamisch verändert werden kann. Um dies jedoch realisieren zu können, kann die Last nur innerhalb einer gemeinsam verwalteten VoIP-Umgebung verteilt werden, da dabei auch Konfigurationsdaten ausgetauscht werden müssen. Dabei spielt die physikalische Entfernung zwischen Endpunkten und zugeordneten Gatekeepern bei der Weiterleitung von Endpunkten eine untergeordnete Rolle. Bei der Weiterleitung von Gatekeepern zwischen Zonen ist jedoch auch die physikalische Entfernung der beteiligten Gatekeeper bzw. Gatekeeper-Cluster von Bedeutung, da die Gatekeeper eines Clusters zum einen Zugang zu den Konfigurationsdaten der Endpunkte der Zone benötigen und zum anderen Zustandsdaten austauschen müssen. Der Austausch der Zustandsdaten hängt dabei auch von der Granularität der Lastverteilung sowie von dem angewandten Lastverteilungsverfahren ab. Daher muss auch für weitergeleitete Gatekeeper gewährleistet sein, dass sie auf diese Daten mit entsprechend kurzen Verzögerungen zugreifen können. Wenn diese Verzögerungen zu groß sein sollten, kann dies zur Verschlechterung der Leistungsfähigkeit des gesamten Clusters führen.

Eine mögliche Gefahr durch die Interzonen-Lastverteilung könnte darin bestehen, dass eine lokale Überlastsituation, deren Ursache sich in einer Zone befindet, durch diese Verteilung auf andere Zonen weiter verbreitet wird und somit deren Diensterbringung beeinträchtigen kann. Im schlimmsten Fall könnte dies zu einer Kettenreaktion führen, die alle Zonen der VoIP-Umgebung betreffen würde. Um dies zu verhindern, müssen geeignete Maßnahmen getroffen werden, wie es z. B. durch geeignete Konfiguration der Lastindikatoren der Gatekeeper-Cluster erfolgen kann.

Schließlich muss bei der Interzonen-Lastverteilung beachtet werden, dass sie in die Strukturierung der VoIP-Umgebung eingreift, da sie die Aufteilung der Umgebung in Zonen verändert.

Da diese Aufteilung in der Regel durch entsprechende Netzplanung optimiert wurde, sollte eine Interzonen-Lastverteilung nur notwendig sein, wenn unerwartete Ereignisse, wie z. B. der Ausfall bestimmter Einheiten, oder Änderungen der Randbedingungen, wie z. B. das Teilnehmerverhalten bezüglich spezieller Komponenten, auftreten. Um die Struktur einer Zone trotzdem zu erhalten, könnte eine bestimmte Anzahl von Reserve-Gatekeepern bereit gehalten werden, die in überlastete Zonen weitergeleitet werden, um die Diensterbringung zu sichern. Diese Reserve-Gatekeeper könnten in den Zeiten, in denen sie nicht benötigt werden, weniger zeitkritische Aufgaben bearbeiten, so dass ihre Ressourcen ebenfalls effizient ausgenutzt werden.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wurden Verfahren für die optimierte Steuerung von VoIP-Netzen untersucht. Mit diesen Verfahren soll eine effiziente Ausnutzung der verfügbaren Ressourcen erzielt werden. Die dabei vorgeschlagenen Verfahren entstammen sowohl der Tele- als auch der Datenkommunikation und spiegeln damit auch die Konvergenz der Netze wider, die durch VoIP einen weiteren Antrieb erfährt.

Als Basis für die durchgeführten Untersuchungen wurde in Kapitel 2 zunächst VoIP mit den dabei angewandten Protokollen beschrieben. Dazu wurden in diesem Kapitel relevante Grundlagen der Kommunikationstechnik sowie Architektur und Protokolle von IP-basierten Netzen vorgestellt. Darüber hinaus wurde auf die Konvergenz der Kommunikationsnetze eingegangen. Anschließend wurde der Austausch der Nutzdaten für VoIP sowie dabei auftretende Probleme und existierende Lösungsansätze zur Unterstützung einer entsprechenden Dienstgüte beschrieben. Bei der Vorstellung der Steuerung von VoIP-Netzen mittels entsprechender Signalisierprotokolle wurde die VoIP-Signalisierung nach der ITU-T-Empfehlung H.323 ausführlich beschrieben, da sie die Grundlage für die weiteren Untersuchungen darstellt. Des Weiteren wurden relevante Unterschiede zwischen der Signalisierung für die klassische, leitungsgebundene Kommunikation und der Signalisierung für VoIP-Dienste aufgezeigt.

In Kapitel 3 wurden verschiedene Verfahren zur optimierten Steuerung von VoIP-Netzen, die auf der Empfehlung H.323 basieren, vorgestellt. Dazu wurden zunächst mögliche Ziele der Optimierung beschrieben, wobei die wichtigsten sicherlich eine Maximierung des Durchsatzes bei gleichzeitiger Einhaltung vorgegebener Grenzen der Antwortzeiten und einer möglichst geringen Anzahl fehlgeschlagener Anforderungen sind. Des Weiteren wurde der prinzipielle Ablauf einer Steuerungsoptimierung abgeleitet, der aus der Bestimmung des aktuellen Lastzustands mittels entsprechender Lastindikatoren, der Verteilung der Last auf die bearbeitenden Komponenten sowie der Überlastabwehr besteht.

Bei der Einordnung dieser Arbeit konnte festgestellt werden, dass zum einen die dynamische Lastverteilung, wie sie innerhalb von Clustern in der Datenkommunikation angewandt wird, in der Telekommunikation kaum Verwendung findet. Zum anderen spielen Überlastabwehrmaßnahmen, die fester Bestandteil von Telekommunikationssystemen sind, in der Datenkommuni-

kation bisher kaum eine Rolle. Dort beschränkt sich die Überlastabwehr in der Regel auf das Verwerfen einzelner Nachrichten, ohne den Kontext dieser Nachrichten zu beachten. Daher stellen die in dieser Arbeit durchgeführten Untersuchungen bezüglich der Lastverteilung in einem Gatekeeper-Cluster und zwischen verschiedenen Clustern sowie der Überlastabwehr in den einzelnen Gatekeepern eine Erweiterung der bisher angewandten Verfahren zur Steuerungsoptimierung dar.

Für verschiedene Ressourcen einer VoIP-Umgebung wurden in Abschnitt 3.4 mögliche Verfahren zur optimierten Steuerung beschrieben, die eine effiziente Nutzung dieser Ressourcen erlauben. In der Regel werden diese Verfahren im Gatekeeper durchgeführt, da er für die Steuerung und Verwaltung einer Zone zuständig ist.

Durch die zentrale Rolle des Gatekeepers und durch seine Aufgaben bei der Steuerung von VoIP-Diensten ist er ein möglicher Ort der Überlastung und sollte daher besonders effizient seine Ressourcen nutzen. Des Weiteren ist die Wirkbreite seiner Überlastung groß, da dabei alle Endpunkte einer Zone betroffen sind. Aus diesen Gründen wurden für die optimierte Steuerung für die Gatekeeper-Ressourcen entsprechende Verfahren abgeleitet und detailliert beschrieben:

- Zur Ermittlung der aktuellen Belastung eines Gatekeepers wurden verschiedene Lastindikatoren definiert, wobei ein neuer Lastindikator, „Gewichtete Verbindungszustände“, vorgeschlagen wurde. Des Weiteren ist eine Kombination von mehreren Lastindikatoren möglich, um z. B. für bestimmte Lastbereiche eine entsprechend große Auflösung der Lastanzeige zu erreichen. Dies wird beispielsweise für die Bestimmung der Belastung eines Gatekeeper-Clusters verwendet.

- Um eine Verteilung der Last auf mehrere Gatekeeper zu ermöglichen, wurde die Bildung von Gatekeeper-Cluster vorgeschlagen, die gemeinsam die Administration einer Zone durchführen. Neben einer höheren Verfügbarkeit gegenüber einem alleinstehenden Gatekeeper erlaubt ein Gatekeeper-Cluster eine sukzessive Erweiterung der Leistungsfähigkeit, so dass er leicht an Änderungen der Randbedingungen angepasst werden kann. Als Nachteil muss der Aufwand für die Datenverwaltung und den Zugriff auf gemeinsam benötigte Zustandsdaten genannt werden. Zur Ausnutzung der zur Verfügung stehenden Gatekeeper-Ressourcen wurden verschiedene Intrazonen-Lastverteilungsverfahren vorgestellt. Des Weiteren wurde auf die Granularität der Lastverteilung eingegangen, die den Aufwand für die Verwaltung der gemeinsam benötigten Daten beeinflusst. Für die Lastverteilung über Zonengrenzen hinweg wurde ein Interzonen-Lastverteilungsverfahren vorgeschlagen, dass die dynamische Veränderbarkeit der Struktur einer VoIP-Umgebung ausnutzt.

- Wenn die Last durch die Gatekeeper bzw. Gatekeeper-Cluster nicht mehr vollständig bearbeitet werden kann, werden Überlastabwehrmaßnahmen angewandt, um die Diensterbringung soweit als möglich sicherzustellen. Dazu wurden in dieser Arbeit verschiedene

Verfahren, die für die Überlastabwehr in Signalisiernetzen der Telekommunikation Anwendung finden und die in einem Gatekeeper anwendbar sind, abgeleitet.

Bei der Betrachtung einiger Realisierungsaspekte der genannten Verfahren wurde auf die Reihenfolge der Ressourcenbewertung und ihre Auswirkungen eingegangen. Darüber hinaus wurden Einschränkungen bezüglich der gemeinsamen Verwendung der verschiedenen Verfahren beschrieben.

In einem Kommunikationsnetz, das Sprach- und Datenkommunikation integriert, müssen die zur Verfügung stehenden Ressourcen entsprechend verwaltet werden. In dieser Arbeit wurde dies berücksichtigt, indem auf die Steuerungsoptimierung eines integriert verwalteten Unternehmensnetzes eingegangen wurde. Dabei wurde vorgeschlagen, dass die Aufgaben eines Gatekeepers erweitert werden, so dass er die Ressourcenverwaltung sowohl für die VoIP- als auch für alle weiteren Dienste übernimmt.

Für die Bewertung der Verfahren zur Steuerungsoptimierung wurden die in Kapitel 4 beschriebenen Untersuchungsmethoden angewandt. Zum einen wurden verschiedene Überlastabwehrmaßnahmen prototypisch im sog. *PreServer* implementiert, so dass sie in einem H.323-basierten VoIP-Testbett zusammen mit einem existierenden Gatekeeper untersucht werden konnten. Zum anderen wurde ein Simulationswerkzeug entwickelt, das Untersuchungen von Lastindikatoren, Intra- und Interzonen-Lastverteilungsverfahren sowie von Überlastabwehrmaßnahmen sowohl im stationären als auch im instationären Fall erlaubt. Dabei wurden die Signalisierprotokolle der Empfehlung H.323 detailliert nachgebildet, um beispielsweise die Granularität der Lastverteilung entsprechend untersuchen zu können.

In Kapitel 5 wurden schließlich die Ergebnisse der Untersuchungen für die Steuerungsoptimierung für die Gatekeeper-Ressourcen präsentiert und bewertet.

