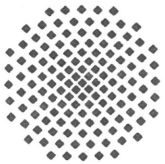


Universität Stuttgart

Institut für Nachrichtenvermittlung und Datenverarbeitung

Prof. Dr.-Ing. P. Kühn

52. Bericht über verkehrstheoretische Arbeiten

KOPPLUNG VON KOMMUNIKATIONSNETZEN: ARCHITEKTUREN, LEISTUNGSUNTERSUCHUNGEN UND EINE BEISPIELREALISIERUNG

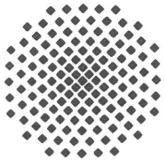
von

Martin Bosch

1992

Druck: E. Kurz & Co., Druckerei + Reprografie GmbH., Stuttgart

ISBN 3-922403-62-X



University of Stuttgart

Institute of Communications Switching and Data Technics

Prof. Dr.-Ing. P. Kühn

52th Report on Studies in Congestion Theory

INTERCONNECTION OF COMMUNICATION NETWORKS: ARCHITECTURES, PERFORMANCE EVALUATIONS, AND A REALIZATION EXAMPLE

by

Martin Bosch

1992

Summary

Especially during the last ten years a great variety of communication networks has evolved due to missing standards during their development times or due to the optimization for specific applications. This heterogeneity makes the development towards Computer Integrated Manufacturing (CIM) more difficult. One research field to overcome this problem is the *interconnection of communication networks* by Interworking Units (IWUs), which is the subject of this report. The state of the art of IWUs, their architectures, traffic models, and performance evaluations, as well as the realization example for a specific interconnection problem are dealt with in Chapters 3 to 5 of this report.

Chapter 1 Introduction

One of the most important features of modern society is the exchange of information among all its members. Communication networks are an adequate means to carry this information over large distances. Besides the traditional telephone network, other networks, especially for data communication, are becoming more and more indispensable. Today we have to provide the means to interconnect all the evolving networks, such as Local Area Networks (LANs), Wide Area Networks (WANs) or Metropolitan Area Networks (MANs). The main focus of this report is the interconnection of LANs with other networks (LANs, MANs, WANs).

Chapter 2 Fundamental Principles of Communication Networks

In this chapter the technical terms used throughout this report are introduced. The framework for communication networks from the International Organization for Standardization (ISO) is explained and protocol profiles are presented.

Chapter 3 Architectural Aspects of Internetworking

This chapter deals with IWU architectures and the building of large internetworks. The state of the art, documented in hundreds of references, is reviewed and the various architectures are classified systematically. The purpose of this chapter is to be a guide for network planners, who wish to learn the typical qualitative properties of different IWUs, together with their advantages and disadvantages for specific kinds of use, in order to make correct strategic decisions.

Chapter 4 Traffic Models and Performance Evaluations

The main part of the report is this Chapter 4, which deals with the modelling of IWUs and of some details, as well as quantitative investigations. In the first step the necessary tools from the fields of modelling, traffic simulation, and traffic theory are assembled. Concerning nonstationary simulations, an interesting phenomenon is shown which has not been detected up to now.

It can be seen that there are configurations in which protocol mechanisms can degrade the performance, although the opposite is intended. The causes for this behaviour are clarified and rules for the reasonable use of such mechanisms are established. Different design alternatives concerning the locations of protocol mechanisms, implementation variants, and adjustment of parameters are compared to each other. Some modifications of protocol mechanisms (depending on adaptive timers or the state of an IWU) are suggested, which improve the performance, provide an overload control for the IWU, and reduce its needed buffer space.

For another class of IWUs the related traffic model is analyzed by mathematical methods. An approximate iterative algorithm is presented, which takes into account the limited buffer space of the IWU, which is subdivided dynamically between several queues. The validation of the analytical results is done with the help of a universal simulation program, which has been developed especially for this purpose. Moreover, an exact mathematical solution for a specific selection of parameters is given.

Chapter 5 MAP–Gateway as a Gate to an Open Communication in a Factory

An adequate migration strategy is very important for modern companies, which have used proprietary communication networks in their factories and now want to integrate new, standardized components into their existing environment. Central elements in this strategy are MAP–Gateways, which are specific IWUs to solve this problem. The performance evaluation, design, and realization of such a MAP–Gateway is subject of this chapter. The way to do the performance evaluation and the realization is also representative for the design of other IWUs.

Chapter 6 Conclusion and Outlook

The results obtained are summarized in this last chapter which is completed by an outlook toward future research topics in the field of internetworking.

Inhaltsverzeichnis

Abkürzungen

ACSE	Association Control Service Element
AE	Application–Entity
AFI	Authority and Format Identifier
AP	Application–Process
AP	Automation Protocol
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
BMFT	Bundesministerium für Forschung und Technologie
BPDU	Bridge Protocol Data Unit
Catanet	Concatenated network
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CIM	Computer Integrated Manufacturing
CNMA	Communications Network for Manufacturing Applications
Codec	Coder/decoder
CS	Circuit Switching
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
D	Deterministic
D–Bit	Delivery Confirmation Bit
DFN	Deutsches ForschungsNetz
DNA	DIGITAL Network Architecture
DQDB	Distributed Queue Dual Bus
DSP	Domain Specific Part
E	Entwarnschranke
EHKP	Einheitliche Höhere KommunikationsProtokolle
EPA	Enhanced Performance Architecture
EPHOS	European Procurement Handbook for Open Systems
ES	End System
ESPRIT	European Strategic Programme for Research and Development in Information Technology
ESH	End System Hello
FDDI	Fiber Distributed Data Interface
FMS	Fieldbus Message Specification
FTAM	File Transfer, Access and Management
G	General
GOSIP	Government Open Systems Interconnection Profile
GreatSPN	Great Stochastic Petri Nets

HSLAN	High Speed Local Area Network
IDI	Initial Domain Identifier
IDP	Initial Domain Part
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IS	International Standard
IS	Intermediate System
ISH	Intermediate System Hello
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
IWU	InterWorking Unit
LAN	Local Area Network
LLC	Logical Link Control
M	Markovian
MAC	Media Access Control
MAN	Metropolitan Area Network
MAP	Manufacturing Automation Protocol
MIB	Management Information Base
MMS	Manufacturing Message Specification
MO	Managed Object
OSI	Open Systems Interconnection
PAI	Protocol Addressing Information
PBX	Private Branch EXchange
PCI	Protocol Control Information
PDU	Protocol Data Unit
PROFIBUS	**PRO**cess **F**ield **BUS**
PS	Packet Switching
RD	ReDirect
SAP	Service Access Point
SDL	Functional Specification and Description Language
SDU	Service Data Unit
SINEC	**SI**emens **NE**tzwerk Ar**C**hitektur für Automatisierung und Engineering
SNA	Systems Network Architecture
SRT	Source Routing Transparent
TCP	Transmission Control Protocol
TOP	Technical and Office Protocols
VLSI	Very Large Scale Integration
W	Warnschranke
WAN	Wide Area Network

Formelzeichen

Die typische Verwendung einiger Formelzeichen soll am Beispiel der kontinuierlichen Zufallsvariablen T_{Zi} für ein Zeitintervall dargestellt werden, so daß es anschließend genügt, nur die Zufallsvariablen selbst anzugeben:

T_{Zi}	Zeitintervall
$F_{Zi}(t) = \mathrm{P}\{T_{Zi} \leq t\}$	Verteilungsfunktion (Wahrscheinlichkeit, daß $T_{Zi} \leq t$ ist)
$f_{Zi}(t) = F'_{Zi}(t)$	Verteilungsdichtefunktion
$\Phi_{Zi}(s)$	Laplace–Transformierte der Verteilungsdichtefunktion
$\mathrm{E}[T_{Zi}] = z_i$	Mittelwert oder erstes gewöhnliches Moment
c_{Zi}	Variationskoeffizient
$\lambda_{Zi} = 1/z_i$	Ankunftsrate, falls T_{Zi} einen Ankunftsabstand repräsentiert
$\mu_{Zi} = 1/z_i$	Bedienrate, falls T_{Zi} eine Bedienzeit repräsentiert
$\varrho_{Zi} = \lambda/\mu_{Zi}$	Auslastung einer Bedieneinheit

A_B	Angebot an einer Bridge
B	Verlustwahrscheinlichkeit
$\hat{B}(y)$	Bedingte Verlustwahrscheinlichkeit
C	Proportionalitätsfaktor
$\delta(t - t_0)$	Dirac–Impuls zur Zeit $t = t_0$
$i,\ j,\ k$	Indizes, Laufvariablen oder Anzahlen
N_B	Pufferspeicherplätze in einer Bridge
N_E	Pufferspeicherplätze in einer Bearbeitungseinheit
N_{Si}	Pufferspeicherplätze in einer Netzeinheit
n	Anzahl
Ω_E	Mittlere Warteschlangenlänge in einer Empfangswarteschlange
Ω_{Si}	Mittlere Warteschlangenlänge in einer Sendewarteschlange
$p(x) = \mathrm{P}\{X = x\}$	Zustandswahrscheinlichkeit (eindimensional)
$\hat{p}(x \mid y) = \mathrm{P}\{X = x \mid Y = y\}$	Bedingte Zustandswahrscheinlichkeit (eindimensional)
$\tilde{p}(x, y) = \mathrm{P}\{X = x, Y = y\}$	Zustandswahrscheinlichkeit (zweidimensional)
p_{a_i}	Wahrscheinlichkeit für i Ankünfte
p_{b_i}	Wahrscheinlichkeit für i Bedienungen (Übertragungen)
p_{wi}	Weiterbearbeitwahrscheinlichkeit
p_{ei}	Anteil des Externverkehrs
q_i	Verzweigungswahrscheinlichkeit
S_E	Warteplätze in einer Empfangswarteschlange

S_{Si}	Warteplätze in einer Sendewarteschlange
s	Komplexe Variable einer Laplace–Transformierten
$\sigma(t - t_0)$	Sprungfunktion mit Sprung zur Zeit $t = t_0$
T_{Ai}	Ankunftsabstand an einer Empfangswarteschlange
T_B	Durchlaufzeit durch eine Bridge
T_E	Ersatzbedienzeit einer Bearbeitungseinheit
T_F	Filterzeit
T_H	Bedienzeit
T_{Si}	Sendezeit
T_W	Weiterbearbeitzeit
T_{Ui}	Umschaltzeit
T_{Vi}	Abwesenheitszeit (Vacation Time) der Sendeberechtigung
t	Zeit
t_0	Zeitpunkt
X_B	Zufallsvariable für den Zustand einer Bridge
X_E	Zufallsvariable für den Zustand einer Bearbeitungseinheit
X_{Si}	Zufallsvariable für den Zustand einer Netzeinheit
x	Ausprägung einer Zufallsvariablen
Y	Zufallsvariable für eine Anzahl
y	Ausprägung einer Zufallsvariablen

Kapitel 1

Einleitung

1.1 Kommunikationsnetze zum Austausch von Informationen

Die moderne Gesellschaft wird geprägt von Kommunikationsakten aller Art zwischen ihren Mitgliedern. Der Austausch von Informationen ist deshalb eines der wichtigsten Merkmale zwischenmenschlicher Beziehungen. Zu den herkömmlichen Möglichkeiten, dies auch über größere Entfernungen hinweg zu tun, haben sich vor allem im nun zuendegehenden 20. Jahrhundert solche auf elektrischer und optischer Basis gesellt. Die sich entwickelnden Kommunikationsnetze wurden zunächst im wesentlichen zur Sprachkommunikation verwendet. Das *Telefonnetz* ist heute die größte Maschine der Welt. Im inzwischen angebrochenen Informationszeitalter ist aber auch der Bedarf an Möglichkeiten zum Austausch von Daten immer akuter geworden, so daß neben dem Telefonnetz separate *Datennetze* entstanden sind. Ein Kommunikationsnetz zeigt sich dem Benutzer durch die angebotenen *Dienste* wie Fernsprechen (Telefon), Datenkommunikation (Datex), Fernschreiben (Telex), Bürofernschreiben (Teletex), Fernkopieren (Telefax), Bildschirmtext (Btx) und andere.

Je nach geographischer Ausdehnung unterscheidet man *Weitverkehrsnetze* (Wide Area Networks, WANs) und *Lokale Netze* (Local Area Networks, LANs) beziehungsweise *Nebenstellenanlagen* (Private Branch EXchanges, PBXs). Während WANs öffentliche Netze sind, welche in der Regel von den nationalen Postverwaltungen betrieben werden, sind die beiden anderen in privatem Besitz und dürfen die Privatgrundstücke, auf welchen sie installiert sind (beispielsweise Firmengelände oder Universitäts–Campus), nicht verlassen. Dabei sind Nebenstellenanlagen vor allem zur (kontinuierlichen) Sprachkommunikation und Lokale Netze vorwiegend zur (büschelförmigen) Datenkommunikation geeignet. Lokale Netze arbeiten typischerweise mit einer Übertragungsgeschwindigkeit von einigen $MBit/s$, was für manche

neuere Anwendungen (beispielsweise Bewegtbildkommunikation) nicht ausreichend ist. Deshalb wird zur Zeit intensiv an *Lokalen Hochgeschwindigkeitsnetzen* (High Speed Local Area Networks, HSLANs) geforscht, welche Übertragungsgeschwindigkeiten von 100 $MBit/s$ und mehr haben. Sie eignen sich auch dazu, im öffentlichen Bereich eingesetzt zu werden, um verschiedene Zentren einer Firma oder Universität innerhalb einer Stadt oder sogar in einem noch größeren Gebiet miteinander zu verbinden. Man spricht dann von *Nahverkehrsnetzen* (Metropolitan Area Networks, MANs).

Die vor allem in den letzten Jahren entstandene Vielfalt der Kommunikationsnetze und ihre Heterogenität (aufgrund fehlender Standards während der Entwicklung oder wegen der Optimierung für bestimmte Einsatzfälle) erschwert die Integration aller Datenverarbeitungsanlagen eines Unternehmens in Büro und Fabrik. Diese wird aber zur Verbesserung der Flexibilität und Effektivität immer wichtiger. Um dem Ziel einer computerintegrierten Fertigung (Computer Integrated Manufacturing, CIM) näher zu kommen, lassen sich zwei Tendenzen beobachten:

- Damit zukünftige multifunktionale Endgeräte an einem Netz angeschlossen werden können, und um mit einem Minimum an Kabel auszukommen, ist man bemüht, viele unterschiedliche Dienste auf *einem* Netz anzubieten, wodurch die Anforderungen an dieses Netz relativ groß werden. Analog dazu wird zur Zeit im öffentlichen Bereich das diensteintegrierende Digitalnetz (Integrated Services Digital Network, ISDN) eingeführt.
- Die verschiedenen existierenden Netze müssen gekoppelt werden, um die Kommunikation zweier Partner an verschiedenen Netzen, und damit ihre Kooperation, zu ermöglichen.

Die im zweiten Punkt angesprochene *Netzkopplung* mit Hilfe von *Netzkoppeleinheiten* (Inter-Working Units, IWUs) ist Gegenstand der vorliegenden Arbeit. Die Wichtigkeit und Aktualität dieser Thematik läßt sich eindrucksvoll an der Vielzahl von Veröffentlichungen [21, 50], Sonderheften von Fachzeitschriften [211, 212, 213] und einer Sammlung von Veröffentlichungen [214] ablesen, von welchen der größte Teil in den letzten fünf Jahren entstanden ist. Im Literaturverzeichnis dieser Arbeit sind nur die zitierten Referenzen enthalten, welche den kleineren Teil der Gesamtliteratur darstellen. Eine nette Satire zur Einstimmung in die Problematik der Netzkopplung, welche auch als historischer Rückblick betrachtet werden kann, ist in [137] enthalten.

1.2 Ziele der Arbeit

Ziele dieser Arbeit sind die folgenden Aspekte des *Protocol Engineering*:

- Systematische Darstellung des Standes der Technik auf dem Gebiet der Netzkopplung und Klassifikation der verschiedenen Architekturen.
- Erörterung der Einsatzmöglichkeiten von unterschiedlichen Netzkoppeleinheiten für die *Netzplanung.*
- Entwicklung charakteristischer Verkehrsmodelle für die Leistungsuntersuchung typischer Netzkoppeleinheiten.
- Untersuchung der Auswirkung von Protokollmechanismen in gekoppelten Netzen.
- Aufstellen von Regeln für die Optimierung von einstellbaren Parametern.
- Sinnvolle Modifikation mancher Protokollmechanismen zur Verbesserung der Leistungsfähigkeit oder zur Überlastabwehr.
- Bereitstellen von universell einsetzbaren mathematischen oder simulativen Analyseprogrammen zur Untersuchung konkreter Kopplungsprobleme und zur vergleichenden Bewertung unterschiedlicher Entwurfsalternativen.
- Leistungsuntersuchung und prototypische Realisierung einer speziellen Netzkoppeleinheit.

Dabei liegt der Schwerpunkt dieser Arbeit auf der Kopplung von Lokalen Netzen mit anderen Kommunikationsnetzen (LAN, MAN, WAN). Die Kopplung verschiedener Weitverkehrsnetze ist vor allem Gegenstand von Empfehlungen des Comité Consultatif International Téléphonique et Télégraphique (CCITT) und soll deshalb hier nicht speziell untersucht werden. Viele allgemeine Aussagen dieser Arbeit gelten jedoch auch für die Kopplung von WANs.

1.3 Übersicht über die Arbeit

Nach diesen einführenden Worten über das Umfeld und die Motivation dieser Arbeit soll nun zur besseren Orientierung der rote Faden aufgezeigt werden, welcher sich durch die restlichen Kapitel zieht:

- Im Kapitel 2 werden die im Rahmen dieser Arbeit verwendeten Fachbegriffe eingeführt und der von den Standardisierungsgremien inzwischen vorgegebene Rahmen für Kommunikationsnetze erläutert.
- Das Kapitel 3 beschäftigt sich mit der Architektur von Netzkoppeleinheiten und dem Aufbau von komplexen Netzen. Hier werden vor allem die Erkenntnisse aus dem Studium der umfangreichen Literatur systematisch aufbereitet und verschiedene Kriterien zur Klassifikation eingeführt. Dieses Kapitel ist insbesondere für Netzplaner gedacht, welche daraus die typischen Eigenschaften und Einsatzfälle unterschiedlicher Netzkoppeleinheiten, sowie deren Vor- und Nachteile entnehmen können.

- Die Modellierung typischer Netzkoppeleinheiten oder einzelner Details sowie deren Leistungsuntersuchung ist Gegenstand des Kapitels 4. Dazu müssen zunächst die benötigten Hilfsmittel aus den Gebieten Modellierungstechnik, Verkehrssimulation und Nachrichtenverkehrstheorie zusammengestellt werden. Insbesondere wird hier auf ein interessantes Phänomen bei der instationären Simulation hingewiesen, welches bisher noch nicht beobachtet wurde.

  Es wird gezeigt, daß manche Protokollmechanismen bei bestimmten Konfigurationen das Gegenteil dessen bewirken, was beabsichtigt ist, und die Leistungsfähigkeit verschlechtern. Die Ursachen dafür werden herausgearbeitet und Regeln für den sinnvollen Einsatz solcher Mechanismen aufgestellt. Unterschiedliche Entwurfsalternativen bezüglich der Lage von Protokollmechanismen, der Art ihrer Realisierung und der Einstellung der Parameter werden miteinander verglichen. Es werden Modifikationen von Protokollmechanismen (mit Hilfe von adaptiven Zeitbegrenzungen oder in Abhängigkeit vom Zustand einer Netzkoppeleinheit) vorgestellt, welche die Leistungsfähigkeit verbessern oder die Netzkoppeleinheit vor einer Überlastung schützen und ihren Pufferspeicherbedarf reduzieren.

  Für eine andere Art von Netzkoppeleinheiten wird das dazugehörende Verkehrsmodell mit Hilfe mathematischer Methoden analysiert. Es wird ein approximativer iterativer Algorithmus vorgestellt, welcher den dynamisch zwischen verschiedenen Warteschlangen aufgeteilten, begrenzten Pufferspeicher der Netzkoppeleinheit berücksichtigt. Die Ergebnisse werden mit Hilfe eines dafür entwickelten universellen Simulationsprogramms validiert. Für einen Spezialfall wird eine exakte Lösung angegeben.

- Im Kapitel 5 wird ein spezielles Kopplungsproblem betrachtet. Nach einer Leistungsuntersuchung wird auf die systemtechnische Realisierung eines Prototyps eingegangen. Es werden anschließend Verifikationsaspekte erwähnt und ein Ausblick auf Erweiterungen für das Netzmanagement gegeben.

- Die gewonnenen Ergebnisse dieser Arbeit werden schließlich im Kapitel 6 nocheinmal zusammengefaßt und um einen Ausblick auf zukünftige Forschungsschwerpunkte auf dem Gebiet der Netzkopplung ergänzt.

Kapitel 2

Grundlagen von Kommunikationsnetzen

2.1 Grundbegriffe

In diesem Abschnitt sollen Grundbegriffe eingeführt werden, welche in den späteren Kapiteln dieser Arbeit verwendet und dort als bekannt vorausgesetzt werden.

2.1.1 Netztopologie

Kommunikationsnetze können mit unterschiedlichen Topologien aufgebaut werden, welche in reiner Form meist nur bei relativ kleinen Netzen vorkommen. Solche elementaren Topologien sollen zunächst gemeinsam mit typischen Beispielen aufgezählt werden:

- Die *sternförmige Topologie* wird bei einer Nebenstellenanlage verwendet, welche die zentrale Vermittlungsfunktion für die angeschlossenen Teilnehmer wahrnimmt.
- Bei *maschenförmigen Topologien* ist eine vollständige oder unvollständige Vermaschung möglich. Die Zentralvermittlungsstellen im öffentlichen Fernsprechnetz der Deutschen Bundespost sind vollständig vermascht.
- Die *ringförmige Topologie* eignet sich besonders zum Aufbau von Glasfasernetzen, da benachbarte Stationen Punkt–zu–Punkt miteinander verbunden werden können.
- Die *linien–* oder *busförmige Topologie* wird bei Lokalen Netzen verwendet und erlaubt eine relativ einfache, passive Stationsankopplung.
- Die *baumförmige Topologie* kann durch Verbindung mehrerer Lokaler Netze mit linienförmiger Topologie entstehen. Bei einem breitbandigen Medium nimmt die Wurzel

des Baumes die Aufgabe der Remodulation wahr, das heißt das Regenerieren ankommender Signale und ihr Umsetzen von der Sendefrequenz auf die Empfangsfrequenz. Dies ist wegen gerichteten analogen Verstärkern auf dem Medium notwendig.

Bei großen Netzen sind in der Regel Mischformen dieser elementaren Topologien zu beobachten. Sie sind meist hierarchisch strukturiert, wobei auf jeder Netzebene eine andere Topologie sinnvoll sein kann. Ein typisches Beispiel ist die in Bild 2.1 dargestellte Kopplung von unterschiedlichen Netzen, wie LANs oder Nebenstellenanlagen, über ein ringförmiges Hintergrundnetz.

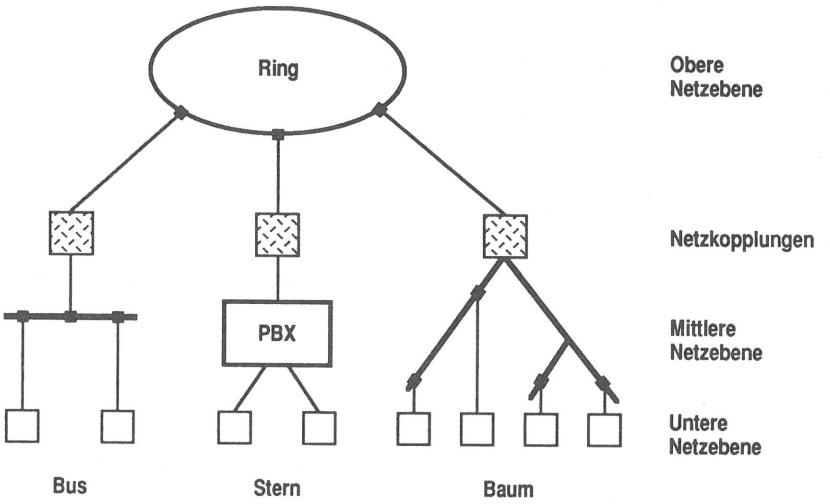


Bild 2.1: Verschiedene elementare Topologien in einem großen Netz

2.1.2 Übertragungstechnik

Als *Medien* für den Aufbau der obigen Topologien kommen vor allem verdrillte Zweidrahtleitungen, Koaxialkabel und Glasfasern in Frage.

Für die Darstellung eines Bits auf dem Medium wird bei LANs im *Basisband* eine störunempfindliche Variante der Manchestercodierung verwendet, bei welcher jede Null durch einen Signalwechsel am Bitanfang repräsentiert wird. Durch einen weiteren Signalwechsel in der Mitte eines jeden Bits kann der Empfänger leicht den Takt regenerieren, und der Gleichanteil des Signals verschwindet im Mittel. Weil dabei die Taktfrequenz doppelt so groß ist wie die Bitfrequenz, wird bei MANs, welche ausschließlich Glasfasermedien mit dem Träger Licht

verwenden, stattdessen beispielsweise ein Byte durch zwei Zeichen mit je 5 Bit codiert. Die Redundanz, welche nach dem Abzug von Steuerzeichen noch übrigbleibt, wird so gewählt, daß auch hier eine Taktrückgewinnung möglich ist. Bei der *Breitbandtechnik* wird die Information auf einen Träger moduliert, beispielsweise unter Verwendung eines Frequenzumtastverfahrens [169].

Durch die Verwendung geeigneter Multiplextechniken kann ein Medium von mehreren Kommunikationsbeziehungen gleichzeitig benützt werden [131]:

- Beim *Kanalmultiplex* steht jeder Beziehung eine garantierte Bandbreite zur Verfügung, welche beim *Raummultiplex* durch eine eigene Leitung, beim *Frequenzmultiplex* durch ein bestimmtes Frequenzband und beim *synchronen Zeitmultiplex* durch eine feste Zeitlage in einem Pulsrahmen garantiert wird.
- Beim *Adreßmultiplex* wird eine Kommunikationsbeziehung nicht mehr durch ihre Zeitlage identifiziert. Stattdessen werden die Nachrichten in Blöcke aufgeteilt, welche in ihrem Kopf die Adresse des Empfängers oder die Identifikation der dazugehörenden Verbindung (siehe Abschnitt 2.1.5) enthalten. Die Blöcke mehrerer Kommunikationsbeziehungen können auf dem Medium beliebig gemischt werden. Beim *asynchronen Zeitmultiplex* haben die Blöcke eine konstante Größe, und sie folgen unmittelbar aufeinander [63]. Sie werden hier als *Zellen* bezeichnet. Dabei kann man zur Synchronisation beispielsweise statistisch verteilte Leerzellen, einen unterlagerten Pulsrahmen oder Informationen aus den Zellköpfen verwenden. Beim *Paketmultiplex* werden Blöcke mit variabler Länge verwendet, welche in dieser Arbeit als Pakete bezeichnet werden. Da zwischen den einzelnen Paketen unterschiedlich lange Pausen auftreten, muß jedes Paket mit einem Synchronisationsmuster beginnen (Präambel), damit auf jedes einzeln synchronisiert werden kann.

2.1.3 Vermittlungsverfahren

Bezüglich der Vermittlungstechnik kann man zwei grundsätzliche Verfahren unterscheiden, welche eng mit der verwendeten Multiplextechnik verbunden sind:

- Bei der *Durchschaltevermittlung* (Circuit Switching, CS) wird einer Verbindung (siehe Abschnitt 2.1.5) während ihrer Dauer ein Kanal mit garantierter Bandbreite exklusiv zur Verfügung gestellt (Kanalmultiplex). Das hat den Nachteil, daß während einer Übertragungspause der Kanal nicht von einer anderen Verbindung verwendet werden kann und so das Medium nicht optimal ausgenützt wird. Die Durchschaltevermittlung eignet sich vor allem für die Sprachkommunikation.
- Bei der *Speichervermittlung* erfolgt die Übertragung abschnittsweise aufgrund von Informationen im Nachrichtenkopf. Solange das abgehende Medium anderweitig belegt ist,

ist eine Zwischenspeicherung notwendig. Normalerweise werden die Nachrichten nicht am Stück, sondern in Form von einzelnen Paketen übermittelt (Paketmultiplex). Man spricht dann von einer *Paketvermittlung* (Packet Switching, PS). Die Paketvermittlung eignet sich vor allem für die Datenkommunikation.

Neben den reinen Vermittlungsverfahren, gibt es auch die *hybride Vermittlung*, welche auf einem Pulsrahmen basiert, der in Zeitschlitze konstanter Länge unterteilt ist. Diese Zeitschlitze werden auf eine geeignete Art und Weise dem CS– oder PS–Verkehr zugeteilt [69, 86].

Das beim zukünftigen Breitband–ISDN vorgesehene Übermittlungsverfahren ATM (Asynchronous Transfer Mode) [63] hat sowohl Eigenschaften der Paket– als auch der Durchschaltevermittlung, wobei versucht wird, die Vorteile beider Vermittlungsprinzipien zu kombinieren. Als Multiplextechnik wird das asynchrone Zeitmultiplex verwendet, bei welchem die Zellen, unter Einhaltung der Reihenfolge innerhalb jeder Kommunikationsbeziehung, einzeln übertragen und vermittelt werden. Diese Flexibilität entspricht derjenigen der Paketvermittlung. Vor der Datentransferphase einer Kommunikationsbeziehung wird eine virtuelle Verbindung (siehe Abschnitt 2.1.5) auf der Bitübertragungsschicht aufgebaut, wobei sichergestellt werden muß, daß die neue virtuelle Verbindung unter Einhaltung der geforderten Dienstqualität (beispielsweise kein Überschreiten einer maximal zulässigen Zellverlustwahrscheinlichkeit) noch angenommen werden kann. Ist dies nicht auf dem gesamten Weg vom Sender zum Empfänger gewährleistet, so wird der Verbindungsaufbauwunsch abgelehnt. Jede Zelle trägt im Kopf eine Identifikation für ihre virtuelle Verbindung, wobei deren Bitrate während der Datentransferphase überwacht wird. Dadurch kann, wie bei der Durchschaltevermittlung, die geforderte Dienstqualität einer virtuellen Verbindung nicht durch andere virtuelle Verbindungen gefährdet werden.

2.1.4 Medienzugangsverfahren bei Lokalen Netzen

Bei paketvermittelnden Netzen muß sichergestellt werden, daß nicht zwei Stationen gleichzeitig auf denselben Übertragungskanal zugreifen können. Bei LANs und MANs wird *ein* gemeinsamer Übertragungskanal von allen Stationen genützt. Da diese Stationen gleichberechtigt sind, wird anstelle der zentralen Steuerung des Medienzugangs eine verteilte eingesetzt.

Prinzipiell lassen sich dabei Wettbewerbs– und Reservierungsverfahren unterscheiden:

- Das bekannteste Beispiel eines *Wettbewerbsverfahrens* ist bei Ethernet realisiert, welches mit leichten Modifikationen unter der Bezeichnung *CSMA/CD* (Carrier Sense Multiple Access with Collision Detection) [106] standardisiert wurde. Dabei hört jede Station das Medium ab und beginnt erst dann mit dem Senden eines Paketes, wenn das Medium als frei erkannt ist. Aufgrund von Signallaufzeiten kann es zu einer Kollision

kommen, welche durch das Mithören des eigenen gesendeten Paketes als Verfälschung erkannt wird. Eine solche Kollision wird dadurch behoben, daß der Sendevorgang abgebrochen und der erneute Sendewunsch um eine zufällige Zeit in die Zukunft verschoben wird.

- *Reservierungsverfahren* lassen sich weiter unterteilen in solche, bei denen freie Zeitschlitze eines Pulsrahmens belegt werden können und solche, bei denen eine Sendeberechtigung auf einem logischen Ring von Station zu Station weitergegeben wird. Ein Beispiel für den ersten Fall ist der PS–Teil von *DQDB* (Distributed Queue Dual Bus) [91] und Beispiele für den zweiten Fall sind *Token–Passing Bus* [107], *Token Ring* [108] und der PS–Teil von *FDDI–II* (Fiber Distributed Data Interface) [3].

2.1.5 Verbindungskonzept

Aus der Durchschaltevermittlung sind drei Phasen einer Verbindung bekannt:

- *Verbindungsaufbauphase* zur Reservierung eines Kanals zwischen Sender und Empfänger,
- *Datentransferphase* und
- *Verbindungsabbauphase* zur Freigabe des reservierten Kanals.

Bei der Paketvermittlung kann man zwischen verbindungsloser und verbindungsorientierter Kommunikation unterscheiden.

Im ersten Fall erfolgt die abschnittsweise Vermittlung durch Auswertung der Zieladresse, welche in jedem Paketkopf enthalten ist. Dabei sind unterwegs keine Betriebsmittel reserviert, so daß mit Verlusten gerechnet werden muß. Die Wege vom Sender zum Empfänger können von Paket zu Paket verschieden sein, weshalb auch die ursprüngliche Reihenfolge beim Empfänger nicht garantiert werden kann.

Bei der verbindungsorientierten Kommunikation wird, ähnlich zur Durchschaltevermittlung, vor der Datentransferphase eine *virtuelle Verbindung* aufgebaut. Dabei werden auf dem Weg vom Sender zum Empfänger die angeforderten Betriebsmittel bereitgestellt, der Verbindung auf jedem Übertragungsabschnitt eine Identifikation zugeteilt und aufeinanderfolgende Übertragungsabschnitte mit Hilfe von Tabellen einander zugeordnet. Die Pakete der Datentransferphase brauchen dann nicht mehr die vollständige Zieladresse mit sich führen, sondern es genügt die jeweilige Verbindungsidentifikation. Dadurch sind die Wege vom Sender zum Empfänger bei jedem Paket gleich, so daß die ursprüngliche Reihenfolge beim Empfänger mit Hilfe von Fehlererkennungs– und –behebungsmechanismen garantiert werden kann. Verluste können durch reservierte Betriebsmittel und Datenflußsteuerungen (siehe Abschnitt 2.2.2) auf den Verbindungen gering gehalten werden.

2.1.6 Verkehrslenkung

Bei LANs und MANs werden Pakete mit vollständiger Zieladresse ausgesandt und der jeweilige Empfänger kopiert sich die an ihn adressierten Pakete in seinen Pufferspeicher.

Ansonsten ist bei der verbindungsorientierten Kommunikation beim Verbindungsaufbau, und bei der verbindungslosen Kommunikation bei jedem Paket, eine Verkehrslenkung notwendig, um den nächsten Übertragungsabschnitt auszuwählen. Dazu wird die Zieladresse im Paketkopf ausgewertet und mit Hilfe einer Tabelle dieser Übertragungsabschnitt ermittelt.

Für den Aufbau solcher Verkehrslenkungstabellen und für ihre ständige Aktualisierung existieren unterschiedliche Algorithmen, welche teilweise auch die momentane Verkehrssituation berücksichtigen. In Abschnitt 3.3.3.1 wird auf den für LANs vorgesehenen Algorithmus etwas näher eingegangen.

Bei hierarchisch strukturierten Netzen sind in der Regel auch die Adressen hierarchisch strukturiert. Für die Verkehrslenkung brauchen dann nur die Teile der Adressen ausgewertet werden, welche der jeweiligen Netzebene zuzuordnen sind.

2.2 Das Basisreferenzmodell zur Verbindung offener Systeme

2.2.1 Das Schichtungsprinzip

Zur systematischen Gliederung der vielfältigen Kommunikationsaufgaben, und als Grundlage für die weitere Standardisierung, wurde nach siebenjähriger Arbeit von der International Organization for Standardization (ISO) 1984 das Basisreferenzmodell zur Verbindung offener Systeme (Open Systems Interconnection, OSI) als internationaler Standard (International Standard, IS) verabschiedet [96]. In diesem Modell sind sieben Schichten definiert, welche jeweils unabhängig von der darunterliegenden Schicht sind und deren Dienst durch den Austausch von *Dienstprimitiven* über einen ihrer *Dienstzugangspunkte* (Service Access Points, SAPs) in Anspruch nehmen können. Die Funktionalität des angebotenen Dienstes nimmt von Schicht zu Schicht zu, ähnlich wie dies auch aus der Rechnertechnik durch Systemprogrammschalen um die Hardware bekannt ist.

Zwischen zwei kommunizierenden *Instanzen* derselben *Schicht N* in verschiedenen Stationen wird ein *Protokoll* abgewickelt. Dazu tauschen diese Instanzen *Protokolldateneinheiten* (Protocol Data Units, PDUs) aus, siehe Bild 2.2. Diese setzen sich zusammen aus den *Dienstdateneinheiten* (Service Data Units, SDUs), welche der sendenden Protokollinstanz mit Hilfe von Dienstprimitiven übergeben wurden, und den eigentlichen *Protokollsteuerinformationen*

(Protocol Control Information, PCI) dieser Schicht. Die SDUs stellen für die betrachtete Schicht lediglich Nutzdaten dar, welche nicht weiter ausgewertet, sondern lediglich transportiert werden. Die PCI enthalten neben den *Protokolladressierungsinformationen* (Protocol Addressing Information, PAI) alle Parameter, welche zur Abwicklung des Protokolls notwendig sind. Zum physikalischen Transport der PDUs wird der Dienst der darunterliegenden Schicht in Anspruch genommen. Als Oberbegriff für PDU, SDU und Dienstprimitiv wird im Rahmen dieser Arbeit, unabhängig von der Schicht, die Bezeichnung *Paket* verwendet, wenn die Unterschiede im jeweiligen Zusammenhang nicht relevant sind; in der Literatur ist dieser Begriff traditionell auch häufig auf die Vermittlungsschicht (siehe unten) beschränkt.

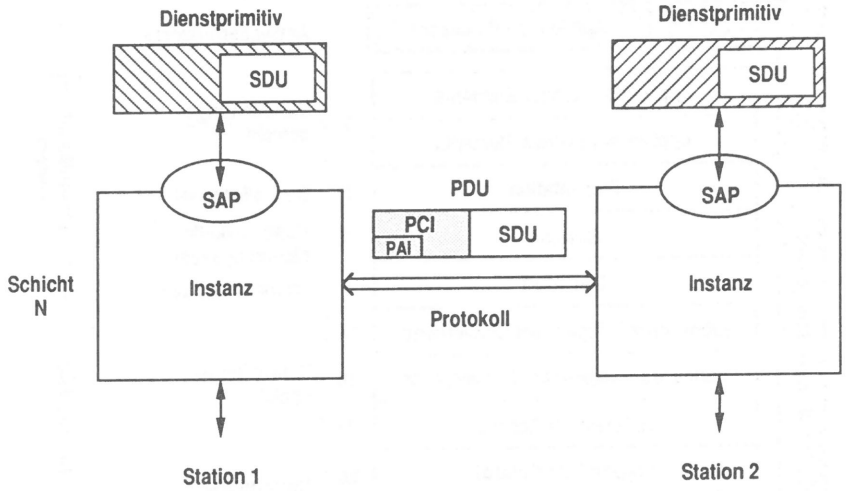


Bild 2.2: Begriffsbestimmung für das Schichtungsprinzip

Im Laufe der folgenden Jahre wurden für alle sieben Schichten von verschiedenen Gremien [71] detaillierte Standards erarbeitet und von der ISO verabschiedet, wobei für jede Schicht aufgrund unterschiedlicher Anforderungen an die Kommunikation mehrere Alternativen von Standards oder zumindest mehrere Klassen *eines* Standards existieren. Ein Beispiel dafür sind die Medienzugangsverfahren CSMA/CD, Token Ring und Token–Passing Bus bei LANs, welche gleichberechtigt nebeneinander standardisiert sind. Standards repräsentieren Protokolle, die den technischen Möglichkeiten zur Zeit ihrer Entstehung entsprechen. Um den technischen Fortschritt nicht aufzuhalten, werden laufend neue Protokolle entwickelt, welche möglicherweise später wieder in Standards münden. So wird beispielsweise in verschiedenen Standardisierungsgremien zur Zeit daran gearbeitet, obige Standards um solche für weitere Medienzugangsverfahren (für MANs) zu ergänzen.

Bei diesem fortschreitenden Standardisierungsprozeß hat sich auch gezeigt, daß die zu lösen-

den Kommunikationsaufgaben nicht gleichmäßig auf die sieben Schichten verteilt sind. Während die volle Funktionalität mancher Schichten oft nicht ausgenützt wird, hat sich in anderen Schichten, insbesondere für LANs und MANs, eine weitere Unterteilung in Teilschichten als notwendig und sinnvoll herausgestellt.

Das verfeinerte Basisreferenzmodell ist in Bild 2.3 dargestellt. Es enthält die englischen Bezeichnungen für die Schichten und Teilschichten aus den Standards. Oberhalb der Verarbeitungsschicht sind die verschiedenen Anwenderprozesse angedeutet, welche ein Kommunikationsbedürfnis haben. Einer davon gehört zum Netzmanagement. Er kann auch direkt auf lokale Objekte aller Schichten zugreifen und diese bei Bedarf modifizieren.

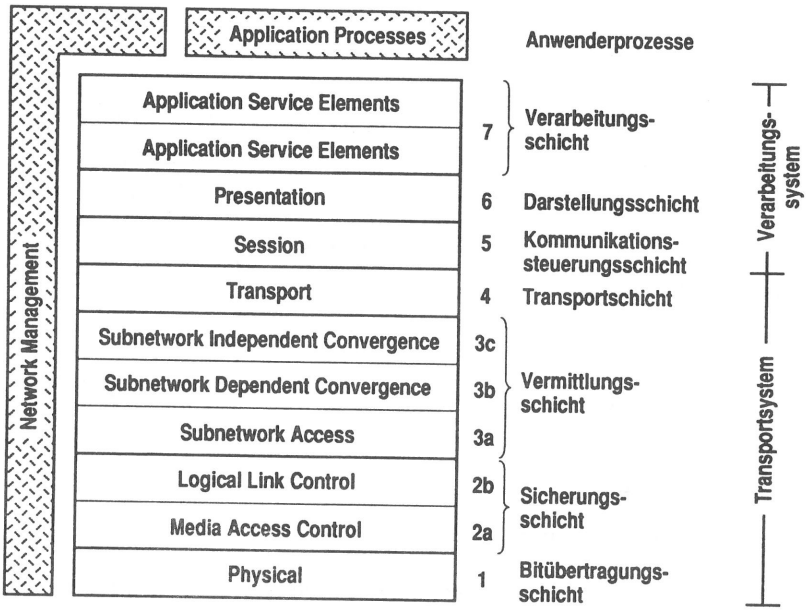


Bild 2.3: Das verfeinerte Basisreferenzmodell

Die Protokolle des Verarbeitungssystems hängen stark von dem darauf zugreifenden Anwenderprozeß ab. Deshalb kann die *Verarbeitungsschicht* verschiedene Verarbeitungsdienstelemente (Application Service Elements, ASEs) enthalten, welche jedem Anwenderprozeß die Schnittstelle bereitstellen, die er zur Kommunikation benötigt. Manche Funktionen wären in mehreren ASEs notwendig, weshalb sie als selbständige ASEs standardisiert wurden, auf welchen andere ASEs aufbauen können. Dieser Sachverhalt ist in Bild 2.3 durch zwei Teilschichten angedeutet. Als Beispiel für eine solche allgemein notwendige Funktion sei die Verbindungssteuerung erwähnt, welche als Association Control Service Element (ACSE) [102, 103] separat standardisiert wurde.

Eng mit den ASEs verbunden ist die Aufgabe der *Darstellungsschicht*. In neueren Standards für ASEs werden die PDUs mit Hilfe der Datenbeschreibungssprache ASN.1 (Abstract Syntax Notation One) [109] beschrieben. Der Implementierer eines ASEs muß diese abstrakte Syntax auf seinem System in eine lokale Syntax mit Hilfe einer problemorientierten Programmiersprache umsetzen. Dazu können teilweise auch ASN.1–Übersetzer verwendet werden [72]. Die Darstellungsschicht muß nun diese lokalen und teilweise recht komplexen Datenstrukturen in eine neutrale Transfersyntax, die konkrete Syntax, umsetzen. Dazu werden normalerweise die ebenfalls standardisierten Codier–Regeln zu ASN.1 [110] verwendet. Falls die Daten für den Transfer verschlüsselt werden sollen, bietet sich hierzu ebenfalls die Darstellungsschicht an.

Die Funktionalität der *Kommunikationssteuerungsschicht* wird in der Regel nicht voll ausgenützt. Sie kann eine Dialogsteuerung durchführen und dadurch festlegen, wer als nächstes senden darf, oder sie erlaubt die Definition von Wiederaufsetzpunkten, um bei einem Fehler im Anwenderprozeß des Empfängers, nach dessen Behebung, am letzten Wiederaufsetzpunkt mit der Übertragung fortzufahren. Letztendlich haben alle ihre Aufgaben mit der Synchronisation von Sender und Empfänger zu tun.

Die Protokolle des Transportsystems garantieren einen zuverlässigen Transport von Paketen vom Sender zum Empfänger unter Einhaltung einer geforderten Dienstqualität. Dies wird insbesondere durch das *Transportprotokoll*, welches als das unterste Protokoll mit Ende–zu–Ende–Signifikanz vorgesehen ist, mit Hilfe vieler möglicher Protokollmechanismen gewährleistet.

Die *Vermittlungsschicht* ist mittlerweile in drei Teilschichten unterteilt [101, 172], wobei die obere noch teilnetzunabhängig und insbesondere für die Verkehrslenkung zuständig ist. Die untere Teilschicht regelt den Zugang zum Teilnetz und die mittlere paßt diese beiden Teilschichten aneinander an, sofern das notwendig ist. Durch diese Konstruktion werden die einzelnen Übertragungsabschnitte miteinander verkettet. Die Vermittlungsschicht eignet sich deshalb auch insbesondere dafür, verschiedene Teilnetze miteinander zu verbinden.

Die *Sicherungsschicht* ist für LANs und MANs in zwei Teilschichten unterteilt. Ihre ursprüngliche Aufgabe, die Erkennung von Übertragungsfehlern auf einem Übertragungsabschnitt, ist in der Logical Link Control (LLC) Teilschicht [104] enthalten. Das Medienzugangsverfahren wird in der unterlagerten Media Access Control (MAC) Teilschicht realisiert.

In der *Bitübertragungsschicht* werden die mechanischen, elektrischen oder optischen Eigenschaften des Mediums und die Darstellung der Bits auf dem Medium festgelegt. Sie überträgt die ihr von der Sicherungsschicht angebotenen Pakete bitweise und muß deshalb auch eine Parallel/Seriell–Wandlung durchführen.

2.2.2 Elementare Mechanismen in Kommunikationsprotokollen

Die Protokolle des Basisreferenzmodells enthalten elementare Mechanismen, welche in verschiedenen Schichten immer wieder vorkommen und deshalb hier kurz definiert werden. Dabei soll auf Protokollmechanismen zur Fehlererbehandlung im Rahmen dieser Arbeit nicht eingegangen werden.

2.2.2.1 Protokollmechanismen zur Anpassung unterschiedlicher Paketgrößen

Außer dem einfachen Zusammenhang zwischen PDUs und SDUs, welcher in Bild 2.2 dargestellt ist, gibt es auch noch die Möglichkeit, die Paketgrößen zu ändern.

Ist eine Schicht nicht in der Lage, die ihr in Form einer SDU angebotene Paketgröße zu bearbeiten, so kann sie diese SDU durch *Aufteilen* in mehrere PDUs einbetten. Die Partnerinstanz ist für das *Vereinigen* zuständig.

Könnten auch größere Pakete verarbeitet werden als die, welche in Form von SDUs in einer Schicht ankommen, so können mehrere SDUs durch *Blocken* in *eine* PDU eigebettet werden. Die Partnerinstanz ist für das *Entblocken* zuständig.

Ist bekannt, daß die unterlagerte Schicht größere Pakete verarbeiten kann, so können für den physikalischen Transport mehrere PDUs durch *Verketten* zu *einer* SDU für diese unterlagerte Schicht zusammengefaßt werden. Die Partnerinstanz ist für das *Trennen* zuständig.

Durch Blocken und Verketten werden sowohl die Prozessorkapazitäten von Prozessoren für tiefere Schichten als auch die Fenstergrößen von Flußkontrollen dort (siehe nächster Abschnitt) effektiver ausgenützt. Beides führt zu einer Erhöhung der Stabilitätsgrenze, falls dadurch der Engpaß des Gesamtsystems beeinflußt wird. Dieser positive Effekt ist beim Blocken noch dadurch verstärkt, daß die obigen Argumente bereits für die betrachtete Schicht selbst gelten. Darüberhinaus können bei beiden Protokollmechanismen Protokollsteuerinformationen eingespart werden.

2.2.2.2 Protokollmechanismen zur Anpassung unterschiedlicher Bearbeitungs- und Übertragungsgeschwindigkeiten

Bei verbindungsorientierten Protokollen können unterschiedliche Geschwindigkeiten bei der Bearbeitung von Paketen in Sender und Empfänger sowie bei der Übertragung aneinander angepaßt werden.

Zur *Datenflußsteuerung* werden in der Regel verbindungsindividuelle Fenster–Mechanismen eingesetzt, welche es erlauben, daß eine begrenzte Anzahl noch nicht quittierter Pakete gleichzeitig unterwegs ist. Ist diese begrenzte Anzahl erreicht, so stauen sich weitere zu übertragende Pakete beim Sender, bis sich das Fenster wieder öffnet. Das üblichste Verfahren

ist hier die Folgenummern–Steuerung, bei welchem die Blindlast, durch *Sammelquittierung* und *Piggybacking* von Quittungen auf Datenpakete der Rückwärtsrichtung, reduziert werden kann. Neben diesen Fenster–Mechanismen gibt es auch noch andere Möglichkeiten zur Datenflußsteuerung, wie beispielsweise das in Abschnitt 4.3.5.1 beschriebene vorübergehende Anhalten des Senders mit Hilfe von Steuerpaketen. Anstelle von *Datenflußsteuerung* wird auch häufig der Begriff *Flußkontrolle* verwendet.

Durch *Multiplexen* mehrerer Verbindungen auf *eine* Verbindung der darunterliegenden Schicht kann eine bestehende Verbindung dieser darunterliegenden Schicht zum selben Empfänger ausgenützt werden, so daß ein erneuter Verbindungsaufbau entfallen kann. Das *Demultiplexen* erfolgt beim Empfänger. Dies bietet sich vor allem an, wenn die Verbindung der darunterliegenden Schicht sowieso noch nicht voll ausgelastet ist.

Durch *Verbindungsaufspaltung* kann die zur Verfügung stehende Bandbreite des Übertragungskanals vergrößert werden, wenn dort Engpässe bestehen. Deshalb kann sich die Stabilitätsgrenze des Gesamtsystems erhöhen, insbesondere wenn die effektive maximale Fenstergröße einer unterlagerten Flußkontrolle dadurch vervielfacht wird. Wartezeiten im Netz können sich durch die Ausnützung mehrerer paralleler Wege und Betriebsmittel reduzieren. Die *Verbindungssammlung* beim Empfänger muß von einem *Sequentialisieren* gefolgt werden, da aufgrund verschiedener Übertragungskanäle Überholungen nicht ausgeschlossen werden können. Dazu werden in der Regel die Folgenummern der Datenflußsteuerung mitverwendet.

2.2.3 Adressierungskonzept bei offenen Systemen

Im Zusammenhang mit der Adressierung müssen drei Begriffe unterschieden werden [167, 172, 184]:

- Der *Name* eines Objektes gibt an, *was* angesprochen wird,
- die *Adresse*, *wo* es sich befindet,
- und der *Weg* beschreibt, *wie* man dorthin kommt.

Eine Instanz wird durch ihren Namen eindeutig gekennzeichnet. Davon wird vor allem auf der Verarbeitungsschicht Gebrauch gemacht. Als Name wird dort der AE–Title (Application–Entity–Title) verwendet, welcher sich aus dem Namen des dazugehörenden Anwenderprozesses, dem AP–Title (Application–Process–Title), und einer weiteren Kennzeichnung (AE–Qualifier) zusammensetzt. Dabei ist jeder Verarbeitungsinstanz eindeutig *ein* Anwenderprozeß und *eine* Darstellungsadresse zugeordnet. Die Abbildung des AE–Titles auf die Darstellungsadresse erfolgt üblicherweise mit Hilfe einer Tabelle. Dadurch kann der Anwender mit logischen Namen arbeiten, während auf den für den Anwender unsichtbaren, unterlagerten Schichten vorwiegend Adressen verwendet werden.

Gemäß dem Standard [97] kennzeichnet eine Adresse eine Menge von SAPs zwischen denselben zwei Instanzen. Für den Benutzer spielt es dabei keine Rolle, über welchen der SAPs seine Kommunikation abgewickelt wird, falls mehrere zur Verfügung stehen. Jedem SAP ist eindeutig *eine* Instanz der darüberliegenden Schicht zugeordnet, wobei diese aber auch auf andere darunterliegende SAPs zugreifen darf. Dadurch kann eine Instanz mit Hilfe der Adresse der darunterliegenden Schicht eindeutig adressiert werden. Die Unterscheidung einzelner Verbindungen mit derselben Adresse erfolgt durch eine zusätzliche Identifikation.

Meistens ergibt sich die Adresse einer Schicht aus der Adresse der darunterliegenden Schicht durch Erweiterung. Die zweite Abbildung mit Hilfe einer Tabelle, neben derjenigen auf der Verarbeitungsschicht, ist in der Regel auf der Vermittlungsschicht notwendig, um die Vermittlungsadresse auf die teilnetzspezifische Adresse für den nächsten Übertragungsabschnitt abzubilden.

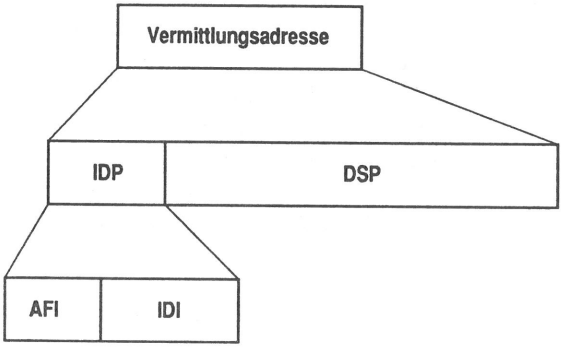


Bild 2.4: Hierarchische Stuktur der Vermittlungsadresse

Die Vermittlungsadresse ist normalerweise hierarchisch strukturiert. Im Standard [99] sind drei Felder vorgesehen, wobei der Inhalt des ersten Feldes (Authority and Format Identifier, AFI) von der ISO festgelegt wird (Beispiel: 36 für das paketvermittelnde öffentliche Datennetz, binär codiert in *einem* Byte). Es kennzeichnet die adreßvergebende Organisation (Beispiel: CCITT), das Dokument, in welchem das Format des zweiten Feldes beschrieben wird (Beispiel: CCITT–Empfehlung X.121) und dessen Länge (Beispiel: bis zu 14 Dezimalziffern), sowie Länge und Syntax des Dritten Feldes (Beispiel: 24 Dezimalziffern). Das zweite Feld (Initial Domain Identifier, IDI) kennzeichnet das Teilnetz zu welchem die teilnetzspezifische Adresse gehört und damit indirekt deren Vergabestelle, und im dritten Feld ist diese teilnetzspezifische Adresse (Domain Specific Part, DSP) enthalten. Die ersten beiden Felder zusammen werden auch als Initial Domain Part (IDP) bezeichnet, wie das in Bild 2.4 dargestellt ist. Bei LANs wird üblicherweise AFI=49 verwendet. Dabei ist kein IDI vorgesehen und für den Inhalt des DSP gibt es keine Vorschriften, so daß hier private Adressen mit einer Länge von 15 Byte verwendet werden können.

2.2.4 Netzmanagement

Unter Netzmanagement versteht man die Verwaltung eines Netzes sowohl bei der Inbetriebnahme als auch später im laufenden Betrieb. Es deckt die Bereiche Konfigurierung, Leistungsfähigkeitsüberwachung, Fehlerbehandlung, Abrechnung und Sicherheit ab. Prinzipiell kann man drei Kategorien des Netzmanagements unterscheiden [98]:

- Funktionen, welche zwar dem Netzmanagement dienen, aber *in normalen Protokollen* enthalten sind (sie beziehen sich auf *eine* Instanz),
- *Schichten–Management–Funktionen*, welche in separaten Protokollen definiert sind (sie beziehen sich auf eine ganze Schicht) und
- *System–Management–Funktionen*, welche die komfortable Verwaltung eines ganzen Netzes auf allen sieben Schichten ermöglichen.

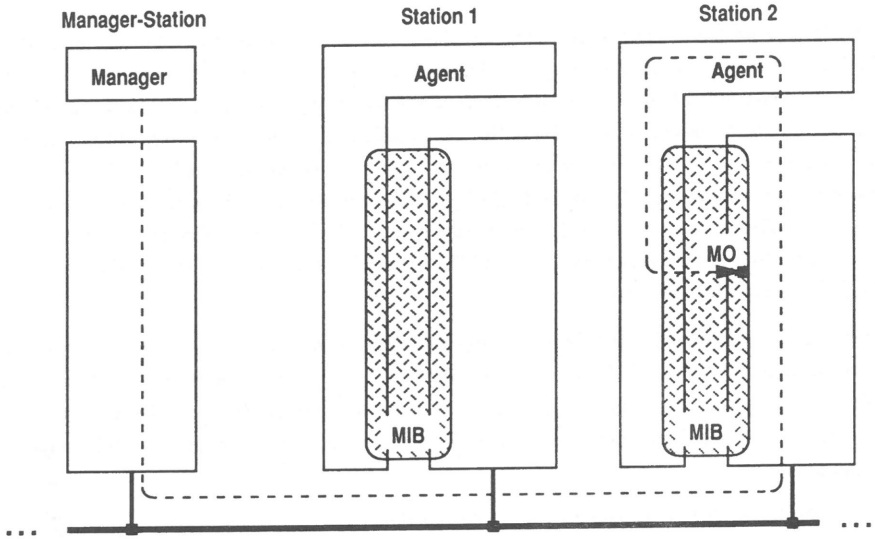


Bild 2.5: Zugriff auf ein zu verwaltendes Objekt von einem Manager–Prozeß aus

Da die letzte Kategorie bei weitem die wichtigste ist, soll darauf noch etwas genauer eingegangen werden. Zur Verwaltung eines Netzes, welches ein verteiltes System darstellt, werden *Manager–Prozesse* auf zentralen *Manager–Stationen* eingesetzt. Diese Manager–Prozesse kommunizieren mit *Agent–Prozessen*, welche in allen Stationen als Kommunikationspartner vorhanden sein müssen. Die Agent–Prozesse wiederum haben Zugriff auf ihre lokalen zu verwaltenden Objekte (Managed Objects, MOs) in allen Schichten, welche zu einer speziellen Datenbank (Management Information Base, MIB) zusammengefaßt sind. Zum Austausch von

Netzmanagement–Informationen zwischen Manager– und Agent–Prozessen wurde ein neues ASE definiert, das Common Management Information Service Element (CMISE) [113, 114]. Der genaue Aufbau der Verarbeitungsinstanz für das Netzmanagement ist beispielsweise in [34] zu finden. In Bild 2.5 ist der Mechanismus verdeutlicht, wie ein Manager–Prozeß auf ein MO in irgendeiner Schicht auf einer anderen Station zugreifen kann. Er kann solche Objekte erzeugen oder löschen sowie deren Attribute lesen oder verändern. Darüberhinaus können Agent–Prozesse von sich aus Alarmmeldungen an Manager–Prozesse verschicken, wenn bestimmte vordefinierte Ereignisse (Beispiel: Überschreitung von Schwellwerten) eingetroffen sind, und sie können auf Anforderung eines Manager–Prozesses Tests ausführen und deren Ergebnisse an diesen zurückmelden.

2.3 Protokollprofile für spezielle Anwendungen

Unmittelbar nach der Standardisierung des Basisreferenzmodells bestand das Problem, daß die internationale Standardisierung der einzelnen Protokolle, insbesondere auf den höheren Schichten, mit dem akuten Bedarf der Anwender nach Möglichkeiten zur Datenkommunikation nicht Schritt halten konnte. Als nationale Zwischenlösung der Bundesrepublik Deutschland wurden deshalb für die Zeit bis zum Vorliegen internationaler Standards Einheitliche Höhere KommunikationsProtokolle (EHKP) für die oberen vier Schichten definiert, welche die Protokollvielfalt dort einschränken sollten und vor allem in der öffentlichen Verwaltung und bei Bildschirmtext eingesetzt wurden [202]. Die Anpassung an die fortschreitende Standardisierung erfolgte stufenweise.

Mittlerweile existieren für alle Schichten des Basisreferenzmodells internationale Standards. Jetzt besteht das Problem darin, daß auf jeder Schicht mehrere unterschiedliche Standards oder zumindest unterschiedliche Klassen *eines* Standards vorliegen. Deshalb ist es normalerweise für ein beliebiges Paar standardkonformer Stationen nicht möglich, miteinander zu kommunizieren.

Dieses Problem wird für Beschaffungen im öffentlichen Bereich von den Regierungen Englands [42] und der Vereinigten Staaten von Amerika [163, 174] beispielsweise dadurch reduziert, daß sie die Spezifikation von GOSIPs (Government Open Systems Interconnection Profiles) angeordnet haben, welche die sinnvollen Kombinationen von Protokollen und Klassen [172] enthalten. Dort, wo die Standards Freiheitsgrade offen lassen, werden diese durch Implementierungsvorschriften beseitigt. Das englische GOSIP ist darüberhinaus die Basis für ein europäisches Beschaffungshandbuch mit der Bezeichnung EPHOS (European Procurement Handbook for Open Systems), an dem seit 1990 gearbeitet wird [42].

Für spezielle Anwendungen in abgeschlossenen Bereichen werden von Anwendergruppen dedizierte *Protokollprofile* festgelegt, welche möglichst keine Alternativen mehr zulassen. So

haben sich die Protokollprofile TOP (Technical and Office Protocols) für die Büroautomatisierung und MAP (Manufacturing Automation Protocol) für die Fertigungsautomatisierung herauskristallisiert. Dabei kann es auch vorkommen, daß auf einer Schicht für die betrachtete Anwendung noch kein geeignetes standardisiertes Protokoll existiert, so daß ein neues spezifiziert und standardisiert werden muß. Auf diese Art und Weise ist beispielsweise auch das Verarbeitungsdienstelement MMS (Manufacturing Message Specification) für MAP entstanden, welches mittlerweile in einen internationalen Standard [111] gemündet ist. Die Arbeiten an GOSIP, EPHOS, MAP und TOP werden eng koordiniert.

Das Protokollprofil MAP, an welchem auch in Europa im Rahmen des ESPRIT–Projektes (European Strategic Programme for Research and Development in Information Technology) CNMA (Communications Network for Manufacturing Applications) seit 1986 mitgewirkt wird, ist in Bild 2.6 neben zwei weiteren Protokollprofilen für die Fertigungsautomatisierung dargestellt. Dabei sind in jeder Schicht die wichtigsten Eigenschaften erwähnt und die wesentlichen Standards der ISO zitiert, sofern solche existieren.

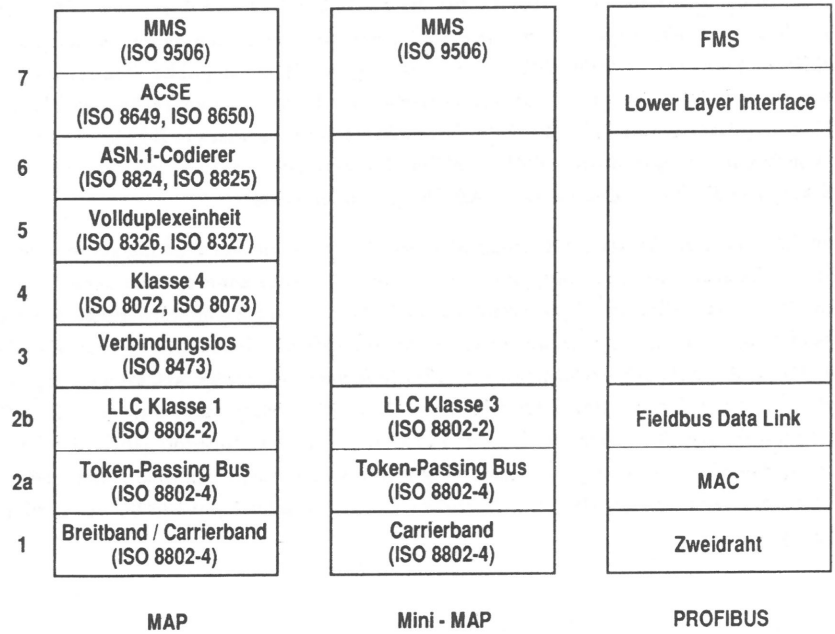


Bild 2.6: Protokollprofile von MAP, Mini–MAP und PROFIBUS

Die Implementierung aller Schichten des Basisreferenzmodells führt zu einem sehr komfortablen, aber aufgrund seiner Komplexität auch zu einem recht langsamen Kommunikationssystem. MAP in seiner vollständigen Variante kommt deshalb vor allem auf den höheren

Ebenen einer hierarchisch strukturierten Fabrik zum Einsatz. Wegen den extrem unterschiedlichen Anforderungen an die Netze auf den verschiedenen Ebenen einer Fabrik muß selbst in dem abgeschlossenen Bereich der Fertigungsautomatisierung die Idealvorstellung eines einheitlichen Protokollprofils für alle kommunizierenden Stationen aufgegeben werden. Um den Anforderungen an die Reaktionszeit auf den unteren Ebenen gerecht zu werden, wird in der MAP–Spezifikation eine Enhanced Performance Architecture (EPA) spezifiziert, welche neben dem vollständigen MAP auch noch Mini–MAP enthält. In Mini–MAP werden die Protokolle der Schichten 3 bis 6 nicht verwendet, wie das auch in Bild 2.6 dargestellt ist. Die Codierung der Daten für die Übertragung wird von der Verarbeitungsschicht in einer einfachen Form miterledigt, wobei die Datenstrukturen nicht mitübertragen werden und der Empfänger wissen muß, wie er die Daten zu interpretieren hat.

Um einzelne Sensoren und Aktoren anzusprechen, sind Medium und Anschaltung auch von Mini–MAP zu teuer. Außerdem sind die dort verwendeten Protokolle für diesen Einsatzfall nicht optimiert. Deshalb soll hier der Feldbus eingesetzt werden, welcher dieselbe Architektur wie Mini–MAP hat, also nur Protokolle für die Schichten 1, 2 und 7 enthält. Für diese Protokolle gibt es allerdings zur Zeit viele konkurrierende Vorschläge und es ist noch kein einheitlicher Standard in Sicht [201]. Der Vorschlag der Bundesrepublik Deutschland im Rahmen eines BMFT–Projektes (Bundesministerium für Forschung und Technologie) ist der PROFIBUS (**PRO**cess **FI**eld **BUS**) [120], dessen Architektur ebenfalls in Bild 2.6 dargestellt ist. Sein Verarbeitungsprotokoll FMS (Fieldbus Message Specification) ist stark an MMS angelehnt, was die Integration in eine MAP–Umgebung erleichtert.

Neben den bisher erwähnten Protokollprofilen, welche durch die Verwendung standardisierter Protokolle gekennzeichnet sind, gibt es auch noch zahlreiche andere, meist herstellerspezifische Protokollprofile, die überwiegend vor und während des Standardisierungsprozesses entstanden sind und sich mittlerweile weit verbreitet haben. Hierzu zählen beispielsweise DNA (DIGITAL Network Architecture), SINEC (SIemens NEtzwerk ArChitektur für Automatisierung und Engineering), SNA (Systems Network Architecture) und die TCP/IP–Protokollumgebung (Transmission Control Protocol / Internet Protocol) [166]. Es ist zu erwarten, daß diese Protokollprofile langfristig durch standardisierte abgelöst werden oder zu diesen migrieren. Stellvertretend für andere wird in Kapitel 5 auf SINEC etwas näher eingegangen.

Kapitel 3

Architekturaspekte bei der Netzkopplung

Nachdem die Grundlagen von Kommunikationsnetzen bereitgestellt sind, können nun die Architekturaspekte von Netzkoppeleinheiten und Netzen beleuchtet werden. In diesem Kapitel werden auch unterschiedliche Netzkoppeleinheiten vorgestellt und ihre Eigenschaften miteinander verglichen. Ein Netzplaner bekommt dadurch das notwendige Wissen vermittelt, welches er benötigt, um die richtigen Entscheidungen für bestimmte Komponenten und Netzstrukturen zu treffen. Die Implementierungsaspekte im letzten Abschnitt sind dagegen vor allem für Entwickler von Netzkoppeleinheiten gedacht.

3.1 Verschiedene Kopplungstypen

3.1.1 Voraussetzungen

Geht man bei der Kopplung zweier Netze davon aus, daß die Protokolle in diesen Netzen gemäß dem Basisreferenzmodell geschichtet sind, so ergibt sich bei einer Gegenüberstellung der Protokollprofile zweier Stationen dieser Netze die Konfiguration nach Bild 3.1. Dabei sollen die Protokolle der Schicht N–1 unterschiedlich und von der Schicht N an aufwärts identisch sein (die Protokolle der Schichten 1 bis N–2 dürfen, soweit vorhanden, identisch oder verschieden sein). Die Adressen und Namen der Protokolle oberhalb der Kopplungsschicht müssen im gekoppelten Netz eindeutig sein. Ist dies nicht der Fall, müssen vor der Kopplung in mindestens einem Netz entsprechende Umbenennungen durchgeführt werden.

Bei der Kopplung mit einem Netz, dessen Protokolle nicht gemäß dem Basisreferenzmodell geschichtet sind, kann keine Schicht N gefunden werden, auf und oberhalb derer die Protokolle in beiden Netzen identisch sind. Die Palette der Kopplungstypen beschränkt sich

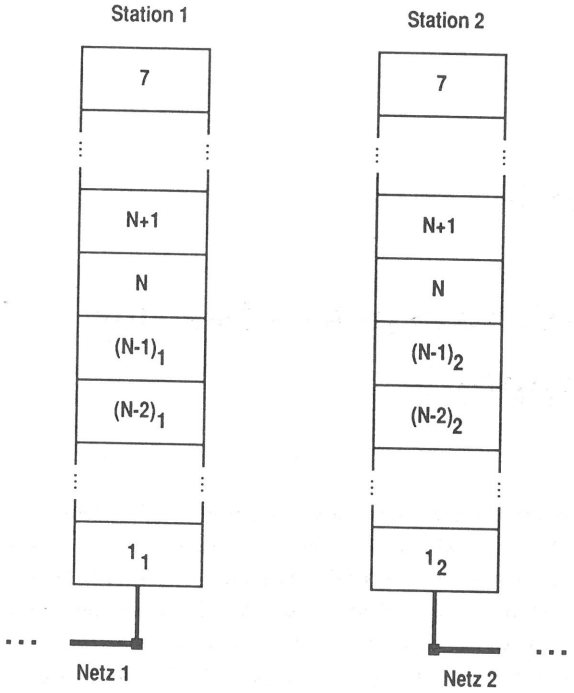


Bild 3.1: Protokollprofile zweier zu verbindender Netze

dann zum einen auf die Verwendung *eines* Netzes als Transitnetz, siehe Abschnitt 3.1.5, und zum anderen auf die Transformation von Dienstprimitiven der jeweils höchsten Schicht der Protokollprofile, was in Abschnitt 3.1.4 beschrieben wird. Ein Beispiel für den zweiten Fall ist in Kapitel 5 zu finden.

Man kann nun vier Kopplungstypen mit individuellen Vor– und Nachteilen unterscheiden, welche zum Teil auch in [196] am Beispiel einer speziellen Netzkoppeleinheit erwähnt werden. Diese Kopplungstypen werden nun in der Reihenfolge zunehmender Komplexität der Kopplungsaufgabe vorgestellt. In [23, 191] wird die Einteilung etwas anders vorgenommen. Dort wird der Schwerpunkt darauf gelegt, ob die Kopplung an einer Schichtgrenze oder innerhalb einer Schicht vorgenommen wird, was im folgenden als Kopplungstyp und seine Variante unterschieden werden soll. Die Kopplungstypen werden hier so gewählt, daß die Protokollinstanzen in der Netzkoppeleinheit identisch zu den entsprechenden Instanzen der übrigen Stationen sind. Wenn Modifikationen in Protokollinstanzen notwendig werden, so liegen Varianten von Kopplungstypen vor.

3.1.2 Netzkopplung durch transparentes Durchreichen

Der einfachste Kopplungstyp ist die Netzkopplung durch transparentes *Durchreichen* [189]. Hierzu bietet sich vor allem die Schicht N als Kopplungsschicht an; es sind dafür aber auch höhere Schichten denkbar, sofern dies aus anderen Gründen wünschenswert oder notwendig ist (beispielsweise aus Sicherheitsgründen, zur Isolation der Netze auf einer höheren Schicht). In Bild 3.2 ist dieser Kopplungstyp für die Kopplungsschicht N dargestellt. Dienstprimitive der Schicht N werden auf die jeweils andere Seite einfach durchgereicht. Erkauft wird dieser relativ einfache Kopplungstyp durch den Verlust der Ende–zu–Ende–Signifikanz des Protokolls auf der Schicht N und durch die vergleichsweise große Anzahl von Schichten in der Netzkoppeleinheit, was eine entsprechend große Transferzeit durch die Netzkoppeleinheit zur Folge hat. Die gemeinsame Schicht N in der Netzkoppeleinheit sollte (muß aber nicht) aus Aufwandsgründen als *eine* Protokollinstanz realisiert sein, welche von zwei Protokollprofilen verwendet wird.

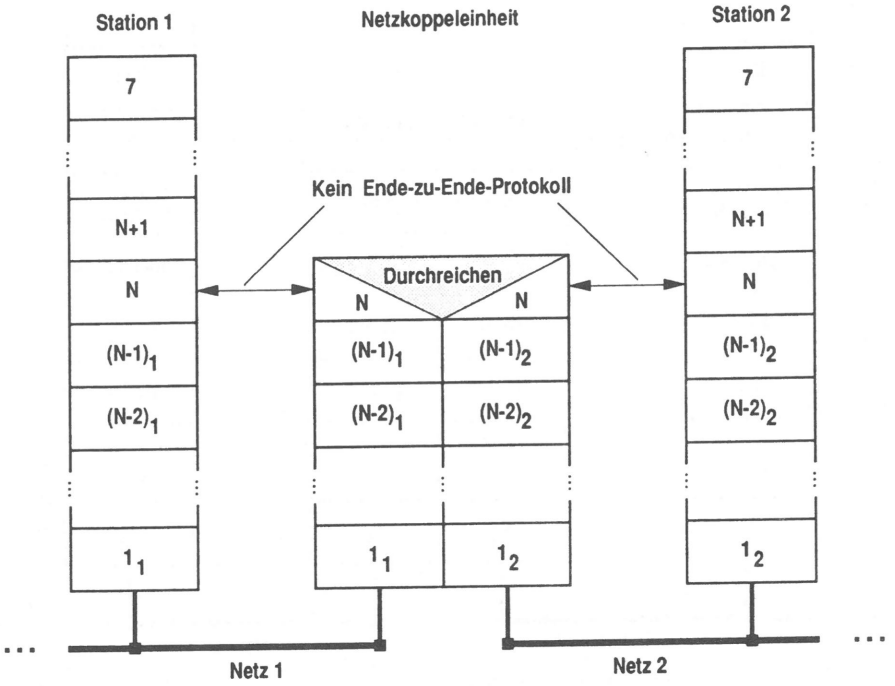


Bild 3.2: Netzkopplung durch transparentes Durchreichen

Als Variante dieses Kopplungstyps kann die Schicht–N–Instanz der Netzkoppeleinheit so modifiziert werden, daß sie keine Dienstprimitive für die (nicht existierende) Schicht N+1

erzeugt, sondern stattdessen direkt die PDUs, also Nutzdaten *und* Protokollsteuerinformationen der Schicht N, über die Durchreichesoftware zur jeweils anderen Seite durchreicht. Dazu müssen allerdings auch auf der Kopplungsschicht selbst die Adressen und Namen im gekoppelten Netz eindeutig sein. Diese Protokollmodifikation erhöht den Implementierungsaufwand [25], erspart aber unnötige Bearbeitungszeiten im Betrieb und kann bei Bedarf auch eine Ende–zu–Ende–Signifikanz des Schicht–N–Protokolls ermöglichen.

3.1.3 Netzkopplung über ein globales Protokoll

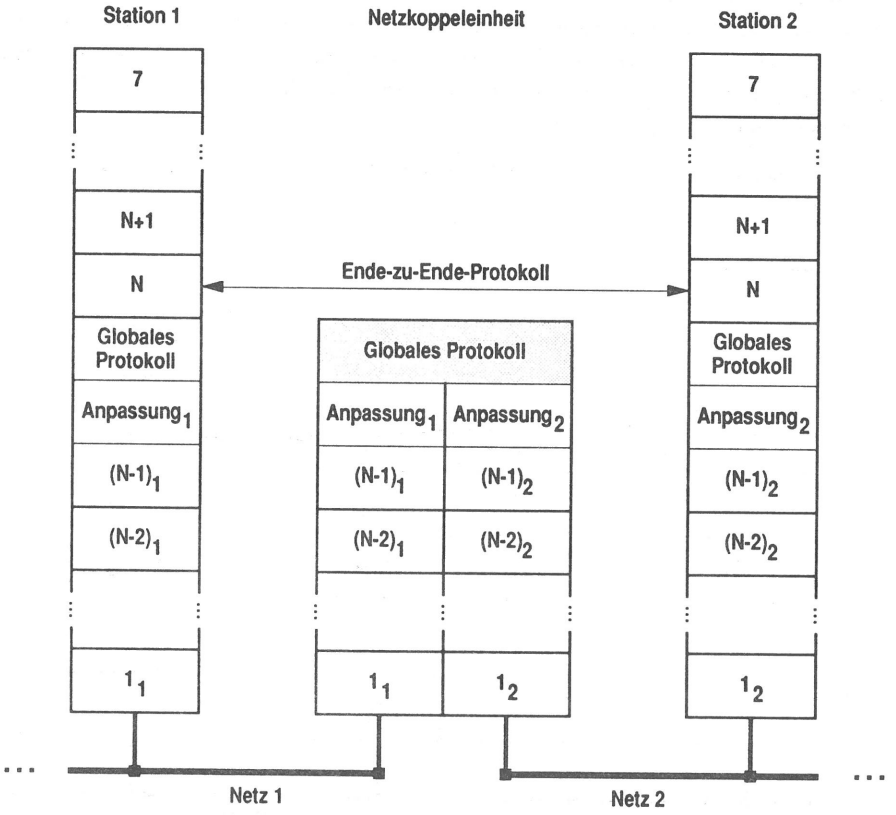


Bild 3.3: Netzkopplung über ein globales Protokoll

Ein zweiter Kopplungstyp, bei welchem PDUs einfach durchgereicht werden können, ist in Bild 3.3 dargestellt. Hier wird bereits auf der Schicht N–1 ein *globales Protokoll* in allen Stationen von Netz 1 und 2 auf die teilnetzspezifischen Protokolle dieser Schicht aufgesetzt,

welches in der Netzkoppeleinheit als Hauptaufgaben die globale Verkehrslenkung und das Durchreichen realisiert. Die Adressen und Namen dieses globalen Protokolls müssen im gekoppelten Netz eindeutig sein. Zur Anpassung der teilnetzspezifischen Protokolle an das globale Protokoll ist jeweils eine Anpassungsschicht notwendig, welche je nach Bedarf den Dienst des teilnetzspezifischen Protokolls anheben oder vermindern kann, um zu einer einheitlichen Schnittstelle zu kommen.

Dieser Kopplungstyp hat gegenüber dem ersten den Vorteil, daß er bereits eine Schicht tiefer eingesetzt werden kann und so die Ende–zu–Ende–Signifikanz des Schicht–N–Protokolls nicht beeinträchtigt wird. Sogar das globale Protokoll selbst kann Eigenschaften eines Ende–zu–Ende–Protokolls aufweisen. Als Nachteil dieses Kopplungstyps ist anzumerken, daß die Protokollprofile in den Stationen der Netze 1 und 2 entsprechend erweitert werden müssen, falls das globale Protokoll nicht sowieso schon in einem oder beiden Netzen vorhanden ist. Er sollte deshalb sinnvollerweise nur bei neuen Installationen zum Einsatz kommen oder dort, wo ein solches globales Protokoll schon vorher vorhanden war. Dieser Kopplungstyp wird vor allem auf der Vermittlungsschicht verwendet, was in Abschnitt 3.3.3 genauer beschrieben wird.

3.1.4 Netzkopplung durch Transformation

Führt man die Kopplung noch tiefer durch, nämlich direkt auf der letzten unterschiedlichen Schicht, so wird die Kopplungsaufgabe wesentlich schwieriger. Anstelle des Durchreichens ist dann eine *Transformation* der Dienstprimitive (vertikale Transformation) notwendig, wie das in Bild 3.4 dargestellt ist. Bei der Transformation ist oft ein Verlust an Funktionalität nicht zu vermeiden, da nicht immer alle Primitive des einen Netzes ein Analogon im anderen besitzen, so daß nur die Schnittmenge, welche beiden Netzen gemeinsam ist, umgesetzt werden kann. Im allgemeinen sind hier oft komplizierte Szenarien notwendig, um durch die Abbildung *eines* Dienstprimitives auf eine ganze Sequenz von Dienstprimitiven, oder umgekehrt, fehlende Analogien zumindest teilweise nachzubilden und so den Funktionalitätsverlust minimal zu halten. Dadurch werden bei verbindungsorientierten Protokollen auch unterschiedliche Zustandsautomaten für zusammengehörende Verbindungsabschnitte aneinander angepaßt. Diese Zustandsautomaten sind dann lose, über Meldungen, gekoppelt und werden synchronisiert, indem die Meldungen (hier: Dienstprimitive) so transformiert werden, daß der jeweils andere Zustandsautomat sich so verhält, wie das für eine funktionsfähige Netzkopplung notwendig ist. Man spricht deshalb auch von einer *Protokolltransformation.*

Was die Parameter in den Dienstprimitiven anbetrifft, so kann man unterscheiden zwischen

- Parametern, welche in beiden Netzen dieselbe Bedeutung haben und deshalb nicht transformiert werden müssen,

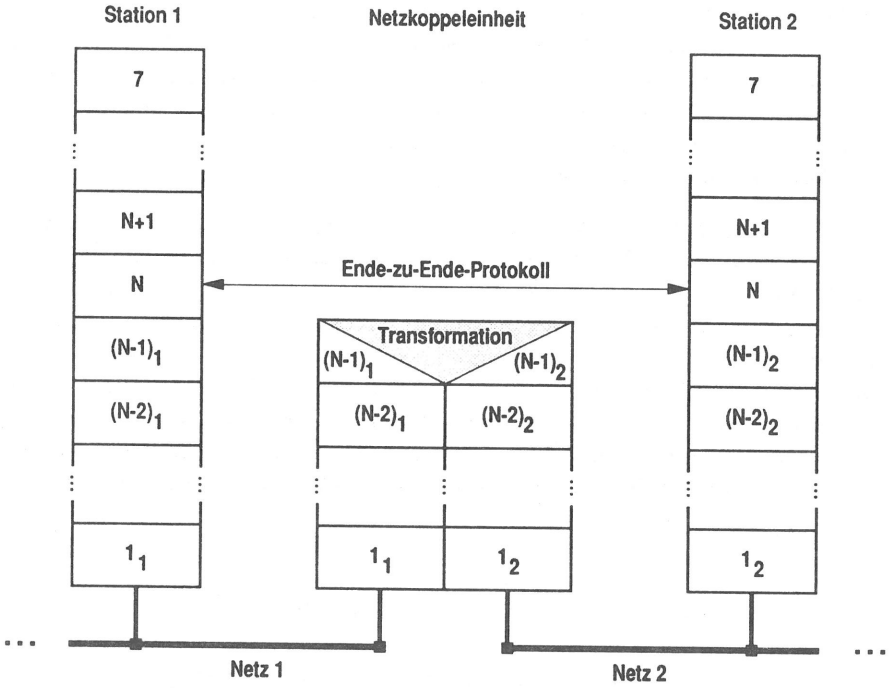


Bild 3.4: Netzkopplung durch Transformation

- Parametern, welche in den beteiligten Netzen unterschiedliche Bedeutungen haben und deshalb transformiert werden müssen,
- Parametern, welche nur im ersten Netz definiert sind und somit in der Netzkoppeleinheit entfernt werden müssen (sie können eventuell noch als Zusatzinformation für die Netzkoppeleinheit verwendet werden) und
- Parametern, welche nur im zweiten Netz definiert sind und somit in der Netzkoppeleinheit, zum Beispiel mit Hilfe von Tabellen, erzeugt werden müssen (Projektierung).

In der Transformationssoftware dürfen keine Protokollmechanismen, die einen Partnermechanismus benötigen, eingesetzt werden, da es keine Instanz gibt, welcher dieser Partnermechanismus eindeutig zugeordnet werden könnte.

Vorteile dieses Kopplungstyps sind die Ende–zu–Ende–Protokolle von der Schicht N an aufwärts, die Tatsache, daß die Protokollprofile in den Stationen der Netze 1 und 2 nicht modifiziert werden müssen und die minimale Anzahl von Schichten in der Netzkoppeleinheit, was eine minimale Transferzeit durch die Netzkoppeleinheit zur Folge hat.

Ein Beispiel für die Transformation von Dienstprimitiven ist in Kapitel 5 zu finden.

Eine Variante dieses Kopplungstyps ist die direkte Transformation der PDUs (horizontale Transformation), bei der die Kopplung der Zustandsautomaten zusammengehörender Verbindungen (falls welche existieren) enger als oben ist. Hier werden nicht mehr nur die an der Schnittstelle sichtbaren Meldungen (Dienstprimitive) transformiert, sondern es können hier auch interne Meldungen einer Protokollinstanz (PDUs) zur Meldungskopplung mitverwendet werden. Die Kopplungsaufgabe ist hier wesentlich komplexer, weil die Anzahl der Primitive (und gegebenenfalls der Zustände) größer ist. Außerdem müssen neben den Daten auch die Protokollsteuerinformationen des ersten Netzes entfernt und durch die entsprechenden Protokollsteuerinformationen des zweiten Netzes ersetzt werden. Diese Variante hat eine Verwischung der Instanzgrenzen auf der Kopplungsschicht zur Folge und ist schwieriger zu implementieren, weil keine existierende Schnittstelle verwendet wird. Sie setzt deshalb auch eine größere Ähnlichkeit der Protokolle auf den Schichten N–1 der Netze 1 und 2 voraus, als dies bei allen bisher erwähnten Kopplungstypen und Varianten der Fall war. Dafür ist sie bezüglich eines möglichen Einflusses auf Fluß– oder Fehlerkontrolle flexibler. Außerdem können auch PDUs (wie beispielsweise TEST auf der Sicherungsschicht) umgesetzt werden, welche den Dienstzugangspunkt der Schicht N–1 gar nicht erreichen. Desweiteren kann versucht werden, eine Ende–zu–Ende–Signifikanz wenigstens einiger Protokollmechanismen zu erreichen. Alle Adressen und Namen müssen hier auch auf der Kopplungsschicht eindeutig sein. Bezüglich der Parameter, Szenarien und Protokollmechanismen gilt für die PDUs das oben für Dienstprimitive gesagte.

3.1.5 Netzkopplung durch Einbettung

Der letzte Kopplungstyp ist auch bei völlig inkompatiblen Netzen anwendbar. Dabei wird eine Konstellation vorausgesetzt, bei welcher eine Station des Netzes 1 mit einer anderen Station an einem Netz 2 kommunizieren möchte. Diese Netze können räumlich weit voneinander entfernt sein. Zur Kommunikation wird ein Netz 3 als *Transitnetz* verwendet, wie das in Bild 3.5 dargestellt ist. An den Netzgrenzen sind jeweils *halbe Netzkoppeleinheiten* vorhanden, welche immer paarweise benützt werden müssen. Das kann auch aus organisatorischen Gründen ein sinnvoller Kopplungstyp sein, da dann die Betreiber der Netze 1 und 2 jeweils für ihre halben Netzkoppeleinheiten zuständig sind.

Die Protokolle in den Netzen 1 und 2 müssen ab der höchsten Schicht für diese Netze in den halben Netzkoppeleinheiten identisch sein, da keine Transformation durchgeführt wird. Deshalb müssen die Netze 1 und 2 dort auch einen gemeinsamen Adreßraum verwenden, da sie aus der Sicht dieser Protokolle *ein* erweitertes Netz bilden. Bei Zeitüberwachungen muß man hier die Laufzeiten durch das Transitnetz mit berücksichtigen.

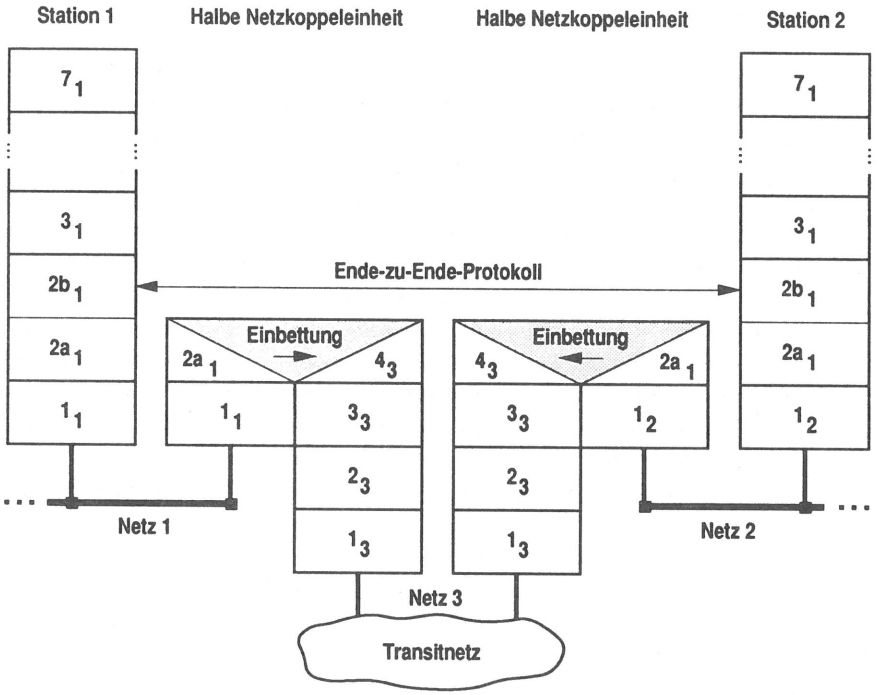


Bild 3.5: Netzkopplung durch Einbettung

An der ersten Netzgrenze werden PDUs des Netzes 1 als reine Nutzdaten aufgefaßt, mit Protokollsteuerinformationen für das Transitnetz (Netz 3) ergänzt und dadurch in Pakete des Transitnetzes *eingebettet* [27]. Diese Pakete werden durch das Transitnetz zur zweiten Netzgrenze übermittelt. Dort werden die Protokollsteuerinformationen für das Transitnetz wieder entfernt, so daß die ursprünglichen PDUs wieder zum Vorschein kommen, welche über das Netz 2 die Station 2 erreichen.

Es muß bei diesem Kopplungstyp beachtet werden, daß aufgrund der fehlenden Transformation keine Kommunikation mit normalen Stationen an Netz 3 möglich ist. Dies wäre nur denkbar bei modifizierten Netz–2–Stationen am Transitnetz, welche die Schnittstellenfunktion zwischen den Netzen 3 und 2 noch zusätzlich zu ihren eigenen Protokollen implementiert haben.

Die höchste Schicht des Transitnetzes muß nicht die Transportschicht sein, wie dies in Bild 3.5 dargestellt ist. Die MAC–Teilschicht wäre zum Beispiel auch eine sinnvolle Alternative.

Als Transitnetze zur Überbrückung größerer Entfernungen werden Weitverkehrsnetze verwendet, welche im Extremfall auch Satellitenstrecken [85] sein können. Für mittlere Entfernungen

werden zunehmend MANs als Hintergrundnetze eingesetzt.

Ein anderer Anwendungsfall für diesen Kopplungstyp ist die Kopplung von Lokalen Netzen über einen Großrechner, zu welchem Stationen aus beiden Netzen sowieso Zugang haben, so daß die dafür notwendigen Verkabelungen schon vorhanden sind [83]. Dabei werden die Pakete der Netze 1 und 2 in die oft noch herstellerspezifischen Pakete für den Großrechner eingebettet.

Wenn die höchste Schicht einer halben Netzkoppeleinheit auf der Seite der Netze 1 und 2 die MAC–Teilschicht ist (dieses Beispiel ist auch in Bild 3.5 dargestellt), so spricht man von einer *Remote Bridge*. In Abschnitt 3.3.2.5 wird darauf noch näher eingegangen.

Bei einer Kopplung oberhalb der MAC–Teilschicht in den Netzen 1 und 2 werden SDUs in die Pakete des Transitnetzes eingebettet. Die Einbettung von PDUs stellt dann eine Variante dar, welche eine Modifikation der entsprechenden Protokollinstanzen notwendig macht.

3.2 Merkmale von Netzen und deren Einfluß auf die Netzkopplung

In Abschnitt 3.1 wurden verschiedene Kopplungstypen und Varianten dazu vorgestellt. Dabei wurde nicht im einzelnen auf die Teilaufgaben eingegangen, welche eine Netzkoppeleinheit zu bewältigen hat. Die für die Netzkopplung interessanten Merkmale von zu verbindenden und gekoppelten Netzen [14, 17, 187] werden in diesem Abschnitt behandelt. Es werden Kopplungsprobleme aufgezeigt und prinzipielle Lösungsmöglichkeiten vorgestellt. Zur vergleichenden Bewertung unterschiedlicher Lösungsmöglichkeiten bezüglich ihrer quantitativen Auswirkungen wird auf Abschnitt 4.3 verwiesen.

3.2.1 Netztopologie

Die Kopplung von Netzen mit unterschiedlichen Topologien bereitet keine speziellen Kopplungsprobleme. Bezüglich der Leistungsfähigkeit des gekoppelten Netzes ist es allerdings sinnvoll, als Netzkoppeleinheit eine ausgezeichnete Station zu verwenden. Bei einem Netz mit sternförmiger Topologie bietet sich zur Kopplung die zentrale Station und bei baumförmiger Topologie die Wurzel des Baumes an, um die mittleren Übertragungswege möglichst kurz zu halten. Bei einer bus– oder ringförmigen Topologie gibt es keine Station, welche von der Topologie her ausgezeichnet wäre. Sind unterschiedliche Stationsprioritäten möglich, so sollte die Netzkoppeleinheit eine hohe erhalten, da sie einen potentiellen Engpaß darstellt. Bei CSMA/CD kann man dies beispielsweise auch dadurch erreichen, daß der VLSI–Baustein (Very Large Scale Integration) 82586 von Intel für den Medienzugang so programmiert wird,

daß er Pakete, welche zur Übertragung bereitstehen, erschöpfend abarbeitet bevor anderen Stationen wieder ein Zugriff auf das Medium ermöglicht wird [53, 155, 203]. Realisiert wird dies so, daß die Netzkoppeleinheit etwas früher als andere Stationen wieder auf das Medium zugreift, nachdem dieses als frei erkannt ist. Im Falle einer Kollision wird der Sendewunsch nicht zurückgestellt und die Netzkoppeleinheit sendet alle bereitstehenden Pakete unmittelbar hintereinander, so daß andere Stationen das Medium ständig als belegt betrachten und nicht darauf zugreifen können, solange die Netzkoppeleinheit noch etwas zu senden hat. Diese Modifikation des CSMA/CD–Protokolls ist aufwärtskompatibel mit herkömmlichen CSMA/CD–Implementierungen.

Bei Netzkopplungen unterhalb der Vermittlungsschicht muß man darauf achten, daß die resultierende Topologie des gekoppelten Netzes keine Schleifen bildet, da diese Netzkoppeleinheiten nicht explizit adressiert werden und dadurch endlos kreisende Pakete entstehen können. Eine Alternative ist die Abbildung der zunächst beliebigen physikalischen Topologie auf einen logischen Baum, was in Abschnitt 3.3.2.3 näher beschrieben wird.