Zunächst wurden die Verfahren der Steuerungsoptimierung für einen einzelnen Gatekeeper untersucht. Die Ergebnisse der Untersuchungen mittels der prototypischen Implementierung im *PreServer* zeigen die Wirksamkeit der einzelnen Überlastabwehrmaßnahmen, ohne dass sich eines der Verfahren sehr deutlich von den anderen abhebt. Die Untersuchungen mittels Simulationen erlauben eine detailliertere Betrachtung sowohl der Lastindikatoren als auch der Überlastabwehrmaßnahmen:

- Die Ergebnisse für die untersuchten Lastindikatoren „Anzahl offener Anfragen“, „Warteschlangenlänge“ und „Gewichtete Verbindungszustände“ zeigen sowohl für den stationären als auch für den instationären Fall keine wesentlichen Unterschiede bei der Bestimmung des aktuellen Lastzustands. Durch seine einfache Realisierbarkeit besitzt der Lastindikator „Warteschlangenlänge“ jedoch Vorteile gegenüber den anderen beiden.
- Bei Anwendung jeder der untersuchten Überlastabwehrmaßnahmen kann im Vergleich zum Fall ohne Überlastabwehr ein deutlich höherer Durchsatz sowie die Einhaltung der Ant-

wortzeiten bis zu einer hohen Überlast erreicht werden. Des Weiteren wurden im betrachteten Lastbereich keine fehlgeschlagenen Verbindungsanforderungen auf Grund verlorener oder verspäteter Nachrichten beobachtet. Beim Vergleich der Überlastabwehrmaßnahmen wurden keine deutlichen Unterschiede festgestellt, wobei wiederum der jeweilige Realisierungsaufwand mit betrachtet werden sollte.

Wenn sich beispielsweise durch Einführung neuer zusätzlicher Dienste die Verkehrscharakteristika ändern, muss dies bei der Parametrisierung der Verfahren beachtet werden, da sich dadurch der Ressourcenverbrauch des Gatekeepers pro Verbindung ändert. Bei Anwendung eines Anpassungsfaktors, der beispielsweise durch kontinuierliche Messungen im Gatekeeper aktualisiert wird, können neben dem Durchsatz auch die Antwortzeiten weitgehend eingehalten werden, ohne dass ein Umkonfigurieren der Lastzustandsbestimmung notwendig ist.

Bei den Verfahren der Steuerungsoptimierung für einen Gatekeeper-Cluster wurde zunächst die Granularität der Lastverteilung untersucht. Dabei konnte festgestellt werden, dass auch bei sehr geringem Aufwand für den Zugriff auf die Zustandsdaten die Verteilung der Last auf Verbindungsebene am vorteilhaftesten ist, da damit der höchste Durchsatz erzielt wird. Jedoch werden die Unterschiede zur Verteilung auf Verbindungsphasen- und Nachrichtenebene geringer, wenn dieser Aufwand kleiner wird.

Die Untersuchung der Intrazonen-Lastverteilungsverfahren zeigt, dass die statische Lastverteilung durch ihre fehlende dynamische Anpassungsfähigkeit bei entsprechend ungleichmäßiger Belastung der Cluster-Mitglieder am schlechtesten abschneidet. Wenn keine Überlastabwehr durchgeführt wird, erzielt das verteilt gesteuerte „Sender-Receiver“-Verfahren ab einem bestimmten Angebot sogar einen höheren Durchsatz als ein einzelner Gatekeeper, der über die gleiche Leistungsfähigkeit verfügt wie die Summe alle Cluster-Mitglieder. Dies wird dadurch erreicht, dass die Lastverteilung nur dann durchgeführt wird, wenn ein Cluster-Mitglied angezeigt hat, dass es weitere Anforderungen anderer Cluster-Mitglieder übernehmen kann. Damit wird eine Verteilung einer Überlast verhindert. Wenn neben der Intrazonen-Lastverteilung auch Überlastabwehrmaßnahmen in den Gatekeepern eines Clusters angewendet werden, ist diese Eigenschaft jedoch kaum von Bedeutung, so dass die zentral gesteuerten Verfahren „Least-Loaded“ und „Round-Robin“ ähnlich abschneiden wie das verteilt gesteuerte „Sender-Receiver“-Verfahren. Die Untersuchung des instationären Verhaltens bei einem Rechteckimpuls des Angebots zeigt, dass eine ausreichende Reaktionsfähigkeit auch bei der Kombination von Intrazonen-Lastverteilung und Überlastabwehr gewährleistet ist, wobei diese Untersuchungen nur für das „Round-Robin“- und das „Sender-Receiver“-Verfahren durchgeführt wurden. Jedoch ergeben sich keine deutlichen Unterschiede zwischen den Verfahren.

Bei der Betrachtung der Realisierung der Intrazonen-Lastverteilungsverfahren muss für die zentral gesteuerten Verfahren beachtet werden, dass diese eine zentrale Instanz für die Durchführung der Lastverteilung benötigen, was bei den Verfahren mit verteilter Steuerung nicht

notwendig ist. Dagegen können bei den zentral gesteuerten Verfahren Standard-Gatekeeper verwendet werden, wenn die Lastverteilung mit einer entsprechenden Granularität erfolgt. Bei den Verfahren mit verteilter Steuerung müssen dagegen alle Cluster-Mitglieder über die Funktionalität zur Lastverteilung verfügen.

Für die Interzonen-Lastverteilung wurde das instationäre Verhalten untersucht, um den Verlauf der Lastverteilung zu ermitteln. Dabei wurden die Auswirkungen verschiedener Parameter der Lastindikator-Bestimmung eines Gatekeeper-Clusters und der Interzonen-Lastverteilung selbst betrachtet. Ein wichtiger Parameter ist der Ressourcenanteil, der für die Durchführung der Interzonen-Lastverteilung zur Verfügung steht: Je größer dieser Anteil ist, desto schneller werden die entsprechenden Maßnahmen durchgeführt und führen somit zu einer Entlastung der überlasteten Zone. Jedoch sind die Auswirkungen auf die Verbindungssteuerung ebenfalls größer, so dass die Dienstgüte dadurch beeinträchtigt werden kann. Des Weiteren wurde festgestellt, dass die Weiterleitung einzelner Endpunkte von einer überlasteten in eine weniger belastete Zone kaum Wirkung zeigt. Jedoch würde die Weiterleitung von Endpunkt-Gruppen oder von zentralen Endpunkten, wie z. B. Gateways, zu einer Entlastung einer Zone führen. Durch die Weiterleitung eines Gatekeepers kann dagegen eine deutliche Entlastung der überlasteten Zone erreicht werden, wobei keine Auswirkungen in der weniger belasteten Zone durch die reduzierte Anzahl von Cluster-Mitgliedern sichtbar wurden. Bei der Interzonen-Lastverteilung ist zu beachten, dass die Gefahr einer Verbreitung einer lokalen Überlastsituation besteht. Dies muss durch eine entsprechende Konfiguration des Verfahrens verhindert werden.

Insgesamt betrachtet, werden durch die verschiedenen Verfahren zur Steuerungsoptimierung deutliche Verbesserungen im Bereich des Durchsatzes, der Einhaltung von Antwortzeiten sowie bei der Zahl fehlschlagender Anforderungen erzielt. Des Weiteren konnte eine ausreichende Reaktionsfähigkeit der Verfahren auch bei sich sehr schnell ändernden Lastprofilen festgestellt werden. Auch in hohen Überlastbereichen ist somit die Diensterbringung und die Stabilität der VoIP-Umgebung sichergestellt.

Für einige der durchgeführten Untersuchungen mussten bezüglich des Ressourcen- und Zeitbedarfs für die Verfahren zur Steuerungsoptimierung Parameterwerte geschätzt werden. Um diese zu bestätigen, sind daher weitere Untersuchungen an realen Implementierungen notwendig. Dies würde auch eine noch genauere Differenzierung zwischen den einzelnen Verfahren erlauben. Da bezüglich der Intra- und Interzonen-Lastverteilung bei Gatekeepern bisher keine Realisierungen existieren, sind als nächster Schritt für weitere Untersuchungen prototypische Implementierungen der Verfahren notwendig. Dies wäre zumindest für die zentral gesteuerten Intrazonen-Lastverteilungsverfahren mit begrenztem Aufwand zu realisieren, da hierbei Standard-Gatekeeper, die nur bezüglich der Datenverwaltung entsprechend angepasst werden müssen, verwendet werden könnten. Als Basis für den dabei notwendigen Dispatcher könnte der *PreServer* benutzt werden.

Wie bereits erwähnt, kann durch einen Gatekeeper-Cluster eine höhere Verfügbarkeit der Gatekeeper-Funktionalität als durch einen alleinstehenden Gatekeeper erreicht werden. Um diese höhere Verfügbarkeit zu erhalten, müssen jedoch geeignete Verfahren entwickelt werden, die zum einen das Erkennen des Ausfalls eines Cluster-Mitglieds und zum anderen die Übernahme seiner Aufgaben durch andere Cluster-Mitglieder ermöglichen. Dabei ist insbesondere die Übernahme von bereits existierenden Signalisierbeziehungen von Bedeutung. Als Basis für weitere Untersuchungen in diesem Bereich kann beispielsweise Anhang R der Empfehlung H.323 dienen.

Ein Aspekt, der die Sicherheit der Steuerungskomponenten betrifft und damit über die Überlastabwehr hinausgeht, sind sog. *Denial of Service* (DoS) Angriffe, die beispielsweise das Funktionieren von Web-Servern verhindern wollen. Dabei wird eine große Anzahl von Verbindungsanforderungsnachrichten an die Gatekeeper gesendet, ohne dass die Antworten der Gatekeeper anschließend weiter bearbeitet werden. Dadurch werden viele Ressourcen der Gatekeeper für Verbindungsanforderungen gebunden, die nicht erfolgreich bearbeitet werden können. Dies kann so weit gehen, dass andere Verbindungsanforderungen nicht mehr bearbeitet werden können, oder gar das System instabil wird. Um dies zu verhindern, müssen geeignete Maßnahmen zur Erkennung und zur Bekämpfung derartiger Angriffe entwickelt werden. Dabei könnten bereits bekannte Verfahren aus dem Bereich der Web-Server in adaptierter Form für Gatekeeper angewandt werden.

Schließlich sollten die Verfahren für ein zukünftiges, integriert verwaltetes Kommunikationsnetz erweitert werden, so dass eine ganzheitliche Verwaltung aller Ressourcen eines Kommunikationsnetz durchgeführt werden kann. Damit könnten die verfügbaren Ressourcen effizient genutzt werden, so dass die Diensterbringung für unterschiedlichste Tele- und Datenkommunikationsdienste auch in Hoch- und Überlastsituationen gewährleistet ist.

Literaturverzeichnis

[1] ARON, M. ; SANDERS, D. ; DRUSCHEL, P. ; ZWAENEPOEL, W.: „Scalable Content-aware Request Distribution in Cluster-based Network Servers“, *Proceedings of the USENIX 2000 Annual Technical Conference*, San Diego, June 2000

[2] ARVIDSSON, A. ; PETTERSSON, S. ; ANGELIN, L.: „Congestion Control in Intelligent Networks for Real Time Performance and Profit Optimisation“, *Proceedings of the 10th ITC Specialists Seminar on Control in Communications*, Lund, Sweden, Sep. 1996, pp. 347-358

[3] VAN AS, H. R.: *Modellierung und Analyse von Überlast-Abwehrmechanismen in Paketvermittlungsnetzen*, Dissertation, Universität Stuttgart, 1984

[4] BHATTI, N. ; FRIEDRICH, R.: „Web Server Support for Tiered Services“, IEEE Network Magazine, Vol. 13, No. 5, Sep./Oct. 1999, pp. 64-71

[5] BLAKE, S. ; BLACK, D. ; CARLSON, E. ; DAVIES, E. ; WANG, Z. ; WEISS, W.: *An architecture for differentiated services*, RFC 2475, IETF, Dec. 1998

[6] BODAMER, S.: *Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknoten*, Dissertation (eingereicht), Universität Stuttgart, 2002

[7] BRADEN, R. ; CLARK, D. ; SHENKER, S.: *Integrated services in the Internet architecture: an overview*, RFC 1633, IETF, June 1994

[8] BRADEN, R. ; ZHANG, L. ; BERSON, S. ; HERZOG, S. ; JAMIN, S.: *Resource ReSerVation Protocol (RSVP)*, RFC 2205, IETF, Sep. 1997