Um die Anzahl der denkbaren Protokollumsetzungsvarianten zu reduzieren, kann es sinnvoll sein, die Umsetzung auf ein neutrales Netz vorzunehmen, welches auch als Hintergrundnetz betrieben werden kann und von dem aus dann über eine zweite Netzkoppeleinheit das Zielnetz erreicht wird. Für das neutrale Netz bietet sich heute ein standardisiertes, offenes Protokollprofil an [73]. Im Bereich der Fertigungsautomatisierung kann hier MAP verwendet werden und in [13] wird das Deutsche ForschungsNetz (DFN) als neutrales WAN zur Kopplung privater LANs vorgeschlagen. Jeder Hersteller muß dann nur eine Kopplung seiner herstellerspezifischen Produkte mit dem neutralen Netz anbieten, um die Kommunikation mit irgendeinem anderen herstellerspezifischen Netz zu ermöglichen. Die Anzahl der denkbaren unterschiedlichen Netzkoppeleinheiten sinkt dann von $n(n-1)/2$ auf n, wenn n die Anzahl der unterschiedlichen Netze ist. Die Leistungsfähigkeit der Netzkopplung ist dabei aber nicht optimal, da zur Kommunikation zweier Stationen meistens zwei Netzkoppeleinheiten und drei Netze beansprucht werden. Eine solche indirekte Kopplung macht deshalb nur dann einen Sinn, wenn der Externverkehr der Netze die seltene Ausnahme darstellt. Bei hohem Verkehr zwischen zwei Netzen ist eine direkte Kopplung vorzuziehen.

3.2.2 Paketgröße

Oberhalb der Kopplungsschicht sind die Protokolle in beiden Netzen identisch und die Paketgrößen deshalb auch kompatibel. Auf der Kopplungsschicht und darunter sind in beiden Netzen unterschiedliche Paketgrößen möglich. Gegebenenfalls kann oder muß dann eine Paketgrößenanpassung durchgeführt werden.

Wenn die angebotenen Pakete einer Schicht die bearbeitbare maximale Größe überschreiten, so ist spätestens in dieser Schicht jede SDU durch Aufteilen in kleinere PDUs zu zerle-

gen. Ist dies in beiden Netzen auf der Kopplungsschicht oder darunter notwendig, so kann man alternativ dazu auch das Aufteilen bereits oberhalb der Kopplungsschicht durchführen, um das Vereinigen und erneute Aufteilen in der Netzkoppeleinheit zu vermeiden, siehe Abschnitt 4.3.2. Ansonsten sollte das Aufteilen immer so spät wie mögich erfolgen, um die Anzahl der einzelnen Pakete pro Instanz minimal zu halten. Bei der Verwendung eines verbindungslosen globalen Protokolls, welches ein Aufteilen durchführt, darf das Vereinigen erst beim Empfänger und nicht schon in einer Netzkoppeleinheit unterwegs erfolgen, da die Segmente *eines* Paketes unterschiedliche Wege nehmen können [172] und erst beim Empfänger wieder alle Segmente zusammenkommen. Für das Vereinigen ist in der betreffenden Station genügend Speicher zu reservieren, damit alle Segmente *eines* Paketes aufeinander warten können und keine Verklemmungen entstehen.

Wenn die angebotenen Pakete einer Schicht kleiner sind als die Größe, welche von ihr und den unterlagerten Schichten bearbeitet werden könnte, so besteht die Möglichkeit, durch Blocken mehrerer SDUs zu einer PDU oder durch Verketten mehrerer PDUs die Anzahl der einzelnen Pakete zu reduzieren. In Abschnitt 4.3.2 wird gezeigt, daß dies bei gekoppelten Netzen nur sinnvoll ist, wenn dadurch der Engpaß der Kommunikationsbeziehung beeinflußt wird.

Das Hinzufügen von *Füllbits* stellt eine weitere Anpassungsmögichkeit dar, welche verwendet werden kann, wenn die Paketgröße eine vorgeschriebene untere Schranke unterschreitet. Mit Hilfe eines Längenindikators können diese Füllbits wieder erkannt und entfernt werden. Diese Methode wird beispielsweise bei CSMA/CD verwendet, um Mindestpaketgrößen zu erreichen, welche eine Kollisionserkennung auch im ungünstigsten Fall ermöglichen.

3.2.3 Übertragungsgeschwindigkeit

Die kleinste in allen beteiligten Netzen und der Netzkoppeleinheit gerade noch erreichbare *Transfergeschwindigkeit* ist die maximale Transfergeschwindigkeit, welche im gekoppelten Netz möglich ist. Wird diese vom Sender nicht überschritten, so ist im Mittel der Zeit keine Komponente überlastet, sofern bei der Bestimmung dieses Maximalwertes der Verkehr in beiden Richtungen und die Belastung durch andere Kommunikationsbeziehungen berücksichtigt wird.

Um Lastspitzen auszugleichen, ist in der Netzkoppeleinheit ein Pufferspeicher notwendig, welcher einerseits groß genug sein muß, um die Verluste klein zu halten, der andererseits aber auch nicht zu groß sein darf, um die Transferzeiten durch die Netzkoppeleinheit zu begrenzen.

Zur Beseitigung von Verstopfungssituationen und Duplikaten kann es sinnvoll sein, bereits gespeicherte Pakete mit Hilfe eines Alterungsmechanismusses aus dem Pufferspeicher zu entfernen [8, 67]. Dadurch wird die Wartezeit der restlichen Pakete kleiner. Der Verlust muß in höheren Schichten erkannt und behoben werden. Dabei sollten möglichst Pakete aus den

direkt angeschlossenen Netzen entfernt werden und nicht solche Pakete, die bereits einen weiteren Weg hinter sich und viele Betriebsmittel belegt haben.

Die *Übertragungsgeschwindigkeit* kann in den zu verbindenden Netzen unterschiedlich sein. Der Pufferspeicher in der Netzkoppeleinheit enthält im wesentlichen Pakete, welche in Richtung zum langsameren Netz unterwegs sind. Sie können auf dem schnelleren Netz zeitweise kurz hintereinander eintreffen und auch nach der Bearbeitung die Netzkoppeleinheit nur relativ langsam wieder verlassen, während Pakete in der umgekehrten Richtung ihren Pufferspeicherplatz in der Netzkoppeleinheit nach der Bearbeitung rasch freigeben, da sie über das schnellere Netz abfließen.

Bei verbindungslosen Protokollen sind die Möglichkeiten begrenzt, den ankommenden Verkehr zu drosseln, weil die einzelnen Pakete keine Beziehung zueinander haben, von unterschiedlichen Sendern kommen können und die Sender normalerweise keine Quittungen erwarten. Beim standardisierten, verbindungslosen Vermittlungsprotokoll gibt es die Möglichkeit, in einem Paket ein Bit zu setzen, welches dem Empfänger anzeigt, daß unterwegs eine Verstopfungssituation aufgetreten ist [118]. Der Standard läßt allerdings offen, wie der Empfänger in diesem Fall den Sender bremsen soll. Auch wird nicht festgelegt, nach welchen Kriterien die Netzkoppeleinheit dieses Bit in Paketen zu setzen hat.

Bei verbindungsorientierten Protokollen werden zur Vermeidung einer Überlastung der einzelnen Übertragungsabschnitte und der jeweiligen Empfänger die Flußkontrollmechanismen dieser Protokolle in beiden Netzen ausgenützt [204]. Die Wirksamkeit unterschiedlicher Kombinations– und Modifikationsmöglichkeiten wird in den Abschnitten 4.3.4 und 4.3.5 untersucht.

Die Anpassung unterschiedlicher Übertragungsgeschwindigkeiten kann bei verbindungsorientierten Protokollen auch durch weitere Protokollmechanismen unterstützt werden. Im schnelleren Netz können durch das Multiplexen von Verbindungen Betriebsmittel eingespart werden. Andererseits gibt es die Möglichkeit, im langsameren Netz durch Verbindungsaufspaltung die Betriebsmittel mehrerer Verbindungen auszunützen und so die Unterschiede in den Übertragungsgeschwindigkeiten zu kompensieren. Dabei sind auch physikalisch getrennte Wege zum Empfänger möglich, um die Bandbreite des zur Verfügung stehenden Mediums zu erhöhen.

3.2.4 Vermittlungsverfahren

Bei der Datenkommunikation spielt es für Protokolle der höheren Schichten keine wesentliche Rolle, ob die Pakete auf einem paketvermittelnden oder auf einem durchschaltevermittelnden Netz übermittelt werden. Der Unterschied wird lediglich repräsentiert durch die Protokolle, welche auf den unteren Schichten verwendet werden und hat keinen speziellen Einfluß auf

die Kopplung solcher Netze. Allerdings haben die Netze unterschiedliche Eigenschaften, deren Auswirkungen auf die Dienstqualität beachtet werden müssen. Während bei einem durchschaltevermittelnden Netz einer Kommunikationsbeziehung unabhängig von der Last eine feste Bandbreite zur Verfügung steht, nimmt diese bei einem paketvermittelnden Netz mit zunehmender Auslastung des Netzes ab. Dies hat dann auch Auswirkungen auf die Wartezeit beim Sender und damit auf die Transferzeit, was bei der Datenkommunikation aber normalerweise nicht kritisch ist.

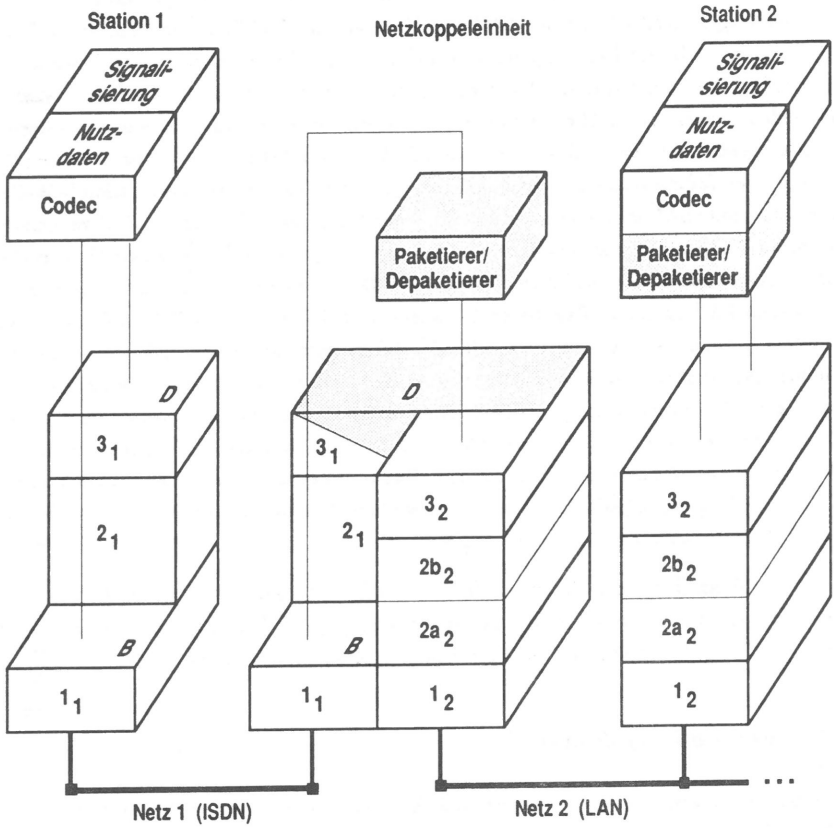


Bild 3.6: Unterschiedliche Vermittlungsverfahren bei der Sprachkommunikation

Durchschaltevermittelnde Netze werden in der Regel verwendet, um Sprache zu übertragen, da hier Anforderungen an die Dienstqualität gestellt werden, welche paketvermittelnde Netze nur schwer oder gar nicht erfüllen können. Die Kopplung unterschiedlicher Vermittlungsverfahren wird für die Sprachkommunikation anhand der Konfiguration in Bild 3.6 erläutert.

Als Vertreter eines durchschaltevermittelnden Netzes 1 werde das ISDN angenommen, wobei die Nutzdaten über einen durchgeschalteten Basiskanal (B–Kanal) fließen; Netz 2 sei ein paketvermittelndes LAN mit einem verbindungsorientierten Vermittlungsprotokoll. Die Netzkoppeleinheit ist an einem T–Bezugspunkt direkt oder an einem S–Bezugspunkt indirekt (beispielsweise über eine ISDN–fähige Nebestellenanlage) am ISDN angeschlossen [26, 64]. Sie nimmt die Aufgabe einer Endgeräteanpassung wahr. Vor der eigentlichen Sprachkommunikation muß der B–Kanal zunächst durchgeschaltet werden. Dazu ist ein spezieller paketvermittelnder Signalisierkanal (D–Kanal) vorhanden, über welchen Signalisierinformationen übertragen werden können. Ist keine Unteradresse im ISDN möglich, so muß die ISDN–Rufnummer in der Netzkoppeleinheit auf der Vermittlungsschicht, beispielsweise mit Hilfe einer Tabelle, in die Adresse der Vermittlungsschicht im Empfänger am LAN transformiert werden. Dabei darf nicht die gesamte ISDN–Rufnummer zur Adressierung der Netzkoppeleinheit benötigt werden, weil sonst nur *eine* Station am LAN erreicht werden könnte. Die nichtbenötigten Ziffern sind dann der Tabelleneingang für die Adreßtransformation [37]. Das paketvermittelnde LAN wird hier praktisch als Transitnetz (siehe Abschnitt 3.1.5) verwendet. Die Sprache (Nutzdaten), welche von einem Codec (Coder/decoder) in einen kontinuierlicher Bitstrom umgewandelt und auf dem B–Kanal des ISDN auch als solcher übertragen wird, muß vor dem LAN in einem Paketierer/Depaketierer paketiert und in Pakete für das LAN eingebettet werden. Auf der anderen Seite des LANs wird aus diesen Paketen wieder der kontinuierliche Bitstrom regeneriert. Dabei sind im Depaketierer künstliche Verzögerungen notwendig, welche einerseits so groß sein müssen, daß der regenerierte Bitstrom keine Unterbrechungen aufweist und andererseits so klein, daß die Verzögerungen für einen menschlichen Benutzer nicht als störend empfunden werden. Dadurch sind auch die Anforderungen an das LAN zur Übertragung paketierter Sprache sehr groß, welche dort sinnvollerweise eine höhere Priorität als Datenpakete erhalten sollte [179].

Bei der Kopplung mit einem hybriden (CS und PS) Netz sollte die Vermittlungstechnik ausgewählt werden, welche durch das andere Netz vorgegeben ist, so daß die Eigenschaften des gekoppelten Netzes bezüglich der Dienstqualität möglichst homogen sind.

3.2.5 Verbindungskonzept

Bei der Transformation zwischen unterschiedlichen verbindungsorientierten Protokollen sind die Zustandsautomaten der Verbindungssteuerung aneinander anzupassen. Diese Anpassung kann relativ aufwendige Szenarien erzwingen, wenn so unterschiedliche Netze wie LAN, das paketvermittelnde öffentliche Datennetz nach der CCITT–Empfehlung X.25 oder das ISDN zu verbinden sind. Die Lösung der verteilten Steuerungsaufgabe muß für jedes Paar von unterschiedlichen Netzen in Form von Szenarien individuell ausgearbeitet werden [142].

Beim Übergang von einem verbindungsorientierten auf ein verbindungsloses Protokoll müssen

bei jedem Verbindungsaufbau die Adressen von Sender und Empfänger gespeichert werden, da die nachfolgenden Pakete auf der Verbindung diese Informationen in der Regel nicht mehr beinhalten. Diese Adressen müssen dann jedem Paket, welches an das verbindungslose Protokoll weitergereicht wird, mitgegeben werden.

Beim Übergang von einem verbindungslosen auf ein verbindungsorientiertes Protokoll ist der Pufferspeicher in der Netzkoppeleinheit nicht nur für das Ausgleichen von Lastspitzen notwendig, sondern auch aus drei weiteren Gründen:

- Bei jedem Paket muß überprüft werden, ob das gewünschte Ziel über eine bereits bestehende Verbindung erreicht werden kann oder nicht. Dabei muß jede Quittung der Verbindung zugeordnet werden, auf welcher das dazugehörende Datenpaket gekommen ist. Wenn keine geeignete Verbindung existiert, muß die Netzkoppeleinheit das Paket zwischenspeichern, die benötigte Verbindung aufbauen und anschließend das Paket über diese Verbindung übertragen. Es stellt sich dann die Frage, wann eine so aufgebaute Verbindung wieder abgebaut werden soll, da dies ja nicht von dem verbindungslosen Protokoll aus angestoßen werden kann. Hier bietet es sich an, die Dauer der Datentransferphase einer Verbindung mit einer Uhr zu überwachen, welche bei jedem Paket, das über diese Verbindung übertragen wird, zurückgesetzt wird [207]. Wenn nun die Verbindung lange nicht mehr benötigt worden ist, läuft die Uhr ab und die Verbindung wird von der Netzkoppeleinheit wieder abgebaut. Die optimale Einstellung dieser Uhr hängt von der Tarifstruktur des Netzes mit dem verbindungsorientierten Protokoll ab. In [191] wird beispielsweise für das paketvermittelnde öffentliche Datennetz bei Verbindungen mit einem Empfänger im europäischen Ausland ein Richtwert von wenigen Minuten angegeben, während in [94] für dasselbe Netz, auf Inlandsverbindungen bezogen, der Abbau und mögliche anschließende Wiederaufbau einer Verbindung generell als nicht lohnend bezeichnet wird. Der gegenteilige Extremfall wäre, eine Verbindung nach jedem Paket sofort wieder abzubauen [25].
- In der Netzkoppeleinheit muß ein *Sequentialisieren* durchgeführt werden, damit die Pakete, welche infolge des verbindungslosen Protokolls in beliebiger Reihenfolge ankommen können, auf der abgehenden Verbindung in der richtigen Reihenfolge ausgesandt werden. Der Pufferspeicher ist daher auch notwendig, um Pakete zwischenzuspeichern, bis die jeweils noch fehlenden Pakete mit niedrigerer Sequenznummer in der Netzkoppeleinheit eingetroffen, bearbeitet und wieder ausgesandt sind.
- Auch durch die fehlende Möglichkeit einer effizienten Flußkontrolle im verbindungslosen Protokoll wird der Pufferspeicher in der Netzkoppeleinheit stärker als sonst belastet. Verstärkt wird dieser Effekt möglicherweise noch durch eine strenge Flußkontrolle im abgehenden, verbindungsorientierten Netz.

Bei der Netzkopplung über ein globles Protokoll nach Abschnitt 3.1.5 wird die Anpassung

eines verbindungsorientierten an ein verbindungsloses Protokoll üblicherweise durch die dazwischenliegende Anpassungsschicht vorgenommen.

3.2.6 Adressierungskonzept

Bevor Netze miteinander verbunden werden können, muß gegebenenfalls durch Umbenennungen dafür gesorgt werden, daß die Adressen und Namen in Ende–zu–Ende–Protokollen eindeutig sind. Die Menge der definierten Adressen vergrößert sich durch die Netzkopplung, was eine Erweiterung der Adreßabbildungstabellen in jeder Station erforderlich macht, wenn eine Kommunikation mit Stationen am anderen Netz ermöglicht werden soll.

Bei der Kopplung von LANs auf der MAC–Teilschicht wird eine Adressierungsart mit einheitlichem Adreßraum im Gesamtnetz bereits auf der Kopplungsschicht verwendet, welche in Abschnitt 3.3.2.1 genauer erläutert wird. Der Sender adressiert direkt den Empfänger, ohne zu wissen wo dieser sich befindet. Die Netzkoppeleinheiten empfangen zunächst alle Pakete der angeschlossenen Netze und stellen mit Hilfe von Filtertabellen fest, welche Pakete wohin weiterzugeben sind. Dabei werden die Adressen also *durchgereicht.*

Für die weiteren Adressierungsarten muß man berücksichtigen, daß Adressen, je nach Schicht und Protokoll, hierarchisch strukturiert oder flach sein können.

Hierarchische Adressen (vorwiegend auf der Vermittlungsschicht verwendet) enthalten Informationen über die Lage der Stationen, welche bei der Verkehrslenkung ausgewertet werden. Deshalb müssen Teile der Stationsadresse geändert werden, wenn eine Station mit einer solchen hierarchischen Adresse an einem anderen Netz angeschlossen werden soll, als das bisher der Fall war. Je nach Hierarchieebene, welcher eine betrachtete Netzkoppeleinheit angehört, wird ein bestimmter Teil der Adresse eines Paketes ausgewertet, um den Empfänger oder die nächste Netzkoppeleinheit zu adressieren. Dabei müssen Tabellen für die Verkehrslenkung nur diese Adreßteile enthalten. Jede Netzkoppeleinheit ist logisch betrachtet eine Hierarchieebene höher angesiedelt als die Stationen, welche direkt über sie kommunizieren. Es gibt auch die Möglichkeit, bisher flache Adressen der zu verbindenden Netze um eine Netzidentifikation zu erweitern, welche in einer Netzkoppeleinheit ausgewertet werden kann, so daß für das gekoppelte Netz eine hierarchische Adresse zur Verfügung steht [189]. Dies setzt jedoch Modifikationen an allen Stationen der zu verbindenden Netze voraus.

Flache Adressen haben den Vorteil, daß Stationen an einer beliebigen Stelle des gekoppelten Netzes eingesetzt werden können, ohne daß Umbenennungen notwendig wären. Dies ist insbesondere in mobilen Stationen wichtig, welche drahtlos mit jeweils einem der Netzsegmente verbunden sind. Es handelt sich also um gerätespezifische Adressen, wie sie auf der MAC–Teilschicht üblich sind. Aus ihnen können aber keine relevanten Informationen für die Verkehrslenkung abgeleitet werden. Man unterscheidet ferner zwischen einer einstufigen und einer mehrstufigen Adressierung.

- Bei der *einstufigen Adressierung* werden die Stationen am anderen Netz mit Hilfe von Adressen, welche bisher nicht verwendet worden sind (Aliasadressen), adressiert. In der Netzkoppeleinheit wird bei jedem Paket mit Hilfe einer Abbildungstabelle die Aliasadresse des Empfängers in die tatsächliche Adresse (oder in eine weitere Aliasadresse, falls der Empfänger nicht direkt am nächsten Netz angeschlossen ist) *transformiert*. Analog wird die Adresse des Senders in die Aliasadresse transformiert, über welche der Sender von diesem Netz aus angesprochen werden würde. Manchmal ist es auch möglich, auf die Abbildungstabelle zu verzichten, wenn ein bisher unbenützter Parameter so umdefiniert werden kann, daß er die Adresse des nächsten Übertragungsabschnitts führt. Dann muß allerdings in jeder sendenden Station des betreffenden Netzes dieser Parameter ausgefüllt werden. Bei einer Kopplung auf der Verarbeitungsschicht werden Namen von Verarbeitungsinstanzen (AE–Titles) mit Hilfe der Abbildungstabelle aufeinander abgebildet, welche jeweils netzintern wieder auf Adressen abgebildet werden. Der Aufbau der Adressen ist in jedem Netz unabhängig von dem im anderen Netz, was diese Methode zu einer universell einsetzbaren Möglichkeit macht. Die Adressierung eines Empfängers ist bei dieser Methode allerdings abhängig von dem Netz, an welchem der Sender angeschlossen ist, und nicht einheitlich. Außerdem ist die Aktualität und Konsistenz der Tabellen in verschiedenen Netzkoppeleinheiten sehr schwierig zu überwachen und zu gewährleisten. Sie muß in der Regel manuell wieder hergestellt werden, wenn sich an der Konfiguration etwas ändert.

- Bei der *mehrstufigen Adressierung* wird bereits beim Sender eine Folge von Adressen in jedes Paket eingetragen, welche in der richtigen Reihenfolge die zu durchlaufenden Netzkoppeleinheiten, Netze und den Empfänger repräsentieren. Diese Folge von Adressen kann deshalb auch als eine im gekoppelten Netz eindeutige, hierarchische Adresse des Empfängers interpretiert werden, wobei, logisch betrachtet, jede Teiladresse in die vorhergehende *eingebettet* ist [191]. Auf seinem Weg wird nun in jedem Paket diese Folge von Adressen sequentiell abgearbeitet, um so letztendlich den Empfänger zu erreichen. Die Intelligenz wird hier von den Netzkoppeleinheiten zu den Sendern verlagert. Man spricht deshalb auch von einem *Source Routing* [118, 188]. Die mehrstufige Adressierung hat bezüglich der Verkehrslenkung sehr einfache Netzkoppeleinheiten, ohne Abbildungstabellen, zur Folge. Stattdessen muß nun jede Station zu jeder Zeit genau die Konfiguration des Netzes kennen, den Weg bereits beim Senden eines Paketes auswählen und diesem mitgeben. Dies ist zum Beispiel sinnvoll bei militärischen Netzen, wenn der Sender den Weg einer Nachricht von vorn herein festlegen möchte, ist aber auch für die Kopplung von Token Ring LANs als Standarderweiterung vorgesehen (siehe Abschnitt 3.3.2.4).

3.2.7 Dienstqualität

Da eine Netzkoppeleinheit den Zugang zu vielen Stationen im zweiten Netz ermöglicht, sollte ihre *Zuverlässigkeit* und ihre *Verfügbarkeit* größer sein als die einer normalen Station.

Falls die Daten in einem Sender verschlüsselt werden, so sollte dies nach Möglichkeit oberhalb der Kopplungsschicht und damit Ende–zu–Ende erfolgen, damit die Nutzdaten in der Netzkoppeleinheit nicht im Klartext vorliegen müssen und die *Datensicherheit* nicht beeinträchtigt wird. Es bietet sich hierzu vor allem die Darstellungsschicht an.

Viele Dienstqualitätsmerkmale sind in einem heterogenen Netz schlechter als in einem homogenen:

- Durch die Bearbeitungszeiten in einer Netzkoppeleinheit entstehen zusätzliche Verzögerungen, welche die *Transferzeit* erhöhen und die Leistungsfähigkeit verschlechtern.
- Die Ende–zu–Ende–Signifikanz der Protokolle unterhalb der Kopplungsschicht geht verloren, wodurch die *Behandlung von Fehlern* weniger zuverlässig [22] und die Flußkontrollen weniger wirkungsvoll werden.
- Wenn die Protokolle auf der Kopplungsschicht so unterschiedlich sind, daß nicht alle Pakete des einen Protokolls ein Analogon im anderen haben, so entsteht außerdem ein Verlust an *Funktionalität*.

Andere Dienstqualitätsmerkmale können durch Zusatzmaßnahmen bei der Netzkopplung erhalten oder sogar verbessert werden:

- Die Netzkoppeleinheit kann den Zugang zu einem Netz überwachen und nur autorisierte Verbindungsaufbaupakete durchlassen, so daß dieser *Sicherheitsaspekt* verbessert wird. Dazu werden die Adresse des Senders und ein Geheimwort überprüft. Diese Aufgabe kann auch von der Netzkoppeleinheit an einen zentralen Autorisierungsdienst im Netz weitergegeben werden [66]. Derselbe Mechanismus kann auch verwendet werden, um zu verhindern, daß lokale Stationen mit Stationen an einem anderen Netz kommunizieren und dadurch unerlaubterweise Gebühren in diesem anderen Netz verursachen. Eine Alternative ist das Überprüfen eines *Stempels* in der Netzkoppeleinheit, welcher jedem Paket mitgegeben wird. Der Stempel wird aus der Prüfsumme des Paketes und aus einem *Visum* [62] berechnet. Dieses Visum wird an den Sender und an die Netzkoppeleinheit von einem zentralen Autorisierungsdienst vergeben, wenn vor Beginn der eigentlichen Kommunikation die Kontrolle des Sender/Empfänger–Paares und weiterer Parameter (beispielsweise eines Geheimwortes) erfolgreich verlief.
- Durch Verbindungsaufspaltung in einem langsameren Netz kann eine geforderte *Bandbreite* auch dann bereitgestellt werden, wenn eine einzelne Verbindung auf diesem Netz diese Bandbreite nicht zur Verfügung stellen könnte.

3.2.8 Netzmanagement

Da in einem gekoppelten Netz Fehler erwünschterweise oft auf Teilnetze begrenzt bleiben, sind sie dafür von einem Netzbetreiber nicht mehr so ohne weiteres zentral zu beobachten und zu beheben. Deshalb ist besonders bei gekoppelten Netzen ein leistungsfähiges Netzmanagement unverzichtbar, durch welches Alarmmeldungen an eine zentrale Manager–Station geschickt werden können und dieser die Abfrage weiterer Informationen ermöglicht.

Um Netzmanagementinformationen der Netzkoppeleinheit selbst einer zentralen Manager–Station zugänglich zu machen, ist in der Netzkoppeleinheit ein dafür geeignetes Protokollprofil und der übliche Agent–Prozeß notwendig. Dasselbe Protokollprofil muß auch in der Manager–Station vorhanden sein. Prinzipiell gibt es hier zunächst zwei Möglichkeiten:

- Unabhängig von der Kopplungsschicht in der Netzkoppeleinheit das vollständige Protokollprofil implementieren, welches in der Manager–Station sowieso vorhanden ist.
- Ein unvollständiges Protokollprofil unter dem Agent–Prozeß verwenden, welches dann auch in die Manager–Station als alternativer Netzzugang eingefügt werden muß.

Die zweite Möglichkeit lohnt sich insbesondere dann, wenn mehrere Stationen (Netzkoppeleinheiten oder Protokoll–Analysatoren) mit solchen unvollständigen Protokollprofilen mitverwaltet werden sollen. Wenn in der Netzkoppeleinheit kein geeigneter Agent–Prozeß mit dem dazu passenden Protokollprofil implementiert werden kann oder soll, so gibt es noch eine dritte Möglichkeit. Dabei wird der Agent–Prozeß für die Netzkoppeleinheit auf einer normalen Station als zweiter Agent–Prozeß implementiert. Statische Informationen über die Netzkoppeleinheit können dort fest in die MIB eingetragen werden. Veränderliche Informationen muß diese Station mit Hilfe eines geeigneten (möglicherweise herstellerspezifischen) Protokolls periodisch von der Netzkoppeleinheit abfragen.

In einer Netzkoppeleinheit sollten neben den zu verwaltenden Objekten einer normalen Station spezielle MOs definiert werden [88], welche Aufschluß über ihre korrekte Arbeitsweise geben oder beim Lokalisieren aufgetretener Fehler eine Hilfe sein können. Beispiele dafür sind Zustandsinformationen der Netzkoppeleinheit, Pufferspeicherfüllstand, Verkehrslenkungstabellen oder Teile davon. Darüberhinaus müssen Funktionen bereitgestellt werden, welche diese MOs ständig auf dem aktuellen Stand halten und beim Auftreten definierter Ereignisse diese der Manager–Station mitteilen können. Der Manager–Prozeß muß dann so erweitert werden, daß er die Zusatzinformationen sinnvoll ausnützen und richtig interpretieren kann.

Je nach der Philosophie für das globale Netzmanagement gibt es noch weitere Aufgaben für eine Netzkoppeleinheit. Wenn jedes Netz seine eigene Manager–Station behält, sollte eine Kooperation dieser Manager–Stationen ermöglicht werden. Dazu müssen die Manager–Stationen entweder direkt miteinander oder mit einer übergeordneten Manager–Station kommunizieren können. Werden unterschiedliche Protokolle für das Netzmanagement verwendet, so muß die

Netzkoppeleinheit, neben ihren normalen Aufgaben, auch noch eine Transformation dieser Protokolle ermöglichen. Soll das Gesamtnetz von *einem* Netz aus verwaltet werden, so ist in der Netzkoppeleinheit neben ihrem eigenen Agent–Prozeß ein zweiter Agent–Prozeß notwendig, welcher die Agent–Prozesse aller Stationen im anderen Netz ersetzt [34]. Die Aussagen am Ende des vorletzten Absatzes gelten hier entsprechend. In Abschnitt 5.6 wird ein Beispiel dafür angesprochen. Eine weitere Möglichkeit ist die, die Netzkoppeleinheit direkt als eine zweisprachige Manager–Station zu betreiben, welche einen direkten Zugriff auf die Stationen an beiden (direkt angeschlossenen) Netzen hat [197].

Zum Schichten–Management gehört bei Netzkoppeleinheiten, neben den normalen Aufgaben, das Abwickeln des Spanning Tree Algorithmusses, welcher in Abschnitt 3.3.2.3 näher erläutert wird, und das Austauschen von Verkehrslenkungsinformationen, siehe Abschnitt 3.3.3.1.

3.3 Klassifikation von Netzkoppeleinheiten

Netzkoppeleinheiten und gekoppelte Netze haben, je nachdem auf welcher Schicht die Kopplung durchgeführt wird, unterschiedliche Eigenschaften. In diesem Abschnitt sollen Netzkoppeleinheiten nach ihrer Kopplungsschicht klassifiziert und diese Eigenschaften beleuchtet werden. Die Benennung der Netzkoppeleinheiten ist in der Literatur nicht immer einheitlich. Im folgenden werden die englischen Bezeichnungen verwendet, wie sie beispielsweise in [168] definiert sind und sich auch in der deutschsprachigen Literatur eingebürgert haben.

3.3.1 Repeater (Netzkopplung auf der Bitübertragungsschicht)

3.3.1.1 Grundform

Erfolgt die Kopplung auf der Bitübertragungsschicht, wie das in Bild 3.7 dargestellt ist, so spricht man von einem *Repeater*. Er dient in der Regel dazu, Netze, deren räumliche Abmessungen aus physikalischen Gründen (Dämpfung) begrenzt sind, zu erweitern. In einem Repeater werden die einzelnen Bits regeneriert und nach einer kurzen Verzögerung wieder ausgesandt. Wenn die Präambel eines Paketes eine vorgeschriebene Länge unterschreitet, werden die fehlenden Bits auf dem abgehenden Netz eingefügt [183]. Das Medium und die Darstellung der Bits auf dem Medium dürfen verschieden sein. Im Repeater wird dann die entsprechende Transformation vorgenommen. Die zu koppelnden Netze müssen dieselbe Übertragungsgeschwindigkeit besitzen, da der Repeater keine Pakete zwischenspeichern kann, um Lastspitzen auszugleichen. Es können keine Informationen ausgewertet werden, so daß auch alle Adressen im gekoppelten Netz einem gemeinsamen Adreßraum entnommen sein müssen. Man erhält praktisch *ein* Gesamtnetz, welches aus mehreren Netzsegmenten aufgebaut ist. Dadurch wird auch der Internverkehr eines Netzsegmentes unnötigerweise an das

andere Segment weitergegeben. Die entstehende Topologie darf keine Schleifen bilden, da ein Repeater immer alles weitergibt, was er empfängt.

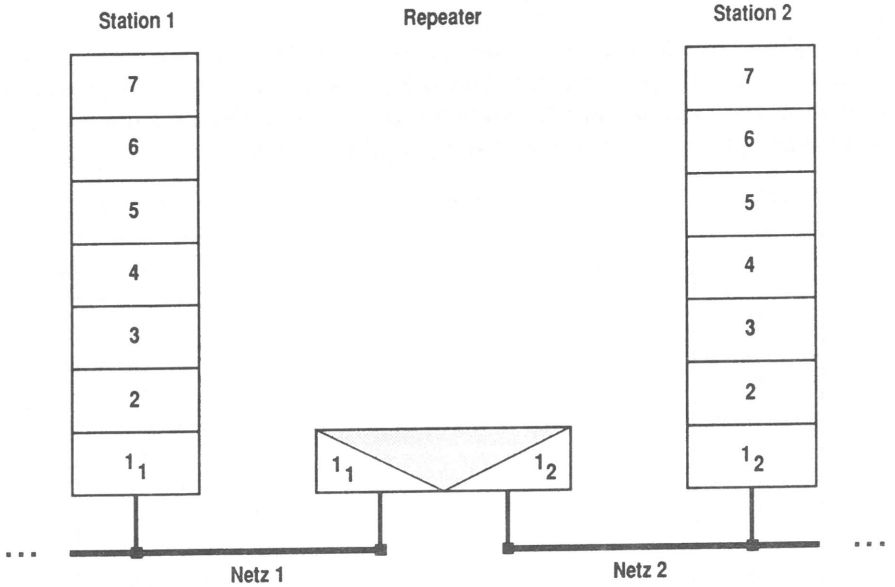


Bild 3.7: Netzkopplung über einen Repeater

Repeater werden insbesondere verwendet, um CSMA/CD–Netzsegmente miteinander zu verbinden [89]. In diesem Fall haben Repeater auch entdeckte Kollisionen in Form eines sogenannten Jam–Signals weiterzugeben. Um zu verhindern, daß sich häufige Kollisionen eines fehlerhaften Netzsegmentes über das gekoppelte Netz ausbreiten, kann ein Repeater automatisch ein als fehlerhaft erkanntes Netzsegment vom Gesamtnetz isolieren und dieses auch automatisch, nach Feststellen seiner korrekten Arbeitsweise, wieder mit dem Gesamtnetz verbinden.

3.3.1.2 Sonderformen

Wenn ein Repeater mehr als zwei Netzanschlüsse hat, so spricht man von einem *Multiport Repeater*. Dieser muß alle Bits, welche von einem Netz empfangen werden, regenerieren und auf allen anderen Netzen wieder aussenden.

Zur Überbrückung größerer Entfernungen können zwei Repeater über eine Punkt–zu–Punkt–Verbindung, zum Beispiel über eine Glasfaserstrecke, miteinander verbunden werden. Bei dieser Konfiguration spricht man von einem *Remote Repeater*.

3.3.2 Bridge (Netzkopplung auf der Sicherungsschicht)

3.3.2.1 Allgemeine Eigenschaften

Zur Kopplung zweier Netze auf der Sicherungsschicht wird eine *Bridge* verwendet. In der Regel arbeitet eine Bridge auf der MAC–Teilschicht, so daß bereits die LLC–Teilschicht eine Ende–zu–Ende–Signifikanz hat. Eine solche Bridge ist auch in Bild 3.8 dargestellt. Ein Durchreichen auf der LLC–Teilschicht ist aber prinzipiell auch möglich.

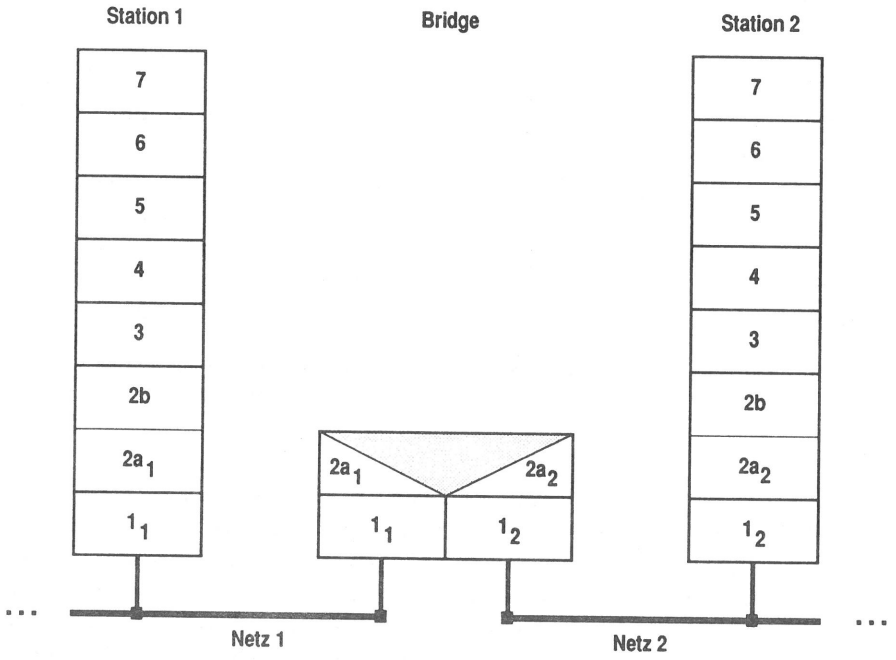


Bild 3.8: Netzkopplung über eine Bridge

Im Gegensatz zum Repeater werden in einer Bridge Pakete anstelle von Bits bearbeitet. Diese Pakete werden zum Ausgleich von Lastspitzen zwischengespeichert [8], so daß die Netze auch unterschiedliche Übertragungsgeschwindigkeiten haben dürfen. Speziell bei einer CSMA/CD–Bridge gibt es die Möglichkeit, bei einem Paketverlust ein Jam–Signal zu senden, um dem Sender eine Kollision vorzutäuschen und somit eine Wiederholung bereits auf der MAC–Teilschicht zu veranlassen. Die Paketgrößen müssen aufgrund der für die Sicherungsschicht vorgesehenen Aufgaben auf beiden Seiten einer Bridge kompatibel sein.

Eine Bridge kann zur Erweiterung der räumlichen Ausdehnung eines Gesamtnetzes verschiedene Netze auch dann noch miteinander verbinden, wenn

- aufgrund einer sonst zu großen Ende–zu–Ende–Laufzeit auf der Sicherungsschicht (vor allem kritisch bei Medienzugangsverfahren mit konkurrierendem Zugriff),
- wegen einer zu großen Anzahl von Stationen (vor allem kritisch bei Medienzugangsverfahren welche Sendeberechtigungen vergeben) oder
- wegen der resultierenden Auslastung des Gesamtnetzes

kein Repeater mehr eingesetzt werden darf.

Die Medienzugangsverfahren der Netze arbeiten unabhängig voneinander und dürfen deshalb auch unterschiedlich sein. Bei gleichen Medienzugangsverfahren werden die PDUs einfach durchgereicht, bei unterschiedlichen müssen sie transformiert werden. Welche Aufgaben die Transformation bei konkreten Kopplungsbeispielen von standardisierten LANs hier bewältigen muß, wird im einzelnen in [20] beschrieben. Beispielsweise ist die Reihenfolge der Bits in den Adressen bei CSMA/CD genau umgekehrt wie bei Token Ring [122]. Bei völlig inkompatiblen Medienzugangsverfahren kommt nur noch eine Einbettung von PDUs in solche eines Transitnetzes in Frage [196]. Ist kein Durchreichen möglich, so muß auch die Prüfsumme der Pakete neu berechnet werden, wodurch ihre Ende–zu–Ende–Signifikanz verlorengeht und Fehler, welche die Bridge bei ihrer Bearbeitung verursacht, beim Empfänger nicht mehr erkannt werden können. Ansonsten sollte die Prüfsumme in einer Bridge beim Empfangen nicht ausgewertet und beim Senden nicht neu berechnet, sondern stattdessen transparent durchgereicht werden, was allerdings nur bei wenigen VLSI–Standardbausteinen für den Medienzugang möglich ist [79]. Bei Medienzugangsverfahren, welche Sendeberechtigungen vergeben, muß darauf geachtet werden, daß diese nicht über die Bridge ins andere Netz weitergegeben werden, insbesondere auch dann nicht, wenn auf der LLC–Teilschicht der verbindungslose quittierte Datagramm–Dienst verwendet wird.

Eine Bridge wird, aufgrund der fehlenden Möglichkeit einer effizienten Verkehrslenkung auf der Sicherungsschicht, nicht explizit adressiert. Sie empfängt normalerweise alle Pakete von den angeschlossenen Netzen, welche jeweils direkt die in der Regel flache und im Gesamtnetz eindeutige Adresse des Empfängers beinhalten. Dadurch werden in einer Bridge hohe Anforderungen an die Rate von Paketen, welche sie bearbeiten können muß, gestellt. Mit Hilfe eines Filtermechanismusses kann sie in der Regel dafür sorgen, daß ein Netz nicht durch den Internverkehr des anderen Netzes belastet wird. Dieser Filtermechanismus verwendet meist Adreßtabellen, deren Einträge die Zieladressen der Pakete repräsentieren, welche weiterzugeben sind (positiver Filter) oder derjenigen welche zu verwerfen sind, weil sie aus der Sicht der Bridge zum Internverkehr eines Netzes gehören (negativer Filter). Diese Filtertabellen können statisch sein und beim Hochfahren der Bridge von einer Datei geladen werden, dynamisch im Betrieb aufgebaut und ständig aktualisiert werden oder einen statischen sowie einen dynamischen Teil enthalten.

Filtertabellen können in Software realisiert sein, wobei zum Durchsuchen unterschiedlich effiziente Algorithmen eingesetzt werden können, welche sich stark in Zeit– und Speicherplatz-

bedarf unterscheiden. Eine Filtertabelle sollte sortiert sein, da sie bei jedem ankommenden Paket durchsucht werden muß und bei einer unsortierten Tabelle beispielsweise erst nach einem vollständigen Überprüfen aller Einträge festgestellt werden kann, daß sich die gesuchte Adresse nicht in der Tabelle befindet. Das lineare oder binäre Durchsuchen [178] kommt mit einem Minimum an Speicherplatz aus, ist aber vergleichsweise langsam. Dabei erfordert das lineare Durchsuchen deutlich mehr Suchschritte (bis zu n bei n Einträgen) als das binäre (nur bis zu $\log_2 n$), setzt aber keine sortierte Filtertabelle voraus und ist deshalb auch leichter in Hardware zu realisieren. Die gesuchte Adresse direkt als Index für die Filtertabelle zu verwenden, wäre zwar sehr schnell, würde aber bei weitem zu viel Speicherplatz reservieren, obwohl nur ein Bruchteil davon belegt wäre. Als Kompromiß bietet sich die Verwendung einer Hash–Tabelle an, bei welcher die Adressen mit Hilfe einer arithmetischen Funktion (Modulorechnung) in den Index für die Filtertabelle umgerechnet werden [135]. Dies hat jedoch den Nachteil, daß unterschiedliche Adressen auf denselben Index abgebildet werden können, obwohl auch hier der größte Teil der Tabelle nicht belegt ist, was die Effizienz des Filtermechanismusses reduziert. Eine schnellere Lösung ist die Verwendung einer speziellen Hardware zum (binären) Durchsuchen der (sortierten) Tabelle [79] oder eines inhaltsadressierbaren Speichers [193], bei welchem die gesuchte Adresse parallel mit allen Speicherzelleninhalten verglichen werden kann. Mittlerweile gibt es einen speziellen inhaltsadressierbaren Speicher von Advanced Micro Devices, welcher für diese Anwendung optimiert ist und bis zu 256 Adressen (je 48 Bit) aufnehmen und bearbeiten kann [52]. In allen Fällen wird aufgrund der flachen Adressen verhältnismäßig viel Speicherplatz benötigt, da es hier nicht reicht, nur bestimmte Teile der Adresse abzuspeichern und zu überprüfen.

In die Filtertabelle können auch Sicherheitsaspekte hineingearbeitet werden, so daß beispielsweise bestimmte Stationen nicht vom anderen Netz aus erreicht werden können. Außerdem besteht die Möglichkeit, auch Protokollsteuerinformationen höherer Schichten in die Tabelle mit aufzunehmen, was allerdings die Philosophie des Basisreferenzmodells verletzt. Man spricht dann von einem Protokollfilter. Dadurch kann beispielsweise erreicht werden, daß solche Pakete verworfen werden und dadurch das andere Netz nicht belasten, die eine Protokollsteuerinformation einer höheren Schicht enthalten von der bekannt ist, daß in diesem Netz keine Station mit diesem höheren Protokoll arbeitet [80]. Umgekehrt ist es auch möglich, nur Pakete für bestimmte höhere Protokolle durchzulassen.

Durch den Filtermechanismus eignet sich eine Bridge auch dazu, den bereichsübergreifenden Verkehr zwischen zwei oberhalb der Sicherungsschicht identischen (im wesentlichen durch Internverkehr belasteten) Netzen abzuwickeln. Es ist dann eine gleichzeitige Übertragung von Internverkehr in beiden Netzen möglich, was die Leistungsfähigkeit des Gesamtnetzes gegenüber einem durchgehenden oder über Repeater gekoppelten Gesamtnetz beträchtlich vergrößert [173]. Bei sinnvoller Aufteilung eines Gesamtnetzes in kleinere Netze, sollte der *Externverkehr beider Netze möglichst klein und die Auslastung der Teilnetze ungefähr gleich* sein. Dann sind auch die Anforderungen an den Durchsatz einer Bridge nicht allzu kritisch.