[9] BRANDT, A. ; BRANDT, M. ; SPAHL, G. ; WEBER, D.: „Modelling and Optimization of Call Distribution Systems“, *Proceedings of the 15th International Teletraffic Congress (ITC 15)*, Washington, Jun. 1997, pp. 133-144

[10] BRESLAU, L. ; KNIGHTLY, E. W. ; SHENKER, S. ; STOICA, I. ; ZHANG, H.: „Endpoint Admission Control: Architectural Issues and Performance“, *Proceedings of ACM SIGCOMM 2000*, Stockholm, Aug. 2000

[11] BRYHNI, H. ; KLOVNING, E. ; KURE, O.: „A Comparison of Load Balancing Techniques for Scalable Web Servers“, *IEEE Network Magazine*, Vol. 14, No. 4, July/Aug. 2000, pp. 58-64

[12] BURGSTAHLER, L.: *Messverfahren zur Unterstützung von dienstgüteorientierter Wegesuche in verbindungslosen Datennetzen*, Monographie, Universität Stuttgart, 2003

[13] CALDERON, M.: *Integration of a Multipoint Conference System in a VoIP Test Environment*, Student thesis project No. 1771, Institute for Communication Networks and Computer Engineering, University of Stuttgart, 2003

[14] CARDELLINI, V. ; CASALICCHIO, E. ; COLAJANNI, M. ; TUCCI, S.: „Mechanism for quality of service in Web clusters“, *Computer Networks*, Vol. 37, No. 6, Dec. 2001, pp. 761-771

[15] CASAVANT, T. L. ; KUHL, J. G.: „A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems“, *IEEE Transactions on Software Engineering*, Vol. 14, No. 2, Feb. 1988, pp. 141-154

[16] CASNER, S. ; JACOBSON, V.: *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*, RFC 2508, IETF, Feb.1999

[17] CHARZINSKI, J.: *IP Based Networks and Applications*, Manuskript zur Vorlesung an der Universität Stuttgart, 2002

[18] CHERKASOVA, L. ; PHAAL, P.: „Hybrid and Predictive Admission Strategies to Improve the Performance of an Overloaded Web Server“, *HP Laboratories Report No. HPL-98-125R1*, July 1998

[19] CHERKASOVA, L. ; PHAAL, P.: „Session Based Admission Control: a Mechanism for Improving the Performance of an Overloaded Web Server“, *HP Laboratories Report No. HPL-98-119*, June 1998

[20] CIARDO, G. ; RISKA, A. ; SMIRNI, E.: „EQUILOAD: a load balancing policy for clustered web servers“, *Performance Evaluation*, Vol. 46, No. 2-3, Oct. 2001, pp. 101-124

[21] CISCO: *Cisco CallManager System Guide, Release 3.3(2)*, Cisco Systems, 2002

[22] COMER, D. E.: *Internetworking with TCP/IP – Volume I: Principles, Protocols, and Architecture*, 2nd edition, Prentice Hall, Upper Saddle River,1991

[23] DAISENBERGER, G. ; ÖHLERICH, J. ; WEGMANN, G.: „STATOR - Statistical Overload Regulation - and TAIL - Time Account Input Limitation - Two Concepts for Overload Regulation in SPC Systems“, *Proceedings of the 11th International Teletraffic Congress (ITC 11)*, Kyoto, Japan, 1985, Paper 2.1B-4

[24] DEERING, S. ; HINDEN, R.: *Internet protocol, version 6 (IPv6) specification*, RFC 2460, IETF, Dec. 1998

[25] ELGEBALY, H.: „Characterization of Multimedia Streams of an H.323 Terminal“, *Intel Technology Journal*, 2nd quarter, 1998

[26] FIELDING, R. ; GETTYS, J. ; MOGUL, J. ; FRYSTYK, H. ; MASINTER, L. ; LEACH, P. ; BERNERSLEE, T.: *Hypertext transfer protocol HTTP/1.1*, RFC 2616, IETF, Jun. 1999

[27] FINEBERG, V.: „A Practical Architecture for Implementing End-to-End QoS in an IP Network“, *IEEE Communications Magazine*, Vol. 40, No. 1, Jan. 2002, pp. 122-130

[28] FOX, A. ; GRIBBLE, S. D. ; CHAWATHE, Y. ; BREWER, E. A. ; GAUTHIER, P.: „Cluster-Based Scalable Network Services“, *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, St. Malo, France, Oct. 1997

[29] GRAY, C. G. ; CHERITON, D. R.: „Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency“, *Proceedings of 12th ACM Symposium on Operating Systems Principles*, Dec. 1989

[30] GUÉRIN, R. ; AHMADI, H. ; NAGHSHINEH, M.: „Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks“, *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 7, Sep. 1991, pp. 968-981

[31] HAAS, M.: *Mechanismen zur Leistungsregelung von Rechensystemen*, Dissertation, Universität Karlsruhe, 1990

[32] HÄNDEL, R. ; HUBER, M. N. ; SCHRÖDER, S.: *ATM Networks – Concepts, Protocols, Applications*, Addison-Wesley Publishing Company, 1994

[33] HANSELKA, P. ; ÖHLERICH, J. ; WEGMANN, G.: „Adaptation of the Overload Regulation Method STATOR to Multiprocessor Controls and Simulation Results“, *Proceedings of the 12th International Teletraffic Congress (ITC 12)*, Torino, June 1988, pp. 395-401

[34] HOUCK, D. ; MEEMPAT, G.: „Call admission control and load balancing for voice over IP“, *Performance Evaluation*, Vol. 47, No. 4, Mar. 2002, pp. 243-253

[35] HUBIG, W. ; WEBER, D.: „Overload Control in ISDN PABXs“, *Proceedings of the 14th International Teletraffic Congress (ITC 14)*, Antibes Juan-les-Pins, June 1994, pp. 243-252

[36] ISO: *Information technology – Open systems interconnection – Basic reference model: The basic model*, ISO/IEC 7498-1, International Organisation for Standardization, 1994

[37] ITU: *ITU Internet Reports 2001: IP Telephony*, International Telecommunication Union, 2001

[38] ITU-T: *Traffic and congestion control requirements for SS No. 7 and IN-structured networks*, Recommendation E.744, International Telecommunication Union, 1996

[39] ITU-T: *One-way transmission time*, Recommendation G.114, International Telecommunication Union, 2000

[40] ITU-T: *Call signalling protocols and media stream packetization for packet-based multimedia communication systems*, Recommendation H.225.0, International Telecommunication Union, 2000

[41] ITU-T: *Call signalling protocols and media stream packetization for packet-based multimedia communication systems, Amendment 1*, Recommendation H.225.0 Amendment 1, International Telecommunication Union, 2002

[42] ITU-T: *Security and encryption for H-Series (H.323 and other H.245-based) multimedia terminals*, Recommendation H.235, International Telecommunication Union, 2000

[43] ITU-T: *Control protocol for multimedia communication*, Recommendation H.245, International Telecommunication Union, 2003

[44] ITU-T: *Interworking of H.Series multimedia terminals with H.Series multimedia terminals and voice/voiceband terminals on GSTN and ISDN*, Recommendation H.246, International Telecommunication Union, 1998

[45] ITU-T: *Gateway control protocol*, Recommendation H.248, International Telecommunication Union, 2000

[46] ITU-T: *Gateway control protocol: Media gateway overload control package*, Recommendation H.248.11, International Telecommunication Union, 2002

[47] ITU-T: *Packet-based multimedia communications systems*, Recommendation H.323 (Version 4), International Telecommunication Union, 2000

[48] ITU-T: *Generic functional protocol for the support of supplementary services in H.323*, Recommendation H.450.1, International Telecommunication Union, 1998

[49] ITU-T: *B-ISDN ATM Adaptation Layer (AAL) Functional Description*, Recommendation I.362, International Telecommunication Union, 1993

[50] ITU-T: *Methods for subjective determination of transmission quality*, Recommendation P.800, International Telecommunication Union, 1996

[51] ITU-T: *ISDN user-network interface layer 3 specification for basic call control*, Recommendation Q.931, International Telecommunication Union, 1998

[52] ITU-T: *Data protocols for multimedia conferencing*, Recommendation T.120, International Telecommunication Union, 1996

[53] ITU-T: *Specification and Description Language (SDL)*, Recommendation Z.100, International Telecommunication Union, 2002

[54] JAMJOOM, H. ; REUMANN, J. ; SHIN, K. G.: „QGuard: Protecting Internet Servers from Overload“, *Technical Report CSE-TR-427-00*, University of Michigan, Ann Arbor, MI, 2000

[55] JENNINGS, B. ; ARVIDSSON, A. ; CURRAN, T.: „A Token-based Strategy for Coordinated, Profit-optimal Control of Multiple IN Resources“, *Proceedings of the 17th International Teletraffic Congress (ITC 17)*, Salvador da Bahia, Brazil, Dec. 2001

[56] KASERA, S. ; PINHEIRO, J. ; LOADER, C. ; KARAUL, M. ; HARI, A. ; LAPORTA, T.: „Fast and Robust Signaling Overload Control“, *9th International Conference on Network Protocols (ICNP 2001)*, Riverside, CA, Nov. 2001, pp. 323-331

[57] KATZSCHNER, L.: *Digitale Vermittlungstechnik*, Manuskript zur Vorlesung, Universität Stuttgart 2000

[58] KELLY, F. P. ; KEY, P. B. ; ZACHARY, S.: „Distributed Admission Control“, *IEEE Journal On Selected Areas In Communications*, Vol. 18, No. 12, Dec. 2000, pp. 2617-2628

[59] KIHL, M.: *Overload Control Strategies for Distributed Communication Networks*, Ph.D. thesis, Lund Institute of Technology, Lund University, 1999

[60] KLEINROCK, L.: „Distributed Systems“, *Communications of the ACM*, Vol. 28, No. 11, Nov. 1985, pp. 1200-1213

[61] KLEINROCK, L.: „Power and Deterministic Rules of Thumb for Probabilistic Problems in Computer Communications“, *15th IEEE International Conference on Communications (ICC’79)*, Vol. 2, Boston, June 1979

[62] KÖSTER, G.: „Improving the automatic congestion control functionality in SS7-signaling networks“, *Computer Networks*, Vol. 36, No.5/6 , Aug. 2001, pp. 617-624

[63] KOCHER, H.: *Entwurf und Implementierung einer Simulationsbibliothek unter Anwendung objektorientierter Methoden*, Dissertation, Universität Stuttgart, 1994

[64] KOCHER, H. ; LANG, M.: „An Object-Oriented Library for Simulation of Complex Hierarchical Systems“, *Proceedings of the Object-Oriented Simulation Conference (OOS ‘94)*, 1994, pp. 145-152

[65] KORPI, M. ; KUMAR, V.: „Supplementary Services in the H.323 IP Telephony Network“, *IEEE Communications Magazine*, Vol. 37, No. 7, July 1999, pp. 118-125

[66] KRÖNER, H. ; KÜHN, P. J. ; RENGER, T.: „Management von ATM-Netzen“, *Informationstechnik und Technische Informatik (it+ti)*, Feb. 1997

[67] KÜHN, P. J.: *Communication Networks I*, Manuskript zur Vorlesung, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 2002

[68] KÜHN, P. J.: *Communication Networks II*, Manuskript zur Vorlesung, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 2002

[69] KÜHN, P. J. ; SCHOPP, M.: „Signalling Networks for ISDN, IN and Mobile Networks - Modelling, Analysis, and Overload Control“, *Proceedings of the 10th ITC Specialists Seminar on Control in Communications*, Lund, Sweden, Sep. 1996, pp. 35-49

[70] KUMAR, V. ; KORPI, M. ; SENGODAN, S.: *IP Telephony with H.323*, John Wiley & Sons, 2001

[71] LAW, A. L. ; KELTON, W. D.: *Simulation modelling and analysis*, McGraw-Hill, New York, 1991