Bridges sind aufgrund der fehlenden Möglichkeit einer intelligenten Verkehrslenkung allerdings ungeeignet zur Kopplung einer sehr großen Anzahl von Netzen. Einerseits wäre die Belastung einer Bridge dadurch sehr groß, daß sie alle Pakete von beiden Netzen empfangen und bearbeiten müßte. Andererseits wäre bei einer unbekannten Empfängeradresse meist ein Rundsenden notwendig, was bei sehr großen Netzen viele Duplikate verursachen oder eine starke Einschränkung der Kommunikationspfade (keine parallelen Wege) erzwingen würde.

3.3.2.2 Kenngrößen einer Bridge

Es gibt zwei Kenngrößen zur Angabe der Leistungsfähigkeit einer Bridge [55]:

- Die *Filterrate* ist diejenige Anzahl von Paketen pro Sekunde, welche die Bridge ohne Paketverluste untersuchen kann. Dabei werden zusätzliche Filter (beispielsweise für die Datensicherheit) normalerweise nicht aktiviert. Sie liegt bei heutigen Bridges in der Größenordnung von 20000 Paketen pro Sekunde [152, 156].
- Die *Übertragungsrate* ist diejenige Anzahl von Paketen pro Sekunde, welche die Bridge maximal von einem Netz zum anderen übertragen kann. Sie liegt bei heutigen Bridges in der Größenordnung von 10000 Paketen pro Sekunde [152].

Auf parallele Bridges zwischen zwei LANs kann man also ohne allzugroße Probleme verzichten, da *eine* Bridge schon so schnell ist, daß sie bei sinnvollen Einsatzfällen den Externverkehr zwischen diesen LANs abwickeln kann. Beim Vergleich verschiedener Bridges sollte man allerdings die verwendete Meßmethode genauer hinterfragen, da bezüglich der Verkehrsaufteilung und der Paketgrößen Variationsmöglichkeiten bestehen. In der Regel wird nur Simplexverkehr bei leerem abgehendem Netz betrachtet, und es wird eine minimale Paketgröße verwendet, so daß die Kenngrößen maximal werden. Zur Bestimmung der Filterrate wird dann nur Internverkehr verwendet, so daß die Bridge nicht mit Übertragungen belastet ist, und zur Bestimmung der Übertragungsrate nur Externverkehr, so daß die Bridge nicht zusätzlich auch Internpakete untersuchen muß. Bei der Kopplung unterschiedlicher Netze sind die Kenngrößen für jede Richtung unterschiedlich und deshalb getrennt zu ermitteln. Desweiteren muß beachtet werden, daß es sich hier nur um Mittelwerte handelt und eine Bridge durch kurz hintereinander ankommende Pakete während eines Zeitintervalls, trotz Einhaltung der erlaubten Mittelwerte, überlastet werden kann [177].

Neben privaten Vorschlägen für die Verkehrslenkung und den Aufbau von Filtertabellen in Bridges, wie beispielsweise in [133], wird vor allem beim IEEE (Institute of Electrical and Electronics Engineers) an zwei Standards für Bridges gearbeitet, welche im folgenden beschrieben werden. Diese Standards regeln insbesondere auch das Zusammenspiel von mehreren Bridges und Netzen.

3.3.2.3 Spanning Tree Bridge

In [88] wird ein allgemeiner Bridge–Standard für alle LANs mit von IEEE standardisierten Medienzugangsverfahren vorgeschlagen, an dem seit 1984 gearbeitet wird. Dieser Standard ist so angelegt, daß normale Stationen der beteiligten Netze nicht modifiziert werden müssen, so daß er aufwärtskompatibel zu bisherigen IEEE–Standards ist. Normale Stationen sollen von der Existenz einer Bridge nichts bemerken. Sie wird deshalb auch als transparente Bridge bezeichnet [7].

Der Kernpunkt dieses Standards ist der sogenannte *Spanning Tree* Algorithmus, durch welchen die beliebige physikalische Topologie eines über Bridges gekoppelten Gesamtnetzes auf einen logischen Baum abgebildet wird, um Duplikate und endlos kreisende Pakete zu vermeiden. Damit stehen allerdings keine alternativen Wege für die Lastaufteilung zur Verfügung, und es kann auch kein optimaler Weg vom Sender zum Empfänger gewählt werden, da es nur *einen* Weg gibt und dieser durch die Baumtopologie vorgegeben ist. Die vorhandenen Betriebsmittel werden also nicht voll ausgenützt. Segmente und Bridges in der Nähe der Wurzel des Baumes werden verkehrsmäßig stärker belastet als diejenigen, welche weiter entfernt sind. Der logische Baum sollte deshalb so aufgebaut werden, daß er bereits vorhandenen hierarchischen Beziehungen unter den Stationen entspricht, um in den meisten Fällen möglichst kurze Kommunikationswege benützen zu können.

Der Aufbau des logischen Baumes ist eine Schichten–Management–Aufgabe und funktioniert völlig automatisch. Jede Bridge, die nicht durch das Netzmanagement deaktiviert wurde (Zustand: *Disabled*), nimmt daran teil. Es handelt sich also um ein selbstkonfigurierendes System. Der Zustand einer Bridge kann temporär für jedes Netz unterschiedlich sein. Ein vereinfachtes Implementierungsbeispiel ist in [145] zu finden. Alle Bridges erhalten neben ihren individuellen Adressen (die für jedes Netz unterschiedlich sein dürfen) eine Gruppenadresse, unter welcher sich auch alle anderen Bridges im gekoppelten Netz angesprochen fühlen. Während des Aufbaus empfängt jede Bridge (Zustand: *Listening*) nur Pakete, welche eine dieser zwei Adressen als Zieladresse enthalten. Die Ursprungsadresse ist dabei die individuelle Adresse der sendenden Bridge (auf diesem Netz). Solche Pakete werden Bridge Protocol Data Units (BPDUs) genannt. Aus der Ursprungsadresse wird eine Priorität abgeleitet, wobei die Bridge mit der höchsten Priorität im Laufe der Konfigurierung automatisch zur Wurzel des Baumes wird. Durch den Austausch von BPDUs werden Schleifen in der Topologie erkannt und die Bridges mit der kleineren Priorität in einer Schleife werden passiv (Zustand: *Blocking*), so daß sie später keine Pakete weitergeben. Sie beobachten aber weiterhin das Netz und beteiligen sich auch am Aufbau eines neuen logischen Baumes, falls sich an der Topologie (beispielsweise durch den Ausfall einer bisher aktive Bridge) etwas verändert. Bei derselben physikalischen Konstellation bildet sich durch den Spanning Tree Algorithmus deterministisch [43] immer der gleiche logische Baum. Die Topologie wird von der Wurzel aus durch Aussenden einer speziellen BPDU periodisch überprüft. Die Zeit zwischen zwei

BPDUs dieser Art wird in jeder nicht deaktivierten Bridge überwacht. Tritt in irgendeiner Bridge etwas Unvorhergesehenes ein, so stößt sie eine neue Konfigurierung an, und es bildet sich ein neuer logischer Baum.

Bis zur Erkennung einer Unregelmäßigkeit können vorübergehend Schleifen existieren, welche *Stürme* von Paketen und unterbrochene Kommunikationsbeziehungen aufgrund inkonsistenter Tabellen zur Folge haben können [61]. Ein Nachteil dieses Algorithmusses ist, daß während jeder neuen Konfigurierung der gesamte Externverkehr des gekoppelten Netzes verlorengeht. Die Verfügbarkeit des Gesamtnetzes kann man dadurch verbessern, daß man mehrere Inseln bildet, in welchen jeweils ein logischer Baum aufgebaut wird. Die Wurzeln der Bäume werden über eine *virtuelle Bridge* verbunden [45]. Ändert sich nun die Topologie in einer Insel, so ist die neue Konfigurierung auf diese Insel begrenzt.

Nach dem Aufbau des logischen Baumes beobachten die Bridges, welche nicht im Zustand *Blocking* sind, noch eine gewisse Zeit (Zustand: *Learning*) die Ursprungsadressen der Pakete auf den direkt angeschlossenen Netzen, um die Lage der aktiven Stationen zu erkennen und mit dieser Information selbständig den dynamischen Teil ihrer Filtertabellen aufzubauen. Anschließend gehen diese Bridges in den Normalbetrieb (Zustand: *Forwarding*) über. Sie empfangen alle Pakete von den angeschlossenen Netzen, tragen weiterhin die Ursprungsadressen in die Filtertabelle für die jeweilige Richtung ein und verwerfen die Pakete deren Zieladressen in derselben Filtertabelle gefunden werden, welche dadurch als Internverkehr erkannt sind. Es erfolgt hier also eine negative Filterung was bedeutet, daß Pakete mit einer bisher unbekannten Empfängeradresse auf jeden Fall weitergegeben werden und so ihr Ziel erreichen. Sie belasten aber unnötigerweise das andere Netz, wenn sie eigentlich zum Internverkehr gehören. Bei einer transparenten Bridge wird also für jedes Paket auf den angeschlossenen Netzen die Filtertabelle nach der Ursprungsadresse *und* nach der Zieladresse durchsucht, weshalb die Verwendung eines inhaltsadressierbaren Speichers (siehe 3.3.2.1) einen enormen Gewinn für die Leistungsfähigkeit bedeutet.

Es empfielt sich, das Alter der Einträge in den Filtertabellen zu überwachen und Adressen, welche längere Zeit nicht mehr als Ursprungsadressen beobachtet wurden, zu entfernen. Dazu kann man die Filtertabelle beispielsweise um eine Zählerspalte ergänzen, wobei bei jedem beobachteten Paket in der Zeile mit seiner Ursprungsadresse der Zähler inkrementiert wird. In periodischen Abständen wird diese Spalte überprüft und die Zähler zurückgesetzt. Zähler, die schon vor dem Zurücksetzen null waren, führen dazu, daß der gesamte Eintrag entfernt wird. Durch diesen Alterungsmechanismus werden die Filtertabellen kleiner, was die Suchzeit verkürzt, und falsche Einträge (beispielsweise weil sich eine Station mittlerweile an einem anderen Segment befindet) verschwinden spätestens nach zwei Perioden. Außerdem kann man dann bei voller Filtertabelle den ältesten Eintrag entfernen, wenn ein neuer aufgenommen werden soll.

3.3.2.4 Source Routing Bridge

In [90] wird eine Erweiterung des bisherigen Token–Ring–Standards vorgestellt, um die Kopplung von Ringen über Bridges mit einzubeziehen. Diese Arbeiten wurden 1986 begonnen. Dabei sind einige Änderungen des bestehenden Standards notwendig, so daß eine solche Bridge nicht von Paketen einer Station, deren Medienzugang nach dem bisherigen Standard für Token Ring implementiert wurde, passiert werden kann [186].

Kernpunkt dieser Erweiterung ist der *Source Routing* Algorithmus. Als Voraussetzung müssen Ringe und Bridges (wegen der Möglichkeit paralleler Bridges zwischen denselben Ringen) bei der Installation manuell numeriert werden. Diese Nummern müssen bei Konfigurationsänderungen gegebenenfalls wieder manuell geändert werden, damit sie dann im neuen gekoppelten Netz wieder eindeutig sind. Die Pakete werden um ein Informationsfeld [59] für die Verkehrslenkung ergänzt, dessen Existenz durch das erste Bit der Ursprungsadresse angezeigt wird, welches vom Standard ursprünglich als Indikator für eine Gruppenadresse vorgesehen war. Dieses Informationsfeld wiederum enthält ein Steuerfeld, in welchem unter anderem der Typ der Adressierung (individuell oder rundsenden) enthalten ist, und eine Sequenz aus Bridge– und Ringnummern (für maximal sieben Übertragungsabschnitte), welche den Weg vom Sender zum Empfänger beschreibt. Der Sender muß diesen Weg einer Tabelle entnehmen und jedem Paket mitgeben. Um ihre Filterfunktion wahrnehmen zu können, muß eine Bridge nur noch überprüfen, ob sie (4 Bit) und der nächste Ring (12 Bit) in dieser Sequenz enthalten ist und nicht mehr die 48–Bit–breiten Tabellen durchsuchen [170]. Es kann durch einen speziellen VLSI–Baustein dafür gesorgt werden, daß das Informationsfeld eines Paketes bereits während des Eintreffens decodiert wird, um abhängig vom Ergebnis des Vergleichs den Rest des Paketes auch noch zu empfangen oder nicht. Durch Verwendung einer einfacheren Hardware, welche alle Pakete empfängt, deren erstes Bit in der Ursprungsadresse gesetzt ist, werden ebenfalls von vorn herein in einer Bridge keine Internpakete empfangen [76, 192]. Auch müssen die Bridges die Lage einer Station nicht mehr durch Beobachten der Ursprungsadressen lernen und in Tabellen eintragen. Die Bridges werden dadurch relativ einfach, billig und schnell [16], und es können beliebige Topologien zugelassen werden, so daß auch eine Lastaufteilung über parallele Wege prinzipiell möglich ist. Sie eignen sich deshalb insbesondere für die Kopplung von Hochgeschwindigkeitsnetzen.

Andererseits wird jede einzelne Station, welche die Grenze ihres eigenen Ringes überschreiten können soll, wesentlich komplizierter und teurer (die Anzahlen von Stationen und Bridges unterscheiden sich normalerweise in ungefähr zwei Zehnerpotenzen). Sie muß eine Tabelle führen, welche die gesamte ihr bekannte Information zur Verkehrslenkung enthält, und diese bei fehlendem Eintrag einer gesuchten Zieladresse selbständig ergänzen. Dazu wird ein spezielles Entdeckungspaket rundgesandt, dessen Informationsfeld für die Verkehrslenkung unterwegs sukzessive in jeder Bridge um deren Nummer und um die des nächsten Ringes aufgefüllt wird. Dabei wird jeweils überprüft, ob die Nummer dieses nächsten Ringes schon

vorher enthalten war. Gegebenenfalls wird das als Duplikat erkannte Entdeckungspaket verworfen, da es sich dann um eine Schleife handelt. Bei diesem Rundsenden wird auch die maximale Paketgröße festgestellt, welche von allen beteiligten Ringen bearbeitet werden kann, und in das Informationsfeld des Entdeckungspaketes eingetragen. Da dieses Entdeckungspaket in jeder Bridge modifiziert wird, muß die Prüfsumme jedesmal neu berechnet werden, so daß sie ihre Ende–zu–Ende–Gültigkeit verliert. Der Empfänger erhält über jeden möglichen Weg genau ein solches Entdeckungspaket und schickt zu jedem auf dem gleichen Weg (die Information zur Verkehrslenkung ist ja jetzt in jedem Entdeckungspaket vollständig vorhanden) eine Quittung zum Sender zurück. Dieser hat dann anschließend mehrere alternative Wege zur Auswahl, von denen er einen, in der Regel den in der ersten zurückkommenden Quittung (dies war offensichtlich der schnellste Weg), in seine Tabelle zur Verkehrslenkung einträgt. Die Verkehrsaufteilung wird dabei aber nicht unbedingt optimal. In die Tabelle wird außerdem die maximale Paketgröße für diesen Weg eingetragen, so daß unterwegs kein Aufteilen notwendig ist.

Der Source Routing Algorithmus eignet sich insbesondere für verbindungsorientierte Protokolle auf höheren Schichten. Entdeckungspakete müssen dann höchstens beim Verbindungsaufbau ausgesandt werden. Ein Schwachpunkt des Source Routing Algorithmusses ist, daß die Topologie, im Gegensatz zum Spanning Tree Algorithmus, nicht überwacht wird, und die Stationen beispielsweise den Ausfall einer Bridge nicht bemerken, so daß sie aufgrund veralteter Tabellen viele Pakete nicht mehr erfolgreich verschicken können, obwohl vielleicht ein alternativer Weg vorhanden wäre.

Je größer ein gekoppeltes Netz wird, umso mehr Duplikate entstehen beim Rundsenden eines Paketes zur Verkehrslenkung. Dies ist bei Source Routing Bridges besonders kritisch, weil sich diese Pakete nicht nur entlang eines logischen Baumes ausbreiten können [206]. Bei sehr großen gekoppelten Netzen sollten deshalb Router (siehe Abschnitt 3.3.3) anstelle von Bridges eingesetzt werden, welche das Rundsenden durch eine intelligentere Verkehrslenkung vermeiden können. Insbesondere bei einer großen Anzahl von über Bridges gekoppelten Ringen kann das Gesamtnetz instabil werden, wenn (zum Beispiel nach einem Stromausfall) viele Bridges gleichzeitig ihre Tabellen durch Rundsenden von Paketen neu aufbauen.

Die Koexistenz von Source Routing und Spanning Tree Bridges muß durch die ersteren ermöglicht werden. Dieses Problem ist aber noch nicht zufriedenstellend gelöst. Es gibt spezielle Bridges, welche eine Transformation der Algorithmen derart vornehmen, daß sie den Stationen auf ihrer einen Seite vortäuschen, daß sich die Stationen auf der anderen Seite am gleichen Segment befinden [82, 121]. Man darf dabei aber nicht beliebige Topologien aus unterschiedlichen Netzen und Bridges aufbauen. Ein neuerer Standardisierungsvorschlag bei IEEE ist die Source Routing Transparent (SRT) Bridge, welche abhängig vom Indikator in der Ursprungsadresse einen der beiden Algorithmen auswählt [74]. Die SRT–Bridge soll normale Source Routing Bridges ersetzen und dadurch das Koexistenzproblem lösen.

3.3.2.5 Sonderformen

Während sich Bridges bei der Kopplung von Netzen mit hohem Internverkehr bewährt haben, verhalten sich Repeater bei niedriger Netzauslastung günstiger, da sie nur eine Verzögerung von wenigen Bits verursachen. In [135, 136] wird eine *Cut–Through Bridge* vorgeschlagen, welche für CSMA/CD–LANs die Vorteile beider Techniken vereinigt. Wenn die Bridge leer und das abgehende Netz frei ist, wird bereits während des Empfangens Bit für Bit regeneriert und weitergesandt. Tritt eine Kollision auf dem abgehenden Netz auf, oder wird beim Auswerten der Zieladresse festgestellt, daß das Paket herausgefiltert werden muß, so wird das Weitersenden abgebrochen und das Paket in der Bridge zwischengespeichert beziehungsweise verworfen. Ansonsten verhält sich die Cut–Through Bridge wie eine normale Bridge und ist insbesondere kompatibel mit den Standards für CSMA/CD und Spanning Tree Bridge. Im Niederlastfall arbeitet diese Bridge fast so gut wie ein Repeater und im Hochlastfall ist sie aufgrund unnötigerweise angefangener Übertragungen leicht schlechter als eine normale Bridge. Diese unnötigerweise angefangenen Übertragungen können umso schneller abgebrochen werden, je schneller die Suche in der Filtertabelle möglich ist.

Wenn eine Bridge mehr als zwei Netzanschlüsse hat, so spricht man von einer *Multiport Bridge* [155, 203]. Dabei muß für jedes Netz eine Filtertabelle vorhanden sein, und Pakete, deren Zieladressen in der eigenen Filtertabelle nicht gefunden werden, müssen an alle Netze (außer an das, von welchem sie gekommen sind) weitergegeben werden. Durch den Zusatzaufwand eines positiven Filtervorgangs mit allen anderen Filtertabellen kann man erreichen, daß die Pakete, deren Zieladresse in einer dieser Filtertabellen gefunden wird, nur auf dem jeweils dazugehörenden Netz weitergegeben werden. Multiport Bridges ermöglichen ganz andere Verkabelungskonzepte, da sie sternförmig viele Netze, beispielsweise auf einer Etage, miteinander verbinden können. Zwei Netzanschlüsse sollten dazu verwendet werden, die Verbindung mit Multiport Bridges in benachbarten Etagen zu realisieren. Auf diese Art und Weise ist die Verbindung zweier beliebiger Netze einer Etage über nur *eine* Multiport Bridge möglich, während bei der Verwendung normaler Bridges oft längere Wege in Kauf genommen werden müssen. Ein Nachteil dieser Lösung ist die geringere Zuverlässigkeit, da hier keine Redundanz vorhanden ist [125].

Bei der Kopplung zweier LANs über ein Transitnetz zur Überbrückung größerer Entfernungen kann man an den Netzgrenzen Pakete der LANs in die des Transitnetzes einbetten, siehe auch Bild 3.5. Erfolgt diese Einbettung auf der Sicherungsschicht des LANs, so werden die halben Netzkoppeleinheiten *Half Bridges* oder *Remote Bridges* genannt. Vor der Einbettung dürfen in diesem Fall auch Pakete aufgeteilt oder geblockt werden, da dies in der anderen Remote Bridge entsprechend wieder rückgängig gemacht werden kann. Bei der Einstellung von Zeitüberwachungen in den höheren Protokollen müssen die relativ großen Laufzeiten durch das Transitnetz berücksichtigt werden, um unnötige Wiederholungen zu vermeiden. Als Transitnetze kommen vor allem WANs [139] und MANs [54, 148, 157] in Frage. In [1]

wird das Breitband–ISDN auf der Basis von ATM als Transitnetz zwischen HSLANs (FDDI) verwendet. Falls ein öffentliches Netz als Transitnetz verwendet wird, sollte ein Teil des Pufferspeichers einer Remote Bridge für den Verkehr in Richtung LAN reserviert werden, damit die über das öffentliche Netz erfolgreich übertragenen Pakete, welche einen langen Weg hinter sich sowie Gebühren verursacht haben, möglichst nicht mehr verlorengehen. Wird das ISDN [138] als Transitnetz eingesetzt, so steht der Remote Bridge nach dem Verbindungsaufbau ein B–Kanal als Zugang zum Transitnetz zur Verfügung. Durch Ausnützung des Protokollmechanismusses Multiplexen auf der Vermittlungsschicht des B–Kanals [207] können über denselben B–Kanal mehrere räumlich getrennte LANs, jeweils hinter ihrer Remote Bridge, erreicht werden. Die Trennung der verschiedenen Richtungen erfolgt in der Ortsvermittlungsstelle der sendenden Remote Bridge. Bezüglich des Filtermechanismusses arbeitet diese Remote Bridge in abgehender Richtung wie eine Multiport Bridge. In der Regel muß eine Remote Bridge große Unterschiede in der Übertragungsgeschwindigkeit ausgleichen (beispielsweise 64 $kBit/s$ auf der WAN–Seite und 10 $MBit/s$ auf der LAN–Seite) [28]. Dabei muß die Ende–zu–Ende–Flußkontrolle auf einer höheren Schicht (zum Beispiel auf der Transportschicht) die Anzahl der Pakete, welche gleichzeitig auf einer Verbindung unterwegs sind, begrenzen. Die Remote Bridge hat keine Möglichkeit, weitere Verbindungen abzulehnen, da sie die Verbindungsaufbaupakete höherer Schichten als reine Datenpakete interpretiert. Der Filtermechanismus vor dem Transitnetz muß sehr effektiv arbeiten, um das Transitnetz möglichst wenig zu belasten und möglichst geringe Gebühren zu verursachen. Der ankommende Verkehr vom Transitnetz in der zweiten Remote Bridge braucht dann nicht mehr gefiltert zu werden. Remote Bridges können auch mehrere Anschlüsse für Transitnetze haben, um bei WANs den Unterschied der Übertragungsgeschwindigkeiten durch Lastaufteilung über parallele Wege zu verringern [78]. Da WANs, im Vergleich zu LANs, relativ unsichere Übertragungsstrecken sind, ist es hier besonders wichtig, die Prüfsumme des ersten Netzes unverändert zum zweiten zu übertragen, indem sie gemeinsam mit dem Rest des LAN–Paketes in ein WAN–Paket eingebettet wird. Während von IEEE auf der LAN–Seite der Spanning Tree Algorithmus vorgesehen ist, gibt es für die Seite des Transitnetzes vom Standard her keine Vorschriften [181]. Implementierungen verschiedener Hersteller müssen nicht als Remote–Bridge–Paar zusammenarbeiten können.

3.3.3 Router (Netzkopplung auf der Vermittlungsschicht)

3.3.3.1 Grundform

Eine Netzkoppeleinheit, welche auf der Vermittlungsschicht zwei oder mehr Netze miteinander verbindet, wird allgemein als *Router* bezeichnet. Solche Netzkoppeleinheiten gehören zu den ältesten überhaupt. Sie wurden bisher vor allem als Vermittlungsknoten zum Aufbau von WANs eingesetzt.

Im Gegensatz zu einer Bridge setzt ein Router das Vorhandensein einer Vermittlungsschicht in beiden Netzen voraus, was bei LANs oft nicht der Fall ist. Stationen, welche mit einem anderen Vermittlungsprotokoll als der Router arbeiten, können nicht über ihn kommunizieren. Ein Router ist deshalb bezüglich der Vermittlungsprotokolle nicht transparent, dafür aber aufgrund seiner Verkehrslenkungsmöglichkeit geeignet, auch sehr große Netze mit beliebiger Topologie aufzubauen. Router werden immer schneller und deshalb mittlerweile auch bei einer kleineren Anzahl von zu koppelnden LANs oder MANs als sinnvolle Alternative zu Bridges betrachtet. Heutige Router sind in der Lage, einige 1000 Pakete pro Sekunde zu übertragen [152]. Die Anzahl der Protokollinstanzen, welche in einem Router durchlaufen werden müssen, ist zwar größer als in einer Bridge, dafür wird aber ein Router nicht durch den Internverkehr der beteiligten Netze belastet [29]. Außerdem müssen Router ihre Verkehrslenkungstabellen nur nach der Zieladresse und nicht auch noch nach der Ursprungsadresse durchsuchen. Bei hohem Internverkehr ist es deshalb durchaus möglich, daß ein Router weniger ausgelastet ist als es eine Bridge an derselben Stelle wäre.

Sind die Vermittlungsprotokolle in den zu verbindenden Netzen unterschiedlich, so ist eine Transformation von Dienstprimitiven als Kopplungstyp notwendig. Adressen müssen in diesem Fall ebenfalls transformiert werden. Üblicherweise ist allerdings zumindest der Teil der Vermittlungsschicht, welcher für die Adressierung und Verkehrslenkung zuständig ist, in beiden Netzen identisch. Im folgenden werden stellvertretend für andere, einige der standardisierten Protokolle betrachtet, welche von der ISO für die Aufgaben der Vermittlungsschicht vorgesehen sind.

Die Vermittlungsschicht ist nach Abschnitt 2.2.1 in drei Teilschichten aufgeteilt, wobei die unterste teilnetzspezifisch und die oberste im gekoppelten Netz global einheitlich ist. Die teilnetzspezifischen Protokolle der beiden Netze können stark unterschiedlich sein, beispielsweise eines verbindungslos und das andere verbindungsorientiert. Sie können auch mit inkompatiblen Paketgrößen arbeiten, was durch geeignete Protokollmechanismen auf der mittleren oder oberen Teilschicht ausgeglichen werden muß und vor allem bei der Kopplung eines LANs mit einem WAN vorkommt [36]. Aufgrund der Dreiteilung der Vermittlungsschicht bietet es sich an, daß ein Router die Netzkopplung über das globale Protokoll in der obersten Teilschicht durchführt, welches insbesondere die Verkehrslenkung im gekoppelten Netz vornimmt. Ein solcher Router, der die PDUs über sein globales Protokoll unverändert auf das richtige abgehende Netz durchreicht, ist in Bild 3.9 dargestellt. Auf eine weitere Untergliederung der Schicht 3c im Router in Protokolle, Verkehrslenkungs– und Durchreichefunktion [115] wird in diesem Bild verzichtet.

Der Router wird mit seiner teilnetzspezifischen Adresse (bei LANs üblicherweise die Adresse der Sicherungsschicht) explizit adressiert, wodurch die Anforderungen an den Router wesentlich geringer sind als bei einer Bridge, welche zunächst alle Pakete empfangen und bearbeiten muß. Dadurch wird auch verhindert, daß Pakete einer defekten Station sich im ganzen Netz

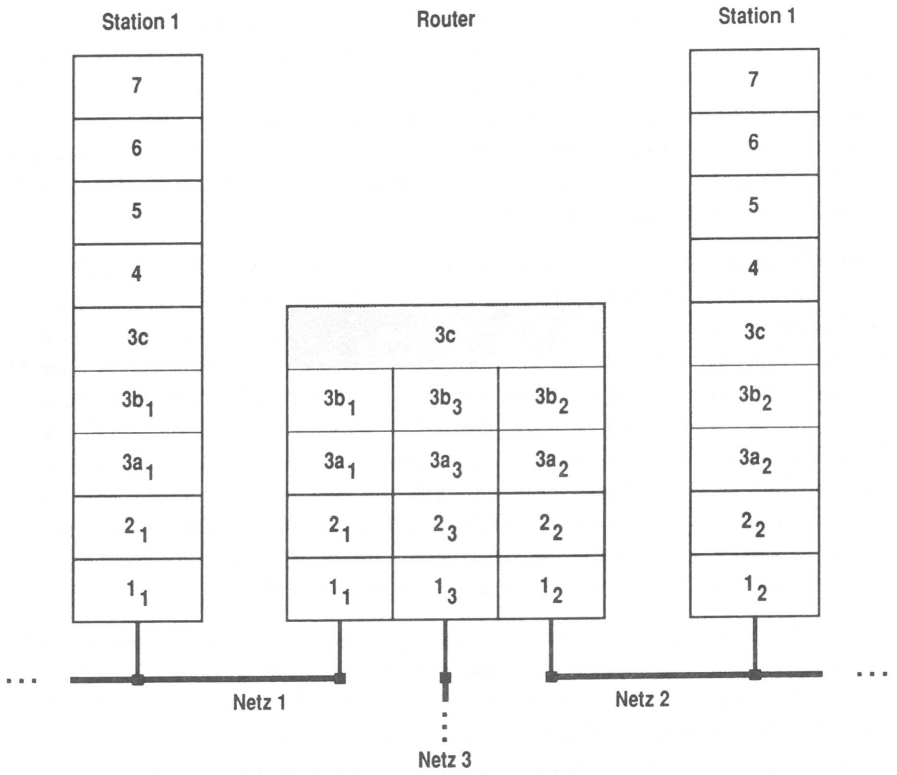


Bild 3.9: Netzkopplung über einen Router (Beispiel für drei zu verbindende Netze)

ausbreiten, nur weil ihre Zieladressen nicht in einer Filtertabelle gefunden werden (Bridge) oder weil sowieso alle Pakete weitergegeben werden (Repeater). Die eigentliche Vermittlungsadresse des Empfängers wird in der oberen Teilschicht des Routers ausgewertet und durchgereicht. Sie ist in der Regel hierarchisch strukturiert [123]. Deshalb müssen möglicherweise Adreßänderungen in den Stationen der beteiligten Netze vorgenommen werden, bevor man die Netze verbinden kann. Jeder Router muß nur die für ihn relevanten Adreßteile in seiner Verkehrslenkungstabelle führen. Dadurch sind diese in einem Router nicht so breit und auch nicht so lang wie die Filtertabellen einer Bridge. Nach der Auswahl des richtigen abgehenden Teilnetzes wird der Empfänger mit seiner teilnetzspezifischen Adresse adressiert.

Für die Verkehrslenkung gibt es viele Algorithmen [192], welche hier nicht im einzelnen diskutiert werden sollen. Beim Ausfall eines Weges oder Routers können oft alternative Wege bestimmt werden, so daß ein über Router gekoppeltes Netz eine hohe Zuverlässigkeit und Verfügbarkeit bietet [182]. Bei Verwendung eines geeigneten Transportprotokolls gehen auch

während der Umkonfigurierung keine Pakete verloren, sondern eine Wiederholung wird beim Sender veranlaßt. Ist der weitere Weg eines Paketes der Verkehrslenkungstabelle nicht zu entnehmen, so wird dieses Paket meist an einen hierarchisch höherstehenden Router weitergegeben. Dieser kann dann gegebenenfalls für spätere Pakete dem ersten Router eine direktere teilnetzspezifische Adresse mitteilen. Stellvertretend für andere wird im folgenden ein Algorithmus zur Verkehrslenkung prinzipiell vorgestellt, welcher insbesondere bei der Kopplung von LANs über Router verwendet wird.

Bei der Inbetriebnahme eines Routers wird zunächst der statische Teil seiner Verkehrslenkungstabelle von einer Datei eingelesen, falls ein solcher existiert. Für den dynamischen Aufbau oder zur Ergänzung des statischen Teils der Verkehrslenkungstabelle und zur ständigen Aktualisierung wurde von der ISO das verbindungslose Vermittlungsprotokoll, welches vor allem in LANs als globales Protokoll verwendet wird, um den Standard für ein *End System to Intermediate System Routeing Exchange Protocol*, oder kurz ES/IS–Protokoll, ergänzt [112]. Eine normale Station wird dort als *End System* (ES) und ein Router als *Intermediate System* (IS) bezeichnet. Dieser Standard regelt den Austausch von für die Verkehrslenkung notwendigen Informationen innerhalb *eines* LANs zwischen normalen Stationen und Routern sowie zwischen einzelnen Stationen untereinander. Das ES/IS–Protokoll kann dem Schichten–Management der Vermittlungsschicht zugeordnet werden.

Es wird für alle normalen Stationen und für alle Router an *einem* LAN je eine Gruppenadresse definiert. Durch spezielle ESH–Pakete (End System Hello), welche die Gruppenadresse der Router als Zieladresse enthalten, unterrichten die normalen Stationen periodisch die Router am selben LAN über ihre Verfügbarkeit. Die Adressen der jeweils unterstützten Dienstzugangspunkte der Vermittlungsschicht werden dann, neben weiteren Informationen, von jedem Router in seine Verkehrslenkungstabelle eingetragen, deren Aktualität ein Alterungsmechanismus garantiert. Analog dazu teilen die Router durch ISH–Pakete (Intermediate System Hello) periodisch den normalen Stationen an jedem direkt angeschlossenen LAN ihre teilnetzspezifischen Adressen mit, welche dort in entsprechende Tabellen eingetragen werden.

Ist die teilnetzspezifische Adresse des Empfängers eines Paketes nicht bekannt, so wird dieses Paket an irgendeinen Router am selben LAN geschickt. Dieser schickt es dann zum Empfänger oder an einen anderen Router weiter. Wenn ein direkterer Weg möglich gewesen wäre, so wird anschließend ein sogenanntes RD–Paket (ReDirect) vom Router an den Sender zurückgeschickt, um ihm für nachfolgende Pakete die teilnetzspezifische Adresse des Empfängers oder des direkteren Routers mitzuteilen. Ist momentan überhaupt kein Router verfügbar oder bekannt, so wird das Paket mit der Gruppenadresse aller normalen Stationen versehen und auf dem LAN ausgesandt, aber nur in der Station, welche den adressierten Dienstzugangspunkt der Vermittlungsschicht enthält, an die Transportschicht weitergegeben. Die erfolgreich adressierte Station teilt daraufhin für nachfolgende Pakete dem Sender durch ein ESH–Paket ihre teilnetzspezifische Adresse mit.

Sollen größere Netze mit Hilfe von Routern aufgebaut werden, bei welchen mehr als ein Router zwischen Sender und Empfänger liegen kann, so ist zusätzlich ein Austausch von für die Verkehrslenkung relevanten Informationen zwischen benachbarten Routern notwendig. Die hierfür benötigten Standards werden zur Zeit erarbeitet. In solchen großen Netzen werden Bereiche für die Verkehrslenkung gebildet [38]. Für den Austausch von Informationen zwischen Routern innerhalb *eines* Bereiches liegt inzwischen ein Vorschlag für einen Standard vor [117]. Immer dann, wenn sich am Zustand der Verbindung zwischen zwei Routern etwas ändert, teilen diese Router die Änderung allen anderen Routern in diesem Bereich mit. Dadurch erhält jeder Router Informationen über alle anderen Router sowie über deren Verbindungen untereinander. Aus diesen Informationen kann jeder Router lokal die kürzesten Wege von sich zu jedem Ziel ermitteln [58].

In [12] werden verschiedene Möglichkeiten zur Kopplung von LANs und WANs über Router vorgestellt. Als sinnvolle Alternativen für das globale Protokoll werden das verbindungslose und das verbindungsorientierte Vermittlungsprotokoll betrachtet, sowie X.25 (Packet Layer Protocol). Letztere Alternative wird aufgrund ihrer weiten Verbreitung in WANs in [56] favorisiert. Beispielhafte Implementierungen von Routern zur Verbindung von LANs über ein MAN [60], über ein WAN [65] oder über Breitband–ISDN [70, 147] sind in der Literatur zu finden. In den letzten drei Referenzen wird das verbindungslose Vermittlungsprotokoll eingesetzt, dem allerdings teilweise wieder ein verbindungsorientiertes Protokoll unterlagert ist. Die Kopplung verschiedener öffentlicher WANs auf der Vermittlungsschicht ist Gegenstand von CCITT–Empfehlungen [46, 47, 48, 49].

3.3.3.2 Sonderformen

Da Router, im Gegensatz zu Bridges, für die Protokolle der Vermittlungsschicht nicht transparent sind und auf demselben Medium von verschiedenen Stationen unterschiedliche Vermittlungsprotokolle verwendet werden können, werden auch Router angeboten, welche mehr als ein Vermittlungsprotokoll bearbeiten können. Die Erkennung des verwendeten Protokolls erfolgt anhand der Protokollidentifikation in der Protokollsteuerinformation eines jeden Paketes [36].

Neuerdings gibt es auch Router, welche zunächst alle Pakete auf den angeschlossenen Netzen empfangen. Diejenigen Pakete, welche durch ihre teilnetzspezifische Adresse an den Router adressiert sind, werden von der Sicherungsschicht an die Vermittlungsschicht weitergegeben und wie in einem normalen Router bearbeitet. Die übrigen Pakete werden wie in einer Bridge behandelt und abhängig von einer Filtertabelle auf der MAC–Teilschicht durchgereicht oder verworfen. Man spricht deshalb auch von einem *Brouter* [180] (Achtung: dieser Begriff wird von anderen Autoren teilweise auch für Bridges verwendet, welche eine Kopplung auf der LLC–Teilschicht durchführen). Er hat den Vorteil, für bestimmte Vermittlungsprotokolle eine

effiziente Verkehrslenkung durchzuführen und für alle anderen transparent zu sein. Insofern vereint er die Vorteile von Bridge und Router, hat aber gegenüber einem reinen Router den Nachteil, auch durch den Internverkehr der direkt angeschlossenen Netze belastet zu werden.

3.3.4 Gateway (Netzkopplung oberhalb der Vermittlungsschicht)

Netzkoppeleinheiten, welche auf der Transportschicht oder höher die Kopplung vornehmen, werden als *Gateway* bezeichnet. Es geht hier die Ende–zu–Ende–Signifikanz des Transportprotokolls verloren. Aufgrund dessen Definition im Basisreferenzmodell wird das Gateway gemeinsam mit dem gesamten Netz 2 vom Netz 1 aus als *ein* verteiltes Endsystem angesehen.

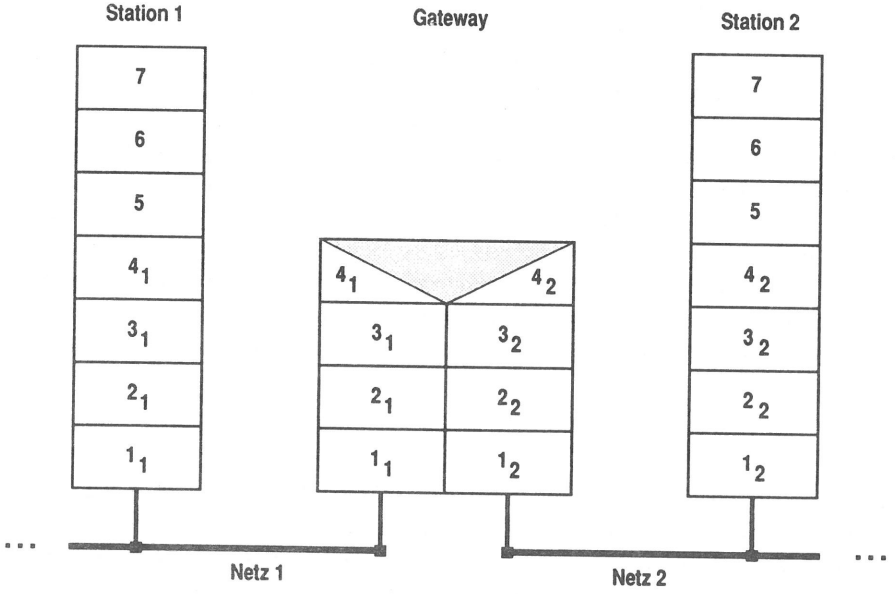


Bild 3.10: Netzkopplung über ein Gateway

Gateways werden ausschließlich zur Kopplung unterschiedlicher Netze eingesetzt, da sie im homogenen Fall gegenüber anderen Netzkoppeleinheiten in allen Bereichen unterlegen sind. Sie führen in der Regel eine Transformation von Dienstprimitiven [57] auf der Transportschicht [93], siehe Bild 3.10, oder auf der Verarbeitungsschicht durch und sind, mehr als andere Netzkoppeleinheiten, nur auf einen ganz speziellen Anwendungsfall zugeschnitten. Ein Verlust an Funktionalität ist normalerweise unvermeidbar und Transferzeiten durch ein Gateway sind wegen der großen Anzahl von zu durchlaufenden Protokollinstanzen vergleichsweise hoch. Ein Gateway wird wie ein Router explizit adressiert. In der Regel ist aufgrund der unterschiedlichen Protokolle auf der Kopplungsschicht eine Adreßtransformation notwendig.

Ein spezielles Beispiel für ein Gateway, welches auf der Verarbeitungsschicht eine Transformation von Dienstprimitiven vornimmt, wird in Kapitel 5 detailliert beschrieben. Ein solches Gateway ist immer dann notwendig, wenn die Protokolle auf der Verarbeitungsschicht unterschiedlich sind, oder wenn die Protokolle eines Netzes nicht gemäß dem Basisreferenzmodell geschichtet sind, also insbesondere auch zur Kopplung unterschiedlicher herstellerspezifischer Netze. Andere Beispiele für Gateways sind in [24] enthalten.

3.4 Netzkoppeleinheiten als Komponenten zum Aufbau einer komplexen Netzstruktur

Mit Hilfe der in Abschnitt 3.3 vorgestellten Netzkoppeleinheiten kann eine komplexe Netzstruktur aufgebaut werden, welche in [149] auch als Catanet (Concatenated network) bezeichnet wird. Eine solche Struktur soll nun anhand eines Beispiels erläutert werden, ausgehend von einem MAP–Netz in einer Fabrik [31]. Dabei wird der typische Einsatz unterschiedlicher Netzkoppeleinheiten nocheinmal zusammenfassend verdeutlicht und die unterschiedlichen Gründe, welche zum Einsatz von Netzkoppeleinheiten führen, werden erwähnt [15, 128]. Insbesondere für die Netzplanung ist es wichtig, die Vor– und Nachteile der einzelnen Netzkoppeleinheiten, sowie ihr typisches Einsatzgebiet in Verbindung mit den möglichen Problemen, zu kennen.

3.4.1 Segmentierung des homogenen MAP–Netzes

Die Segmentierung des homogenen MAP–Netzes ist notwendig, wenn seine *räumliche Ausdehnung*, die *Anzahl der angeschlossenen Stationen* oder der *Verkehr auf dem Netz* bestimmte Grenzwerte überschreiten würden. Diese Grenzwerte haben physikalische Ursachen wie Dämpfung, Reflexionen oder Laufzeit. Die Segmentierung größerer Netze führt darüberhinaus auch zu einer besseren *Fehlerisolation* und damit zur Vergrößerung der *Zuverlässigkeit* und *Verfügbarkeit* der einzelnen Netze. Außerdem kann die *Sicherheit* verbessert werden, wenn die Netzkoppeleinheiten Zugangsberechtigungskontrollen durchführen. Sind nur wenige Netzkoppeleinheiten in einem Netz vorhanden, über welche der Netzzugang von außen auf Stationen diseses Netzes möglich ist (analog zu Stadttoren in einer Stadtmauer), so ist der Netzzugang von außen relativ leicht zu überschauen und zu überwachen.

Repeater erlauben eine starke Kopplung weniger Netzsegmente. Insbesondere dürfen nicht mehr als zwei Repeater zwischen miteinander kommunizierenden Stationen liegen. Sie eignen sich vor allem dazu, solche Netzsegmente miteinander zu verbinden, die logisch gesehen zusammengehören, so daß ein Großteil des Verkehrs sowieso über die Netzgrenzen hinweg abgewickelt werden muß.

Ist eine größere Lokalität der Kommunikationsbeziehungen vorhanden oder ist kein weiterer Repeater mehr erlaubt, so bieten sich Bridges zur Segmentierung an. Aufgrund ihrer Filtereigenschaft können sie die Lokalität der Kommunikationsbeziehungen ausnützen und den Internverkehr der einen Seite von der anderen fernhalten. Die Lage einer Bridge sollte nach Abschnitt 3.3.2.1 so gewählt werden, daß der Verkehr über die Bridge minimal wird und der Internverkehr auf beiden Seiten der Bridge ungefähr gleich groß ist. Dadurch spiegelt sich in der Regel die Organisationsform des Betreibers in der Aufteilung der Netze und deren Kopplung wider. Durch die gleichzeitig mögliche Übertragung von Internverkehr auf beiden Seiten einer Bridge ist eine beträchtliche Leistungssteigerung des Catanets möglich. Dies wird noch verstärkt dadurch, daß die LANs auf jeder Seite der Bridge kleiner sind als das nichtsegmentierte Netz und deshalb effektiver arbeiten. Bridges sollten anstelle von Repeatern insbesondere dort eingesetzt werden, wo Rechner ohne eigene Festplatte am Netz betrieben werden, da diese einen hohen Verkehr verursachen, welcher sinnvollerweise von unbeteiligten Netzteilen fernzuhalten ist.

Die Anzahl der verwendeten Bridges zur Segmentierung sollte ebenfalls nicht allzu groß sein, da bei unbekannten Empfängeradressen, die Blindlast durch Rundsendepakete und ihre Duplikate überproportional mit der Anzahl der Bridges zunimmt. Zur Segmentierung großer Netze sind aufgrund ihrer Verkehrslenkungsmöglichkeit Router besser geeignet, falls die Anzahl der unterschiedlichen Vermittlungsprotokolle, welche dieser unterstützen muß, nicht zu groß ist.

Bei der Koexistenz von Bridges und Routern in einem Catanet muß man normalerweise darauf achten, daß sie nicht parallel zueinander zwischen zwei Netzen installiert werden [205], da ESH–Pakete nur für Router bestimmt sind, aber beispielsweise auch von einer normalen Spanning Tree Bridge weitergegeben werden, weil ihre Zieladresse nie als Ursprungsadresse auftaucht und deshalb auch nicht in der Filtertabelle der Bridge eingetragen ist. Dadurch erhält der Router dasselbe ESH–Paket auf beiden Seiten je ein Mal, und er weiß nicht, an welchem Netz die betreffende Station angeschlossen ist. Entsprechend würde jedes ISH–Paket des Routers auch auf beiden Netzen erscheinen und dadurch Verwirrung stiften, daß die enthaltene teilnetzspezifische Routeradresse nur für eine Seite richtig ist. Abhilfe wäre durch einen speziellen Filter möglich, welcher dafür sorgt, daß diese für die Verkehrslenkung benötigten Pakete die Bridge nicht passieren können.

3.4.2 Anbindung von herstellerspezifischen Netzen

Nach der Einführung von MAP entsteht die Forderung, daß Stationen mit herkömmlichen, *herstellerspezifischen Protokollen* mit den neuen Stationen kommunizieren können müssen. Dazu ist an der Schnittstelle zwischen den beiden unterschiedlichen Netzen ein spezielles Gateway notwendig, welches die Kopplung auf der Verarbeitungsschicht realisiert, da

herkömmliche Protokollprofile noch nicht mit MMS arbeiten. Ein *MAP–Gateway* ist ein Gateway, welches auf der einen Seite das vollständige MAP–Profil und auf der anderen Seite ein anderes Protokollprofil enthält. Ein solches MAP–Gateway ist Gegenstand des Kapitels 5.