[72] LEINER, B. M. ; CERF, V. G. ; CLARK, D. D. ; KAHN, R. E. ; KLEINROCK, L. ; LYNCH, D. C. ; POSTEL, J. ; ROBERTS, L. G. ; WOLFF, S.: *A Brief History of the Internet*, http://www.isoc.org/internet/history/brief.shtml, Aug. 2000

[73] LOVEGROVE, W. P. ; HAMMOND, J. L. ; TIPPER, D.: „Simulation Methods for Studying Nonstationary Behavior of Computer Networks“, *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 9, Dec. 1990, pp. 1696-1708

[74] MATHY, L. ; EDWARDS, C. ; HUTCHISON, D.: „The Internet: A Global Telecommunications Solution?“, *IEEE Network Magazine*, Vol. 14, No. 4, July/Aug. 2000, pp. 46-57

[75] MATSUMURA, R. ; YOSHINO, H. ; HORIGOME, H. ; MIWA, H.: „Traffic Control for Server Overload and Network Congestion by Dynamic Multi-Server System“, *NTT Review*, Vol. 10, No. 2, Mar. 1998

[76] NORTHCOTE, B. S. ; SMITH, D. E.: „Service Control Point Overload Rules to Protect Intelligent Network Services“, *IEEE/ACM Transactions on Networking*, Vol. 6, No. 1, Feb. 1998, pp. 71-81

[77] ODLYZKO, A.: „The Internet and other networks: Utilization rates and their implications“, *Information Economics and Policy*, Vol. 12, No. 4, Dec. 2000, pp.341-365

[78] PADHYE, C. ;CHRISTENSEN, K ; MORENO, W.: „A New Adaptive FEC Loss Control Algorithm for Voice Over IP Applications“, *Proceedings of the 19th IEEE International Performance, Computing, and Communication Conference*, Feb. 2000

[79] PAI, V. S. ; ARON, M. ; BANGA, G. ; SVENDSEN, M. ; DRUSCHEL, P. ; ZWAENEPOEL, W. ; NAHUM, E.: „Locality-Aware Request Distribution in Cluster-based Network Servers“, *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-VIII)*, San Jose, CA, Oct. 1998, pp. 205-216

[80] PETTERSSON, S. ; ARVIDSSON, A.: „A profit optimizing strategy for congestion control in signaling networks“, *Proceedings of the ITC-Seminar*, Bangkok, Nov./Dec. 1995, Paper 39-1

[81] PHAM, X. H. ; BETTS, R.: „Congestion control for intelligent networks“, *Computer Networks and ISDN Systems*, Vol. 26, No. 5, Jan. 1994, pp. 511-524

[82] PÖHNL, M.: *Untersuchung des Konzepts des alternativen Gatekeepers in H.323*, Studienarbeit Nr. 1772, Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart, 2003

[83] POSTEL, J.: *Internet protocol*, RFC 791, IETF, Sep. 1981

[84] POSTEL, J.: *Transmission control protocol*, RFC 793, IETF, Sep. 1981

[85] POSTEL, J.: *User datagram protocol*, RFC 768, IETF, Aug. 1980

[86] POSTEL, J. ; REYNOLDS, J.: *File transfer protocol (FTP)*, RFC 959, IETF, Oct. 1985

[87] REYNOLDS, J. ; POSTEL, J.: *Assigned Numbers*, RFC 1700, IETF, Oct. 1994

[88] RÖSSLER, G. ; STEINERT, T.: „A flexible traffic generator for testing PABX and Call Center performance“, *Proceedings of the 14th IFIP International Conference on Testing Communicating Systems (TestCom 2002)*, Berlin, 2002, pp. 139-147

[89] ROSEN, E. ; VISWANATHAN, A. ; CALLON, R.: *Multiprotocol label switching architecture*, RFC 3031, IETF, Jan. 2001

[90] ROSENBERG, J. ; SCHULZRINNE, H. ; CAMARILLO, G. ; JOHNSTON, A. ; PETERSON, J. ; SPARKS, R. ; HANDLEY, M. ; SCHOOLER, E.: *SIP: Session Initiation Protocol*, RFC 3261, IETF, Jun. 2002

[91] ROSENBERG, J. ; SHOKEY, R.: „The Session Initiation Protocol (SIP): A Key Component for Internet Telephony“, *Computer Telephony*, Vol. 8, No. 6, June 2000

[92] SADKA, A. H.: *Compressed Video Communications*, John Wiley & Sons, 2002

[93] SAITO, H.: *Teletraffic Technologies in ATM Networks*, Artech House, Boston, 1994

[94] SCHROEDER, T. ; GODDARD, S. ; RAMAMURTHY, B.: „Scalable Web Server Clustering Technologies“, *IEEE Network Magazine*, Vol. 14, No. 3, May/June 2000, pp. 38-45

[95] SCHULZRINNE, H. ; CASNER, S. ; FREDERICK, R. ; JACOBSON, V.: *RTP: a transport protocol for real-time applications*, RFC 1889, IETF, Jan. 1996

[96] SCHULZRINNE, H. ; ROSENBERG, J.: „The Session Initiation Protocol: Internet-Centric Signaling“, *IEEE Communications Magazine*, Vol. 38, No. 10, Oct. 2000, pp. 134-141

[97] SCHULZRINNE, H. ; ROSENBERG, J.: „The Session Initiation Protocol: Providing Advanced Telephony Services Across the Internet“, *Bell Labs Technical Journal*, Vol. 3, No. 4, Oct./Dec. 1998, pp. 144-160

[98] SCHULZRINNE, H. ; ROSENBERG, J.: „Internet Telephony: architecture and protocols – an IETF perspective“, *Computer Networks*, Vol. 31, No. 3, Feb. 1999, pp. 237-255

[99] SCHWARZ, A.: *Modellierung und Bewertung von Verfahren zur Last- und Leistungsregelung in Steuereinheiten von B-ISDN/ATM-Vermittlungssystem*, Dissertation, Universität Stuttgart, 2002

[100] SCHWARZ, A.: “Overload Indicators and Strategies for a B-ISDN/ATM Switching System“, *6th Open Workshop on High Speed Networks*, Stuttgart, Oct. 1997

[101] SHAN, Z. ; LIN, C. ; MARINESCU, D. C. ; YANG, Y.: „Modeling and performance analysis of QoS-aware load balancing of Web-server clusters“, *Computer Networks*, Vol. 40, No. 2, Oct. 2002, pp. 235-256

[102] SHEPLER, S. ; CALLAGHAN, B. ; ROBINSON, D. ; THURLOW, R. ; BEAME, C. ; EISLER, M. ; NOVECK, D.: *Network File System (NFS) version 4 Protocol*, RFC 3530, IETF, Apr. 2003

[103] SHIRAZI, B. A. ; HURSON, A. R. ; KAVI, K. M. , Eds.: *Scheduling and Load Balancing in Parallel and Distributed Systems*, IEEE Computer Society Press, Los Alamitos, California, 1995

[104] SHIRAZI, B. ; WANG, M. ; PATHAK, G.: „Analysis and Evaluation of Heuristic Methods for Static Task Scheduling“, *Journal of Parallel and Distributed Computing*, Vol. 10, No.3, 1990, pp. 222-232

[105] SHIVARATRI, N. G. ; KRUEGER, P. ; SINGHAL, M.: „Load Distributing for Locally Distributed Systems“, *Computer*, Vol. 25, No. 12, Dec. 1992, pp. 33-44

[106] SIDI, M. ; LIU, W. Z. ; CIDON, I. ; GOPAL, I.: „Congestion Control Through Input Rate Regulation“, *IEEE Transactions on Communications*, Vol. 41, No. 4, Aug. 1993, pp. 471-477

[107] SIEGMUND, G.: *ATM – Die Technik des Breitband-ISDN*, 2. Auflage, R. v. Decker's Verlag, Heidelberg, 1994

[108] STATHOPOULOS, V. M. ; VENIERIS, I. S.: „ICALB: an integrated congestion avoidance and load balancing algorithm for distributed intelligent networks. Part I: description of ICALB“, *Computer Communications*, Vol. 25, No. 17, Nov. 2002, pp. 1548-1563

[109] STATHOPOULOS, V. M. ; VENIERIS, I. S.: „ICALB: an integrated congestion avoidance and load balancing algorithm for distributed intelligent networks. Part II: Performance evaluation of ICALB“, *Computer Communications*, Vol. 25, No. 17, Nov. 2002, pp. 1564-1574

[110] STEINERT, T. ; RÖSSLER, G.: „Generation of realistic signalling traffic in an ISDN load test system using SDL user models“, *Proceedings of the 13th IFIP International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE 2000)*, Pisa, 2000, pp. 219-234

[111] STEWART, R, ; XIE, Q. ; MORNEAULT, K. ; SHARP, C. ; SCHWARZBAUER, H. ; TAYLOR, T. ; RYTINA, I. ; KALLA, M. ; ZHANG, L. ; PAXSON, V.: *Stream Control Transmission Protocol*, RFC 2960, IETF, Oct. 2000

[112] STEWART, R, ; XIE, Q. ; STILLMAN, M. ; TUEXEN, M.: *Aggregate Server Access Protocol (ASAP)*, Internet Draft draft-ietf-rserpool-asap-07.txt, IETF, May 2003, work in progress

[113] STIDHAM, S.: „Optimal Control of Admission to a Queueing System“, *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 8, Aug. 1985, pp. 705-713

[114] SZE, H. P. ; LIEW, S. C. ; LEE, J. Y. B. ; YIP, D. C. S.: „A Multiplexing Scheme for H.323 Voice-Over-IP Applications“, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 7, Sep. 2002, pp. 1360-1368

[115] TANENBAUM, A. S.: *Computer Networks – Third Edition*, Upper Saddle River, New Jersey, Prentice Hall, 1996

[116] TOGA, J. ; OTT, J.: „ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations“, *Computer Networks*, Vol. 31, No. 3, Feb. 1999, pp. 205-223

[117] UNGER, H. ; DÄNE, B. ; NÜTZEL, J.: „Experiences Simulating the Load Sharing System LYDIA with High Level PN“, *Proceedings of High Performance Computing '98*, Boston, April 1998

[118] VARSHNEY, U. ; SNOW, A. ; MCGIVERN, M. ; HOWARD, C.: „Voice over IP“, *Communications of the ACM*, Vol. 45, No. 1, Jan. 2002, pp. 89-96

[119] VLEESCHAUWER, D. D. ; JANSSEN, J. ; PETIT, G. H. ; POPPE, F.: „Quality bounds for packetized voice transport“, *Alcatel Telecommunication Review*, 1st Quarter 2000

[120] VOIGT, T. ; TEWARI, R. ; FREIMUTH, D. ; MEHRA, A.: „Kernel Mechanisms for Service Differentiation in Overloaded Web Servers“, *Proceedings of 2001 Usenix Annual Technical Conference*, Boston, MA, June 2001

[121] WAHL, M. ; HOWES, T. ; KILLE, S.: *Lightweight Directory Access Protocol (v3)*, RFC 2251, IETF, Dec. 1997

[122] WANG, Z.: *Internet QoS: Architecture and mechanisms for quality of service*, Morgan Kaufmann Publishers, San Francisco, 2001

[123] WELSH, M. ; CULLER, D. ; BREWER, E.: „SEDA: An Architecture for Well-Conditioned, Scalable Internet Services“, *Proceedings of the 18th ACM symposium on Operating systems principles (SOSP-18)*, Banff, Alberta, Canada, Oct. 2001, pp. 230-243

[124] WU, D. ; HOU, T. ; CHAO, H. J. ; LI, B.: „A Per Flow Based Node Architecture for Integrated Services Packet Networks“, *Telecommunication Systems*, Vol. 17, No. 1,2, May/June 2001, pp. 135-160

[125] XIE, Q. ; STEWART, R. ; STILLMAN, M.: *Endpoint Name Resolution Protocol (ENRP)*, Internet Draft draft-ietf-rserpool-enrp-06.txt, IETF, May 2003, work in progress

[126] YANG, C.-S. ; LUO, M.-Y.: „Efficient Support for Content-Based Routing in Web Server Clusters“, *Proceedings of the 2nd USENIX Symposium on Internet Technologies & Systems (USITS'99)*, Boulder, Colorado, USA, Oct. 1999

[127] YU, H. ; BRESLAU, L. ; SHENKER, S.: „A Scalable Web Cache Consistency Architecture”, *Proceedings of ACM SIGCOMM*, Sep. 1999

[128] ZEPF, J.: *Modellierung und Bewertung von Überlastabwehrmechanismen in Signalisiernetzen*, Dissertation, Universität Stuttgart, 1995

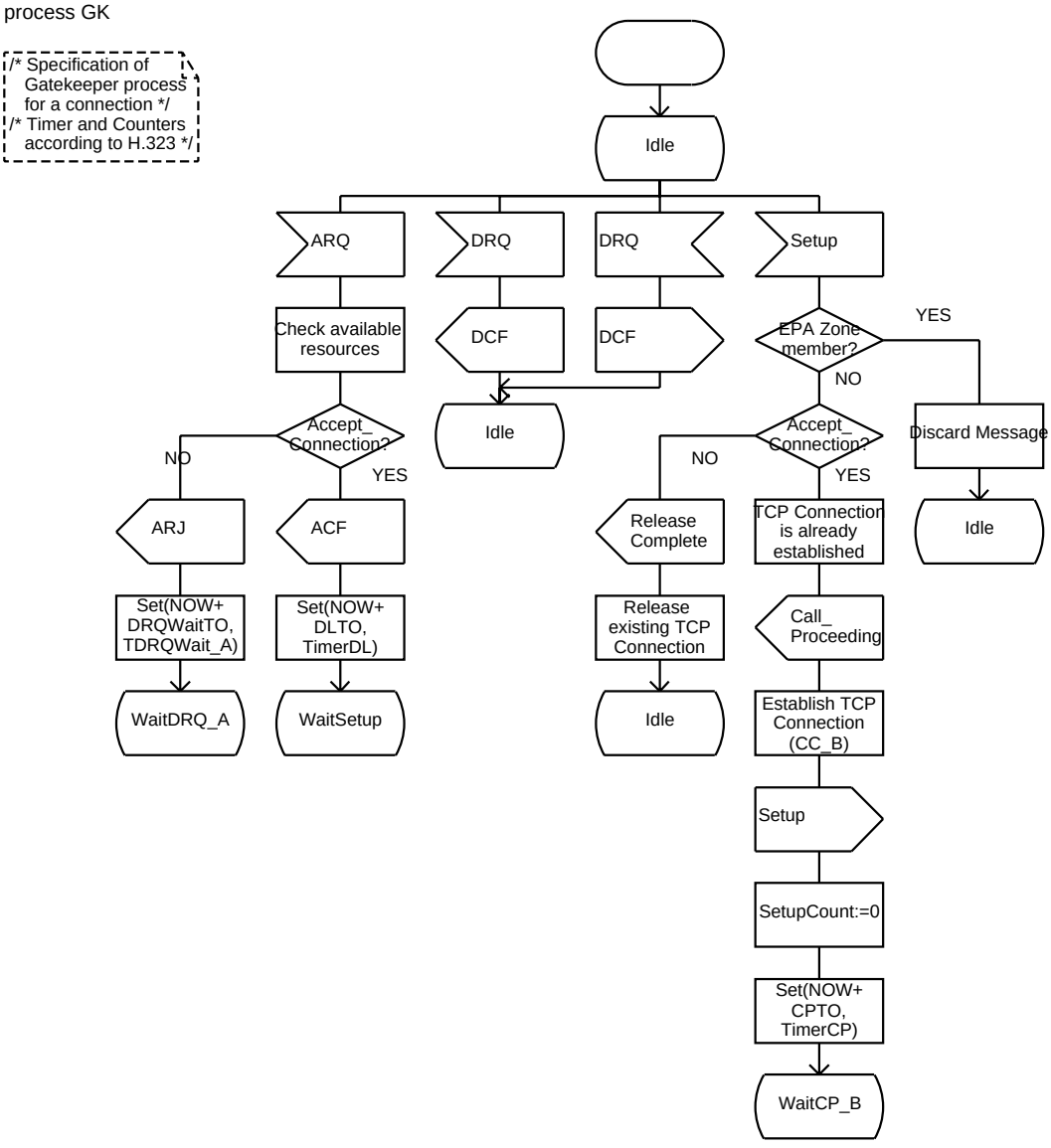
[129] ZHU, H. ; YANG, T. ; ZHENG, Q. ; WATSON, D. ; IBARRA, O. H. ; SMITH, T.: „Adaptive Load Sharing for Clustered Digital Library Servers“, *Proceedings of the 7th IEEE International Symposium on High Performance Distributed Computing (HPDC-7)*, Chicago, 1998, pp. 235-242

Anhang A

SDL-Diagramme

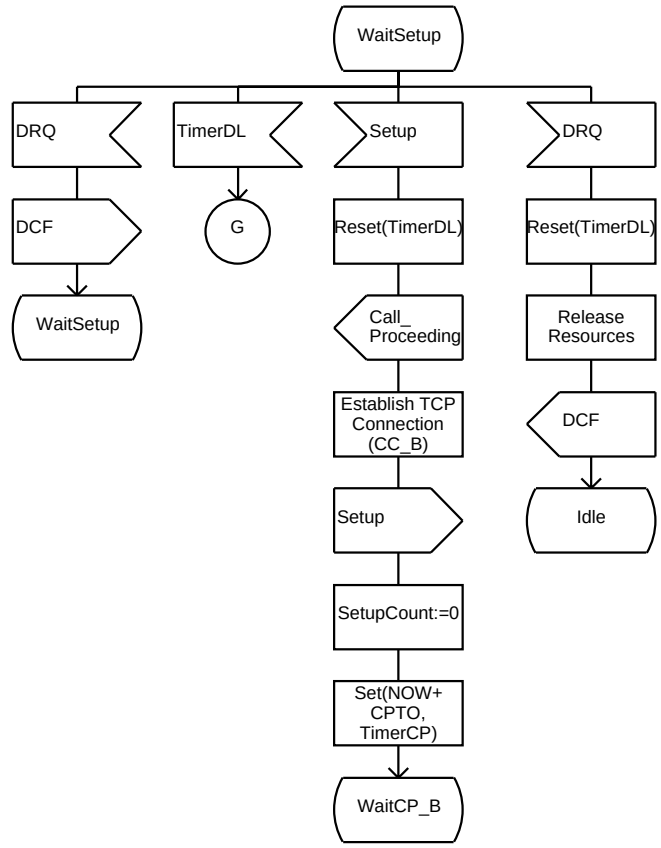
A.1 Spezifikation des Verbindungssteuerungsprozesses innerhalb des Gatekeepers

process GK 1(9)



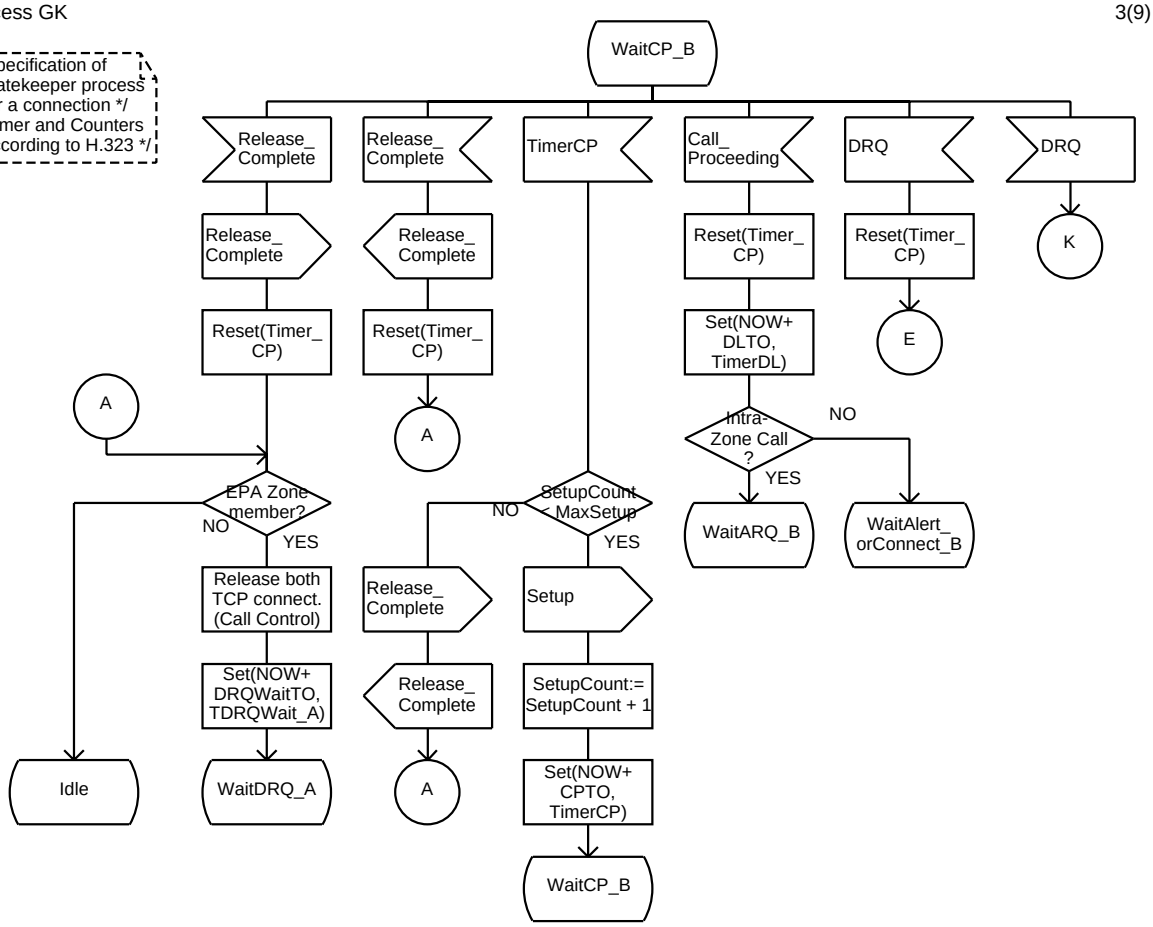
process GK 2(9)

/* Specification of Gatekeeper process for a connection */
/* Timer and Counters according to H.323 */