3.4.3 Anbindung von feldbusähnlichen Netzen

Die Anbindung von Mini–MAP an MAP erfolgt über EPA–Stationen, welche beide Protokollprofile beinhalten. Solche Stationen können mit allen anderen Stationen an beiden Teilnetzen kommunizieren. Sollte eine Kommunikation über die Netzgrenze hinweg notwendig sein, so muß die betreffende EPA–Station eine Gateway–Funktion wahrnehmen und im wesentlichen MMS–Pakete durchreichen.

Die Kopplung eines Feldbusses (Beispiel: PROFIBUS) mit MAP oder Mini–MAP erfolgt über Gateways. Die Transformation auf der Verarbeitungsschicht ist meist sehr einfach, da die Verarbeitungsprotokolle (Beispiel: FMS) eng an MMS angelehnt sind. In der Regel ist eine direkte 1:1–Abbildung möglich.

3.4.4 Anbindung an TOP und an Weitverkehrsnetze

Im Büro hat sich TOP durchgesetzt. MAP und TOP können über einen Router gekoppelt werden, weil insbesondere die Medienzugangsverfahren, wegen *unterschiedlichen Standards*, verschieden sind. Dadurch ist ein Filetransfer über FTAM (File Transfer, Access and Management) [100] möglich, da FTAM bei Bedarf auch in MAP neben MMS auf der Verarbeitungsschicht vorgesehen ist. Zur Einbeziehung räumlich entfernter Rechenzentren oder Unternehmensniederlassungen, sind Remote Bridges, Router oder Gateways zu öffentlichen WANs notwendig.

3.5 Implementierungsaspekte

Jede Kette ist nur so stark wie ihr schwächstes Glied. Wenn eine Netzkoppeleinheit der Engpaß ist, so bestimmt sie im wesentlichen die Stabilitätsgrenze und die maximale Transfergeschwindigkeit auf jeder Kommunikationsbeziehung, an der sie beteiligt ist. Der Engpaß innerhalb einer Netzkoppeleinheit ist in der Regel die Protokollabwicklung und nicht die Transformation, das Durchreichen oder die Einbettung. Zur Verbesserung ihrer Leistungsfähigkeit sind bei der Implementierung einer Netzkoppeleinheit einige Grundregeln zu beachten.

Während es bei normalen Stationen sinnvoll ist, die Empfangsrichtung vor der Senderichtung zu priorisieren, ist es in einer Netzkoppeleinheit genau umgekehrt. Eine normale Station sollte

solche Pakete bevorzugt behandeln, die bereits Betriebsmittel des Netzes belegt haben, um Verluste (und dadurch bedingte Wiederholungen mit erneuter Belegung derselben Betriebsmittel) zu vermeiden. Außerdem kann es sich bei ankommenden Paketen in einer normalen Station auch um Quittungen handeln, welche belegte Pufferspeichersegmente der Station freigeben können, so daß ihre Priorisierung nicht nur aus der Sicht des Netzes, sondern auch aus der Sicht dieser Station günstig ist. Im Gegensatz dazu ist eine Netzkoppeleinheit nur eine Durchgangsstation, welche zwischengepufferte Pakete möglichst schnell auf dem abgehenden Netz wieder aussenden sollte, um den belegten Speicherplatz für anschließend ankommende Pakete wieder freizugeben [19]. Darüberhinaus sind auszusendende Pakete einer Netzkoppeleinheit im Durchschnitt bereits länger unterwegs als ankommende und sollten auch deshalb bevorzugt behandelt werden. Innerhalb einer Richtung sollte die Priorität mit dem Alter eines Paketes steigen.

Die Bearbeitungsgeschwindigkeit einer Netzkoppeleinheit ist abhängig von Prozessortyp und internem Kommunikationsmedium. Zur Vermeidung des zweiten Engpasses ist es sinnvoll, die Adresse eines Pufferspeichersegmentes, welches ein Paket enthält, von Schicht zu Schicht weiterzugeben, anstatt das Paket mit geringfügigen Modifikationen in ein neues Pufferspeichersegment zu kopieren. Damit auch bei notwendigen Transformationen das Umkopieren vermieden werden kann, muß das Paket mit einem solchen relativen Abstand zum Anfang des Pufferspeichersegmentes abgelegt werden, daß auch die längste in der Netzkoppeleinheit denkbare Protokollsteuerinformation noch in demselben Pufferspeichersegment untergebracht werden kann [190].

Auch bei Netzkoppeleinheiten mit mehreren Prozessoren sollte nach Möglichkeit das Umkopieren vermieden werden. Alle beteiligten Prozessoren sollten deshalb Zugriff auf einen gemeinsamen Speicher haben. Programme und lokale Variablen dürfen sich aber in lokalen Speichern der einzelnen Prozessoren befinden. Die Aufteilung der Arbeit auf die einzelnen Prozessoren kann beispielsweise so erfolgen, daß jede Richtung und jede Seite der Netzkoppeleinheit von einem eigenen Prozessor bearbeitet wird oder daß sogar noch eine weitere Aufteilung ensprechend der Protokollschichtung vorgenommen wird. Für den fehlerfreien Datentransfer nach einem Verbindungsaufbau kann auch pro Schicht eine spezielle Hardware eingesetzt werden, so daß durch *Pipelining* eine Parallelarbeit realisiert wird [84]. Pufferspeichersegmente können während dieser Phase mit bereits teilweise ausgefüllten Protokollsteuerinformationen vorausschauend zur Verfügung gestellt werden.

Zur Ausnützung von parallel bearbeitbaren Aufgaben an einem Paket innerhalb einer Schicht oder in der Transformationssoftware wird in [208, 209, 210] ein Transputernetz gewinnbringend eingesetzt. Die zugrundeliegende Hardware muß bereits bei der Erstellung der Software in einer geeigneten Programmiersprache, durch Definition der parallel bearbeitbaren Aufgaben, berücksichtigt werden.

Ein anderer Ansatz zur Steigerung der Bearbeitungsgeschwindigkeit wird in [81, 160] durch

eine verteilte Architektur für die Kopplung von HSLANs vorgeschlagen. Als internes Kommunikationsmedium für verschiedene Bearbeitungselemente wird ein Ring verwendet, auf welchem eine Sendeberechtigung oder ein Pulsrahmen von Bearbeitungselement zu Bearbeitungselement weitergegeben wird. Der Ring kann aufgrund der kurzen Entfernungen parallel realisiert werden, um die notwendige Übertragungsgeschwindigkeit *einer* Leitung zu reduzieren. Solche Architekturen werden auch in Paketvermittlungsknoten verwendet [86]. Die Anforderungen sind allerdings unterschiedlich, da dort viele, (noch) relativ langsame Leitungen miteinander verbunden werden, während hier wenige, sehr schnelle Netze gekoppelt werden sollen.

Damit der Speicherverschnitt wegen Paketen stark unterschiedlicher Größe nicht zu groß wird, kann man Pakete in gleichgroße Segmente (beispielsweise mit der minimalen Paketgröße) zerlegen und diese unabhängig voneinander abspeichern. Zu jedem Paket existiert dann eine Liste der dazugehörenden Pufferspeichersegmentadressen. Weitergereicht wird diese Liste oder wiederum nur die Adresse des Pufferspeichersegments, welches diese Liste enthält. Protokollmechanismen wie Aufteilen, Blocken oder Verketten kann man dann direkt auf diese Liste anwenden.

Kapitel 4

Verkehrsmodelle und Leistungsuntersuchungen

Nach den qualitativen Aussagen zur Netzkopplung im vorhergehenden Kapitel ist dieses Kapitel den quantitativen Untersuchungen gewidmet. Dazu müssen geeignete Verkehrsmodelle entwickelt und diese mit mathematischen oder simulativen Analysemethoden bewertet werden. Die dafür notwendigen Grundlagen werden im ersten Abschnitt bereitgestellt. Anschließend werden zwei Detailmodelle beschrieben. Die letzten beiden Abschnitte behandeln Leistungsuntersuchungen, insbesondere von Routern und Bridges. Das Vorgehen zur Bewertung von Gateways ist anhand eines konkreten Beispiels in Abschnitt 5.3 zu finden. Verkehrstheoretische Leistungsuntersuchungen an Repeatern sind nicht notwendig, da es sich hierbei praktisch um Verstärker handelt, welche eine konstante Verzögerung von wenigen Bits verursachen.

4.1 Grundlagen

4.1.1 Verkehrsmodelle

Verkehrsmodelle bilden die prinzipielle Funktionalität und das zeitliche Verhalten realer Systeme nach. Deshalb dürfen die Teile realer Systeme, welche sich nicht signifikant auf das zeitliche Verhalten auswirken, bei der Modellierung vernachlässigt werden. Die anderen Teile müssen unter Umständen, je nachdem was untersucht werden soll, sehr detailliert modelliert werden. Als Verkehrsmodelle haben sich insbesondere Petri–Netz– und Warteschlangenmodelle bewährt. Für zeitbehaftete Modellkomponenten werden zunächst einige Grundlagen zu stochastischen Prozessen zuammengestellt.

4.1.1.1 Stochastische Prozesse

Der Abstand zweier Ereignisse einer zeitbehafteten Modellkomponente wird mit Hilfe einer *Zufallsvariablen* T_{Zi} beschrieben. Sie hat die *Verteilungsfunktion*

$$F_{Zi}(t) = \mathrm{P}\{T_{Zi} \leq t\} \tag{4.1}$$

und die *Verteilungsdichtefunktion*

$$f_{Zi}(t) = F'_{Zi}(t) = \frac{\mathrm{d}F_{Zi}(t)}{\mathrm{d}t} \tag{4.2}$$

mit deren *Laplace-Transformierten*

$$\Phi_{Zi}(s) = \int_{0-}^{\infty} e^{-st} f_{Zi}(t)\, \mathrm{d}t\ . \tag{4.3}$$

Häufig sind von einer charakteristischen Zeitgröße durch Messung nur der *Erwartungswert*

$$\mathrm{E}[T_{Zi}] = z_i \tag{4.4}$$

und der *Variationskoeffizient*

$$c_{Zi} = \frac{\sqrt{\mathrm{VAR}[T_{Zi}]}}{\mathrm{E}[T_{Zi}]} = \sqrt{\frac{\mathrm{E}[T_{Zi}^2]}{\mathrm{E}^2[T_{Zi}]} - 1} \tag{4.5}$$

der Zufallsvariablen T_{Zi} bekannt, und die Verteilungsfunktion muß gegebenenfalls damit approximiert werden (siehe Abschnitt 4.1.3.1). Die in diesen beiden Gleichungen auftretenden *gewöhnlichen Momente* sind dabei folgendermaßen definiert:

$$\mathrm{E}[T_{Zi}^n] = \int_{0-}^{\infty} t^n f_{Zi}(t)\, \mathrm{d}t\ . \tag{4.6}$$

Im Rahmen dieser Arbeit werden drei stochastische Prozesse verwendet, deren charakteristischen Eigenschaften nun zusammengestellt werden sollen:

- Oft wird ein *allgemeiner stochastischer Prozeß* benötigt, über welchen keine genaueren Aussagen gemacht werden. Er wird mit der Abkürzung G (General) gekennzeichnet.

- Messungen haben gezeigt, daß der *Poisson–Prozeß* oft eine gute Approximation des Ankunftsprozesses aus sehr vielen unabhängigen Quellen ist. Er ist auch der einzige zeitkontinuierliche Prozeß, der die Eigenschaft der *Gedächtnisfreiheit* besitzt, welche auch als *Markoff–Eigenschaft* (Markovian, M) bezeichnet wird. Dies bedeutet, daß der Prozeßverlauf nur vom momentanen Zustand und nicht von seiner Vorgeschichte abhängt. Deshalb sieht ein externer Beobachter an zufälligen Zeitpunkten, als Verteilungsfunktion für das jeweilige Restintervall bis zum nächsten Ereignis, exakt die Verteilungsfunktion des Poisson–Prozesses selbst. Diese Tatsache führt bei der mathematischen Analyse von Verkehrsmodellen oft zu erheblichen Vereinfachungen oder macht sie überhaupt erst möglich. Der Poisson–Prozeß besitzt *negativ exponentiell verteilte*, unabhängige Ereignisabstände und hat folgende charakteristische Größen:

$$F_{Zi}(t) = 1 - e^{-\lambda_{Zi} t} \tag{4.7}$$

$$f_{Zi}(t) = \lambda_{Zi}\, e^{-\lambda_{Zi} t} \tag{4.8}$$

$$\Phi_{Zi}(s) = \frac{\lambda_{Zi}}{s + \lambda_{Zi}} \tag{4.9}$$

$$\mathrm{E}[T_{Zi}] = \frac{1}{\lambda_{Zi}} \tag{4.10}$$

$$c_{Zi} = 1\,. \tag{4.11}$$

Dabei ist λ_{Zi} die Ankunftsrate, wenn man als Beispiel annimmt, daß T_{Zi} einen Ankunftsabstand beschreibt.

- Bedienzeiten sind häufig konstant eine bestimmte Zeit t_0 lang, so daß sie *deterministisch verteilt* (Deterministic, D) sind und die folgenden charakteristischen Größen besitzen:

$$F_{Zi}(t) = \sigma(t - t_0) \tag{4.12}$$

$$f_{Zi}(t) = \delta(t - t_0) \tag{4.13}$$

$$\Phi_{Zi}(s) = e^{-st_0} \tag{4.14}$$

$$\mathrm{E}[T_{Zi}] = t_0 \tag{4.15}$$

$$c_{Zi} = 0\,. \tag{4.16}$$

Dabei sind die Sprungfunktion $\sigma(t - t_0)$ und der Dirac–Impuls $\delta(t - t_0)$ durch die folgenden Gleichungen definiert:

$$\sigma(t - t_0) = \begin{cases} 0 & \text{für } t < t_0 \\ 1 & \text{für } t \geq t_0 \end{cases} \tag{4.17}$$

$$\delta(t - t_0) = 0 \quad \text{für } t \neq t_0 \tag{4.18}$$

$$\int_{0-}^{\infty} \delta(t - t_0)\mathrm{d}t = 1 \,. \tag{4.19}$$

4.1.1.2 Petri–Netzmodelle

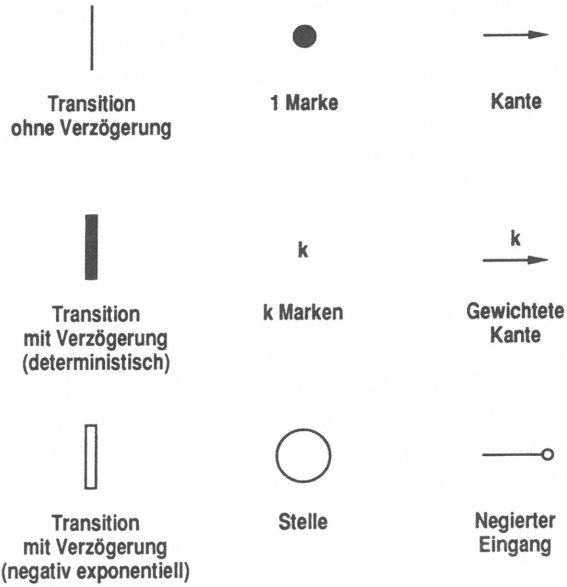


Bild 4.1: Symbole der Klasse der deterministischen und stochastischen Petri–Netze

Petri–Netzmodelle sind besonders geeignet, die *Funktionalität* von Teilen komplexer, realer Systeme mit mehreren gleichzeitig ablaufenden Vorgängen zu beschreiben, da bereits ihre Grundform Synchronisationsmechanismen enthält. Diese Grundform besteht im wesentlichen aus *Stellen*, deren Gesamtheit den aktuellen Zustand des Petri–Netzmodells repräsentiert, *Transitionen* zur Beschreibung von Ereignissen und *gerichteten Kanten* dazwischen. Stellen können durch *Marken* belegt werden, welche nicht voneinander unterscheidbar sind. Eine Transition kann immer genau dann *feuern*, wenn in allen ihren Eingangsstellen mindestens die jeweils benötigte Anzahl von Marken vorhanden ist. Diese werden dabei entfernt und jede Ausgangsstelle dieser Transition wird mit der spezifizierten Anzahl von Marken versehen. Es existieren viele Erweiterungen der Grundform in verschiedenen Richtungen, wovon im Rahmen dieser Arbeit aber nur eine angesprochen werden soll.

Um auch das zeitliche Verhalten zumindest teilweise beobachten zu können, werden zeitbehaftete Modellelemente eingeführt. *Eine* Möglichkeit ist die, an den Transitionen eine definierte *Verzögerung für das Feuern* zuzulasen. Während dieser Verzögerungszeit ist es möglich, daß Marken aus Eingangsstellen durch andere Transitionen abgezogen werden und dadurch das Feuern einer bereits aktivierten Transition verhindert wird. Die Klasse der *deterministischen und stochastischen Petri–Netze* [2] verwendet genau diese zeitbehafteten Modellelemente. Die Verzögerungszeit darf dabei deterministisch oder negativ exponentiell verteilt sein. Desweiteren sind bei dieser Klasse von Petri–Netzen auch *Verzweigungswahrscheinichkeiten* nach Stellen und *negierte Eingänge* an Transitionen zugelassen, welche ein Feuern genau dann ermöglichen, wenn die dazugehörenden Eingangsstellen keine Marke enthalten. Die Symbole für deterministische und stochastische Petri–Netze sind in Bild 4.1 zusammengestellt.

Für die mathematische Analyse solcher Petri–Netzmodelle ist die Einschränkung notwendig, daß durch *eine* Markierung nicht gleichzeitig zwei Transitionen mit deterministischer Verzögerung aktiviert werden dürfen [2] .

4.1.1.3 Warteschlangenmodelle

Um das *zeitliche Verhalten* realer Systeme zu untersuchen und charakteristische Größen zu ermitteln, haben sich vor allem Warteschlangenmodelle bewährt. Sie enthalten bereits in ihrer Grundform *zeitbehaftete Bedieneinheiten* und Möglichkeiten, die Einhaltung von bestimmten Reihenfolgen zu gewährleisten. Die Ankunftsabstände und Bedienzeiten werden üblicherweise mit Hilfe ihrer (möglicherweise approximierten) Verteilungsfunktion beschrieben. Die häufigsten Symbole von Warteschlangenmodellen sind in Bild 4.2 dargestellt.

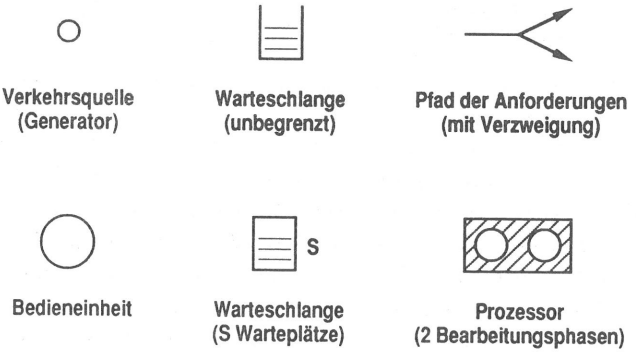
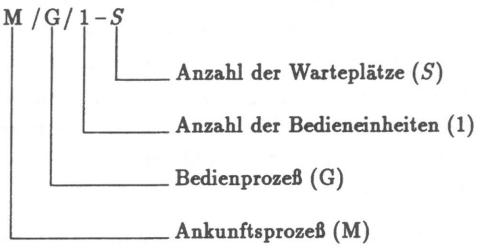


Bild 4.2: Häufigste Symbole in Warteschlangenmodellen

An *Verzweigungen* werden in der Regel Verzweigungswahrscheinlichkeiten mit angegeben. Für *Prozessoren* und *Warteschlangen* können, je nach Anwendungsfall, unterschiedliche Be-

arbeitungsreihenfolgen festgelegt werden. Im Rahmen dieser Arbeit werden *Bearbeitungsphasen* in Prozessoren mit nichtunterbrechenden Prioritäten versehen. Bearbeitungsphasen gleicher Priorität werden abwechselnd aktiviert. Die Abfertigungsstrategie von Warteschlangen ist hier ausschließlich die Bearbeitung in Ankunftsreihenfolge. Bei Warteschlangen mit einer begrenzten Anzahl von Warteplätzen gehen alle Anforderungen verloren, welche zu ihrem Ankunftszeitpunkt eine volle Warteschlange antreffen. Je nach Komplexität des zu modellierenden Systems müssen weitere Symbole eingeführt werden, welche zum Teil auch aus Petri–Netzmodellen entnommen sein können.

Für die Grundform eines einstufigen Warteschlangenmodells wurde 1953 von Kendall eine Notation eingeführt [124], welche in einer für begrenzte Warteschlangen erweiterten Form auch im Rahmen dieser Arbeit verwendet wird. Die Bedeutung der einzelnen Komponenten wird anhand eines Beispiels erläutert:



Für eine unbegrenzte Warteschlange wird der Parameter S der Einfachheit halber meist weggelassen.

4.1.2 Verkehrssimulation

Die Verkehrssimulation ist bei allen Leistungsuntersuchungen ein wichtiges Hilfsmittel. Bei komplexen Verkehrsmodellen ist sie oft die einzige Möglichkeit, um überhaupt zu quantitativen Aussagen zu gelangen. Bei einfacheren Verkehrsmodellen, welche einer mathematischen Analyse zugänglich sind, wird die Verkehrssimulation benötigt, um die mathemetischen Ergebnisse zu validieren, da diese oft mit Hilfe von approximativen Lösungsverfahren oder vereinfachenden Modellannahmen gewonnen werden. Nachdem die in dieser Arbeit verwendeten Meßmethoden bei der stationären und instationären Simulation beschrieben sind, soll anschließend auf ein interessantes Phänomen bei der Erfassung von Zeitintervallen während einer instationären Simulation hingewiesen werden, welches bisher noch nicht beobachtet wurde.

4.1.2.1 Stationäre Simulation

Die Verkehrssimulation ist ein experimentelles Verfahren, bei welchem die Ergebnisse durch statistische Messungen gewonnen werden. Es wird im Rahmen dieser Arbeit das *zeittreue, ereignisgesteuerte Simulationsverfahren* verwendet [132].

Das Verkehrsmodell, welches in abstrakter Weise das zu untersuchende System repräsentiert, wird mit Hilfe von komplexen Datenstrukturen auf ein Programm abgebildet. Die Anforderungen, welche das Verkehrsmodell durchlaufen, sind in der Regel ebenfalls komplexe Datenstrukturen, in die, beispielsweise zur Erfassung einzelner Laufzeiten, an definierten Stellen im Verkehrsmodell Zeitstempel eingetragen werden, welche an anderen Stellen auszuwerten sind.

Der Kern eines Simulationsprogramms ist die Ereignisliste, welche man oft auch als *Kalender* bezeichnet, weil sie wie ein Terminkalender verwendet wird und chronologisch geordnet ist. Es werden Ereignisse vorgeplant, indem sie an der richtigen Stelle in diese Liste eingetragen werden, und die Liste wird der Reihe nach abgearbeitet. Dabei entspricht die *simulierte Zeit* der Zeit, welche in dem momentan aktuellen Ereignis eingetragen ist. Nach der vollständigen Bearbeitung eines Ereignisses wird das nächste aus der Liste ausgetragen, und der simulierten Zeit wird ein neuer Wert zugewiesen. Die Zeit zwischen diesen Ereignissen wird dabei übersprungen (simuliert). Im englischsprachigen Raum wird für dieses Simulationsverfahren deshalb der Begriff *Event by Event Simulation* verwendet.

Als Ereignisse kommen naturgemäß alle Enden von definierten Zeitintervallen in Frage. Zeitintervalle werden durch Verteilungsfunktionen beschrieben, und einzelne Ausprägungen der dazugehörenden Zufallsvariablen werden mit Hilfe eines Zufallszahlengenerators ausgewürfelt. Ereignisse können am Anfang eines Zeitintervalls vorgeplant werden, beispielsweise

- Ankunftsereignisse mit Hilfe von Ankunftsabständen,
- Bedienendeereignisse von Bedieneinheiten und Prozessoren mit Hilfe von Bedienzeiten der aktuellen Phasen,
- Zeitüberwachungsereignisse mit Hilfe der konstanten Laufzeiten von Zeitüberwachungen und, bei Bedarf,
- Ereignisse zur Simulationssteuerung mit Hilfe von konstanten Abständen.

Zur Abarbeitung der Ereignisse sind universelle Programmteile notwendig, welche, aus logischer Sicht, teilweise zu den oben erwähnten Datenstrukturen der Modellkomponenten gehören (beispielsweise zum Eintragen einer Anforderung in eine Warteschlange). Bei dieser Abarbeitung werden in der Regel auch wieder neue Ereignisse vorgeplant.

Unmittelbar vor dem eigentlichen stationären Simulationslauf, in welchem die statistischen Größen gemessen werden, ist ein *Vorlauf* notwendig, während dem das ganze System einschwingt. Die anschließende Simulation wird in *Teiltests* unterteilt, so daß durch die Streuung

der Teiltestergebnisse eine Aussage über die statistische Zuverlässigkeit der Simulationsergebnisse, in Form von *Vertrauensintervallen*, gemacht werden kann. Grundlage dafür ist die Tatsache, daß die Teiltestergebnisse statistisch unabhängigen Stichproben entsprechen, welche bei unendlich vielen Ereignissen im Teiltest, nach dem zentralen Grenzwertsatz, normalverteilt wären. Als Simulationsergebnisse werden typischerweise mittlere Transferzeiten, Wartezeiten, Pufferspeicherbelegungen, Prozessorauslastungen, Warteschlangenlängen und Verlustwahrscheinlichkeiten erfaßt. Außerdem ist es als Hilfe für die Dimensionierung von Parametern in realen Systemen oft sinnvoll, neben diesen statistischen Größen auch absolute Maxima oder Minima von Zählern während eines Simulationslaufes festzuhalten.

Um kleine Vertrauensintervalle zu erhalten, sind sehr viele Ereignisse notwendig, was oft zu einer großen Rechenzeit führt. Wenn eine mathematische Analyse ebenfalls möglich und deren Genauigkeit mit Hilfe der Verkehrssimulation für typische Parameterkonstellationen validiert ist, so kann es deshalb bei umfangreichen Parameterstudien sinnvoll sein, nur noch die mathematische Analyse zu verwenden.

4.1.2.2 Instationäre Simulation

Um das Einschwingverhalten von Systemen bei Lastsprüngen und ihre Reaktion auf kurzzeitige Überlastsituationen zu untersuchen, ist eine instationäre Simulation notwendig. Dabei müssen Verkehrsquellen zu bestimmten Zeitpunkten zwischen verschiedenen Ankunftsraten umschalten können. Bei Poisson-Ankunftsprozessen, welche im Rahmen dieser Arbeit ausschließlich verwendet werden, kann man dieses Umschalten relativ einfach dadurch realisieren, daß zum Umschaltzeitpunkt mit Hilfe des neuen Ankunftsabstandes das nächste Ankunftsereignis bestimmt und in die Ereignisliste eingetragen wird, während das bereits vorgplante Ankunftsereignis zu suchen und zu entfernen ist [194]. Dabei wird die Gedächtnisfreiheit des Poisson-Ankunftsprozesses ausgenützt.

Üblicherweise besteht ein *Elementartest* aus einem Paar von komplementären Lastsprüngen. Zur Beobachtung von Simulationsergebnissen über der simulierten Zeit wird der Elementartest in dieser Arbeit in *Meßintervalle* unterteilt. Damit in jedem Meßintervall am Ende der Simulation genügend Meßwerte erfaßt sind, um eine sinnvolle Auswertung zu ermöglichen, muß nach dem Vorlauf eine große Zahl von unabhängigen Elementartests durchgeführt werden. Deshalb darf auch die Dauer der Meßintervalle nicht zu klein sein, andererseits darf sie aber auch nicht zu groß sein, damit sich die Meßwerte innerhalb eines Meßintervalls nicht wesentlich ändern, so daß die Vertrauensintervalle klein sind und der Prozeßverlauf gut aufgelöst werden kann. Wenn man die Dauern der verschiedenen Ankunftsraten so dimensioniert, daß sich das System am Anfang und am Ende eines Elementartests im selben, eingeschwungenen Zustand befindet, so läßt sich der Simulationslauf beispielsweise so organisieren, daß die Elementartests unmittelbar hintereinander (jeweils ohne einen erneuten Vorlauf) gestartet werden können. In diesem Fall bietet es sich an, die Ergebnisse für die verschiedenen

Meßintervalle in eine verkettete Ringliste einzutragen, so daß automatisch nach dem letzten Meßintervall eines Elementartests das erste Meßintervall für den nächsten Elementartest aktiviert wird. An den Meßintervallenden, welche man am einfachsten als spezielle Ereignisse in der Ereignisliste realisiert, muß insbesondere darauf geachtet werden, daß die Statistik abgeschlossen wird, bevor man das nächste Meßintervall aktiviert. Die statistischen Größen werden hier der aktuellen simulierten Zeit am jeweiligen Intervallende zugeordnet, da dann die resultierenden Ergebniskurven in der Regel genau bis zu den jeweiligen Lastsprüngen einen eingeschwungenen Zustand zeigen. Der gesamte Simulationslauf wird wieder, wie bei der stationären Simulation, in *Teiltests* unterteilt, so daß jeder Teiltest viele Elementartests enthält.

4.1.2.3 Beobachtbarkeit von Laufzeiten bei instationärer Simulation

Die Erfassung von Laufzeiten (Wartezeiten, Transferzeiten) über der simulierten Zeit, wie das bei einer instationären Simulation üblich ist, birgt ein besonderes Problem in sich. Da es sich hierbei um Zeitintervalle handelt, können sie naturgemäß nicht eindeutig einem bestimmten Zeitpunkt zugeordnet werden, sondern man hat verschiedene Möglichkeiten zur Auswahl.

Laufzeiten werden in einem Simulationspogramm nach Abschnitt 4.1.2.1 üblicherweise so gemessen, daß an definierten Stellen im Verkehrsmodell Zeitstempel in die Anforderungen eingetragen und an anderen ausgewertet werden. Die Beobachtung einer Laufzeit erfolgt also immer an ihrem *Endezeitpunkt*, so daß es naheliegend ist, sie auch über dieser simulierten Zeit aufzutragen. Wenn als Laufzeit beispielsweise die Transferzeit einer Anforderung von einem Sender zu einem Empfänger gewählt wird, so kann man das Ergebnis folgendermaßen interpretieren: Es handelt sich um die Transferzeit, welche der Empfänger während der jeweiligen simulierten Zeit beobachtet, also um die *Transferzeit aus der Sicht des Empfängers.*

Bei dieser Meßmethode tritt ein interessantes Phänomen auf, welches bisher noch nicht beobachtet wurde. Nach dem Ausschalten einer Verkehrsquelle steigt die bis dahin konstante mittlere Transferzeit an, wobei ihre Asymptote eine Gerade mit der Steigung 1 ist. Dies liegt daran, daß die nach dem Ausschalten beobachteten Anforderungen davor generiert worden sein müssen und ihre Transferzeit deshalb mindestens der Zeit entspricht, welche seit dem Ausschalten vergangen ist. Der Anstieg der mittleren Transferzeit nach dem Ausschalten kommt also daher, daß sich mit zunehmendem Abstand vom Ausschaltzeitpunkt die Verteilung der beobachteten Transferzeiten innerhalb eines Meßintervalls so verändert, daß immer weniger Meßwerte übrigbleiben, und zwar ausschließlich solche, die über der oben erwähnten Asymptote liegen. Noch erstaunlicher sieht das Ergebnis aus, wenn nicht das Ausschalten einer Verkehrsquelle untersucht wird, sondern der Sprung von einer Hochlast auf eine Grundlast. Die mittlere Transferzeit nimmt nach dem transienten Bereich einen niedrigen, konstanten Wert an. Unmittelbar nach dem Lastsprung steigt die mittlere Transferzeit jedoch zunächst, wie beim Ausschalten, wieder an. Sie erreicht einen maximalen Wert und

nimmt dann wieder ab, wenn innerhalb eines Meßintervalls nur noch ein vernachlässigbarer Anteil der großen Meßwerte enthalten ist. Ein Beispiel für dieses *Überschwingen* ist in Bild 5.9 enthalten.

Diese Methode hat den Nachteil, daß die Richtigkeit der Simulationsergebnisse nur schlecht überprüft werden kann. Bei der stationären Simulation kann man beispielsweise die mittlere Transferzeit durch Aufsummieren der entsprechenden mittleren Warte– und Bedienzeiten überprüfen, da diese unabhängig voneinander gemessen werden. Genau das geht aber bei der bisher beschriebenen Methode zur instationären Simulation nicht, da die Warte– und Transferzeiten *einer* Anforderung in verschiedenen Meßintervallen erfaßt werden.

Dieser Nachteil wird behoben, wenn alle Zeitintervalle über dem *Generierungszeitpunkt* der jeweiligen Anforderung aufgetragen werden. Wenn als Beispiel wieder die Transferzeit einer Anforderung betrachtet wird, so kann man das Ergebnis hier folgendermaßen interpretieren: Es handelt sich um die Transferzeit, welche eine soeben generierte Anforderung unterwegs sein wird, also um die vorausschauende *Transferzeit aus der Sicht des Senders.* Realisieren kann man diese Simulationsmethode dadurch, daß in jede Anforderung bei der Generierung die Nummer des dazugehörenden Meßintervalls eingetragen wird. Beim Erfassen von Laufzeiten wird diese Nummer gelesen und die richtige Stelle für den Eintrag des Ergebnisses in der oben erwähnten Ringliste mit Hilfe einer Rückwärtsverkettung gesucht.

Beim Auftragen von Laufzeiten über dem Generierungszeitpunkt der jeweiligen Anforderung tritt das oben beschriebene Phänomen nicht auf. Nach dem Ausschalten einer Verkehrsquelle werden gar keine Meßwerte mehr erfaßt und beim Sprung von einer Hochlast auf eine Grundlast steigen die mittleren Transferzeiten nicht an, bevor sie sich auf den neuen stationären Wert einschwingen. Diese Meßmethode wird gewählt, um die Wirksamkeit von Überlastabwehrmechanismen in Abschnitt 4.3.5 zu untersuchen, da die interessierenden Effekte hier nicht durch das optisch dominante Überschwingen verdeckt werden.

4.1.3 Mathematische Hilfsmittel

In diesem Abschnitt werden einige mathematische Hilfsmittel zusammengestellt [132], welche bei der analytischen Leistungsuntersuchung von Netzkoppeleinheiten verwendet werden.

4.1.3.1 Zweimomentenapproximation

Zur Approximation der Verteilungsfunktion einer Zufallsvariablen T_Z mit Hilfe ihrer ersten beiden Momente kann, abhängig vom Variationskoeffizienten, die *verschobene Exponentialverteilung* (für $0 \leq c_Z \leq 1$) nach Bild 4.3 oder die *hyperexponentielle Verteilung zweiter Ordnung* (für $c_Z > 1$) nach Bild 4.4 verwendet werden [129]. In Bild 4.4 ist q_1 eine Verzwei-

gungswahrscheinlichkeit. Die Approximation erfolgt so, daß die ersten beiden Momente der approximierten und der tatsächlichen Verteilungsfunktion übereinstimmen.

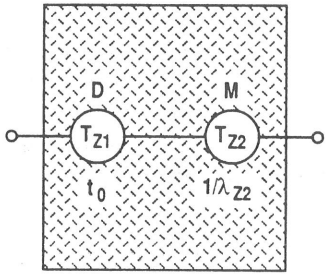


Bild 4.3: Verschobene Exponentialverteilung

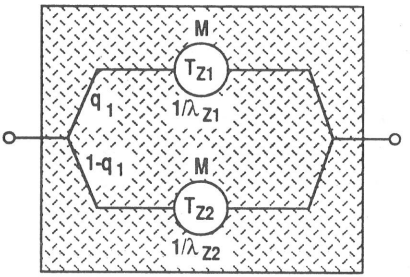


Bild 4.4: Hyperexponentielle Verteilung zweiter Ordnung

Als Verteilungs– beziehungsweise Verteilungsdichtefunktion erhält man für $0 \leq c_Z \leq 1$:

$$F_Z(t) = \sigma(t-t_0) \cdot \left(1 - e^{-\lambda_{Z2}(t-t_0)}\right) \tag{4.20}$$

$$f_Z(t) = \sigma(t-t_0) \cdot \lambda_{Z2}\, e^{-\lambda_{Z2}(t-t_0)} \tag{4.21}$$

mit den Parametern

$$t_0 = (1-c_Z)\mathrm{E}[T_Z] \tag{4.22}$$

$$\lambda_{Z2} = \frac{1}{c_Z \mathrm{E}[T_Z]}\,. \tag{4.23}$$

Entsprechend ergibt sich für $c_Z > 1$:

$$F_Z(t) = q_1\left(1 - e^{-\lambda_{Z1}t}\right) + (1-q_1)\left(1 - e^{-\lambda_{Z2}t}\right) \tag{4.24}$$

$$f_Z(t) = q_1\lambda_{Z1}\, e^{-\lambda_{Z1}t} + (1-q_1)\lambda_{Z2}\, e^{-\lambda_{Z2}t} \tag{4.25}$$

mit den Parametern

$$q_1 = \frac{1}{2}\left(1 + \sqrt{\frac{c_Z^2-1}{c_Z^2+1}}\right) \tag{4.26}$$

$$\lambda_{Z1} = \frac{2q_1}{\mathrm{E}[T_Z]} \tag{4.27}$$

$$\lambda_{Z2} = \frac{2(1-q_1)}{\mathrm{E}[T_Z]}\,. \tag{4.28}$$

4.1.3.2 Laplace–Transformation

Die Laplace–Transformierte einer Verteilungsdichtefunktion ist in Gleichung (4.3) definiert. Sie hat einige Eigenschaften, welche insbesondere bei der Summe von unabhängigen Zufallsvariablen, beispielsweise $T_Z = T_{Z1} + T_{Z2}$, ausgenützt werden können. Die Laplace–Transformierte der Verteilungsdichtefunktion dieser Summenzufallsvariablen ergibt sich als Produkt der einzelnen Laplace–Transformierten $\Phi_{Z1}(s)$ und $\Phi_{Z2}(s)$

$$\Phi_Z(s) = \Phi_{Z1}(s) \cdot \Phi_{Z2}(s) \ , \tag{4.29}$$

was im Zeitbereich einer Faltung entspricht.

Durch Rücktransformation erhält man daraus die Verteilungsdichtefunktion der Summenzufallsvariablen oder man kann zumindest ihren Erwartungswert aus Gleichung (4.4) und den Variationskoeffizienten aus Gleichung (4.5) ermitteln, denn es gilt für das n–te gewöhnliche Moment von T_Z

$$\mathrm{E}[T_Z^n] = (-1)^n \cdot \Phi_Z^{(n)}(s) \mid_{s=0} \ . \tag{4.30}$$

4.1.3.3 Methode der eingebetteten Markoff–Kette

Wenn bei einem einstufigen Warteschlangenmodell der Ankunfts– oder der Bedienprozeß ein Poisson–Prozeß (M) und der andere Prozeß allgemein (G) ist, so kann zu seiner mathematischen Analyse die Methode der *eingebetteten Markoff–Kette* angewandt werden. Der Zustandsprozeß ist zwar gedächtnisbehaftet, er besitzt jedoch ausgezeichnete *Regenerationszeitpunkte*, an welchen er sein Gedächtnis verliert. Diese liegen jeweils am Ende der allgemeinen Phase. Zwischen zwei Regenerationszeitpunkten entwickelt sich der Zustandsprozeß, gemäß dem gedächtnisfreien Poisson–Prozeß, als reiner Geburts– oder Sterbeprozeß und ist deshalb nur noch abhängig vom Zustand am letzten Regenerationszeitpunkt und nicht mehr von dessen Vorgeschichte. Die Zustandswahrscheinlichkeiten an den eingebetteten Regenerationszeitpunkten kann man mit Hilfe von *Übergangswahrscheinlichkeiten* berechnen. Aus ihnen lassen sich die Zustandswahrscheinlichkeiten zu beliebigen Zeitpunkten ermitteln, welche die Voraussetzung für viele charakteristische Größen des Warteschlangenmodells sind.

4.1.3.4 Warteschlangennetze

Mit einstufigen Warteschlangenmodellen als *Knoten* kann man Warteschlangennetze zusammensetzen, welche *offen* (alle Anforderungen kommen von außen an und verlassen das Warte-

schlangennetz auch nach außen wieder), *geschlossen* (alle Anforderungen zirkulieren innerhalb des Warteschlangennetzes) oder *gemischt* sein können.

Für die mehrdimensionalen Zustandswahrscheinlichkeiten existiert eine exakte, geschlossene Lösung, falls die Ankunftsprozesse von außen Poisson–Prozesse sind, wobei die Gesamtankunftsrate von der momentanen Anzahl von Anforderungen im Netz abhängig sein darf. Außerdem müssen die Warteschlangen unbegrenzt sein, und das Warteschlangennetz darf nur Knoten der folgenden vier Kategorien enthalten [10]:

1. Eine oder mehrere Bedieneinheiten, deren Bedienzeiten negativ exponentiell verteilt (M) sind, wobei ihr Mittelwert von der momentanen Anzahl von Anforderungen in diesem Knoten abhängig sein darf. Anforderungen werden in Ankunftsreihenfolge bedient (die umgekehrte Ankunftsreihenfolge oder eine zufällige Reihenfolge wäre darüberhinaus auch möglich).
2. *Eine* Bedieneinheit, deren Bedienzeiten beliebig verteilt (G) sind. Alle sich momentan in diesem Knoten befindlichen Anforderungen teilen die Kapazität der Bedieneinheit gleichmäßig nach dem Zeitscheibenverfahren mit infinitesimal kleinen Zeitscheiben.
3. Unendlich viele Bedieneinheiten, deren Bedienzeiten beliebig verteilt (G) sind.
4. *Eine* Bedieneinheit deren Bedienzeiten beliebig verteilt (G) sind. Neu eintreffende Anforderungen werden sofort bedient, wobei eine eventuell sich gerade in Bedienung befindliche Anforderung dadurch unterbrochen wird. Unterbrochene Anforderungen setzen später ihre Bedienung vom jeweils erreichten Bedienungszustand aus, in umgekehrter Ankunftsreihenfolge, fort.

Die mehrdimensionalen Zustandswahrscheinlichkeiten sind proportional zum Produkt von Termen, welche jeweils nur von *einem* Knoten abhängen (*Produktlösungsform*) [132]. Zur Bestimmung dieser Terme dürfen an jedem Knoten negativ exponentiell verteilte Ankunftsabstände angenommen werden, obwohl dies bei offenen Warteschlangennetzen mit Rückkopplungen und bei geschlossenen Warteschlangennetzen normalerweise nicht den realen Ankunftsprozessen entspricht. Bei offenen, rückkopplungsfreien Warteschlangennetzen nur mit Knoten der Kategorie 1 ist das eine Folge des *Ausgangsprozeßtheorems von Burke* [39], welches besagt, daß bei einem Poisson–Ankunftsprozeß der Ausgangsprozeß eines solchen Knotens wieder ein Poisson–Prozeß ist. Diese Eigenschaft gilt darüberhinaus auch für die drei anderen Kategorien.

4.2 Modellierung und Analyse von Protokollmechanismen

Zur detaillierten Modellierung der in Abschnitt 2.2.2 angesprochenen Protokollmechanismen eignen sich besonders Petri–Netzmodelle, weil dazu auch Synchronisationsmechanismen benötigt werden. Es wird hier die bereits angesprochene Klasse der deterministischen und stochastischen Petri–Netze verwendet, um auch Aussagen über die Leistungsfähigkeit der untersuchten Mechanismen zu erhalten. In [35] werden generische Petri–Netzmodelle präsentiert, von denen zwei zum Verständnis des nächsten Abschnitts hilfreich sind und deshalb hier vorgestellt werden sollen. Diese Petri–Netzmodelle dienen vor allem dazu, die Funktionsweise der modellierten Protokollmechanismen im Detail zu verstehen.

4.2.1 Blocken

Das Blocken von mehreren SDUs zu einer PDU erfolgt *vor* der Bearbeitung eines Paketes in der betrachteten Schicht. Das Petri–Netzmodell für diesen Protokollmechanismus ist in Bild 4.5 dargestellt. Seine Umgebung wird durch die Transitionen *Input* und *Output1* beziehungsweise *Output2* repräsentiert.

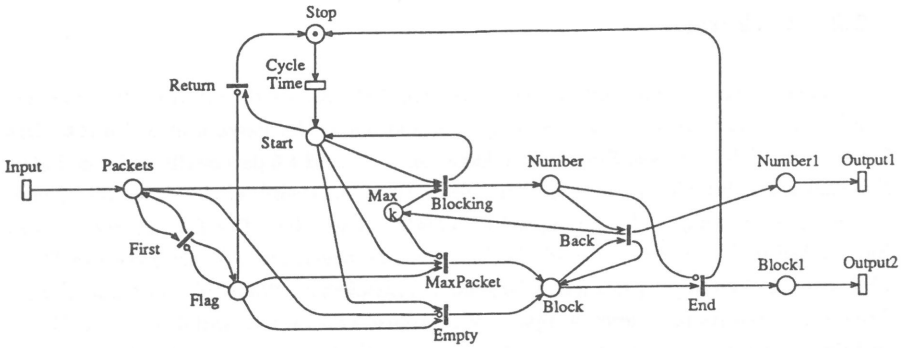


Bild 4.5: Petri–Netzmodell für das Blocken mehrerer SDUs zu *einer* PDU

Die Transition *Cycle Time* repräsentiert die (negativ exponentiell verteilte) Zeit, welche der für die betrachtete Schicht zuständige Prozessor zur Erfüllung anderer Aufgaben (außer dem Blocken) benötigt. Sind diese anderen Aufgaben erledigt, so wird eine Marke in der Stelle *Start* eingetragen. Falls keine SDU während des letzten Zyklusses angekommen ist, wird diese Marke über die Transition *Return* an die Stelle *Stop* weitergegeben und ein neuer Zyklus beginnt. Wenn mindestens eine SDU angekommen ist, also in der Stelle *Packets* sich mindestens eine Marke befindet, so wird nach dem Ende des gegenwärtigen Zyklusses das

Blocken gestartet. Dabei hat die erste ankommende SDU, neben der Marke in der Stelle *Packets*, bereits eine Marke in der Stelle *Flag* erzeugt, um anzuzeigen, daß ein neuer Block zusammengestellt wird. Die maximale Anzahl von SDUs, welche geblockt werden dürfen, ist durch die Anzahl k der Marken in der Stelle *Max* festgelegt. Über die Transition *Blocking* erreichen, begrenzt durch die kleinere der Anzahlen von Marken in den Stellen *Packets* und *Max*, so viele Marken die Stelle *Number*, wie SDUs zu *einer* PDU geblockt werden. Je nachdem wo die Marken zuerst ausgehen, feuert eine der beiden Transitionen *MaxPacket* oder *Empty*, trägt eine Marke, welche die erzeugte PDU repräsentiert, in der Stelle *Block* ein und zieht dabei die Marken aus den Stellen *Start* und *Flag* ab, so daß die nächste ankommende SDU zur ersten SDU des nächsten Blocks wird. Die Transition *Back* dient dazu, die der Stelle *Max* entnommenen Marken wieder zurückzugeben, um einen definierten Ausgangszustand für den nächsten Zyklus zu erreichen. Dabei werden die Marken aus der Stelle *Number* in die Stelle *Number1* kopiert, welche dadurch die Anzahl der SDUs in der resultierenden PDU repräsentiert. Die resultierende PDU selbst erreicht über die Transition *End* die Stelle *Block1* und beendet den Protokollmechanismus durch das Eintragen einer Marke in der Stelle *Stop*. Nachdem der entsprechende Prozessor die Bearbeitung seiner anderen Aufgaben wieder beendet hat, kann auf dieselbe Art und Weise ein Blocken der inzwischen neu eingetroffenen oder noch wartenden SDUs durchgeführt werden.