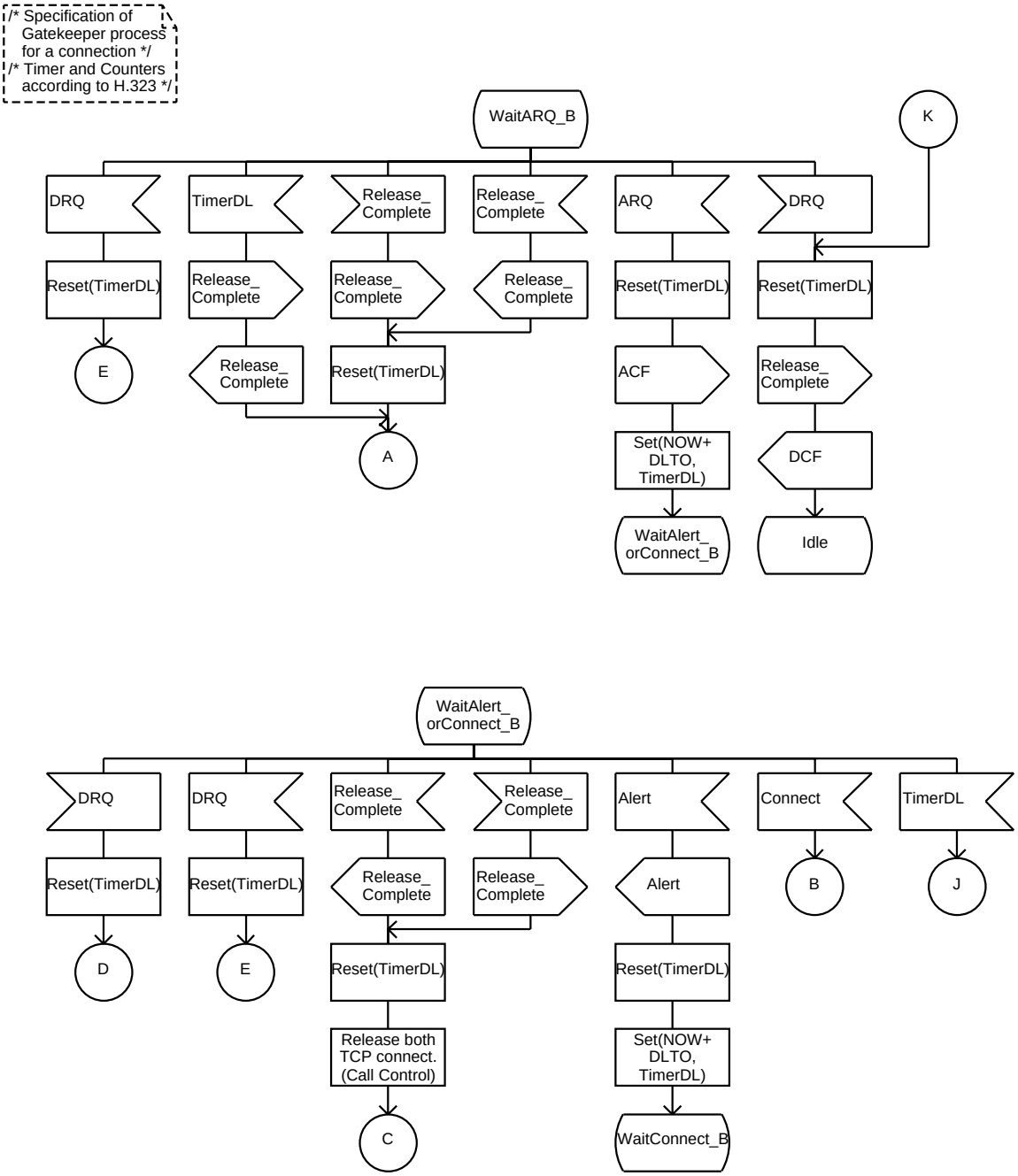


process GK 3(9)

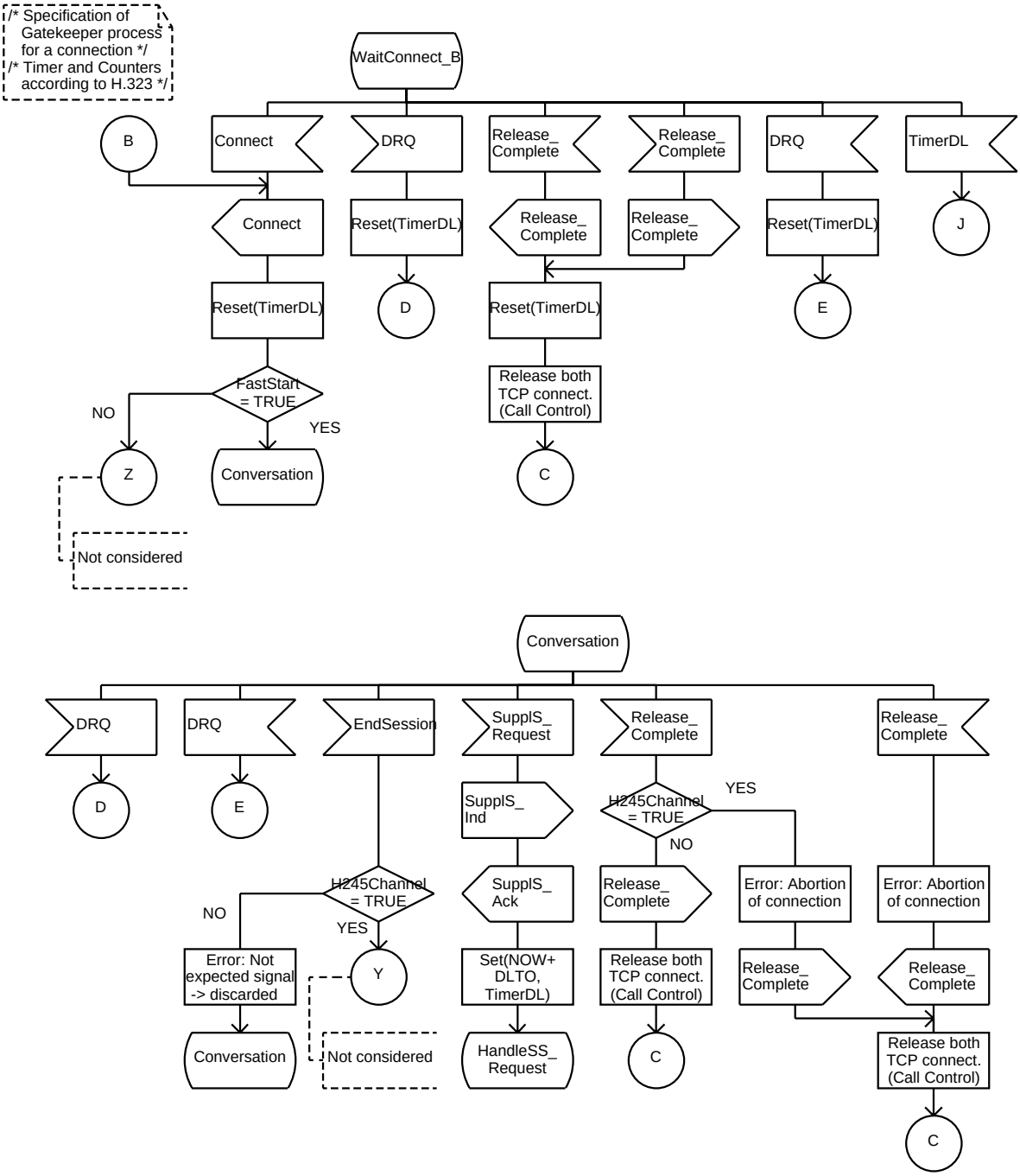
/* Specification of Gatekeeper process for a connection */
/* Timer and Counters according to H.323 */



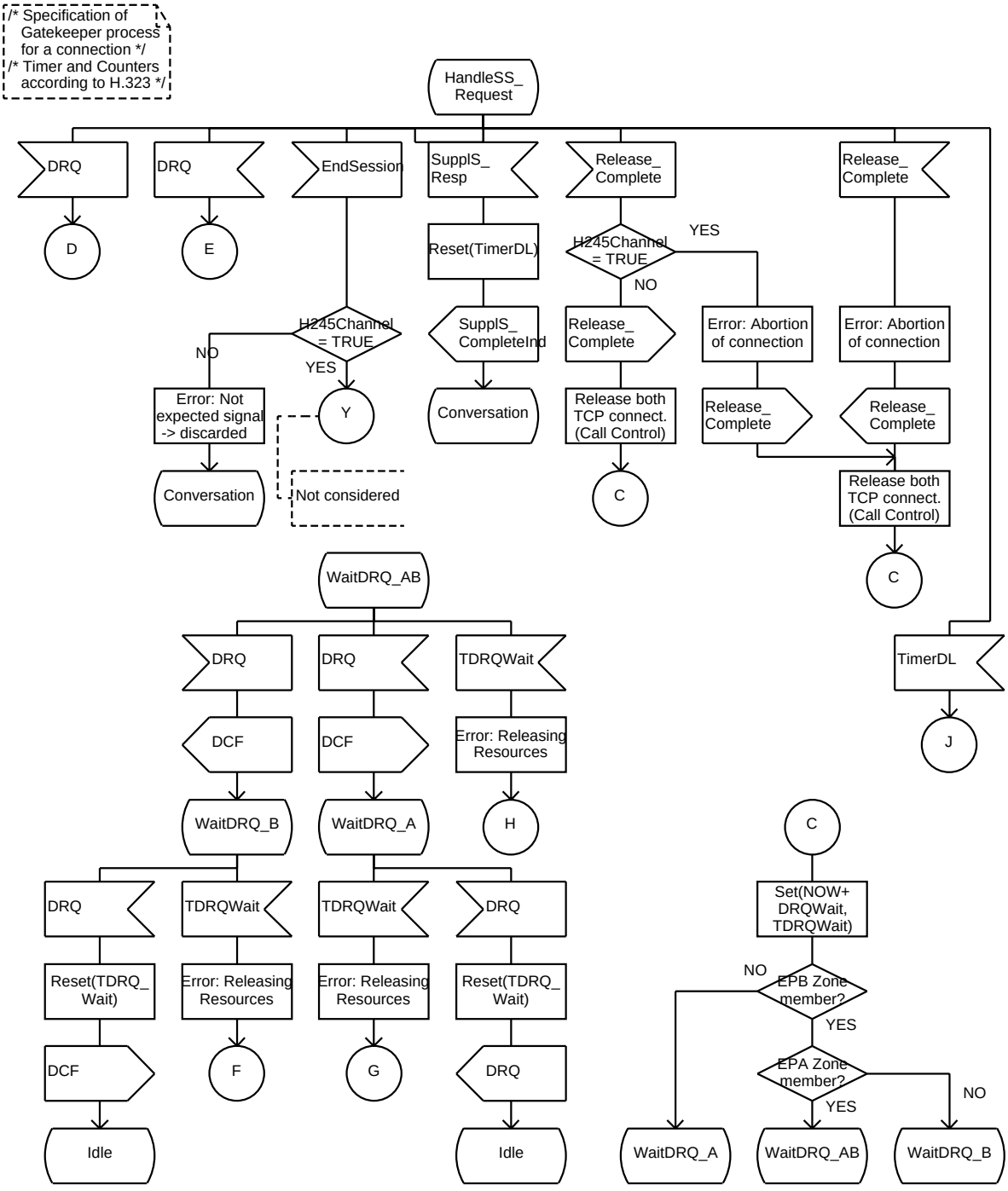
process GK 4(9)