4.2.2 Verketten

Im Gegensatz zum Blocken erfolgt das Verketten mehrerer PDUs zu einer SDU der unterlagerten Schicht *nach* der Bearbeitung eines Paketes in der betrachteten Schicht. Das Petri–Netzmodell für diesen Protokollmechanismus ist in Bild 4.6 dargestellt und dem Petri–Netzmodell für das Blocken sehr ähnlich. Seine Umgebung wird wieder durch die Transitionen *Input* und *Output1* beziehungsweise *Output2* repräsentiert. Der Hauptunterschied zu Bild 4.5 liegt in der Art und Weise wie das Verzögern der zuerst eintreffenden Pakete realisiert wird, um eine Vereinigung mit nachfolgenden zu ermöglichen. Während dies beim Blocken durch die Bearbeitung anderer Aufgaben vom selben Prozessor verwirklicht werden kann, ist beim Verketten eine künstliche Verzögerung nach der Bearbeitung der PDUs notwendig. Um diese Verzögerung auf eine maximale Dauer zu begrenzen, ist hier eine Zeitüberwachung erforderlich. Bild 4.6 kann außerdem als eine zweite Implementierungsvariante des Protokollmechanismusses Blocken angesehen werden.

Solange das eigentliche Verketten noch nicht begonnen hat, erreichen ankommende PDUs über die Transition *Gate* die Stelle *Segments*. Die erste dieser PDUs startet die deterministische Zeitüberwachung, indem über die Transition *First* in die Stelle *Start* eine Marke eingetragen wird. Es gibt jetzt zwei Endekriterien, welche das eigentliche Verketten auslösen und die Anfangsmarke aus der Stelle *Idle* abziehen, so daß anschließend ankommende PDUs

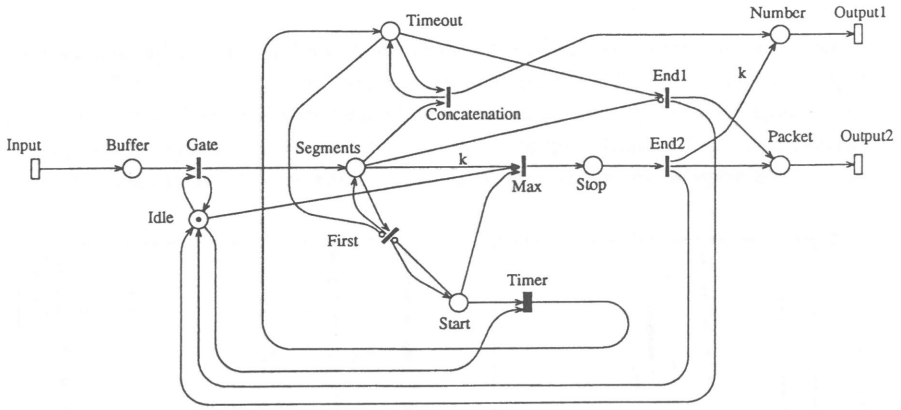


Bild 4.6: Petri–Netzmodell für das Verketten mehrerer PDUs zu *einer* SDU der unterlagerten Schicht

in der Stelle *Buffer* zwischengespeichert werden, bis das aktuelle Verketten abgeschlossen ist: Entweder wird die maximale Anzahl k von PDUs erreicht, welche zu einer SDU verkettet werden dürfen, oder die Verzögerung erreicht die von der Zeitüberwachung festgelegte maximale Dauer. Im ersten Fall feuert die Transition *Max*, an der *ein* Eingang mit der Anzahl k gewichtet ist, und zieht dabei auch die Marke aus der Stelle *Start* ab, um die Zeitüberwachung anzuhalten. Die Marke in der Stelle *Stop* repräsentiert die entstandene verkettete SDU für die unterlagerte Schicht. Sie erreicht über die Transition *End2* die Stelle *Packet*. Dabei erhält die Stelle *Number* k Marken, so daß dieser die Anzahl der verketteten PDUs entnommen werden kann, und durch eine Marke in der Stelle *Idle* wird jetzt die Verkettung nachfolgender Pakete ermöglicht. Im zweiten Fall feuert die Transition *Timer*, welche die Zeitbegrenzung modelliert, und trägt die Marke der Stelle *Start* in die Stelle *Timeout* ein. Daraufhin werden die Marken der Stelle *Segments* über die Transition *Concatenation* in die Stelle *Number* übertragen, so daß dieser wieder die Anzahl der verketteten PDUs entnommen werden kann. Anschließend kann die Transition *End1* feuern und die Marke, welche die resultierende SDU repräsentiert, in die Stelle *Packet* eintragen, sowie jetzt, ebenfalls durch eine Marke in der Stelle *Idle*, die Verkettung nachfolgender Pakete ermöglichen.

4.2.3 Analyse der beiden Protokollmechanismen

Zur funktionellen Simulation (Verifikation) und zum Ermitteln charakteristischer Größen können heute Softwarewerkzeuge wie GreatSPN (Great Stochastic Petri Nets) [51] eingesetzt werden. Aus den ermittelten Wahrscheinlichkeitsverteilungen für die Markierungen des betrachteten Netzes und aus der Ankunftsrate kann man mit Hilfe des Gesetzes von Little [146] charakteristische Zeiten berechnen.

In Bild 4.7 ist die normierte mittlere Wartezeit von SDUs bis zu ihrem Blocken über dem Angebot aufgetragen. Beide Größen sind auf die Zeit bezogen, welche der Prozessor für andere Aufgaben benötigt. Als Parameter wird die maximale Blockgröße k verwendet. Die normierte mittlere Wartezeit steigt mit zunehmendem Angebot und nimmt mit zunehmender maximaler Blockgröße ab, da dann pro Zyklus mehr SDUs geblockt werden können.

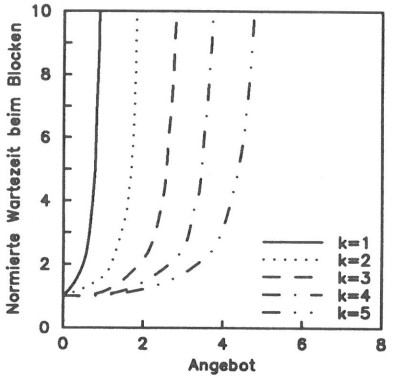


Bild 4.7: Normierte mittlere Wartezeit beim Blocken über dem Angebot

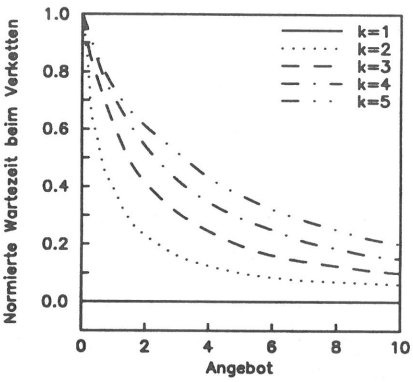


Bild 4.8: Normierte mittlere Wartezeit beim Verketten über dem Angebot

Völlig anders sieht das entsprechende Ergebnis für das Verketten aus. Das Angebot und die normierte mittlere Wartezeit ankommender PDUs (künstliche Verzögerung) in Bild 4.8 sind auf die maximale Zeit für das Verketten bezogen. Parameter ist die maximale Anzahl zu verkettender PDUs. Die normierte mittlere Wartezeit nimmt mit zunehmendem Angebot ab und ist proportional zur maximalen Anzahl zu verkettender PDUs. Für $k = 1$ gibt es keine künstliche Verzögerung, da kein Verketten stattfindet. Die künstliche Verzögerung ist nach oben auf die maximale Zeit für das Verketten (normiert: 1.0) begrenzt.

4.3 Quantitative Untersuchung der Auswirkung von Protokollmechanismen in gekoppelten Netzen

In gekoppelten Netzen tauchen Protokollmechanismen sowohl mit einer abschnittsweisen als auch mit einer Ende–zu–Ende–Signifikanz auf. Sie dienen unter anderem dazu, Inkompatibilitäten in beiden Netzen auszugleichen. Die detaillierten Petri–Netzmodelle von Protokollmechanismen nach Abschnitt 4.2 müssen aufgrund ihrer Komplexität zu neuen Symbolen für Warteschlangenmodelle abstrahiert werden [32], um zu einem handhabbaren Gesamtmodell

zu kommen. An ihnen sind zahlreiche Modifikationen denkbar, welche bei bestimmten Konfigurationen sinnvoll sind. Um ihre quantitativen Auswirkungen auf Pufferspeicherbedarf in der Netzkoppeleinheit und Transferzeit zu untersuchen, ist aufgrund der großen Vielfalt im resultierenden Verkehrsmodell und wegen seiner Komplexität nur die Verkehrssimulation sinnvoll einsetzbar.

4.3.1 Zugrundeliegende Konfiguration

Der Aufbau des Verkehrsmodells entspricht Bild 3.4. Als Kopplungsschicht (N–1) kann die Vermittlungsschicht (Router) oder die Transportschicht (Gateway) gewählt werden. Das Verarbeitungssystem in den Stationen 1 und 2 wird nicht detailliert, sondern als Verkehrsquellen und –senken modelliert. Verkehrsquellen sollen einen Poisson–Ankunftsprozeß erzeugen. Es wird eine Unterteilung der Sicherungsschicht in zwei Teilschichten berücksichtigt, da in heutigen Produkten in der Regel an dieser Stelle die Trennung zwischen Hardware– und Softwarerealisierung liegt. Die Medienzugangsverfahren werden hier nicht explizit modelliert, weil sie einerseits nicht Gegenstand dieser Untersuchungen sein sollen und andererseits bei Routern und Gateways die Laufzeiten in höheren Schichten eindeutig dominant sind. Ein Übertragungskanal wird, einschließlich der Bitübertragungsinstanzen auf beiden Seiten, als konstante Verzögerung modelliert.

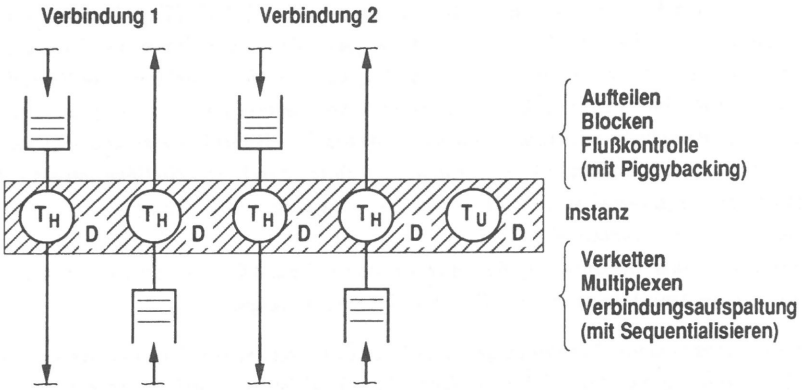


Bild 4.9: Ausschnitt aus dem Warteschlangenmodell

Es wird von verbindungsorientierten Protokollen ausgegangen, da die meisten Protokollmechanismen bei verbindungslosen Protokollen sowieso nicht eingesetzt werden können. In Bild 4.9 ist ein Ausschnitt aus dem Warteschlangenmodell für zwei initialisierte Verbindungen dargestellt. Jede Schicht enthält pro Verbindung und Richtung eine separate Bearbeitungsphase mit eigener Warteschlange. Die Zuordnung von Bearbeitungsphasen zu realen

Prozessoren und ihre Priorisierung kann beliebig gewählt werden. Alle nach dem Basisreferenzmodell auf den verschiedenen Schichten vorgesehenen Protokollmechanismen (außer denen zur Fehlerbehandlung) sind vorhanden, einschließlich einiger Modifikationen, welche in den nächsten Abschnitten beschrieben werden. Ihre Lage bezüglich der dargestellten Instanz ist Bild 4.9 gekennzeichnet. Die entsprechenden Partnermechanismen befinden sich in der jeweiligen Partnerinstanz an derselben Stelle. Da keine Fehlerbehandlung implementiert ist, müssen die Anzahlen von Warteplätzen und damit auch der Pufferspeicher der Netzkoppeleinheit so dimensioniert werden, daß bei den eingestellten Parametern keine Verluste auftreten. Sie müssen also mindestens so groß sein, wie die während eines Simulationslaufs auftretenden Maximalwerte.

Bei der Verkehrssimulation wird der eingeschwungene Zustand betrachtet, in welchem alle benötigten Verbindungen bereits aufgebaut sind, da die Dauer der Datentransferphase einer Verbindung sehr groß ist im Vergleich zu den charakteristischen Zeiten einzelner Pakete. Die Pakete laufen als Anforderungen durch das Verkehrsmodell, wobei angenommen wird, daß die Bedienzeiten unabhängig von den Paketgrößen sind. Der Pufferspeicher der Netzkoppeleinheit sei in Segmente mit der minimalen Paketgröße (Größe einer Quittung) unterteilt. Alle ganzzahlige Vielfache davon sind als Paketgrößen erlaubt, so daß hier kein Speicherverschnitt auftritt.

Schwerpunkt der Untersuchungen dieses Abschnitts ist die Netzkopplung auf der Vermittlungsschicht über Router. Viele qualitative Aussagen gelten jedoch ganz allgemein und insbesondere auch für die Kopplung auf der Transportschicht über Gateways. Deshalb wird hier nicht ein Router wie in Bild 3.9 betrachtet, sondern einer, welcher eine Transformation von Dienstprimitiven zwischen verschiedenen, vollständigen Vermittlungsinstanzen vornimmt. Die Protokollmechanismen auf den unteren drei Schichten arbeiten hier also alle abschnittsweise, sofern nicht auf der Vermittlungsschicht durch eine Modifikation eine Ende–zu–Ende–Signifikanz erreicht wird. In diesem Fall liegt aufgrund des Eingriffs in die Vermittlungsprotokolle des Routers eine Transformation der PDUs vor. Das Transportprotokoll ist entsprechend seiner Bestimmung ein Ende–zu–Ende–Protokoll. Für die Adressierung wird von einer Adreßtransformation mit Hilfe von Tabellen ausgegangen.

Bearbeitungsphasen und Verzögerungen auf einem Übertragungskanal werden alle so gewählt, daß ihre Dauer T_H konstant (D) 3 ms lang ist, falls nicht bei einzelnen Ergebnissen etwas anderes gesagt wird. Diese Bearbeitungsphasen enthalten auch die Zeiten, welche zur Bearbeitung der jeweiligen Protokollmechanismen benötigt werden, nicht aber die von ihnen verursachten Warte– und Verzögerungszeiten. Außerdem wird zwischen zwei aufeinanderfolgenden Bearbeitungsphasen eines Prozessors eine Umschaltphase durchlaufen, welche beispielsweise die Zeit repräsentiert, die das Betriebssystem braucht, um von einem Prozeß auf einen anderen umzuschalten, also insbesondere um Registerinhalte zu retten und mit neuem Inhalt zu laden. Sie wird mit einer konstanten (D) Dauer T_U von 1 ms angenommen.

In Anlehnung an reale Konfigurationen wird davon ausgegangen, daß in jeder Station die Schicht 2a in Hardware realisiert ist und deshalb als separater Prozessor modelliert werden kann. In den Stationen 1 und 2 seien die Protokolle der Schichten 2b bis 4 in Software implementiert, welche in jeder dieser Stationen auf *einem* Prozessor abläuft. Im Router sei, sofern nichts gegenteiliges angegeben wird, für jedes Netz ein separater Prozessor für die Schichten 2b und 3 vorhanden. Die Transformation wird von einem weiteren, dedizierten Prozessor durchgeführt. Zur Priorisierung der einzelnen Bearbeitungsphasen *eines* Prozessors werden die Implementierungsaspekte aus Abschnitt 3.5 berücksichtigt.

Bei den folgenden Untersuchungen werden vor allem drei Größen betrachtet:

- Die *Transferzeit* vom Generieren eines Paketes bis zu dessen Ankunft in der Verkehrssenke oberhalb der Transportschicht in Station 2,
- die *mittlere Anzahl der belegten Pufferspeichersegmente* im Router und
- die *maximale Anzahl der belegten Pufferspeichersegmente* im Router.

Wenn nichts anderes gesagt wird, so wird *eine* Simplex– oder Halbduplex–Verbindung in der Richtung von der Station 1 zur Station 2 betrachtet. Simulationspunkte werden gemeinsam mit ihren 95%–Vertrauensintervallen dargestellt, sofern diese nicht kleiner sind als die verwendeten Symbole.

4.3.2 Paketgrößenanpassung

In Abschnitt 3.2.2 wird darauf hingewiesen, daß das *Aufteilen einer SDU in mehrere PDUs, wenn es schon notwendig ist, normalerweise so spät wie möglich erfolgen sollte, also auf einer möglichst tiefen Schicht.* Bei gekoppelten Netzen kann es allerdings sinnvoll sein, diese Grundregel zu verletzen. Wenn in beiden Netzen ein Aufteilen notwendig ist, so kann das einmalige Aufteilen bereits auf einer Schicht mit Ende–zu–Ende–Signifikanz [185] günstiger sein. In diesem Fall werden zwar höhere Protokollinstanzen unnötigerweise durch eine größere Anzahl von Paketen belastet, dafür kann aber die zusätzliche Zeit für das Vereinigen im Router eingespart werden.

In Bild 4.10 werden diese beiden Möglichkeiten miteinander verglichen. Dabei sei die generierte Paketgröße fünfmal so groß wie die maximal erlaubte Paketgröße auf den Vermittlungsschichten. Neben den mittleren Transferzeiten enthält dieses Bild auch die mittleren Zeiten, welche für das Vereinigen notwendig sind, also die mittleren Wartezeiten einer PDU bis die restlichen PDUs eingetroffen sind, welche zur selben SDU vereinigt werden müssen. Da die Parameter in beiden Netzen gleich sind, sind diese Zeiten in den Vermittlungsschichten des Routers und der Station 2 identisch und aufgrund der geringeren Laufzeitschwankungen wegen dem kürzeren Weg etwas kleiner als in der Transportschicht. Beim Vergleich der mittleren

Transferzeiten zeigt sich, daß bei kleinen Ankunftsraten die eingesparte Zeit für das Vereinigen im Router ausschlaggebend dafür ist, daß das Aufteilen bereits auf der Transportschicht kleinere Werte zur Folge hat. Dagegen ist bei großen Ankunftsraten die Zusatzbelastung der Prozessoren in den Stationen 1 und 2 durch die größere Anzahl von Paketen der dominante Effekt, so daß hier ein Aufteilen auf den Vermittlungsschichten günstiger ist.

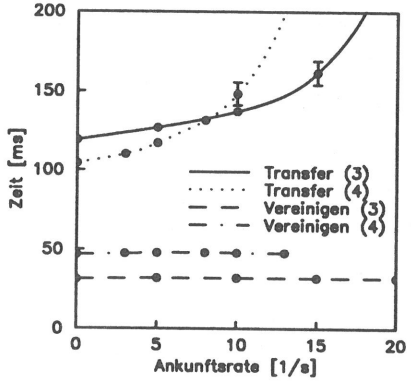


Bild 4.10: Aufteilen auf den Vermittlungsschichten (3) oder auf der Transportschicht (4)

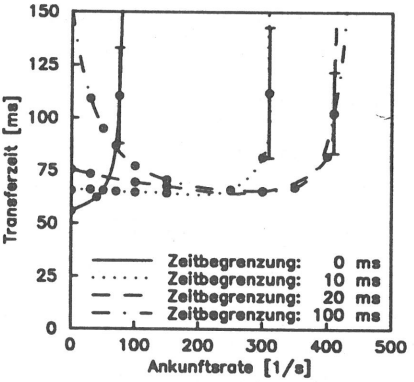


Bild 4.11: Optimierung der Zeitbegrenzung beim Blocken auf der Transportschicht

Jetzt soll der sinnvolle Einsatz der Protokollmechanismen Blocken und Verketten untersucht werden, deren detaillierte funktionelle Modellierung in Abschnitt 4.2 vorgestellt wurde.

Zunächst wird die Optimierung der Zeitbegrenzung für die künstliche Verzögerung beim Verketten beziehungsweise bei der zweiten Implementierungsvariante des Blockens betrachtet, unabhängig von der Problematik der Netzkopplung. Dazu wird auf der Transportschicht ein Blocken von bis zu fünf SDUs zu *einer* PDU erlaubt. Bild 4.11 zeigt, daß mit zunehmender Zeitbegrenzung die mittlere Transferzeit bei kleinen Ankunftsraten steigt. Dafür verschiebt sich die Stabilitätsgrenze wegen der kleineren Anzahl von Paketen in Richtung zu größeren möglichen Ankunftsraten, bis sie aufgrund der maximal erlaubten Blockgröße den fünffachen Anfangswert erreicht hat. *Als optimale Einstellung für die Zeitbegrenzung kann der Wert angesehen werden, welcher einerseits die maximale Ankunftsrate gerade noch zuläßt (meistens kommen während dieser Zeit fünf SDUs an), andererseits aber eine möglichst geringe zusätzliche Verzögerung bei kleinen Ankunftsraten verursacht*, hier also 20 *ms*. Falls auf der Transportschicht ein Verketten statt dem Blocken verwendet würde, ergäbe sich derselbe optimale Wert für die Zeitbegrenzung. Die Stabilitätsgrenze wäre allerdings aufgrund der größeren Anzahl von Paketen in der Transportschicht beim Verketten wesentlich niedriger als beim Blocken auf derselben Schicht. Die Anzahl von Paketen in den Engpaßprozessoren der Sta-

tionen 1 und 2 könnte nämlich dann nicht auf ein fünftel sinken. Beim Vergleich der beiden Implementierungsvarianten des Blockens (mit optimaler Zeitbegrenzung) auf der Transportschicht würde man für beide Varianten dieselbe Stabilitätsgrenze erhalten. Während bei der ersten Implementierungsvariante bei kleinen Ankunftsraten keine künstlichen Verzögerungen aufträten, wäre die zweite bei großen Ankunftsraten leicht überlegen.

Wenn auf den Vermittlungsschichten in beiden Netzen fünfmal so große Pakete wie auf der Transportschicht bearbeitet werden können, so ist es naheliegend, ein Verketten auf der Transportschicht der Station 1 oder ein Blocken auf den Vermittlungsschichten der Station 1 und des Routers einzusetzen. Beide Protokollmechanismen befinden sich an derselben Schichtgrenze. Sie unterscheiden sich nur dadurch, daß das Verketten eine Ende–zu–Ende–Signifikanz hat, während das Blocken abschnittsweise erfolgt. Bild 4.12 zeigt, daß das Verketten auf der Transportschicht dem zweimaligen Blocken nach der zweiten Implementierungsvariante überlegen ist, weil *eine* künstliche Verzögerung wegfällt. Als Zeitbegrenzungen werden jeweils die optimalen Werte von 20 *ms* gewählt. Auf die gestrichelte Kurve wird im nächsten Abschnitt gesondert eingegangen. Die erste Implementierungsvariante des Blockens (ohne künstliche Verzögerung) würde hier übrigens dieselben schlechten Ergebnisse wie die Referenzkurve ohne Protokollmechanismen liefern. Weil dabei keine künstliche Verzögerung verwendet wird, wäre nämlich ein Blocken nur möglich, wenn sich vor der Vermittlungsschicht Pakete stauen könnten, während der dazugehörende Prozessor andere Aufgaben bearbeitet. Genau dies wäre aber bei der gewählten Prozessoraufteilung und Priorisierung in der Station 1 nicht möglich, weil sich die Pakete in Engpaßsituationen bereits vor der Transportschicht stauen und anschließend, aufgrund ihrer gestiegenen Priorität, sofort weiterbearbeitet würden, ohne auf nachfolgende zu warten.

Bild 4.13 zeigt die mittlere Transferzeit beim Blocken von bis zu fünf SDUs zu *einer* PDU auf der Vermittlungsschicht nur im Netz 1. Dabei wird angenommen daß das Netz 2 schneller ist (alle Bedienzeiten und die Verzögerungszeit auf dem Übertragungskanal sind 1 *ms* statt 3 *ms*). Es wird wieder die einzige auf der Vermittlungsschicht sinnvolle zweite Implementierungsvariante des Blockens mit der Zeitbegrenzung als Parameter verwendet. Das Verhalten ist analog zu Bild 4.11. Wenn allerdings das Netz 2 nicht schneller wäre, so wäre es wegen dem Blocken im Netz 1 der alleinige Engpaß. Dann bliebe die Stabilitätsgrenze unverändert und die mittlere Transferzeit wäre trotzdem, aufgrund der künstlichen Verzögerung, erhöht. Deshalb gilt die generelle Grundregel, daß *Protokollmechanismen, welche eine künstliche Verzögerung benötigen, sich nur dann positiv auswirken können, wenn dadurch der Engpaß im System signifikant beeinflußt wird.* Dies wird sich auch noch in Abschnitt 4.3.4 in einem anderen Zusammenhang zeigen. Man kann also nicht pauschal sagen, daß sich solche Protokollmechanismen immer positiv auf die Leistungsfähigkeit in einem gekoppelten Netz auswirken, sondern die Auswirkung ist abhängig von den sonstigen Eigenschaften der Netze und in ungünstigen Konstellationen sind sogar Leistungseinbußen die Folge.

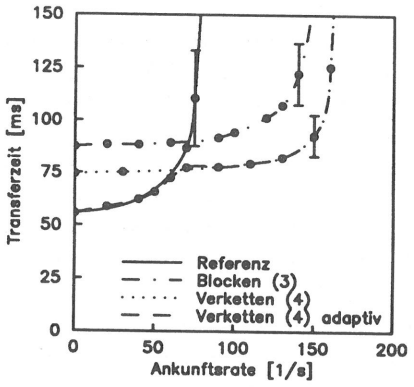


Bild 4.12: Verketten auf der Transportschicht (4) oder Blocken auf den Vermittlungsschichten (3)

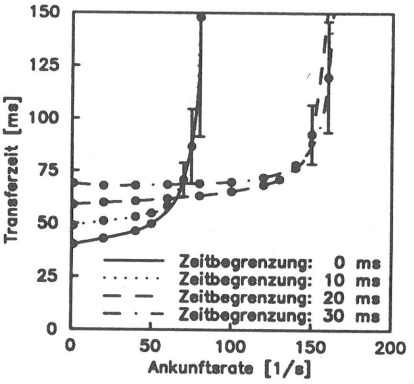


Bild 4.13: Blocken auf der Vermittlungsschicht nur in Netz 1

4.3.3 Modifikation von Verketten und Blocken

Aus Bild 4.12 ist bekannt, daß ein Verketten auf der Transportschicht sich bei großen Ankunftsraten sehr positiv auswirkt und insbesondere die Stabilitätsgrenze erst wesentlich später erreicht wird. Bei kleinen Ankunftsraten ist es dagegen sinnvoller, auf das mögliche Verketten zu verzichten, weil die beteiligten Prozessoren auch so nur wenig ausgelastet sind und sich nur die künstliche Verzögerung, und zwar negativ, auswirkt. Die Entscheidung, ob ein Verketten sinnvoll ist oder nicht, sollte also abhängig von der aktuellen Ankunftsrate getroffen werden.

Dies kann man beispielsweise mit Hilfe einer *adaptiven Zeitbegrenzung* beim Verketten realisieren. Bei einer aktuellen Ankunftsrate unterhalb des Kreuzungspunktes mit der durchgezogenen Referenzkurve wird die Zeitbegrenzung auf 0 *ms* gesetzt, so daß kein Verketten und damit keine künstliche Verzögerung möglich ist. Oberhalb dieses Kreuzungspunktes wird die Zeitbegrenzung auf den optimalen Wert von 20 *ms* erhöht. Die momentane Ankunftsrate wird erfaßt, indem zyklisch während eines Meßintervalls die ankommenden Pakete gezählt werden und die Summe durch die Meßintervalldauer geteilt wird. Als aktuelle Ankuftsrate wird immer der zuletzt berechnete Wert angesehen. Die Intervalldauer muß einerseits so groß sein, daß eine sinnvolle mittlere Ankunftsrate errechnet werden kann, andererseits aber klein genug, damit der als aktuell angesehene Wert nicht zu alt ist. Auf diese Art und Weise erhält man die in Bild 4.12 gestrichelt eingezeichnete Kurve mit einer fast optimalen mittleren Transferzeit für alle Ankunftsraten. Eine analoge Zeitbegrenzung kann man auch bei der zweiten Implementierungsvariante des Blockens einführen.

In realen, neueren Implementierungen wird im Rahmen der Leistungsfähigkeitsüberwachung

des Netzmanagements, vom Agent–Prozeß des Routers, die aktuelle Ankunftsrate in mehreren Schichten sowieso als Attribut von MOs zyklisch ermittelt. Wenn von der Software–Realisierung der obigen Protokollmechanismen aus auf ein solches Attribut lesend zugegriffen werden kann, so ist die beschriebene separate Berechnung der aktuellen Ankunftsrate nicht erforderlich.

4.3.4 Flußkontrollen bei gekoppelten Netzen

In diesem Abschnitt werden Flußkontrollen auf der Basis des üblichen Verfahrens der Folgenummern–Steuerung zugrundegelegt, welches unter anderem in [68] auch mit Hilfe approximativer mathematischer Analysemethoden untersucht wird. Andere Verfahren werden beispielsweise in [77] betrachtet. Flußkontrollen haben den Sinn, bei Engpässen einen Rückstau von Paketen nach Möglichkeit beim Verursacher (Sender) zu erzeugen, anstatt gemeinsame Betriebsmittel im Netz zu belegen. Sie helfen bei der fairen Zuteilung dieser Betriebsmittel auf verschiedene Verbindungen und erlauben bei Bedarf eine Anpassung unterschiedlicher Übertragungs– und Bearbeitungsgeschwindigkeiten. In gekoppelten Netzen haben sie unterschiedliche Auswirkungen, je nachdem, ob sie sich oberhalb der Kopplungsschicht befinden und deshalb eine Ende–zu–Ende–Signifikanz haben, oder nicht. Allen gemeinsam ist jedoch, daß die beteiligten Stationen durch Quittungen zusätzlich belastet werden, wodurch die Stabilitätsgrenze wesentlich früher erreicht und die Bedienung der Pakete verzögert wird. Eine weitere Verzögerung kann durch die eingestellte maximale Fenstergröße hervorgerufen werden [126]. Wenn der Sender der alleinige Engpaß im gekoppelten Netz wäre, so würde ein eventueller Rückstau sowieso dort entstehen und alle Arten von Flußkontrollen hätten in diesem Fall nur negative Auswirkungen.

Zur Ermittlung einer sinnvollen maximalen Fenstergröße, welche einerseits den Datenfluß und die Stabilitätsgrenze möglichst wenig verschlechtert und andererseits trotzdem zu einer effektiven Flußkontrolle führt, kann man folgendermaßen vorgehen: Zunächst ist bei einer sehr großen maximalen Fenstergröße die Stabilitätsgrenze des Gesamtsystems (Pakete pro Sekunde) zu bestimmen. Dann ist eine maximal tolerierbare Quittierungszeit für diese Flußkontrolle festzulegen. Nach dem Gesetz von Little ergibt sich schließlich die *einzustellende maximale Fenstergröße (maximale Anzahl unquittierter Pakete im System) aus dem Produkt von Stabilitätsgrenze und Quittierungszeit. Bei mehreren zulässigen Verbindungen und Richtungen ist dieser Wert entsprechend aufzuteilen.*

Bild 4.14 zeigt den Einfluß verschiedener Flußkontrollen auf die mittlere Transferzeit. Aufgrund des angenommenen Simplex– oder Halbduplexverkehrs kommen die Auswirkungen der Blindlast durch Quittungen extrem zur Geltung, da in der Regel kein Piggybacking von Quittungen auf Datenpakete der Gegenrichtung möglich ist.

Die geringste Verschlechterung der mittleren Transferzeit gegenüber der Referenzkurve ohne Flußkontrolle ergibt sich bei abschnittsweisen Fußkontrollen auf der Vermittlungsschicht, wobei jeweils von einer maximalen Fenstergröße von drei ausgegangen wird. Weil sich die Anzahl der Pakete auf den unteren drei Schichten durch die Quittungen verdoppelt, wird die Stabilitätsgrenze wesentlich früher erreicht. Die abschnittsweisen Flußkontrollen haben den Nachteil, daß sich bei einem Engpaß im Netz 2 die Pakete im Router anstatt beim Sender stauen, weil dieser Engpaß durch die zweite Flußkontrolle in den Router vorverlagert wird. Um dies zu vermeiden ist eine Flußkontrolle mit Ende–zu–Ende–Signifikanz notwendig.

Wenn von der Variante der Transformation von PDUs ausgegangen wird, so kann man dies bereits auf der Vermittlungsschicht erreichen, indem man die abschnittsweisen Flußkontrollen so koppelt, daß in Netz 1 erst dann eine Quittung zurückgeschickt wird, wenn vom Netz 2 eine Quittung angekommen ist. Bei der Verwendung der CCITT–Empfehlung X.25 (Packet Layer Protocol) als Kopplungsschicht kann beispielsweise durch das Delivery Confirmation Bit (D–Bit) vorgegeben werden, ob die Flußkontrollen auf der Kopplungsschicht unabhängig voneinander oder gekoppelt arbeiten sollen [116]. Aufgrund des doppelten Weges für die Flußkontrolle im Netz 1 werden die maximalen Fenstergrößen von drei auf sechs verdoppelt. Der weitere Anstieg der mittleren Transferzeit durch diese Kopplung der Flußkontrollen kommt daher, daß die Quittungen nun auch noch die Transformation durchlaufen müssen. Wenn der Router der Engpaß des Gesamtsystems wäre, so könnte man auf die Kopplung der Flußkontrollen verzichten, weil dann dieser Engpaß durch die erste Flußkontrolle in den Sender vorverlagert würde und das Senden im Router vor dem Empfangen priorisiert ist.

Wird dagegen, wie bisher, von einer Transformation der Dienstprimitive auf der Vermittlungsschicht ausgegangen, so muß die Ende–zu–Ende–Flußkontrolle auf der Transportschicht realisiert werden. Dabei wird die Anzahl der Pakete durch die Quittungen auch noch auf der Transportschicht verdoppelt, was die Stabilitätsgrenze noch mehr zu kleineren Ankunftsraten verschiebt. Die maximale Anzahl von Paketen im Router ist bei Verwendung einer Ende–zu–Ende–Flußkontrolle gleich der Summe der maximalen Fenstergrößen aller erlaubter Verbindungen in jeder Richtung. Alle weiteren Pakete stauen sich direkt bei den jeweiligen Sendern.

Eine Kombination der Flußkontrollen auf den Schichten 3 und 4 sollte nicht verwendet werden, da die mittlere Transferzeit gegenüber der alleinigen Ende–zu–Ende–Flußkontrolle auf der Transportschicht wegen den zusätzlichen Vermittlungsquittungen (in beiden Richtungen!) vergrößert wird, ohne daß dies einen weiteren Nutzen mit sich bringt. Da die Stabilitätsgrenze in der vorliegenden Konfiguration durch die Ende–zu–Ende–Flußkontrolle auf der Transportschicht festgelegt wird, ändert sich daran nichts. Bei Konfigurationen mit unterschiedlichen Stabilitätsgrenzen der letzten beiden Kurven würden diese sich einander annähern, wenn durch eine künstliche Verzögerung mehr Vermittlungsquittungen die Chance erhielten, durch Piggybacking einer Transportquittung mitgegeben zu werden.

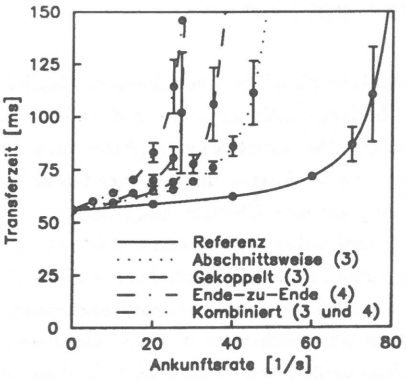


Bild 4.14: Vergleich verschiedener Flußkontrollen

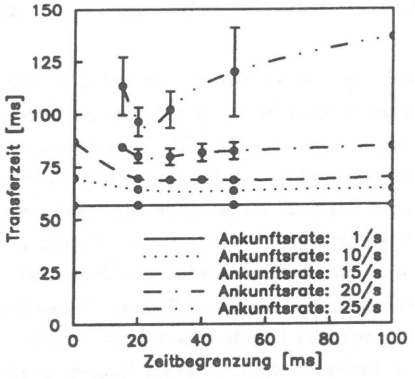


Bild 4.15: Optimierung der Zeitbegrenzung beim Piggybacking

Im realen Betrieb werden Verbindungen duplex betrieben, so daß durch Piggybacking von Quittungen auf Datenpakete der Gegenrichtung die Blindlast und ihre negativen Folgen reduziert werden kann. Es stellt sich die Frage, auf welchen Wert die maximale künstliche Verzögerung der Quittungen in der Empfangsinstanz einer Flußkontrolle sinnvollerweise begrenzt werden soll. Ist dieser zu klein, so wird die Blindlast kaum reduziert, ist er zu groß, so wird der Datenfluß zu stark gebremst weil die Quittungen mit einer zu großen Verzögerung zurückkommen und deshalb in der Sendeinstanz der Flußkontrolle ein Stau entsteht. In Bild 4.15 wird diese Frage anhand einer Ende–zu–Ende–Flußkontrolle auf der Transportschicht untersucht. Es wird hier *eine* symmetrisch belastete Duplexverbindung angenommen, wobei sich die angegebenen Ankunftsraten nur auf die untersuchte Richtung beziehen. Es zeigt sich, daß eine Begrenzung der künstlichen Verzögerung auf 20 *ms* für alle Ankunftsraten zu minimalen mittleren Transferzeiten führt.

Würde das Piggybacking von Quittungen auf Datenpakete der Gegenrichtung bei einer Flußkontrolle auf der LLC–Teilschicht eingesetzt werden, so ergäbe sich bei der vorliegenden Konfiguration durch diesen Protokollmechanismus eine Verschlechterung der mittleren Transferzeiten für alle Ankunftsraten. Die künstliche Verzögerung für das Piggybacking wäre in diesem Fall dominant gegenüber dem Gewinn aus der Reduzierung der Blindlast, zumal hier auch ohne Piggybacking die Vermittlungs– und Transportschichten keine Quittungen bearbeiten müßten. Die Erkenntnis aus Abschnitt 4.3.2 wird also hier bestätigt, daß Protokollmechanismen, welche eine künstliche Verzögerung benötigen, sich nur dann positiv auswirken können, wenn dadurch der Engpaß im System signifikant verbessert wird.

4.3.5 Überlastabwehr in einer Netzkoppeleinheit

In disesem Abschnitt werden mit Hilfe der instationären Simulation verschiedene Möglichkeiten untersucht, wie der Router auf kurzzeitige Überlastsituationen (auf der Basis von Paketen) reagieren kann. Die Vermeidung längerfristiger Überlastsituationen (Ablehnung des Aufbaus weiterer Verbindungen) wird hier nicht betrachtet. Ausgehend von einer Grundlast von 5 Paketen pro Sekunde wird zwei Sekunden lang auf eine Überlast umgeschaltet. Die Umschaltzeitpunkte sind in den Ergebnissen durch senkrechte Striche gekennzeichnet. Die simulierten Punkte werden aus Übersichtlichkeitsgründen nicht mit Symbolen markiert. Sie haben einen Abstand von 250 ms. Die mittlere Transferzeit wird hier über dem Generierungszeitpunkt der Pakete aufgetragen. Das Ziel bei diesen Untersuchungen ist die Minimisierung des Pufferspeicherbedarfs im Router, möglichst ohne dafür einen signifikanten Anstieg der mittleren Transferzeit in Kauf nehmen zu müssen. Beim Einsatz effizienter Überlastabwehrmechanismen kann eine größere Anzahl von Verbindungen zugelassen werden (Überbuchung) als ohne diese Mechanismen. Es wird angenommen, daß Datenpakete zehnmal so groß sind wie reine Quittungen oder Steuerpakete (siehe Abschnitt 4.3.5.1), also jeweils zehn Pufferspeichersegmente belegen. Diese Studie ist natürlich nur bei solchen Konfigurationen notwendig und sinnvoll, bei denen der Router überhaupt in einen Überlastzustand kommen kann.

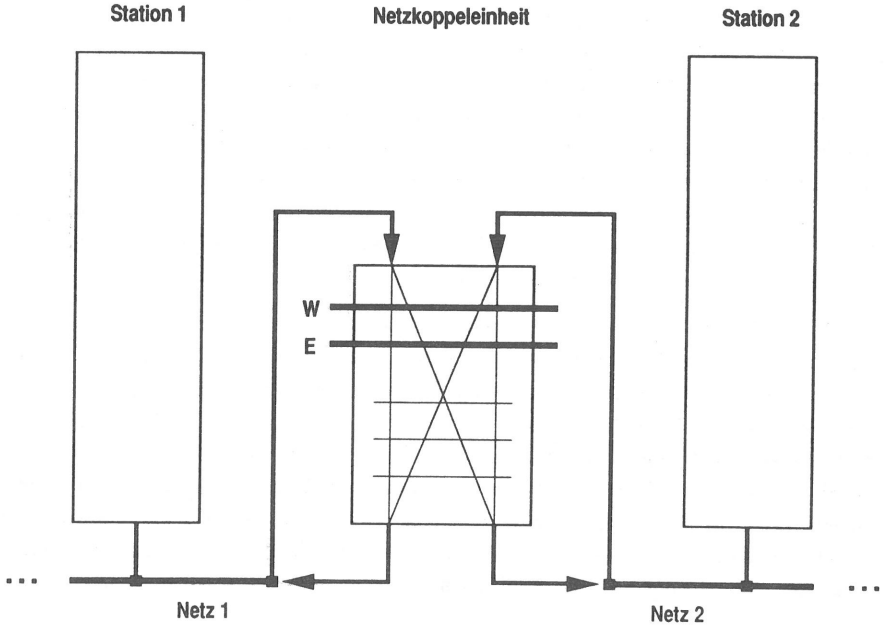


Bild 4.16: Schranken in einer Netzkoppeleinheit zur Überlastabwehr

Zur Überlastabwehr werden im Router zwei Schranken für die Anzahl belegter Pufferspeichersegmente nach Bild 4.16 eingeführt. Sobald die Warnschranke W überschritten wird, befindet sich der Router per Definition im *Überlastzustand*. Dieser bleibt solange bestehen, bis zum ersten Mal die Entwarnschranke E wieder unterschritten wird und der Router dadurch in seinen *Normalzustand* zurückkommt.

4.3.5.1 Verwendung von Steuerpaketen

Zunächst wird eine Konfiguration ohne Fenster–Mechanismen als Flußkontrollen betrachtet. Damit der Router zum Engpaß wird, sei hier der Prozessor für die Transformation zusätzlich auch noch für die Schichten 2b und 3 in beiden Netzen zuständig. Bei der Priorisierung der einzelnen Bearbeitungsphasen werden wieder die Implementierungsaspekte aus Abschnitt 3.5 berücksichtigt. Während des Überlastintervalls ist die Ankunftsrate 100 Pakete pro Sekunde.

Als Überlastabwehrstrategie wird der folgende Mechanismus verwendet: jeder Zustandswechsel des Routers wird den normalen Stationen durch spezielle unquittierte *Steuerpakete* mitgeteilt, welche (wie Quittungen) genau *ein* Pufferspeichersegment belegen. Hat ein Sender zuletzt ein *Warnpaket* vom Router erhalten, so werden dort alle weiteren Pakete solange zwischengespeichert, bis wieder ein *Entwarnpaket* vom Router kommt.

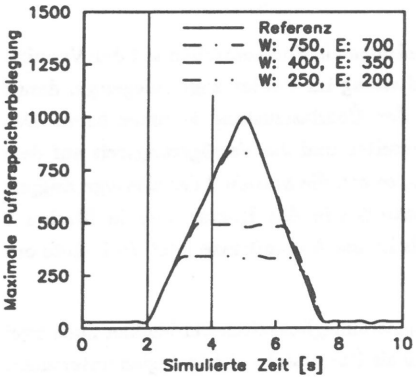


Bild 4.17: Überlastabwehr mit Hilfe von Steuerpaketen

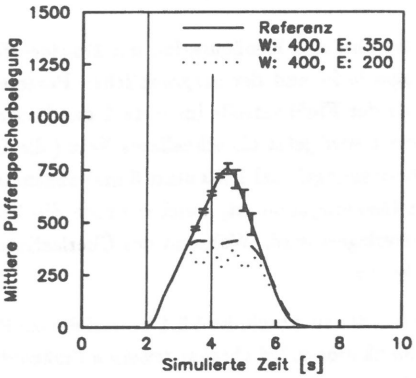


Bild 4.18: Hysterese bei der Überlastabwehr mit Hilfe von Steuerpaketen

In Bild 4.17 wird gezeigt, wie die maximale Anzahl belegter Pufferspeichersegmente, welche eine wichtige Kenngröße zur Dimensionierung des Pufferspeichers ist, durch eine sinkende Warnschranke drastisch reduziert werden kann. Die Entwarnschranke ist hier immer 50 Pufferspeichersegmente unter der Warnschranke. Die Dauer der erhöhten Pufferspeicherbelegung

wird durch den Überlastabwehrmechanismus nicht vergrößert. Der Stau von Paketen wird mit sinkender Warnschranke immer mehr zum Sender vorverlagert (Verursacherprinzip). Dabei wird die mittlere Transferzeit gegenüber den Vergleichswerten ohne Überlastabwehr kaum merklich erhöht. Diese Aussage würde allerdings nicht mehr gelten, wenn der Pufferspeicher wesentlich kleiner wäre und die Warnschranke deshalb einen deutlich niedrigeren Wert annehmen müßte.

Wenn die beiden Schranken relativ weit auseinander sind, so fängt die Pufferspeicherbelegung aufgrund der Hysterese unterhalb der Vergleichskurve an zu schwingen. Dies sieht man besonders schön an der mittleren Anzahl belegter Pufferspeichersegmente in Bild 4.18. In [5] wurde ein solcher Effekt bei einem einfacheren Verkehrsmodell bereits beobachtet. Das Maximum der Kurven kann allerdings durch eine kleinere Entwarnschranke nicht beeinflußt werden. Dieses wird einzig und allein von der Warnschranke festgelegt.

Die Überlastabwehr mit Hilfe von Steuerpaketen würde zwar auch bei mehreren Verbindungen zu derselben drastischen Reduktion des Pufferspeicherbedarfs führen, sie hätte dann aber den Nachteil, daß auch die an der Überlastsituation unschuldigen Verbindungen über eine längere Zeit hinweg deutlich erhöhte mittlere Transferzeiten in Kauf nehmen müßten, was ohne diesen Überlastabwehrmechanismus nicht der Fall wäre. Dieser ist also insofern unfair.

4.3.5.2 Modifikationen an Flußkontrollen

Jetzt wird eine Konfiguration mit Fenster–Mechanismen als Flußkontrollen auf der Vermittlungsschicht und der ursprünglichen Prozessoraufteilung im Router zugrundegelegt, damit trotz der Flußkontrolle im Netz 1 der Router in den Überlastzustand kommen kann. Das Netz 1 wird jetzt als schnelleres Netz (alle Bedienzeiten und die Verzögerungszeit auf dem Übertragungskanal 1 *ms* statt 3 *ms*) angenommen, so daß die Station 2 der alleinige Engpaß im Gesamtsystem ist, welcher durch die Flußkontrollen in den Router oder in Station 1 vorverlagert wird. Während des Überlastintervalls ist die Ankunftsrate jetzt 75 Pakete pro Sekunde.

Neben den in Abschnitt 4.3.4 beschriebenen Flußkontrollmöglichkeiten werden hier noch zwei Modifikationen bei abschnittsweisen Flußkontrollen als Überlastabwehrstrategien untersucht, welche, im Gegensatz zur Kopplung dieser Flußkontrollen, auch bei der Transformation von Dienstprimitiven angewendet werden können:

- *Adaptive maximale Fenstergröße*: Der Router schreibt in jede Quittung seinen aktuellen Zustand als Parameter. Der Sender wertet diesen Parameter aus und setzt die maximale Fenstergröße auf den Wert eins, falls es sich um den Überlastzustand handelt. Er wird dadurch gebremst. Andernfalls wird die maximale Fenstergröße um eins inkrementiert, falls der Defaultwert noch nicht erreicht ist. Ein ähnliches Verfahren wird in [41]

vorgeschlagen, um die Verluste in einer Bridge durch die Ende–zu–Ende–Flußkontrolle auf der LLC–Teilschicht (Klasse 2) zu reduzieren. Es ist mittlerweile auch als Option in die entsprechende Standardisierung eingeflossen [105]. Andere Anwendungen dieser Technik und Varianten dazu werden in [6, 159, 195] untersucht.

- *Quittungen zurückhalten:* Solange der Router sich im Überlastzustand befindet, hält er alle Quittungen zurück, so daß sich die Fenster der Flußkontrollen im Laufe der Zeit vollständig schließen können. Sobald die Schranke E unterschritten wird, wird das Anhalten der Sender durch Sammelquittungen schlagartig wieder aufgehoben. Diese Strategie hat im Gegensatz zu der obigen den Vorteil, daß nur der Router modifiziert werden muß und die normalen Stationen (Sender) unverändert bleiben können.

Bild 4.19 ist das Analogon zu Bild 4.17 für eine adaptive maximale Fenstergröße. Es zeigt, wie die maximale Anzahl belegter Pufferspeichersegmente durch eine sinkende Warnschranke extrem reduziert werden kann. Auch hier wird die Dauer der erhöhten Pufferspeicherbelegung durch die Überlastabwehrstrategie nicht vergrößert. Das Zurückhalten von Quittungen würde, bei der entsprechenden Wahl der beiden Schranken, praktisch dieselben Ergebnisse liefern, so daß auf ein weiteres Bild an dieser Stelle verzichtet werden kann.

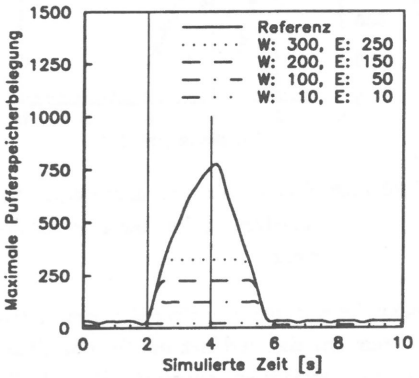


Bild 4.19: Überlastabwehr mit Hilfe einer adaptiven maximalen Fenstergröße

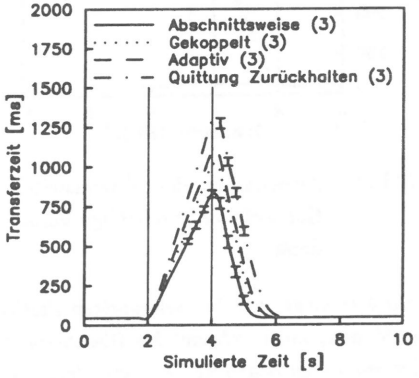


Bild 4.20: Begleiterscheinung der Überlastabwehr mit Hilfe modifizierter Flußkontrollen

In Bild 4.20 ist die negative Begleiterscheinung dieser Reduzierung des Pufferspeicherbedarfs zu sehen. Die mittleren Transferzeiten sind, vor allem bei dem hier gewählten kleinen Wert zehn für beide Schranken (*ein* Datenpaket im Router), gegenüber der Referenzkurve ohne Modifikationen erhöht, insbesondere beim Zurückhalten von Quittungen. Zum Vergleich enthält dieses Bild auch noch die mittlere Transferzeit für eine Kopplung der abschnittsweisen Flußkontrollen. Diese Kurve liegt zwischen der Referenzkurve und den beiden anderen. Der

maximale Pufferspeicherbedarf wird bei der gekoppelten Flußkontrolle allerdings nur auf 40 Pufferspeichersegmente anstelle von 30 beim Zurückhalten der Quittungen und 20 bei der adaptiven maximalen Fenstergröße reduziert.

Jetzt soll noch untersucht werden, wie sich die verschiedenen Überlastabwehrstrategien auf eine erste Verbindung mit der konstanten Ankunftsrate von 20 Paketen pro Sekunde auswirken, wenn die Verkehrsquelle einer zweiten Verbindung, mit derselben Priorität aber separaten Warteschlangen in allen beteiligten Stationen, ausgehend von einer Grundlast von 5 Paketen pro Sekunde, im Überlastintervall eine Ankunftsrate von 55 Paketen pro Sekunde erzeugt.

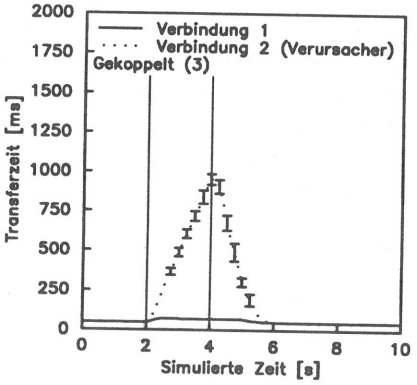


Bild 4.21: Auswirkung der Überlastsituation auf eine unschuldige Verbindung

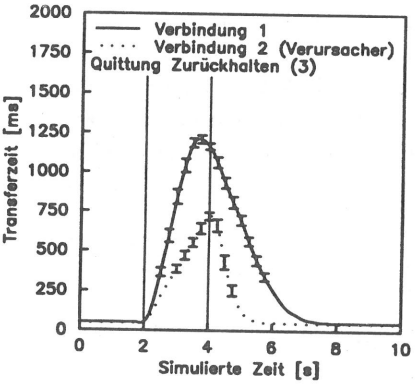


Bild 4.22: Beispiel für eine ungünstige Parameterwahl bei der Überlastabwehr

Bild 4.21 zeigt, daß bei gekoppelten Flußkontrollen die mittlere Transferzeit dieser ersten Verbindung auch während des Überlastintervalls kaum ansteigt und sich die Überlastsituation im wesentlichen nur auf den Verursacher (zweite Verbindung) auswirkt. Da sich die effektive Fenstergröße bei zwei Verbindungen gegenüber den bisherigen Untersuchungen verdoppelt hat, steigt die maximale Anzahl belegter Pufferspeichersegmente von 40 auf 60 an. Fast dieselben Kurven würden sich für die mittleren Transferzeiten auch ergeben, wenn die Flußkontrollen nicht gekoppelt wären und zwar unabhängig davon, ob die maximale Fenstergröße im Netz 1 adaptiv wäre (beide Schranken bei 10 Pufferspeichersegmenten) oder nicht. Bei der adaptiven maximalen Fenstergröße wäre die mögliche Reduzierung des Pufferspeicherbedarfs allerdings bei weitem nicht mehr so stark wie bei *einer* Verbindung nach Bild 4.19, weil die effektive maximale Fenstergröße im Netz 1 nicht mehr unter den Wert zwei gedrückt werden könnte. Beim Zurückhalten der Quittungen (gleiche Schranken) würde sich dagegen der Pufferspeicherbedarf, gegenüber dem entsprechenden Ergebnis für *eine* Verbin-

dung, auf 60 Pufferspeichersegmente lediglich verdoppeln, da die Sender weiterhin vollständig angehalten werden könnten. An der mittleren Transferzeit der ersten Verbindung würde sich auch hier nichts ändern, während die zweite Verbindung (Verursacher) einen um bis zu 50% erhöhten Wert gegenüber Bild 4.21 in Kauf nehmen müßte.

Das in allen diesen Fällen gültige Verursacherprinzip würde allerdings nicht mehr uneingeschränkt gelten, wenn die zweite Verbindung eine wesentlich größere maximale Fenstergröße im Netz 2 zur Verfügung hätte als die erste, da sie dann während des Überlastintervalls den Engpaßprozessor in der Station 2 zu einem größeren Teil auslasten könnte, so daß dadurch auch die mittlere Transferzeit der ersten Verbindung ansteigen würde.

In Bild 4.22 ist zu sehen, wie durch eine ungünstige Parameterwahl die Überlastabwehr zu einem unfairen Verhalten führen kann. Es wird hier angenommen, daß die zweite Verbindung im Netz 1 eine maximale Fenstergröße von zwölf hat und die Überlastabwehr durch das Zurückhalten von Quittungen (beide Schranken bei 10 Pufferspeichersegmenten) realisiert werden soll. Ohne den Überlastabwehrmechanismus würden die mittleren Transferzeiten praktisch dem Ergebnis in Bild 4.21 entsprechen. Durch das Zurückhalten von Quittungen treten vor allem auf der unschuldigen ersten Verbindung während des Überlastintervalls wesentlich erhöhte mittlere Transferzeiten auf, und zwar sogar noch höhere als beim Verursacher der Überlast selbst. Das liegt daran, daß auf der ersten Verbindung nur drei Pakete unterwegs sein können wenn der Router sich im Überlastzustand befindet, während es auf der zweiten Verbindung zwölf sind. Sobald der Router leer ist, können wieder nur bis zu drei beziehungsweise zwölf Pakete gesendet werden, weil der Router beim ersten ankommenden Paket sofort wieder in den Überlastzustand übergeht. Da die zweite Verbindung viermal so viele Pakete senden kann wie die erste, aber selbst im Überlastintervall nicht einmal die dreifache Anzahl von Paketen angeboten bekommt, wird die unschuldige erste Verbindung dadurch zum Engpaß. Sobald sich auch in der zweiten Verbindung beim Sender ein Stau von zwölf Paketen aufgebaut hat, wird die Zeit von der nächsten Sammelquittung bis zum erneuten Anhalten dieser Verbindung durch den Überlastabwehrmechanismus auf ein Minimum reduziert. Je schneller aber der Überlastzustand überwunden wird, umso günstiger ist das für die erste Verbindung. Deshalb nimmt ihre mittlere Transferzeit bereits vor dem Ende des Überlastintervalls wieder ab.

Wie bei der Überlastabwehr durch Verwendung von Steuerpaketen nach Abschnitt 4.3.5.1, können die Sender beim Zurückhalten von Quittungen vollständig angehalten werden. Bei beiden Mechanismen muß man deshalb damit rechnen, daß bei bestimmten Konstellationen auch unschuldige Verbindungen erhöhte mittlere Transferzeiten erfahren, wenn eine Überlastsituation auftritt. Beim Zurückhalten von Quittungen werden besonders Pakete auf Verbindungen mit kleiner maximaler Fenstergröße gebremst, weil das Fenster nach einem Überlastzustand jeweils ganz aufgehen kann. Die Reaktion auf eine Überlastsituation ist bei Verbindungen mit großem maximalem Fenster träge. Im Gegensatz dazu sind bei der

adaptiven maximalen Fenstergröße Verbindungen mit einem großen Wert stärker betroffen, insbesondere wenn er voll ausgeschöpft wird, da die maximale Fenstergröße dann wesentlich stärker reduziert werden kann. Das Verursacherprinzip ist hier also erfüllt.

Als Fazit läßt sich festhalten, daß die *Kopplung von Flußkontrollen bezüglich der mittleren Transferzeiten, der Fairneß und der Reduktion des Pufferspeicherbedarfs den günstigsten Kompromiß darstellt* und deshalb nach Möglichkeit auch realisiert werden sollte, sofern eine Transformation von PDUs überhaupt in Frage kommt. *Wenn nur eine Transformation von Dienstprimitiven möglich ist, so sollte das Zurückhalten von Quittungen wegen geringerem Implementierungsaufwand und Pufferspeicherbedarf bei mehreren Verbindungen der adaptiven maximalen Fenstergröße vorgezogen werden.* Dieser Mechanismus reagiert allerdings empfindlich auf die Einstellung der Parameter, so daß bei einer ungünstigen Wahl die Fairneß darunter leiden kann.

4.4 Leistungsuntersuchungen an Bridges

In diesem Abschnitt wird die Leistungsfähigkeit von Bridges untersucht, wobei insbesondere auch die Sonderform der Remote Bridges dabei berücksichtigt werden soll. Es wird ein iterativer Algorithmus entwickelt, welcher die dynamische Aufteilung des begrenzten Pufferspeichers zwischen den verschiedenen Warteschlangen einer Bridge explizit berücksichtigt. Analytische Ergebnisse werden mit Hilfe eines universellen Simulationsprogramms für Bridges validiert.

4.4.1 Warteschlangenmodell

Es wird der eingeschwungene Zustand betrachtet, in welchem vor allem bei Remote Bridges alle benötigten Verbindungen im Transitnetz bereits aufgebaut sind. Die Topologie ist für die Dauer einer Untersuchung statisch und die BPDUs (zur Überprüfung der Topologie bei Spanning Tree Bridges) sind gegenüber den normalen Paketen vernachlässigbar. Bei Source Routing Bridges wird davon ausgegangen, daß die Wege zu den Empfängern seit dem Verbindungsaufbau höherer Schichten (während der Einschwingphase) bekannt sind und nicht erst durch Rundsendepakete ermittelt werden müssen. Der Unterschied zwischen Spanning Tree und Source Routing Bridges besteht dann nur noch in der Art und Weise wie der Filtermechanismus realisiert ist (mit Hilfe einer Tabelle bei Spanning Tree Bridges oder durch Untersuchen des Informationsfeldes für die Verkehrslenkung bei Source Routing Bridges), was sich aber auf die Struktur des Verkehrsmodells nicht auswirkt. Dadurch gelten alle weiteren Aussagen für beide standardisierte Arten von Bridges, so daß sie bei der Modellierung und Leistungsuntersuchung gemeinsam behandelt werden können.

Es wird die bei mathematischen Analysen meist übliche vereinfachende Annahme getroffen, daß alle Pakete gleich groß sind und jeder Pufferspeicherplatz genau ein Paket aufnehmen kann. Dadurch tritt kein Speicherverschnitt auf und die maximale Anzahl von Paketen, welche im Pufferspeicher der Bridge zwischengespeichert werden kann, ist konstant. Ferner wird explizit berücksichtigt, daß der Pufferspeicher eine begrenzte Größe hat, wobei sich seine Aufteilung zwischen verschiedenen Warteschlangen dynamisch verändern kann. Der Pufferspeicher sollte so dimensioniert werden, daß alle Pakete in der Bridge verlorengehen, deren Zeitüberwachung in einer höheren Schicht des Senders (aufgrund einer momentan erhöhten Quittierungszeit) sowieso eine Wiederholung veranlassen wird. Dadurch werden die Transferzeiten kleiner und auch der Preis und Pufferspeicherbedarf der Bridge wird reduziert. Paketverluste in einer Bridge wirken sich genauso aus, wie die aufgrund von Übertragungsfehlern bei einem Empfänger verworfenen Pakete, so daß keine zusätzlichen Fehlerbehebungsmaßnahmen notwendig sind. Aufgrund der verbindungslosen MAC–Protokolle hat die Bridge auch keine Möglichkeit, bei Pufferspeicherengpässen einen Rückstau beim Sender zu verursachen.

Während bei der Kopplung von LANs auf den höheren Schichten die Medienzugangsverfahren auch bezüglich der Leistungsfähigkeit eine untergeordnete Rolle spielen, müssen sie zur Untersuchung von Bridges explizit berücksichtigt werden, da ihre charakteristischen Zeiten in derselben Größenordnung liegen wie die Bearbeitungszeiten in einer Bridge.

In Bild 4.23 ist das Warteschlangenmodell zweier, über eine Bridge gekoppelter, Netze dargestellt. Stellvertretend für andere Konfigurationen wird die Kopplung eines Token Ring LANs mit dem durchgeschalteten Kanal eines WANs (beispielsweise einem B–Kanal des ISDN) gezeigt, so daß es sich, präziser ausgedrückt, um eine Remote Bridge handelt. Das Verkehrsmodell läßt sich in folgende drei Teilmodelle zerlegen:

- *Bearbeitungseinheit* für das Filtern und die Weiterbearbeitung in der Bridge,
- *Netzeinheit 1* für den Medienzugang in Netz 1 und
- *Netzeinheit 2* für den Medienzugang in Netz 2.

Zur Untersuchung von Simplex– oder Halbduplexverkehr ist die Netzeinheit 2 nicht notwendig, da immer nur eine Richtung aktiv sein kann und die Richtungen deshalb separat untersucht werden dürfen. Diese Netzeinheit wird dann durch einen Generator modelliert, welcher den gesamten Verkehr aller Stationen an Netz 2 (ohne die Bridge) erzeugt. Er setzt sich zusammen aus Paketen und ihren notwendigen Wiederholungen aufgrund abgelaufener Zeitüberwachungen in höheren Schichten, so daß auch die Reaktion der jeweiligen Sender auf Verluste in der Bridge berücksichtigt ist. Da es sich hier um die Überlagerung aus sehr vielen unabhängigen Verkehrsquellen handelt, ist ein Poisson–Ankunftsprozeß an der Bridge mit negativ exponentiell (M) verteilten Ankunftsabständen T_{A1} und Ankunftsrate λ_{A1} für die Richtung zum Netz 1 eine gute Näherung.

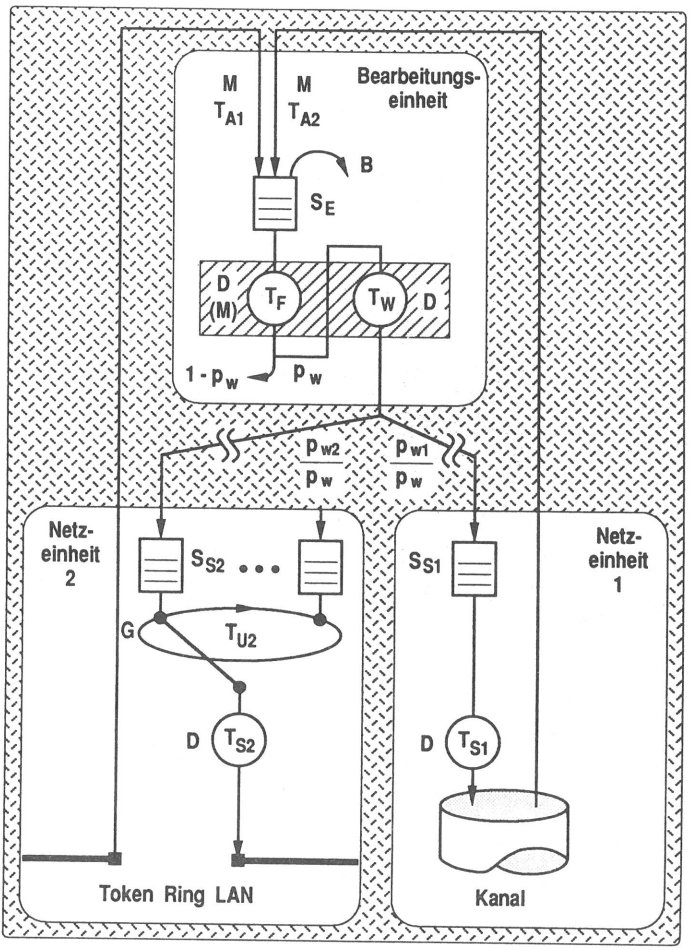


Bild 4.23: Allgemeines Warteschlangenmodell

Bei der Untersuchung von Duplexverkehr müssen auch die entsprechenden Größen T_{A2} und λ_{A2} für die andere Richtung eingeführt werden. Für die mathematische Analyse muß man in beiden Richtungen wieder negativ exponentiell verteilte (M) Ankunftsabstände annehmen. Dann ist auch der Summenankunftsprozeß ein Poisson–Prozeß mit der Ankunftsrate $\lambda_A = \lambda_{A1} + \lambda_{A2}$ [129]. Es können beliebige Netzeinheiten miteinander kombiniert werden.

Im folgenden wird neben den in Bild 4.23 dargestellten Netzeinheiten noch die für ein Token–Passing Bus LAN betrachtet, es wären aber auch Verkehrsmodelle für andere Medienzugangs-

verfahren als Netzeinheiten möglich, wenn deren Analyseverfahren die folgenden Modelleigenschaften erlauben:

- begrenzte Warteschlangen wegen des begrenzten Pufferspeichers in der Bridge und
- unsymmetrische Ankunftsraten an den Stationen, da sich die Ankunftsrate in der Bridge aus dem Verkehr im anderen Netz ergibt und normalerweise nicht derjenigen in einer normalen Station entspricht.

Die *Empfangswarteschlange* der Bridge hat S_E Warteplätze. Bei vollem Pufferspeicher der Bridge gehen ankommende Pakete mit der (wegen obiger M–Annahme) für beide Richtungen gleichen Verlustwahrscheinlichkeit B verloren. Da ein momentan bearbeitetes Paket auch einen Pufferspeicherplatz belegt, welcher aber nicht zur Empfangswarteschlange gehört, hat die *Bearbeitungseinheit* $N_E = S_E + 1$ Pufferspeicherplätze zur Verfügung.

Da Bridges alle Pakete der angeschlossenen Netze empfangen solange ihr Pufferspeicher noch nicht voll ist, muß zunächst für jedes Paket mit Hilfe eines Filtermechanismusses entschieden werden, ob es zum Internverkehr eines Netzes oder zum Externverkehr gehört. Spanning Tree Bridges müssen zusätzlich ihre Filtertabelle aktualisieren. Die Zeit, welche dafür notwendig ist, wird durch die Bedienzeit T_F der *Filterphase* modelliert. Abhängig vom Filtermechanismus kann für sie eine deterministische (D) oder eine negativ exponentielle (M) Verteilungsfunktion gewählt werden. Bei Verwendung einer genügend großen Hash–Tabelle für Spanning Tree Bridges reicht fast immer *ein* Suchschritt pro Adresse aus [135], so daß die erste der beiden Möglichkeiten eine gute Näherung darstellt. Je nach Ergebnis des Filterns wird das Paket aus dem Pufferspeicher entfernt (Internverkehr eines Netzes) oder weiterbearbeitet (Externverkehr). Im Verkehrsmodell ist dieses Ergebnis in Form der *Weiterbearbeitwahrscheinlichkeit* p_w berücksichtigt. Sie ist abhängig von den Ankunftsraten und dem *Anteil des Externverkehrs* p_{ei} für jede Richtung (die für beide Richtungen gleiche Verlustwahrscheinlichkeit B kürzt sich heraus):

$$p_w = \frac{\lambda_{A1} p_{e1} + \lambda_{A2} p_{e2}}{\lambda_A} \,. \tag{4.31}$$

Für die beiden Summanden werden die *richtungsabhängigen Weiterbearbeitwahrscheinlichkeiten* $p_{wi} = \lambda_{Ai}\, p_{ei}/\lambda_A$ eingeführt, welche später noch als Rechengrößen benötigt werden.

Die Weiterbearbeitung (beispielsweise Transformation) eines Paketes bei Externverkehr wird durch die *Weiterbearbeitphase* im selben Prozessor modelliert, welche immer unmittelbar auf die Filterphase folgt, so daß davor keine weitere Warteschlange notwendig ist. Ihre Dauer T_W ist für alle Pakete gleich und besitzt deshalb eine deterministische Verteilungsfunktion (D).

Nach der Weiterbearbeitung verläßt das Paket die Bearbeitungseinheit und erreicht, je nach Richtung, eine der beiden Netzeinheiten. Physikalisch bleibt es dabei im Pufferspeicher der Bridge an derselben Stelle, und es wir nur seine Adresse weitergegeben. Deshalb kann bei dieser Weitergabe kein Verlust mehr auftreten. Dort wartet das Paket in der jeweiligen *Sendewarteschlange* mit S_{S1} beziehungsweise S_{S2} Warteplätzen, bis die vorausgehenden Pakete vollständig übertragen sind und es den Übertragungskanal zugeteilt bekommt.

Wegen der Annahme gleichgroßer Pakete, ist die Dauer T_{S1} beziehungsweise T_{S2} der jeweiligen *Sendephase* hier deterministisch (D). Beim durchgeschalteten Kanal eines WANs wird der belegte Pufferspeicherplatz erst nach dem vollständigen Senden eines Paketes freigegeben, so daß die Sendephase auch einen Pufferspeicherplatz repräsentiert und die Netzeineit 1 somit $N_{S1} = S_{S1} + 1$ Pufferspeicherplätze belegt. Bei Token Ring oder Token–Passing Bus LANs wird angenommen, daß ein Paket seinen Pufferspeicherplatz sofort freigibt und zur Parallel/Seriell–Wandlung in ein Register kopiert wird, sobald es den Übertragungskanal zugeteilt bekommt. Es gilt also $N_{S2} = S_{S2}$. Die Umschaltzeit T_{U2} repräsentiert bei diesen LANs die Zeit zur Weitergabe der Sendeberechtigung zwischen benachbarten Stationen und darf beliebig (G) verteilt sein.

Aufgrund des begrenzten *Pufferspeichers der Bridge* mit N_B Speicherplätzen gilt

$$N_B = N_E + N_{S1} + N_{S2} \,. \tag{4.32}$$

Die im Rahmen dieser Arbeit betrachteten Netzeinheiten werden folgendermaßen modelliert:

- *Token Ring LAN*: Pollingsystem mit begrenzten Warteschlangen und Bedienung von maximal einem Paket pro Warteschlange und Zyklus (Limited–1).
- *Token–Passing Bus LAN*: Pollingsystem mit begrenzten Warteschlangen und Bedienung von maximal n Paketen pro Warteschlange und Zyklus (Limited–n).
- *Durchgeschalteter Kanal eines WANs als Punkt–zu–Punkt–Verbindung (Beispiel: B–Kanal des ISDN)*: M/D/1–S_{Si}–System.

Bei den ersten beiden Netzeinheiten *muß* für die Analyse ein Poisson–Prozeß als Ankunftsprozeß angenommen werden. Die Genauigkeit dieser Näherung könnte durch die Berechnung der Variationskoeffizienten der Ausgangsprozesse der Bearbeitungseinheit für jede Richtung überprüft werden. Bei einem sehr großen Anteil des Internverkehrs, was auch den realistischen Einsatzfall darstellt, erweist sich diese Näherung als gut. Die dritte Netzeinheit könnte auch als zeitdiskretes $G^{[X]}$/D/1–S_{Si}–System modelliert werden, wobei als Ankunftsprozeß der Ausgangsprozeß der Bearbeitungseinheit für diese Richtung eingesetzt wird [140]. X wäre dabei eine diskrete Zufallsvariable, welche durch die Anzahl der Einheiten, aus denen eine Anforderung besteht, deren Größe beschreibt. Diese Analyse erweist sich allerdings als

unnötig aufwendig und wird im Rahmen dieser Arbeit nicht weiter verfolgt, da für realistische Parameter (bei der Richtung zum durchgeschalteten Kanal eines WANs ist der Anteil des Internverkehrs sehr groß) die Ergebnisse der Näherung des M/D/1–S_{Si}–Systems bereits recht gut sind.

In der Literatur sind zahlreiche Untersuchungen von Bridges mit unterschiedlichen Schwerpunkten zu finden, welche aber alle von, gegenüber Bild 4.23, stark vereinfachten Verkehrsmodellen mit separatem Pufferspeicher und Prozessor für jede Richtung ausgehen, oder überhaupt nur eine Richtung betrachten. Für jede Richtung reduziert sich dabei das Verkehrsmodell einer Bridge im Extremfall auf ein einfaches G/M/1–System [95]. Bei anderen Verkehrsmodellen werden teilweise zumindest begrenzte Warteschlangen in der Bridge berücksichtigt. Zur Leistungsuntersuchung wird oft ausschließlich die Verkehrssimulation eingesetzt. Während in [40, 41, 87, 92, 134, 150, 151, 153, 158, 199] Netze mit Ringtopologie und kreisenden Sendeberechtigungen untersucht werden, liegen in [154, 161, 162] CSMA/CD LANs, oder leichte Abwandlungen davon, zugrunde. Außer in [154] wird die Belastung der Bridge durch den Internverkehr der angeschlossenen Netze (in der Praxis über 90%!) in allen Verkehrsmodellen vernachlässigt.

4.4.2 Exakte Lösung für einen Spezialfall

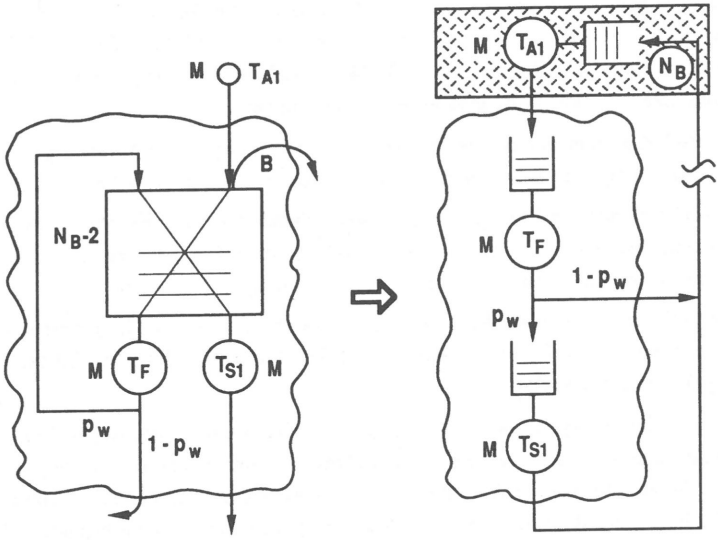


Bild 4.24: Warteschlangenmodell eines Spezialfalls und eine äquivalente Darstellung dazu

Während für den allgemeinen Fall ein iterativer Algorithmus mit einigen Näherungsannahmen notwendig ist, existiert für den Spezialfall des Warteschlangenmodells in Bild 4.24 (links) eine geschlossene, exakte Lösung.

Dieses Warteschlangenmodell ist in einer anschaulicheren Form dargestellt, die den *einen* Pufferspeicher der Bridge mit N_B Pufferspeicherplätzen (Warteschlange und beide Bedieneinheiten) hervorhebt, welcher in Bild 4.23 zwischen verschiedenen Warteschlangen aufgeteilt wird. Es wird Simplex– oder Halbduplexverkehr betrachtet. Die Netzeinheit 1 entspricht dem durchgeschalteten Kanal eines WANs, dessen Sendedauer T_{S1} hier für die Analyse negativ exponentiell (M) verteilt sein soll (einzige Abweichung von den realistischen Annahmen in Bild 4.23). Ihre Bedienrate ist dann μ_{S1}. Die Weiterbearbeitphase sei gegenüber der Filterphase mit der negativ exponentiell (M) verteilten Dauer T_F und der Filterrate μ_F vernachlässigbar (Kopplung gleicher Netze). Zur übersichtlicheren Schreibweise wird im folgenden auf den Index 1, welcher die Richtung zur Netzeinheit 1 anzeigt, verzichtet, da die Netzeinheit 2 hier sowieso nicht benötigt wird. Beispielsweise werden anstelle von λ_{A1} und μ_{S1} also λ_A und μ_S geschrieben.

Da alle Ankunftsabstände und Bediendauern negativ exponentiell verteilt sind, läßt sich der Zustandsprozeß des Pufferspeichers der Bridge mit Hilfe eines zweidimensionalen Markoff–Prozesses beschreiben (in jedem Zustand kann der Prozeß für den Übergang in einen benachbarten Zustand durch einen Markoff–Prozeß beschrieben werden). Die erste Dimension repräsentiert die Anzahl der Pakete in und vor der *Empfangs–Bedieneinheit*, deren Bedienzeit durch die Filterzeit T_F beschrieben wird, und die zweite die Anzahl der Pakete in und vor der *Sende–Bedieneinheit* mit der Sendezeit T_S. Ihre Summe gibt den Füllstand des Pufferspeichers der Bridge an. Die möglichen Zustände und die Übergangsraten zwischen ihnen sind in Bild 4.25 dargestellt. Es gibt folgende Übergänge:

- Der Pufferspeicher der Bridge füllt sich mit der Ankunftsrate λ_A solange, bis alle N_B Pufferspeicherplätze belegt sind. Weitere ankommende Pakete gehen verloren.
- Er leert sich mit der Rate $(1 - p_w)\mu_F$, wenn Pakete nach der Empfangs–Bedieneinheit verworfen werden, weil sie zum Internverkehr eines Netzes gehören, oder mit der Rate μ_S, wenn sie in der Sende–Bedieneinheit vollständig bedient sind.
- Das Ende der Filterphase bei Paketen, welche zum Externverkehr gehören, verursacht einen waagerechten Zustandsübergang mit der Rate $p_w\mu_F$, der den Füllstand des Pufferspeichers nicht verändert.

Da sich das Gleichungssystem aus statistischem Gleichgewicht für die stationären Zustandswahrscheinlichkeiten (eine Gleichung ist redundant) und deren Normierungsbedingung nicht geschlossen für ein allgemeines N_B auflösen läßt, bietet sich hier eine andere Lösungsmöglichkeit an. In Bild 4.24 ist neben der bisher betrachteten linke Seite eine bezüglich des Klemmenverhaltens äquivalente Darstellung als offenes Warteschlangennetz mit zustandsabhängiger Ankunftsrate dargestellt. Die Eigenschaften des Generators sind in dem unterlegten Teil

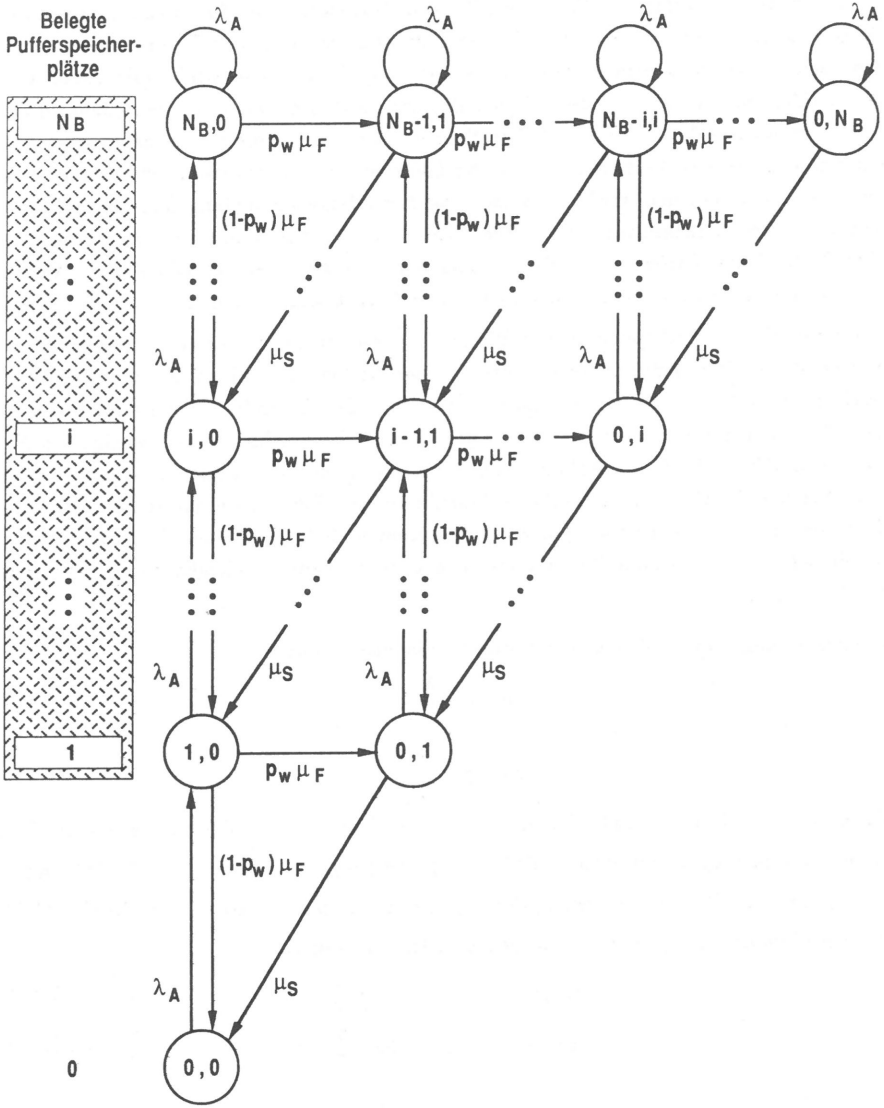


Bild 4.25: Zweidimensionales Zustandsübergangsdiagramm für den Pufferspeicher einer Bridge

enthalten. Solange in der oberen Warteschlange ein Paket vorhanden ist, ist die Ankunftsrate an der Bridge, wie vorher, λ_A. Wird diese Warteschlange leer, so ist die Ankunftsrate an der Bridge $0/s$. In diesem Fall befinden sich aber, aufgrund der konstanten Population des durch den Generator geschlossenen Warteschlangennetzes, N_B Pakete in der Bridge (also im offenen Warteschlangennetz), so daß jedes weitere ankommende Paket sowieso verloren gehen würde. Das bedeutet, daß die Verluste im linken Verkehrsmodell durch die zustandsabhängige Ankunftsrate im rechten Verkehrsmodell nachgebildet werden, so daß die Warteschlangen als unbegrenzt angenommen werden können. Die äquivalente Darstellung führt ebenfalls zu dem Zustandsübergangsdiagramm in Bild 4.25, außer den Übergängen in den jeweils gleichen Zustand am oberen Rand. Diese Übergänge brauchen aber deshalb, weil sie keinen Zustandswechsel verursachen, sowieso nicht weiter berücksichtigt werden.

In diesem offenen, rückkopplungsfreien Warteschlangennetz sind nur Knoten mit negativ exponentiell verteilten Bedienzeiten enthalten, welche die Pakete aus ihren unbegrenzten Warteschlangen in deren Ankunftsreihenfolge bearbeiten. Der Ankunftsprozeß ist ein Poisson–Prozeß mit konstanter Ankunftsrate, solange das offene Warteschlangennetz weniger als N_B Pakete enthält, und mit der Ankunftsrate $0/s$ sonst. Deshalb sind nach Abschnitt 4.1.3.4 die zweidimensionalen Zustandswahrscheinlichkeiten aus Bild 4.25 proportional zum Produkt von Termen, welche jeweils nur von *einem* Knoten abhängen. In dem hier vorliegenden Spezialfall entsprechen diese Terme direkt den Zustandswahrscheinlichkeiten der einzelnen Knoten [132].

Die Auslastungen der Bedieneinheiten werden wie folgt definiert:

$$\varrho_E = \frac{\lambda_A}{\mu_F} \tag{4.33}$$

$$\varrho_S = \frac{p_w \lambda_A}{\mu_S} . \tag{4.34}$$

Dabei entspricht hier p_w nach Gleichung (4.31) direkt dem Anteil des Externverkehrs. Die Zustandswahrscheinlichkeiten $p_E(x) = \mathrm{P}\{X_E = x\}$ und $p_S(y) = \mathrm{P}\{X_S = y\}$ der M/M/1–Knoten ergeben sich für alle x beziehungsweise y jeweils aus ihrem statistischen Gleichgewicht für jeden Zustandsübergang mit anschließender Normierung zu

$$p_E(x) = (1 - \varrho_E)\varrho_E^x \tag{4.35}$$

$$p_S(y) = (1 - \varrho_S)\varrho_S^y . \tag{4.36}$$

Die zweidimensionalen Zustandswahrscheinlichkeiten $\tilde{p}_B(x,y) = \mathrm{P}\{X_E = x, X_S = y\}$ des Pufferspeichers der Bridge können nun als Produktlösungsform angesetzt werden:

$$\tilde{p}_B(x,y) = C \cdot \varrho_E^x \varrho_S^y . \tag{4.37}$$

Dabei muß $0 \leq x + y \leq N_B$ erfüllt sein. $C = \tilde{p}_B(0,0)$ ist der Proportionalitätsfaktor, welcher auch die beiden konstanten Faktoren der Gleichungen (4.35) und (4.36) enthält und sich aus der Normierungsbedingung ergibt:

$$\sum_{x=0}^{N_B} \sum_{y=0}^{N_B - x} \tilde{p}_B(x,y) = 1 \; . \tag{4.38}$$

Insgesamt erhält man als Ergebnis für die zweidimensionalen Zustandswahrscheinlichkeiten des Pufferspeichers der Bridge, unter Berücksichtigung von $0 \leq x + y \leq N_B$,

$$\tilde{p}_B(x,y) = \frac{\varrho_E^x \varrho_S^y}{\sum_{i=0}^{N_B} \sum_{j=0}^{N_B - i} \varrho_E^i \varrho_S^j} \; . \tag{4.39}$$

Durch Einsetzen dieses Ergebnisses in das statistische Gleichgewicht jeder Zustandswahrscheinlichkeit läßt sich der Ansatz als richtig nachweisen. Eine andere Lösungsmöglichkeit wäre die Berechnung des Verkehrsmodells in Bild 4.24 (rechts) als geschlossenes Warteschlangennetz unter Berücksichtigung des Generators als zusätzlichem Knoten. Mit Hilfe der zweidimensionalen Zustandswahrscheinlichkeiten kann man die eindimensionalen Zustandswahrscheinlichkeiten $p_B(x) = \mathrm{P}\{X_B = x\}$ des Pufferspeichers der Bridge für $0 \leq x \leq N_B$ angeben als:

$$p_B(x) = \sum_{i=0}^{x} \tilde{p}_B(x - i, i) \; . \tag{4.40}$$

Insbesondere erhält man wegen des Poisson–Ankunftsprozesses des ursprünglichen Verkehrsmodells in Bild 4.24 (links) für $X_B = N_B$ die Verlustwahrscheinlichkeit

$$B = \sum_{i=0}^{N_B} \tilde{p}_B(N_B - i, i) \; . \tag{4.41}$$

In den folgenden Abschnitten wird eine mathematische Analyse des allgemeinen Verkehrsmodells in Bild 4.23 vorgestellt, ebenfalls unter Berücksichtung des begrenzten Pufferspeichers der Bridge und seiner dynamischen Aufteilung zwischen verschiedenen Warteschlangen. Dabei werden zunächst die Teilmodelle unabhängig voneinander analysiert, und die gegenseitigen Abhängigkeiten werden anschließend durch einen iterativen Algorithmus berücksichtigt.

4.4.3 Analyse der Bearbeitungseinheit

Die Bearbeitungseinheit kann als M/G/1–S_E–System mit der Ersatzbedieneinheit in Bild 4.26 und der Summenankunftsrate λ_A analysiert werden.

Die Laplace–Transformierte $\Phi_E(s)$ der Verteilungsdichtefunktion $f_E(t)$ der Ersatzbedienzeit T_E ergibt sich nach Gleichung (4.29) aus den Laplace–Transformierten $\Phi_F(s)$ und $\Phi_W(s)$ zu

$$\Phi_E(s) = \Phi_F(s) \cdot [p_w \Phi_W(s) + (1 - p_w)] \,. \tag{4.42}$$

Zur Berechnung der Verteilung der Anzahl von Ankünften während einer Ersatzbedienzeit kann Gleichung (4.42) direkt in eine Gleichung aus [143] eingesetzt werden, oder es wird die Verteilungsfunktion $F_E(t)$ mit Hilfe der ersten beiden Momente von T_E (nach Gleichung (4.30) aus Gleichung (4.42)) approximiert, siehe Abschnitt 4.1.3.1, und für diese Berechnung verwendet [175].

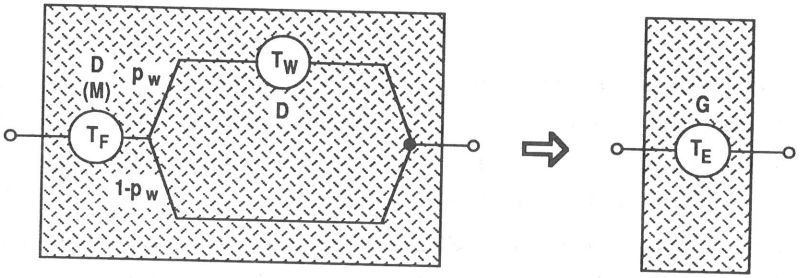


Bild 4.26: Phasenmodell für die Ersatzbedieneinheit der Bearbeitungseinheit

Die Analyse des M/G/1–S_E–Systems erfolgt mit Hilfe der Methode der eingebetteten Markoff–Kette, wobei die Regenerationszeitpunkte jeweils unmittelbar nach dem Ende der Ersatzbedienzeit (also der gedächtnisbehafteten Phase) liegen, was in Bild 4.26 als Punkt gekennzeichnet ist. Die Zustandswahrscheinlichkeiten der Bearbeitungseinheit an den Regenerationszeitpunkten erhält man rekursiv aus dem Gleichungssystem für ihr statistisches Gleichgewicht im Zustandsübergangsdiagramm nach Bild 4.27 mit anschließender Normierung (*eine* Zustandswahrscheinlichkeit darf zunächst beliebig angenommen werden). Dabei sei p_{a_i} die Wahrscheinlichkeit für i Ankünfte seit dem letzten Regenerationszeitpunkt. Der Zustand $X_E = N_E$ wird nicht erreicht, da der Prozeß immer an den Zeitpunkten betrachtet wird, wenn gerade eine Bedienung abgeschlossen ist. Einen Sonderfall stellt der Zustand $X_E = 0$ dar. Hier braucht nur das Zeitintervall ab der nächsten Ankunft betrachtet werden, da sich davor am Prozeßverlauf nichts mehr ändern kann. Erst dann beginnt auch wieder eine neue Ersatzbedienzeit, an deren Ende sich der nächste Regenerationszeitpunkt befindet und während der die Anzahl der weiteren Ankünfte beobachtet wird. Diese erste Ankunft wird also bei der Anzahl der Ankünfte bis zum nächsten Regenerationszeitpunkt nicht mitgezählt.

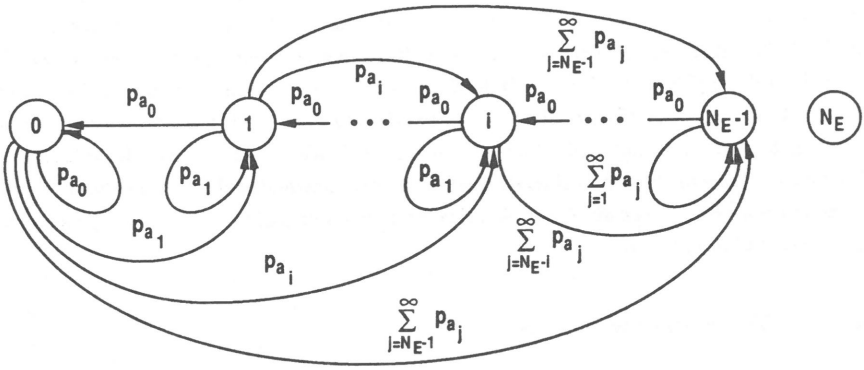


Bild 4.27: Eindimensionales Zustandsübergangsdiagramm für die Bearbeitungseinheit an den Regenerationszeitpunkten

Aus den Zustandswahrscheinlichkeiten der Bearbeitungseinheit an den Regenerationszeitpunkten kann man die Zustandswahrscheinlichkeiten an den Ankunftszeitpunkten berechnen. Ankommende Pakete, welche nicht verloren gehen, sehen dieselben Zustandswahrscheinlichkeiten wie bediente (an den Regenerationszeitpunkten) [75]. Werden auch noch die Pakete mitberücksichtigt, welche verlorengehen, so ist wegen des um den Zustand N_E vergrößerten Zustandsraumes ein Umnormieren notwendig. Der Umnormierungsfaktor kann mit Hilfe des Flußerhaltungssatzes (effektive Ankunftsrate = effektive Bedienrate) ermittelt werden. Wegen des Poisson–Ankunftsprozesses sind diese Antreffwahrscheinlichkeiten gleichzeitig die Zustandswahrscheinlichkeiten $p_E(x) = \mathrm{P}\{X_E = x\}$ der Bearbeitungseinheit für $0 \leq x \leq N_E$ an beliebigen Zeitpunkten [200]. Die Verlustwahrscheinlichkeit ist

$$B = p_E(N_E) \,. \tag{4.43}$$

4.4.4 Analyse der Netzeinheiten

4.4.4.1 Token Ring LAN

Als Verkehrsmodell für ein Token Ring LAN wird ein Pollingsystem verwendet. Für die Analyse muß ein Poisson–Ankunftsprozeß an jeder Warteschlange angenommen werden. Die

Methode für begrenzte Warteschlangen und *einer* Bedienung pro Warteschlange und Zyklus (Limited–1) ist in [175] enthalten. Sie beruht auf einer eingebetteten Markoff–Kette an den Zeitpunkten unmittelbar vor der Ankunft der Sendeberechtigung an der Bridge. Die Verteilung der Zykluszeit wird mit Hilfe ihrer ersten beiden Momente approximiert, siehe Abschnitt 4.1.3.1. Die Abhängigkeiten aufeinanderfolgender Zyklen werden durch *bedingte Zykluszeiten* berücksichtigt [130]. Über die Wahrscheinlichkeiten für n Ankünfte während der Rückwärtsrekurrenzzeit der Zykluszeit werden aus den Zustandswahrscheinlichkeiten an den Regenerationszeitpunkten die Zustandswahrscheinlichkeiten $p_{Si}(x) = P\{X_{Si} = x\}$ an beliebigen Zeitpunkten ermittelt.