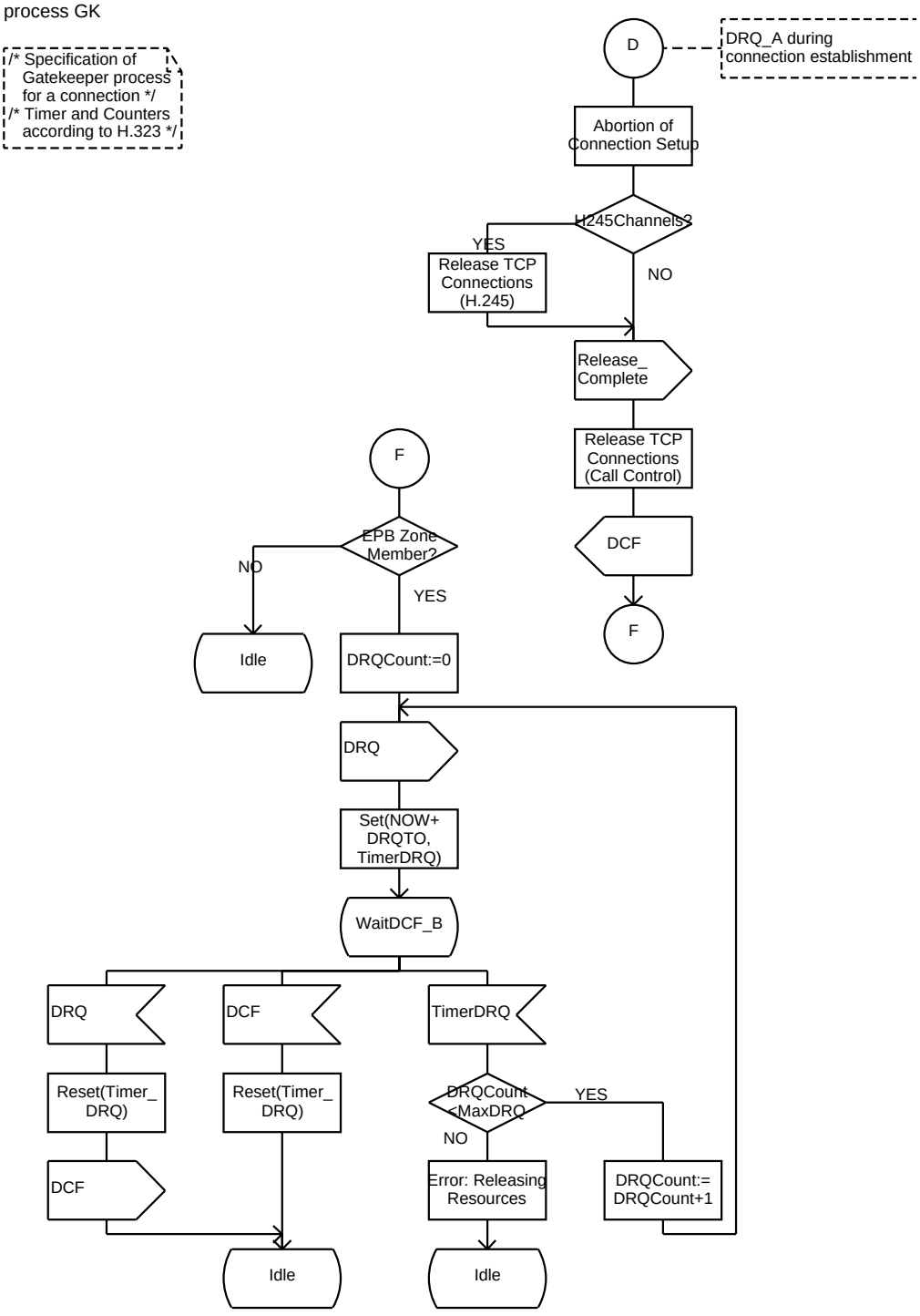
process GK 5(9)



process GK 6(9)

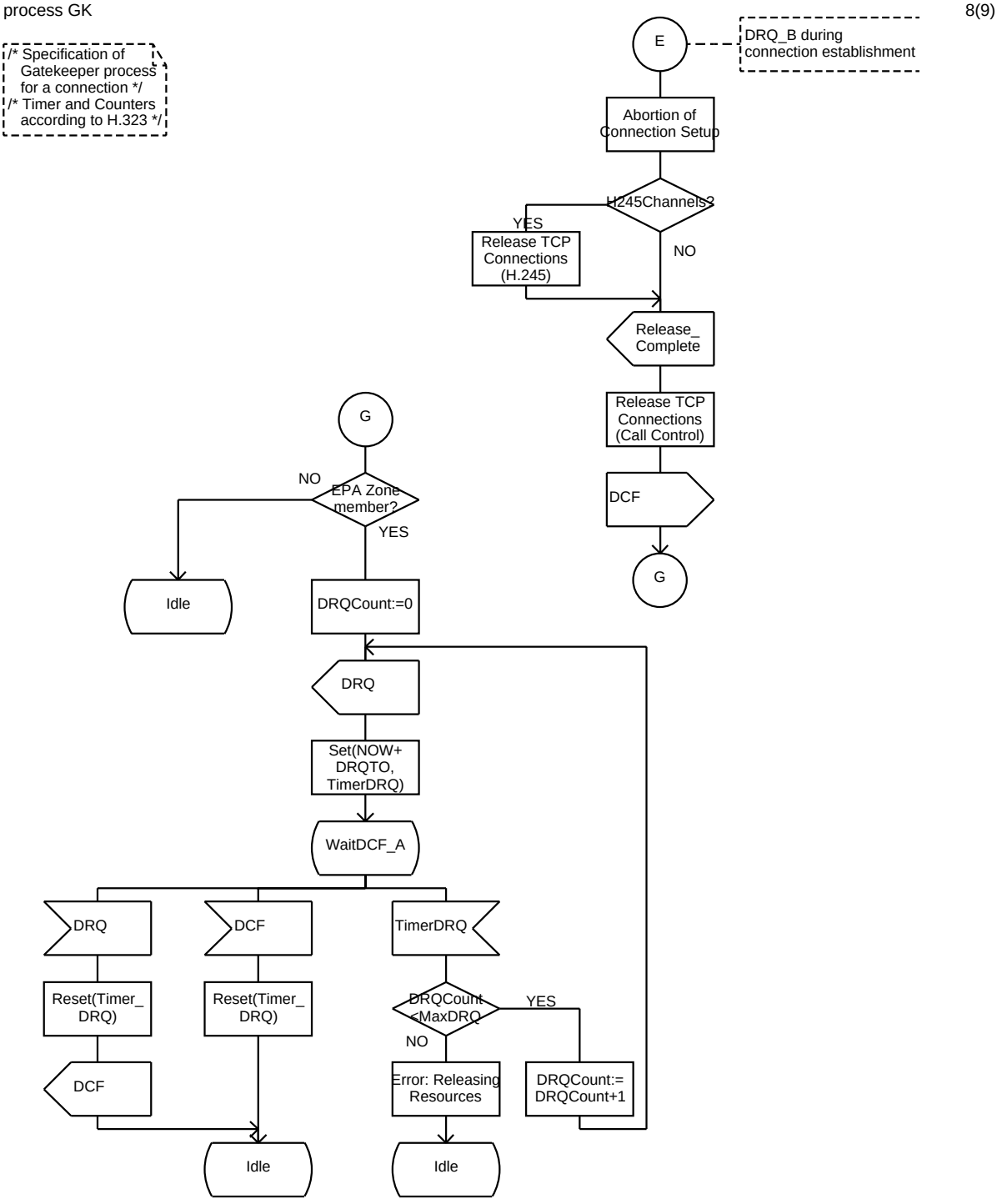


process GK 7(9)

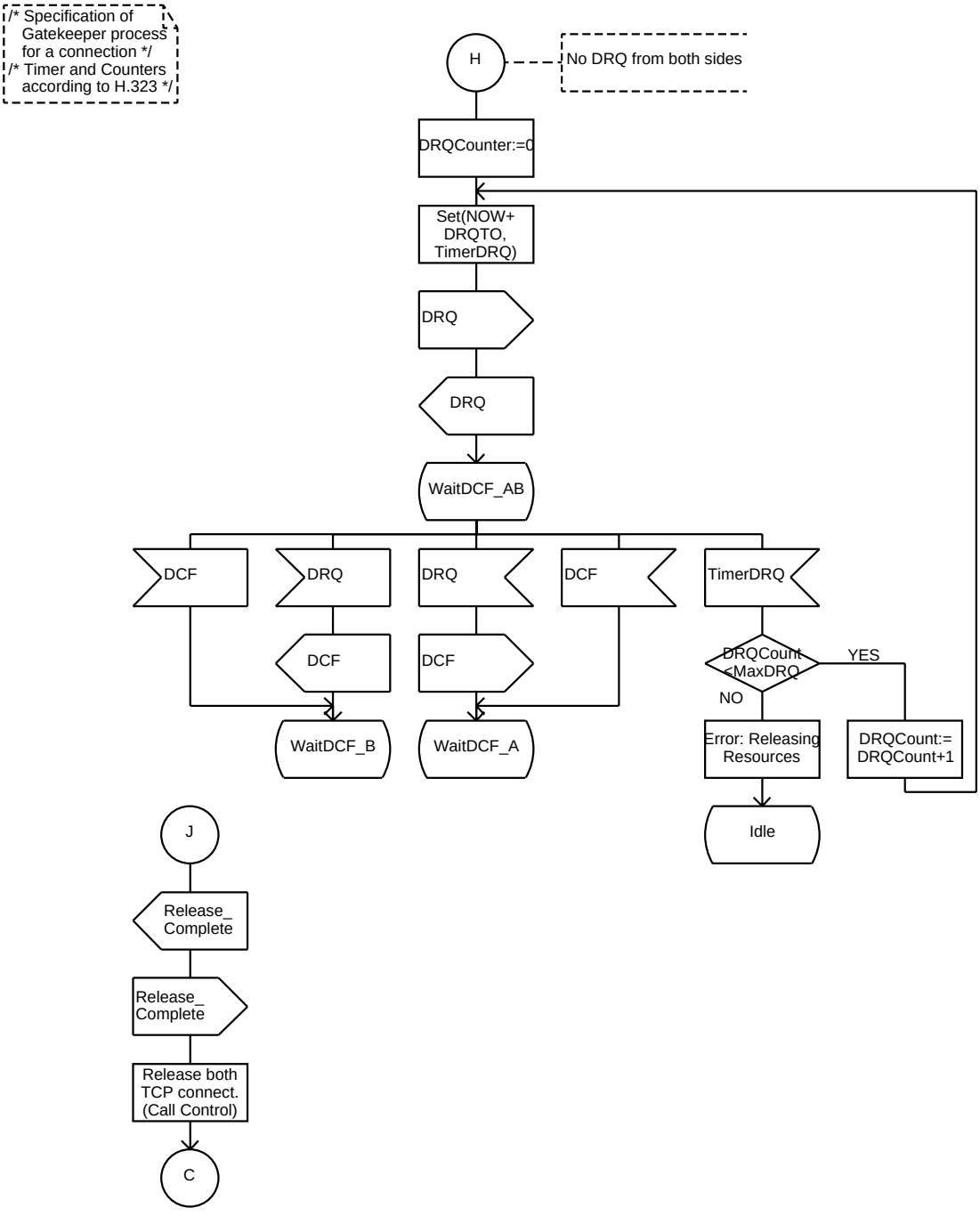


process GK

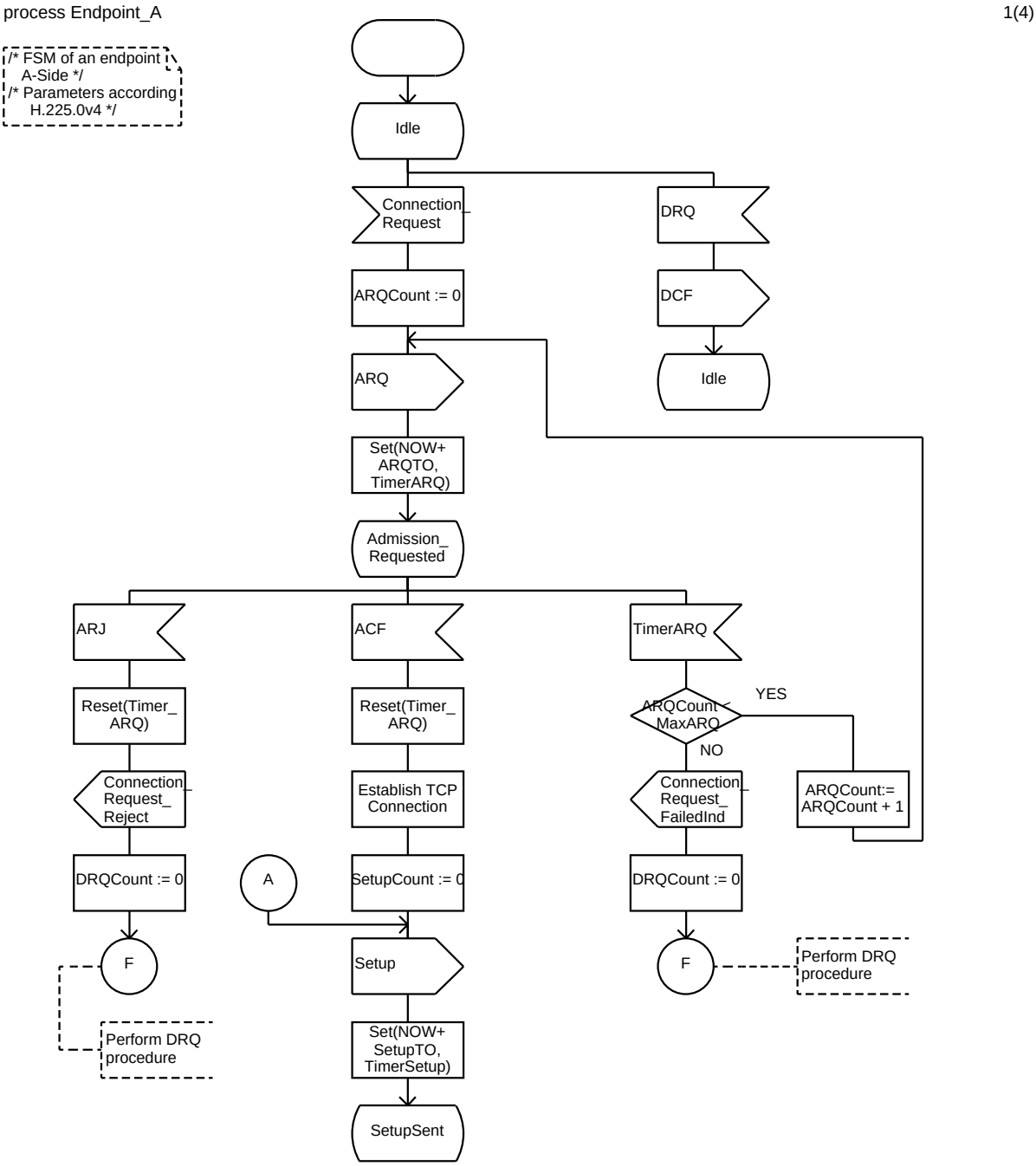
8(9)

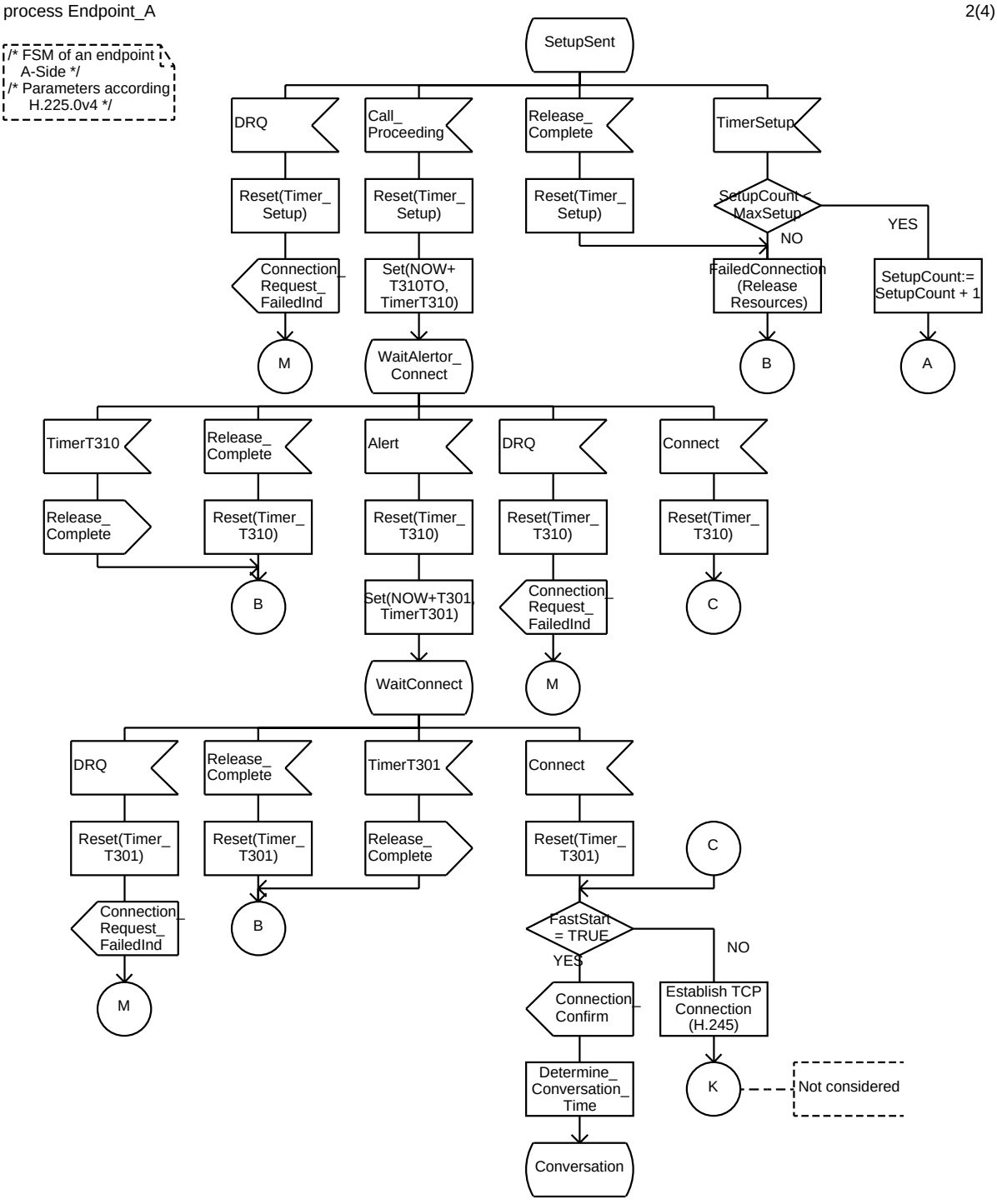


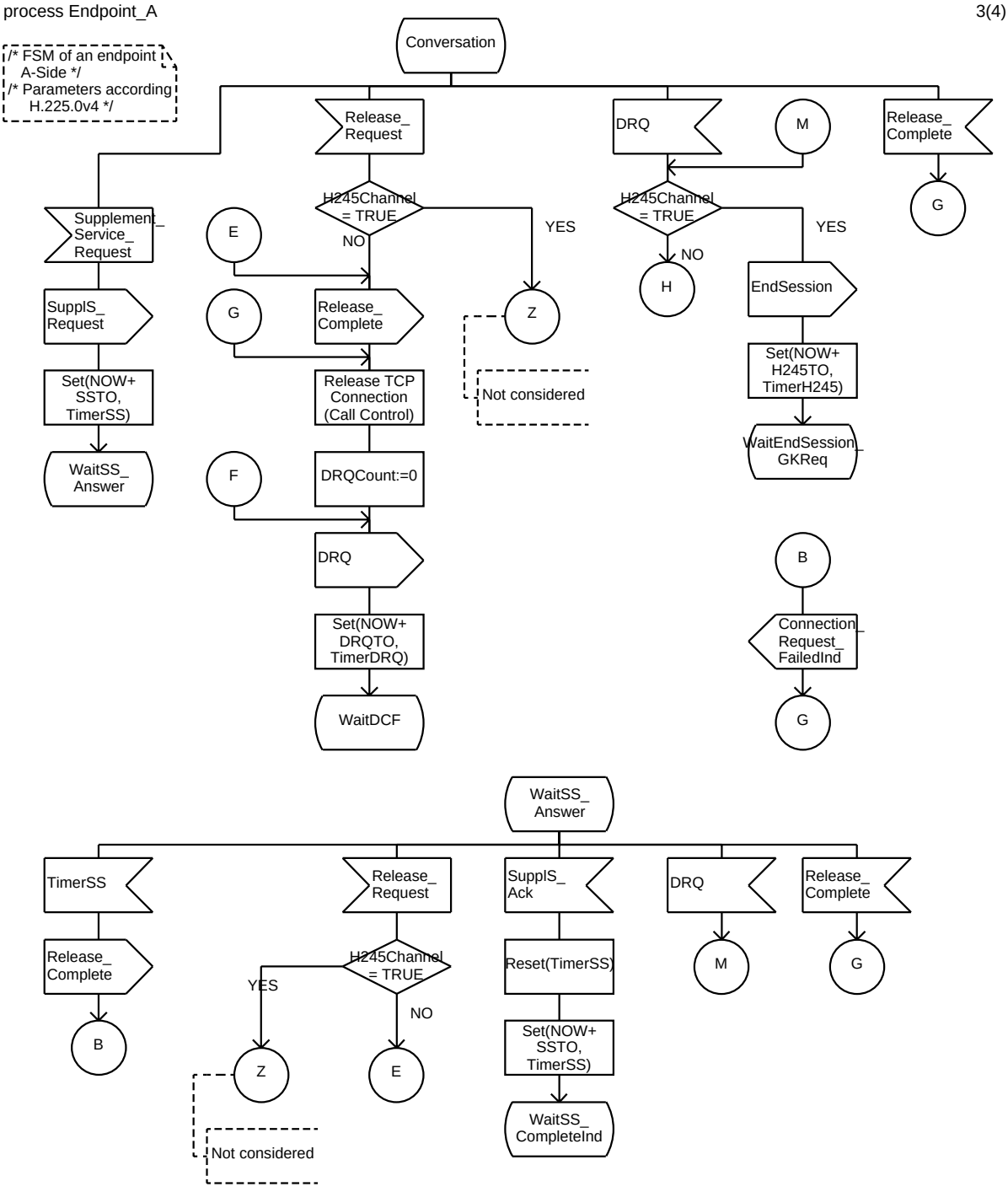
process GK 9(9)



A.2 Spezifikation des Verbindungssteuerungsprozesses innerhalb des Endpunkts A




process Endpoint_A
2(4)
/* FSM of an endpoint A-Side */
/* Parameters according H.225.0v4 */
SetupSent
DRQ
Call_ Proceeding
Release_ Complete
TimerSetup
Reset(Timer_ Setup)
Reset(Timer_ Setup)
Reset(Timer_ Setup)
SetupCount < MaxSetup
YES
NO
Connection_ Request_ FailedInd
Set(NOW+ T310TO, TimerT310)
FailedConnection (Release Resources)
SetupCount:= SetupCount + 1
M
WaitAlertor_ Connect
B
A
TimerT310
Release_ Complete
Alert
DRQ
Connect
Release_ Complete
Reset(Timer_ T310)
Reset(Timer_ T310)
Reset(Timer_ T310)
Reset(Timer_ T310)
B
Set(NOW+T301, TimerT301)
Connection_ Request_ FailedInd
C
WaitConnect
M
DRQ
Release_ Complete
TimerT301
Connect
Reset(Timer_ T301)
Reset(Timer_ T301)
Release_ Complete
Reset(Timer_ T301)
C
Connection_ Request_ FailedInd
B
FastStart = TRUE
NO
YES
M
Connection_ Confirm
Establish TCP Connection (H.245)
Determine_ Conversation_ Time
K
Not considered
Conversation


process Endpoint_A
3(4)
/* FSM of an endpoint A-Side */
/* Parameters according H.225.0v4 */
Conversation
Release_ Request
DRQ
M
Release_ Complete
G
Supplement_ Service_ Request
SupplS_ Request
Set(NOW+ SSTO, TimerSS)
WaitSS_ Answer
E
G
F
H245Channel = TRUE
NO
YES
Release_ Complete
Release TCP Connection (Call Control)
DRQCount:=0
DRQ
Set(NOW+ DRQTO, TimerDRQ)
WaitDCF
Z
Not considered
H245Channel = TRUE
NO
YES
H
EndSession
Set(NOW+ H245TO, TimerH245)
WaitEndSession_ GKReq
B
Connection_ Request_ FailedInd
G
WaitSS_ Answer
TimerSS
Release_ Complete
B
Release_ Request
H245Channel = TRUE
YES
NO
Z
E
Not considered
SupplS_ Ack
Reset(TimerSS)
Set(NOW+ SSTO, TimerSS)
WaitSS_ CompleteInd
DRQ
M
Release_ Complete
G

process Endpoint_A 4(4)