4.4.4.2 Token–Passing Bus LAN

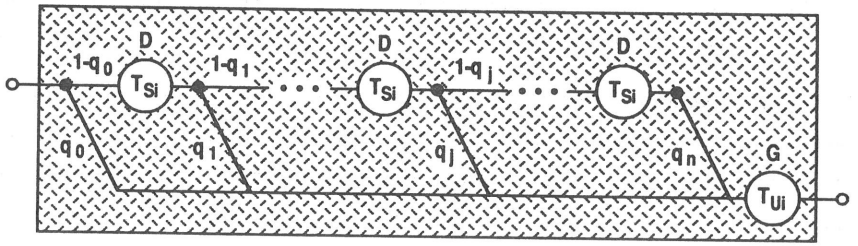


Bild 4.28: Phasenmodell für *eine* Station

Die Analyse des Token Ring LANs kann man erweitern, um die Bedienung von maximal n Paketen pro Warteschlange und Zyklus (Limited–n) zu ermöglichen. Dies entspricht in der höchsten Prioritätsklasse, wegen der Zeitbegrenzung zum Senden (durch den *Hi–Pri–Token–Hold–Timer*) und wegen der angenommenen konstanten Paketgröße, der Analyse eines Token–Passing Bus LANs, wenn dort nur diese höchste Prioritätsklasse verwendet wird. Zu den Regenerationszeitpunkten unmittelbar vor der Ankunft der Sendeberechtigung kommen hier noch weitere Regenerationszeitpunkte, jeweils unmittelbar nach dem Ende einer Sendephase, dazu. Sie sind in dem Phasenmodell nach Bild 4.28 für *eine* Station des LANs als Punkte eingetragen. Aus den Verzweigungswahrscheinlichkeiten q_j (sie können zunächst initialisiert werden und nähern sich dann in einem iterativen Zyklus [141] ihrem wahren Wert) ergibt sich die Wahrscheinlichkeit p_{b_j} für j Übertragungen (Sendephasen) dieser Station pro Zyklus aus

$$p_{b_j} = q_j \cdot \prod_{i=0}^{j-1} (1 - q_i) \; . \tag{4.44}$$

Dabei ist $0 \leq j \leq n$ und das Produkt ohne q_j ist für $j = 0$, wie üblich, als 1 definiert.

Die Laplace–Transformierte $\Phi_{Vi}(s)$ der Verteilungsdichtefunktion $f_{Vi}(t)$ für die Abwesenheitszeit (Vacation Time) T_{Vi} der Sendeberechtigung von einer betrachteten Station k ergibt sich nach Gleichung (4.29) aus den jeweiligen Laplace–Transformierten $\Phi_{Si}(s)$ und $\Phi_{Ui}(s)$ zu

$$\Phi_{Vi}(s) = [\prod_{\forall Stationen} \Phi_{Ui}(s)] \cdot [\prod_{\forall Stationen \neq k} \sum_{j=0}^{n} p_{b_j} \Phi_{Si}^{j}(s)] \,. \qquad (4.45)$$

Dabei wird eine gegenseitige Unabhängigkeit der zu berücksichtigenden Zufallsvariablen und des Ereignisses einer bestimmten Anzahl von Übertragungen pro Station und Zyklus angenommen.

Damit können alle Abstände (Abwesenheitszeiten und Sendezeiten) benachbarter Regenerationszeitpunkte in jeder Station beschrieben werden und die weitere Analyse dieser Netzeinheit der Bridge erfolgt für jeden Regenerationszeitpunkt als M/G/1–S_{Si}–System, analog zur Analyse der Bearbeitungseinheit. Aus den Zustandswahrscheinlichkeiten an den Regenerationszeitpunkten werden die Zustandswahrscheinlichkeiten $p_{Si}(x)$ an beliebigen Zeitpunkten nach [141] basierend auf [144] ermittelt. Für den Spezialfall $n = 1$ erhält man quantitativ dieselben Ergebnisse wie bei der Analyse des Token Ring LANs.

4.4.4.3 Durchgeschalteter Kanal eines WANs

Die Analyse des M/D/1–S_{Si}–Systems ist als Spezialfall in der Analyse der Bearbeitungseinheit enthalten. Für $T_F = T_{Si}$ und $p_w = 0$ erhält man die Zustandswahrscheinlichkeiten $p_{Si}(x)$ an beliebigen Zeitpunkten.

4.4.5 Iterativer Algorithmus zur Ermittlung charakteristischer Größen

In diesem Abschnitt sollen die gegenseitigen Abhängigkeiten der Teilmodelle durch einen iterativen Algorithmus berücksichtigt werden. In einem ersten Schritt wird Simplex– oder Halbduplexverkehr betrachtet, so daß nur zwei Teilmodelle berücksichtigt werden müssen. Im nächsten Schritt kommt das dritte Teilmodell noch dazu, so daß dann das Gesamtmodell nach Bild 4.23 für Duplexverkehr untersucht wird.

4.4.5.1 Simplex– oder Halbduplexverkehr

Ohne Beschränkung der Allgemeinheit wird die Richtung zur Netzeinheit 1 angenommen. Für die Iteration müssen bedingte Zustandswahrscheinlichkeiten für $0 \leq x \leq y$ eingeführt

werden: $\hat{p}_E(x\,|\,y) = \mathrm{P}\{X_E = x\,|\,N_E = y\}$. Sie können nach Abschnitt 4.4.3 für alle sinnvollen Anzahlen N_E von Pufferspeicherplätzen, welche der Bearbeitungseinheit zur Verfügung stehen ($0 \le y \le N_B$), ermittelt werden. Daraus erhält man die dazugehörenden bedingten Verlustwahrscheinlichkeiten

$$\hat{B}(y) = \hat{p}_E(y\,|\,y)\,. \tag{4.46}$$

Bei der Berechnung der Zustandswahrscheinlichkeiten $p_{S1}(x)$ in der Netzeinheit 1 nach Abschnitt 4.4.4 für $0 \le x \le N_B$ wird davon ausgegangen, daß ihr alle N_B Pufferspeicherplätze der Bridge zur Verfügung stehen, da hier keine Verluste mehr auftreten können. Als Ankunftsprozeß wird ein Poisson–Prozeß angenommen, was eine Näherung darstellt, und zwar (wegen der begrenzten Pufferspeichergröße) auch für den Spezialfall nach Bild 4.24. Ein quantitativer Vergleich der resultierenden Verlustwahrscheinlichkeit mit dem exakten Ergebnis aus Abschnitt 4.4.2 hat jedoch gezeigt, daß der Fehler, selbst im ungünstigsten Fall ($N_B = 1$), unter 4% bleibt. Die Ankunftsrate an der Netzeinheit 1 ist

$$\lambda_{S1} = \lambda_{A1}(1 - B)p_{w1}\,. \tag{4.47}$$

p_{w1} entspricht hier direkt dem Anteil des Externverkehrs p_{e1} und nach Gleichung (4.31) auch p_w. Die Verlustwahrscheinlichkeit B ergibt sich nach dem Gesetz von der totalen Wahrscheinlichkeit [132] aus

$$B = \sum_{i=0}^{N_B} \hat{B}(i)\; p_{S1}(N_B - i)\,. \tag{4.48}$$

In Gleichung (4.48) ist wegen der Ankunftsrate an der Netzeinheit 1 nach Gleichung (4.47) auch auf der rechten Seite B enthalten. Eine Auflösung der Gleichung nach B ist beispielsweise für den Spezialfall in Bild 4.24 mit $N_B = 1$ möglich, so daß die Qualität der obigen Näherung leicht überprüft werden kann. Für den allgemeinen Fall erfolgt die Bestimmung von B iterativ, wobei die Iteration normalerweise nach wenigen Schritten abgebrochen werden kann: B wird zu null initialisiert. Damit ergibt sich aus Gleichung (4.47) eine Ankunftsrate, mit welcher die Zustandswahrscheinlichkeiten $p_{S1}(x)$ berechnet werden können. Aus Gleichung (4.48) erhält man eine neue Verlustwahrscheinlichkeit, welche beim nächsten Iterationszyklus in Gleichung (4.47) eingesetzt werden kann.

Nachdem die Zustandswahrscheinlichkeiten $p_{S1}(x)$ bekannt sind, kann man mit Hilfe des Gesetzes von der totalen Wahrscheinlichkeit auch die Zustandswahrscheinlichkeiten $p_E(x)$

der Bearbeitungseinheit für $0 \leq x \leq N_B$ angeben:

$$p_E(x) = \sum_{i=0}^{N_B} \hat{p}_E(x \,|\, i)\, p_{S1}(N_B - i)\ . \tag{4.49}$$

Schließlich kann man mit Hilfe der Zustandswahrscheinlichkeiten neben der bereits bekannten Verlustwahrscheinlichkeit weitere charakteristische Größen ermitteln. Die mittleren Anzahlen belegter Pufferspeicherplätze in Bearbeitungs- und Netzeinheit sind

$$\Omega_E \ = \ \sum_{i=0}^{N_B} i p_E(i) \tag{4.50}$$

$$\Omega_{S1} \ = \ \sum_{i=0}^{N_B} i p_{S1}(i)\ . \tag{4.51}$$

Für die Durchlaufzeit T_B durch die Bridge ergibt sich nach dem Gesetz von Little der Erwartungswert

$$\mathrm{E}[T_B] = b = \frac{\Omega_E}{\lambda_{A1}(1-B)} + \frac{\Omega_{S1}}{\lambda_{S1}}\ . \tag{4.52}$$

4.4.5.2 Duplexverkehr

Jetzt soll auch das dritte Teilmodell (Netzeinheit 2) noch mit berücksichtigt werden, so daß die Bridge bei Duplexverkehr untersucht werden kann. Dabei ergibt sich der Ankunftsprozeß von einem LAN aus den Ankunftsprozessen der aktiven Stationen (ohne die Bridge) an diesem LAN, beeinflußt durch dessen Medienzugangsverfahren. Dies wird bei der Verkehrssimulation explizit berücksichtigt, im Gegensatz zur mathematische Analyse der Bearbeitungseinheit, bei welcher weiterhin die Annahme eines Poisson-Ankunftsprozesses getroffen werden muß. Deshalb ist eine größere Abweichung der analytischen Ergebnisse von den Simulationsergebnissen als bei Simplex- oder Halbduplexverkehr zu erwarten, zumal sich der Fehler durch diese Annahme auch auf den Ankunftsprozeß der Netzeinheit für das abgehende Netz auswirkt.

Die bedingten Zustandswahrscheinlichkeiten $\hat{p}_E(x \,|\, y)$ werden für $0 \leq x \leq y$ wieder nach Abschnitt 4.4.3 und damit die bedingten Verlustwahrscheinlichkeiten $\hat{B}(y)$ nach Gleichung (4.46) für alle sinnvollen Anzahlen von Pufferspeicherplätzen, welche der Bearbeitungseinheit zur Verfügung stehen ($0 \leq y \leq N_B$), berechnet.

Da jetzt zwei Netzeinheiten berücksichtigt werden sollen, müssen auch für sie bedingte Zustandswahrscheinichkeiten $\hat{p}_{Si}(x \,|\, y) = \mathrm{P}\{X_{Si} = x \,|\, N_{Si} = y\}$ für $0 \leq x \leq y$ eingeführt werden.

Zu ihrer Berechnung nach Abschnitt 4.4.4, jeweils für alle sinnvollen Anzahlen N_{Si} von Pufferspeicherplätzen ($0 \leq y \leq N_B$), wird die Ankunftsrate des angenommenen Poisson–Prozesses nach Gleichung (4.47) verwendet. Für die andere Ankunftrsrate gilt eine analoge Definition. *Einer* Netzeinheit stehen alle N_B Pufferspeicherplätze der Bridge zur Verfügung, welche nicht von der jeweils anderen belegt sind, da hier keine Verluste auftreten können. Die Zustandswahrscheinlichkeiten $p_{Si}(x)$ der Netzeinheiten für $0 \leq x \leq N_B$ kann man deshalb (unter Verwendung des Gesetzes von der totalen Wahrscheinlichkeit) aus dem Gleichungssystem (4.53) und (4.54) ermitteln, indem man aufgrund der linearen Abhängigkeit *eine* Gleichung wegläßt, *eine* Zustandswahrscheinlichkeit zunächst zu eins initialisiert und das Ergebnis anschließend, mit Hilfe der Normierungsbedingungen (4.55) und (4.56), nomiert:

$$p_{S1}(x) = \sum_{i=0}^{N_B} \hat{p}_{S1}(x\,|\,i)\; p_{S2}(N_B - i) \tag{4.53}$$

$$p_{S2}(x) = \sum_{i=0}^{N_B} \hat{p}_{S2}(x\,|\,i)\; p_{S1}(N_B - i) \tag{4.54}$$

$$\sum_{i=0}^{N_B} p_{S1}(i) = 1 \tag{4.55}$$

$$\sum_{i=0}^{N_B} p_{S2}(i) = 1\,. \tag{4.56}$$

Daraus kann man die Zustandswahrscheinlichkeiten $p_S(x)$ von beiden Netzeinheiten zusammen für $0 \leq x \leq N_B$, wieder mit dem Gesetz von der totalen Wahrscheinlichkeit, berechnen:

$$p_S(x) = \sum_{i=0}^{x} \hat{p}_{S1}(x - i\,|\,N_B - i)\; p_{S2}(i) = \sum_{i=0}^{x} \hat{p}_{S2}(x - i\,|\,N_B - i)\; p_{S1}(i)\,. \tag{4.57}$$

Bei den bedingten Wahrscheinlichkeiten wird hier insbesondere die Tatsache sichtbar, daß *einer* Netzeinheit alle die Pufferspeicherplätze zur Verfügung stehen, welche nicht von der anderen belegt sind. Die Verlustwahrscheinlichkeit B am Eingang der Bridge ist jetzt

$$B = \sum_{i=0}^{N_B} \hat{B}(i)\; p_S(N_B - i)\,. \tag{4.58}$$

Die Iteration kann nun analog zum Simplex– oder Halbduplexverkehr durchgeführt werden. Dasselbe gilt für die Berechnung der Zustandswahrscheinlichkeiten $p_E(x)$ der Bearbeitungseinheit und der weiteren charakteristischen Größen der Bridge aus den verschiedenen Zustandswahrscheinlichkeiten.

Der Iterationszyklus soll zum Abschluß dieses Abschnitts mit Hilfe einiger Schlüsselworte der problemorientierten Programmiersprache PASCAL nocheinmal zusammengefaßt werden:

BEGIN

$\hat{p}_E(x\,|\,y)$:= [Abschnitt 4.4.3];

$\hat{B}(y)$:= [Gleichung (4.46)];

$B := 0$; {Initialisierung}

REPEAT {Iteration}

λ_{Si} := [Gleichung (4.47)];

IF Duplex **THEN**

BEGIN

$\hat{p}_{Si}(x\,|\,y)$:= [Abschnitt 4.4.4];

$p_{Si}(x)$:= [Gleichungen (4.53) bis (4.56)];

$p_S(x)$:= [Gleichung (4.57)];

B := [Gleichung (4.58)];

END ELSE

BEGIN

$p_{S1}(x)$:= [Abschnitt 4.4.4];

B := [Gleichung (4.48)];

END

UNTIL Konvergenz;

$p_E(x)$:= [Gleichung (4.49)];

[Berechnung weiterer charakteristischer Größen nach Gleichungen (4.50) bis (4.52)];

END.

4.4.6 Ergebnisse

Einige exemplarische Leistungsuntersuchungen mit Hilfe des im letzten Abschnitt beschriebenen iterativen Algorithmusses sollen nun vorgestellt werden. Zur Validierung der Ergebnisse wird ein für diesen Zweck entwickeltes, universelles Simulationsprogramm verwendet [176]. Dieses Simulationsprogramm ist modular aufgebaut. Es enthält Moduln zur Simulation des durchgeschalteten Kanals eines WANs und aller zur Zeit bei der ISO standardisierter Medienzugngsverfahren für LANs (CSMA/CD, Token–Passing Bus und Token Ring), welche beliebig miteinander oder mit einem Generator kombiniert werden können. Ein weiteres

Modul enthält das Verkehrsmodell einer Bridge nach Bild 4.23. Die Simulationsergebnisse werden zusammen mit ihren 95%–Vertrauensintervallen dargestellt, sofern diese nicht kleiner sind als die verwendeten Symbole.

Die folgenden Parameter liegen den Ergebnissen zugrunde: Die konstante Paketgröße sei 1000 *Bit*. Der Pufferspeicher der Bridge ist so dimensioniert, daß er acht Pakete aufnehmen kann. In einer realen Bridge ist er in der Regel größer (beispielsweise 256 *KB* [79, 156]), was aber aufgrund der vielen Zustände dann nicht mehr sinnvoll analysiert werden kann und auch keine prinzipiell neuen Erkenntnisse liefern würde. Außerdem könnten die dann sehr seltenen Verluste durch Verkehrssimulation praktisch nicht mehr validiert werden. Für die Filter– und Weiterbearbeitzeit wird jeweils eine Dauer von 50 μs eingestellt, was einer modernen Bridge (siehe Abschnitt 3.3.2.2) mit einer Filterrate von 20000 Paketen pro Sekunde und einer Übertragungsrate von 10000 Paketen pro Sekunde entspricht. An jedem LAN sind, einschließlich der Bridge, zehn aktive Stationen angeschlossen. Bei den Ergebnissen für Duplexverkehr wird $\lambda_{A1} = \lambda_{A2}$ eingestellt. Das Angebot A_B an der Bridge mit der Pseudoeinheit *Erlang* ist auf die Bearbeitungseinheit bezogen und definiert als

$$A_B = \lambda_{A1}(f + p_{e1}w) + \lambda_{A2}(f + p_{e2}w) \; . \tag{4.59}$$

Dabei sind f und w die Erwartungswerte der Zufallsvariablen T_F und T_W.

Während in den Bildern 4.29 und 4.30 zunächst Simplex– oder Halbduplexverkehr betrachtet wird, liegt den Bildern 4.31 und 4.32 eine Duplexkonfiguration zugrunde.

In Bild 4.29 wird ein Token Ring LAN mit einer Übertragungsgeschwindigkeit von 4 $MBit/s$ als abgehendes Netz verwendet. Die Verlustwahrscheinlichkeit stimmt, insbesondere bei einem Anteil des Externverkehrs von 1%, ausgezeichnet mit den Simulationsergebnissen überein, da der Ankunftsprozeß an der Sendewarteschlange hier tatsächlich fast ein Poisson–Prozeß ist.

Bild 4.30 zeigt die mittlere Anzahl der belegten Pufferspeicherplätze, sowie deren Aufteilung auf Bearbeitungs– und Netzeinheit (Token–Passing Bus LAN mit einer Übertragungsgeschwindigkeit von 10 $MBit/s$). Die Zeitbegrenzung zum Senden ist so eingestellt, daß bis zu acht Pakete pro Station und Zyklus gesendet werden können, und der Anteil des Externverkehrs ist 1%. Auch hier stimmen die Resultate gut mit den Simulationsergebnissen überein. Der größte Teil des Pufferspeichers wird von der Bearbeitungseinheit belegt. Sie stellt also bei dieser Konfiguration den Engpaß innerhalb der Bridge dar. Dies würde wesentlich verstärkt auch für ein HSLAN als abgehendes Netz gelten. Die Pufferspeicheraufteilung für die dem Bild 4.29 zugrundeliegende Konfiguration, mit einem Anteil des Externverkehrs von 10%,

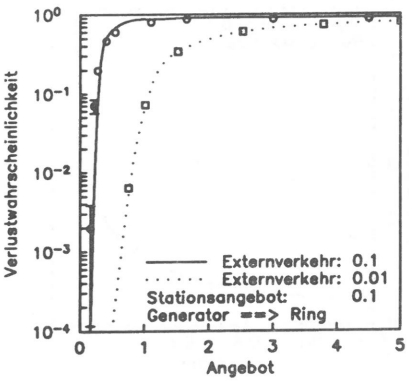


Bild 4.29: Verlustwahrscheinlichkeit über dem Angebot

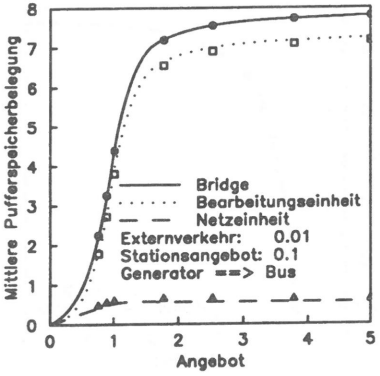


Bild 4.30: Pufferspeicheraufteilung über dem Angebot

würde allerdings vollkommen anders aussehen: Dort würden fast alle Pufferspeicherplätze von der 4 $MBit/s$ Token Ring Netzeinheit belegt werden, so daß dann sie innerhalb der Bridge der Engpaß wäre.

Bild 4.31 zeigt ein Ergebnis für die Kopplung zweier Netze mit stark unterschiedlichen Übertragungsgeschwindigkeiten bei Duplexverkehr. Es werden zwei durchgeschaltete Kanäle von WANs, einer mit der Übertragungsgeschwindigkeit 1 $MBit/s$ und der andere mit 10 $MBit/s$, verwendet. Die Verlustwahrscheinlichkeit erweist sich auch bei der Simulation in beiden Richtungen als gleich groß, und sie stimmt sehr gut mit den analytischen Ergebnissen überein, obwohl jetzt die dynamische Auteilung des Pufferspeichers der Bridge zwischen drei Warteschlangen berücksichtigt werden muß.

Für das letzte Ergebnis wird die Kopplung zweier identischer Token–Passing Bus LANs mit jeweils einer Übertragungsgeschwindigkeit von 10 $MBit/s$ untersucht. Bild 4.32 zeigt die mittlere Durchlaufzeit durch die Bridge, wenn die Ankunftsrate in allen Stationen an beiden LANs symmetrisch von null bis zur Stabilitätsgrenze, welche hier durch die Bridge festgelegt ist, erhöht wird. Die Übereinstimmung der analytischen Ergebnisse mit den Simulationsergebnissen ist jetzt nicht mehr so gut wie bei den anderen drei Bildern, da hier bereits die, für die mathematische Analyse notwendige, Annahme eines Poisson–Prozesses an der Bearbeitungseinheit der Bridge eine Abweichung von der simulierten Realität darstellt.

Im Laufe einer umfangreichen Parameterstudie hat sich herausgestellt, daß die Ergebnisse umso genauer sind und die Iteration umso besser konvergiert, je genauer die Zustandswahrscheinlichkeiten der Netzeinheiten ermittelt werden können. Ist bereits dort, aufgrund von Näherungsannahmen oder schlechter Konvergenz einer inneren Iteration, ein größerer Fehler

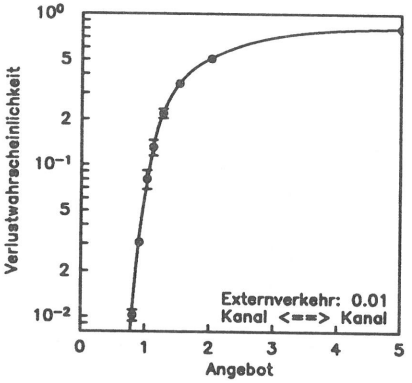


Bild 4.31: Verlustwahrscheinlichkeit über dem Angebot

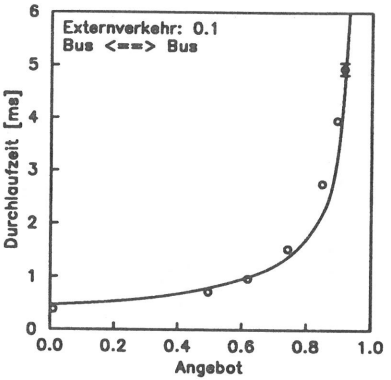


Bild 4.32: Mittlere Durchlaufzeit über dem Angebot

zu erwarten, so kann es auch vorkommen, daß die Iteration zur Bridge–Analyse nicht mehr konvergiert.

Kapitel 5

MAP–Gateway als Tor zur offenen Kommunikation in einer Fabrik

Nach den mehr allgemeinen Untersuchungen zur Netzkopplung soll nun ein konkretes Kopplungsproblem als repräsentatives Beispiel genauer betrachtet werden. Neben einer individuellen Leistungsuntersuchung ist die systemtechnische Realisierung eines Prototyps Gegenstand dieses Kapitels.

5.1 MAP–Gateways als Migrationskomponenten

Bei der Einführung von MAP muß in der Regel Rücksicht darauf genommen werden, daß bereits herstellerspezifische Lösungen des Kommunikationsproblems in einer Fabrik existieren und sich im Betrieb bewährt haben. Diese herstellerspezifischen Netze, welche heute den Materialfluß in einer Fabrik um den notwendigen Informationsfluß ergänzen, können und sollen nicht von heute auf morgen durch standardisierte MAP–Netze ersetzt werden. Stattdessen ist während einer Übergangszeit von bis zu zehn Jahren mit einer Koexistenz von herstellerspezifischen und MAP–Netzen zu rechnen.

Darüberhinaus müssen Hersteller von Kommunikationskomponenten für die Fertigungsautomatisierung heute einen Migrationspfad von ihren herstellerspezifischen Produkten zu zukünftigen MAP–Produkten anbieten. Es gibt zwei Aspekte, die dabei berücksichtigt werden müssen. Zum einen muß die Aufrufschnittstelle für die Anwenderprogramme der MAP–Spezifikation angepaßt werden, damit später dieselbe Anwendersoftware in MAP–Netzen verwendet werden kann, ohne umgeschrieben werden zu müssen. Zum anderen müssen in eine bestehende, herstellerspezifische Umgebung sukzessive flexible Fertigungszellen eingefügt werden können, deren Komponenten über standardisierte Protokolle gemäß der MAP–Spezifikation miteinander kommunizieren, wie dies in Bild 5.1 dargestellt ist. Die Anzahl der

Fertigungszellen im MAP–Netz kann dann in Zukunft immer mehr zunehmen, während im herstellerspezifischen Netz irgendwann keine neuen Fertigungszellen mehr installiert werden sollten. Das anzustrebende Endziel ist die Kommunikation über ein homogenes MAP–Netz, soweit dies aufgrund seiner Leistungsfähigkeit möglich ist und nicht Mini–MAP oder Feldbusse eingesetzt werden müssen. Zentrale Komponenten für die Migration zu MAP sind *MAP–Gateways*, welche die Rolle von Dolmetschern zwischen herstellerspezifischen und standardisierten Stationen einnehmen. Sie vermeiden in der Übergangszeit die Bildung von Kommunikationsinseln, welche dem Prinzip einer computerintegrierten Fertigung widersprechen würden.

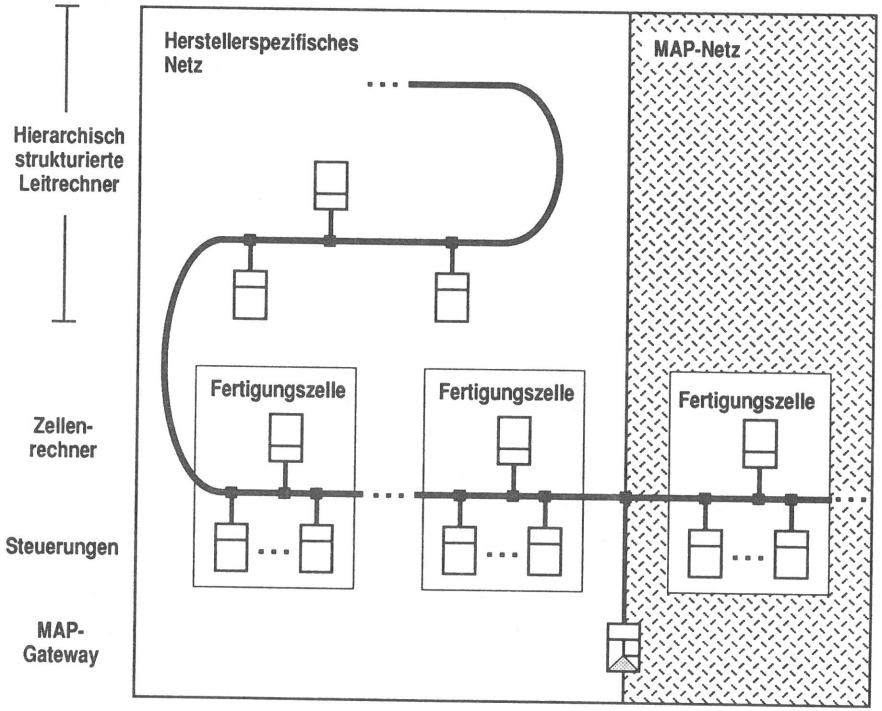


Bild 5.1: Einfügen von MAP–Fertigungszellen in eine herstellerspezifische Umgebung

Ein MAP–Gateway enthält auf der einen Seite das vollständige MAP–Profil mit dem Standardprotokoll MMS auf der Verarbeitungsschicht. Da herkömmliche Protokollprofile für die Fertigungsautomatisierung dieses Verarbeitungsprotokoll noch nicht enthalten, muß die Kopplung auf der Verarbeitungsschicht durchgeführt werden. Dabei bietet es sich an, eine Transformation der Dienstprimitive vorzunehmen. Auf die prinzipielle Notwendigkeit solcher MAP–Gateways wird bereits in einem Anhang der MAP–Spezifikation [149] hingewiesen.

Im folgenden wird beispielhaft ein MAP–Gateway zu dem herstellerspezifischen Netz SINEC näher betrachtet [30]. Das Vorgehen zu seiner Leistungsuntersuchung und Realisierung ist repräsentativ auch für andere Netzkopplungen auf der Verarbeitungsschicht.

5.2 Architektur des MAP–Gateways zu SINEC

In Bild 5.2 ist die Architektur des MAP–Gateways zu SINEC dargestellt. Bei SINEC sind die Aufgaben der Schichten 5 bis 7 nicht geschichtet und werden von dem herstellerspezifischen Automation Protocol (AP) wahrgenommen.

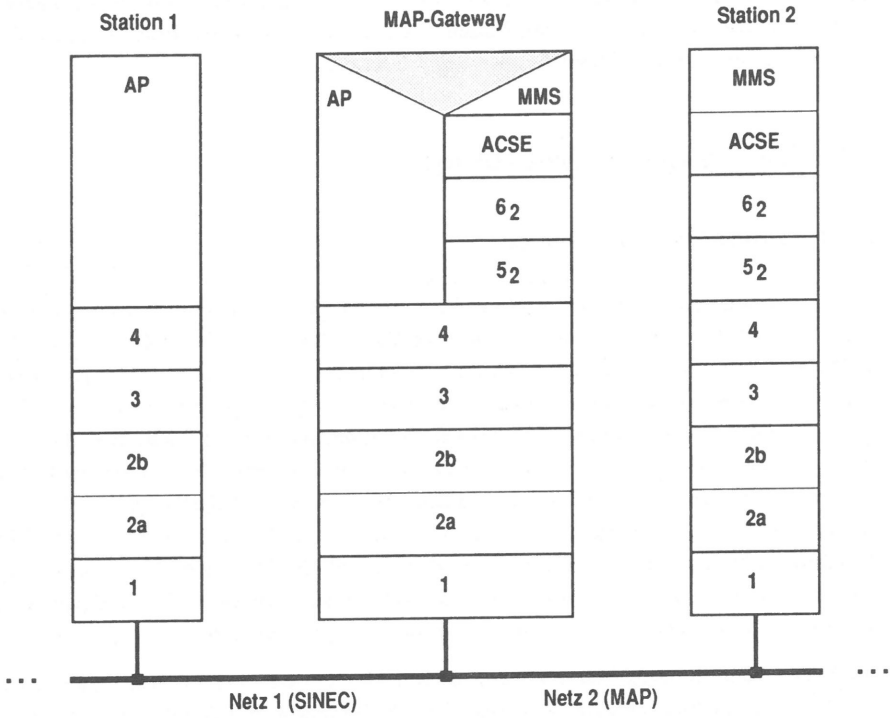


Bild 5.2: Architektur des MAP–Gateways zu SINEC

Auf den unteren vier Schichten werden in beiden Netzen dieselben, von der ISO standardisierten, Protokolle und Medien verwendet. Deshalb können beide Netze physikalisch auf dem gleichen Medium betrieben werden, so daß eine doppelte Verkabelung nicht notwendig ist. Die Kommunikation von einem logischen Netz zum anderen muß jedoch immer über

das MAP–Gateway abgewickelt werden. Die Protokolle des Transportsystems werden auf einer sogenannten Anschaltung ausgeführt. Diese Anschaltung ist aufgrund der identischen Protokolle in beiden Netzen ebenfalls nur einmal notwendig. Dabei werden die Aufgaben der Schichten 1 und 2a in Hardware gelöst, und die Protokolle der Schichten 2b bis 4, welche als Software implementiert sind, werden von einem eigenen Prozessor auf der Anschaltung abgewickelt.

Der eigentliche Prozessor des Kopplungsrechners muß parallel sowohl beide Protokollprofile oberhalb der Transportschicht als auch die Transformation bearbeiten. Die beiden Protokollprofile verwenden unterschiedliche Dienstzugangspunkte der Transportschicht und können deshalb, trotz gemeinsamem Transportsystem, individuell adressiert werden. Wenn der Kopplungsrechner gleichzeitig auch eine normale Station (zum Beispiel einen Zellenrechner) darstellen würde, welche von beiden Netzen aus angesprochen werden kann, so käme die Bearbeitung der normalen Anwendersoftware noch hinzu.

5.3 Leistungsuntersuchung

5.3.1 Modellierungsaspekte

Für die Leistungsuntersuchung des MAP–Gateways zu SINEC wird die Konfiguration in Bild 5.2 zunächst in ein geeignetes Verkehrsmodell abgebildet. Den Engpaß stellen bei der in Abschnitt 5.2 vorgestellten Architektur die Protokolle der Schichten 5 bis 7 dar. Pakete werden auf diesen oberen Schichten nach Aufträgen und Quittungen unterschieden. Messungen in realen Netzen haben gezeigt, daß das Transportsystem durch solche Stationen nur sehr gering ausgelastet werden kann. Es weist auch bei der maximalen Ankunftsrate von Paketen, welche eine solche Station erzeugen kann, für Aufträge und Quittungen je eine konstante Ende–zu–Ende–Verzögerung auf. Deshalb brauchen die unteren vier Schichten nicht detailliert modelliert werden, sondern es genügt, sie jeweils zu einer unendlichen Anzahl von Bedieneinheiten mit konstanter Bedienzeit entsprechend der Ende–zu–Ende–Verzögerung zu aggregieren und für jeden Auftrag eine Transportquittung zu generieren.

Den Schwerpunkt der Modellierung bilden deshalb die Protokolle der Schichten 5 bis 7, sowie die Transformationssoftware. Von diesen Protokollen müssen alle Aspekte berücksichtigt werden, welche sich signifikant auf die Leistungsfähigkeit des gekoppelten Netzes auswirken können. Daraus entsteht die Notwendigkeit eines individuell auf die Kopplung von SINEC und MAP abgestimmten Verkehrsmodells. Das allgemeinere Verkehrsmodell und das dazugehörende Simulationsprogramm aus Abschnitt 4.3 kann deshalb hier nicht verwendet werden, zumal dort nur die Protokolle des Transportsystems explizit modelliert sind, während hier die komplementäre Menge von Protokollen detailliert betrachtet werden muß.

Die Struktur des Verkehrsmodells, dessen Blöcke Warteschlangenmodelle enthalten, ist in Bild 5.3 dargestellt. Dieses Bild enthält alle Protokollinstanzen der beteiligten Stationen, wobei das Transportsystem aus den genannten Gründen aggregiert ist. Ferner ist dem Bild die Zuordnung mehrerer Protokollinstanzen zu einzelnen Prozessoren zu entnehmen, welche auch der im Abschnitt 5.4 beschriebenen Implementierung entspricht. Die Trennung zwischen den Protokollen des Transportsystems und den Schichten darüber bezüglich der Zuordnung zu realen Prozessoren ist eine weitere Voraussetzung dafür, daß die Aggregierung des Transportsystems erlaubt ist. Die Anwendersoftware wird in den Stationen 1 und 2 nicht den Prozessoren zugeordnet welche die Kommunikationsprotokolle bearbeiten, da ihre Bearbeitungszeit sehr stark von der jeweiligen Anwendung abhängt, die hier nicht detailliert modelliert oder gar untersucht werden soll. Ihr Einfluß auf die zu untersuchenden Eigenschaften der Protokollprofile und deren Kopplung wird deshalb vernachlässigt, zumal in zukünftigen Stationen, welche in der realen Produktion eingesetzt werden können, für die Kommunikationsprotokolle ein dedizierter Prozessor zur Verfügung stehen wird.

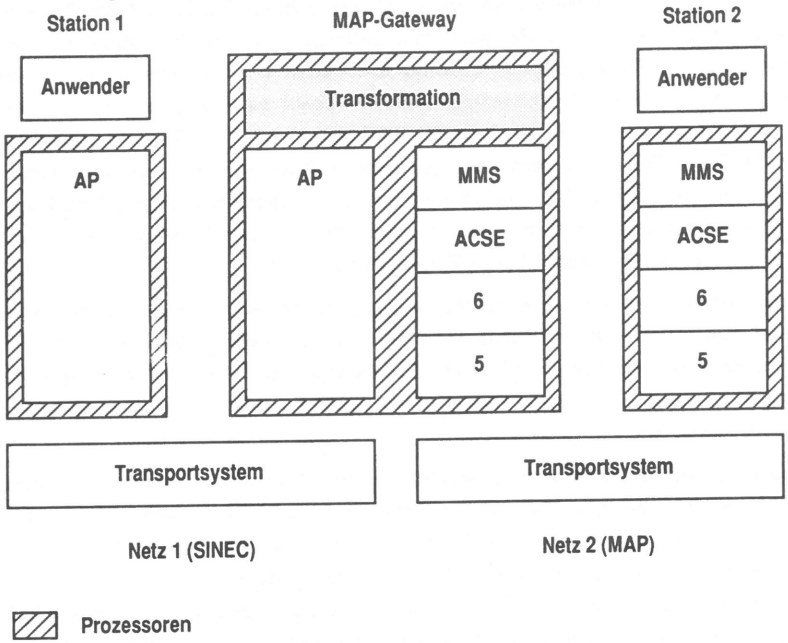


Bild 5.3: Struktur des Verkehrsmodells

Die Priorisierung der Bearbeitungsphasen wird innerhalb eines Prozessors so gewählt, daß grundsätzlich die Priorität mit dem Alter eines Paketes steigt. Da auf den oberen Schichten insbesondere Aufträge und Quittungen voneinander unterscheidbar sind, bedeutet das auch global, daß Quittungen vor Aufträgen priorisiert werden können. Die Zuordnung der

Protokollinstanzen zu einzelnen Prozessoren, sowie die Priorisierung der Bearbeitungsphasen innerhalb eines Prozessors, kann aber im Verkehrsmodell und dem dazugehörenden Simulationsprogramm per Eingabe auch auf eine beliebige andere Art vorgenommen werden. Zwischen zwei aufeinanderfolgenden Bearbeitungsphasen eines Prozessors wird, wie im Abschnitt 4.3, eine Umschaltphase eingeschoben.

Die Protokollinstanzen der Schichten 5 bis ACSE enthalten für jeden Pakettyp je eine Bearbeitungsphase. MMS und AP müssen detaillierter modelliert werden. Die Anwenderprozesse über beiden Protokollen enthalten verbindungsindividuelle Flußkontrollen auf der Basis von Fenster–Mechanismen. Es gibt quittierte und unquittierte Aufträge, wobei im letzteren Fall nicht mehr benötigte Pufferspeichersegmente als lokale Quittungen an die Protokollinstanz zurückgegeben werden müssen, welche sie ursprünglich angefordert hat (geschlossener Pufferkreislauf).

Insbesondere bei AP gibt es darüberhinaus noch einige Besonderheiten, welche bei der Modellierung berücksichtigt werden müssen. Es können mehrere Kanäle auf eine Transportverbindung gemultiplext werden. Bei quittierten Aufträgen wird nach dem Empfang der Transportquittung der Empfang der AP–Quittung überwacht. Wird ein vorgebbares Zeitintervall überschritten, muß der Auftrag wiederholt werden sobald der Prozessor dazu in der Lage ist. Die Anzahl der möglichen Wiederholungen wird durch eine Verwaltungszeit begrenzt, also durch die Zeit während der noch nicht quittierte Aufträge in einer Verwaltungsliste aufbewahrt werden. Außerdem gibt es die Möglichkeit, durch einen Auftrag beim Empfänger einen Reaktionsauftrag anzustoßen, beispielsweise falls die Quittung des ursprünglichen Auftrags eine vorgeschriebene maximale Länge überschreiten würde.

Auf eine detaillierte Darstellung der Warteschlangenmodelle für die beiden Protokollprofile und auf eine genauere Beschreibung soll hier verzichtet werden, da dies den Rahmen der vorliegende Arbeit sprengen würde und zum prinzipiellen Verständnis des folgenden nicht notwendig ist. Der interessierte Leser wird auf entsprechende Veröffentlichungen, beispielsweise [33], verwiesen.

Die teilweise sehr komplexen Szenarien für die Transformation sind in [164] systematisch zusammengestellt. Sie lassen sich für das Verkehrsmodell in Klassen von typischen Szenarien einteilen, wie beispielsweise 1:1–, 1:n– und n:1–Abbildung von Dienstprimitiven. Quittierte Aufträge können abschnittsweise oder Ende–zu–Ende quittiert werden, wobei der letztere Fall meist sinnvoller ist, da er die Abfrage von Daten des Empfängers erlaubt und außerdem die verbindungsindividuellen Flußkontrollen auf eine elegante Art und Weise koppelt, so daß der Pufferspeicherbedarf im MAP–Gateway begrenzt wird.

Ein spezielles Szenario ergibt sich für den Transfer einer Datei, welcher bei MMS und AP vollkommen unterschiedlich realisert wird. In Bild 5.4 ist das Szenario für die Richtung der Datei von der Station 1 zur Station 2 dargestellt. Während bei MMS eine Sequenz der quittierten Aufträge *FileOpen*, mehrere *FileRead* und *FileClose* zum Lesen der Datei durchlaufen

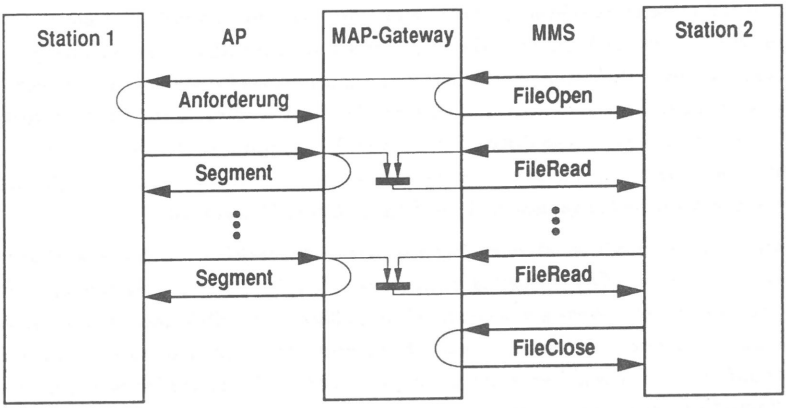


Bild 5.4: Szenario für den Transfer einer Datei von der Station 1 zur Station 2

werden muß, wird bei AP durch eine quittierte Anforderung beim Empfänger ein segmentierter Reaktionsauftrag angestoßen. Dadurch nimmt das MAP–Gateway nach der Anforderung der Datei bei Station 1 eine rein passive Rolle ein: von Station 2 kommen *FileRead*–Aufträge an und von Station 1 die Dateisegmente. Diese können immer dann vom MAP–Gateway an die Station 2 geschickt werden, wenn sowohl mindestens ein *FileRead*–Auftrag von der Station 2 als auch mindestens ein Dateisegment von der Station 1 im MAP–Gateway vorhanden ist. Die Synchronisation wird analog zu einer Transition bei Petri–Netzmodellen modelliert und ist ebenfalls in Bild 5.4 dargestellt. Dieses Szenario ist schneller und benötigt wesentlich weniger Pufferspeicherplatz im MAP–Gateway als die denkbare Alternative, die Datei im MAP–Gateway zunächst vollständig zu rekonstruieren und zwischenzuspeichern, setzt aber eine kompatible Paketgrößen in beiden Netzen voraus.

5.3.2 Stationäre und instationäre Simulation typischer Szenarien

Das Verkehrsmodell kann in ein zeittreues Simulationsprogramm umgesetzt werden, welches sowohl eine stationäre als auch eine instationäre Simulation der erwähnten Szenario–Klassen ermöglicht. Obwohl das Simulationsprogramm wesentlich mehr Freiheiten erlauben würde, werden die Parameter hier so gewählt, daß das Programm auch der im Abschnitt 5.4 beschriebenen Realisierung entspricht. Realistische Einstellungen für die Bearbeitungsphasen werden aus den entsprechenden Prozeduren der Protokollimplementierungen abgeschätzt und können bei Bedarf ebenfalls [33] entnommen werden. Ihre Dauer ist jeweils deterministisch eingestellt, da im wesentlichen bei jedem Paket dieselben Programmteile durchlaufen werden müssen.

Bei den folgenden Simulationsergebnissen wird jeweils nur *eine* Kommunikationsbeziehung

während ihrer Datentransferphase betrachtet. Da die Anwendersoftware nicht detailliert modelliert ist, wird die Verkehrsquelle beim Sender so eingestellt, daß sie Aufträge mit negativ exponentiell verteilten Ankunftsabständen erzeugt, was für viele Anwendungsfälle eine gute Näherung der tatsächlichen Verkehrscharakteristik repräsentiert. Flußkontrollen und Zeitüberwachungen werden so dimensioniert, daß sie im normalen Betriebsfall den Verkehrsfluß nicht behindern beziehungsweise keine zusätzliche Last verursachen. Dadurch stellen die Flußkontrollen bei den folgenden Untersuchungen keinen Engpaß dar.

Die Kurven sind jeweils im stabilen Bereich dargestellt, welcher durch die Auslastung des Engpaßprozessors im MAP–Gateway begrenzt ist. *Der Kehrwert der Summe aller Bedien– und Umschaltphasen, welche pro Auftrag (einschließlich seiner Quittungen) durchlaufen werden müssen, entspricht der maximal möglichen Ankunftsrate.* Im Hochlastfall stauen sich die Pakete aufgrund der gewählten Priorisierung vor allem in der ersten Warteschlange, welche dem Engpaßprozessor zugeordnet ist.

Simulationspunkte sind gemeinsam mit ihren 95%–Vertrauensintervallen dargestellt, sofern diese nicht kleiner sind als die verwendeten Symbole. Die untersuchten charakteristischen Zeiten sind folgendermaßen definiert:

- Unter der *Transferzeit* wird hier die Zeit verstanden, welche von der Generierung eines Auftrags vergeht, bis dieser beim Anwender in der Empfängerstation ankommt.
- Die *Speicherbelegungszeit* beim Sender repräsentiert die Zeit von der Generierung eines Auftrags bis zur Freigabe des belegten Pufferspeichersegments durch die ankommende Quittung beim Anwender der Senderstation. Diese Quittung wird bei unquittierten Aufträgen, angestoßen durch die Transportquittung, lokal erzeugt.
- Die *Dateilesezeit* ist die Zeit, welche für das Szenario in Bild 5.4 benötigt wird, also vom Generieren des Auftrages *FileOpen* bis zum Empfang der Quittung des dazugehörenden Auftrages *FileClose*.

In Bild 5.5 sind Ergebnisse für eine 1:1–Abbildung von unquittierten Aufträgen für die Richtung von MAP zu SINEC dargestellt. Die Stabilitätsgrenze ist bei 15 Aufträgen pro Sekunde, und die mittleren Transferzeiten sind 80 *ms* und größer. Die mittlere Speicherbelegungszeit wird hier nicht vom Engpaßprozessor beeinflußt, da die lokale Quittung beim Sender bereits durch die Transportquittung erzeugt wird.

Im Vergleich dazu werden in Bild 5.6 Aufträge mit Ende–zu–Ende–Quittierung betrachtet. Während sich die zusätzlich zu bearbeitende Quittung nur geringfügig auf die mittlere Transferzeit auswirkt, wird die mittlere Speicherbelegungszeit wesentlich größer, da ein belegtes Pufferspeichersegment erst nach dem Erhalt der dazugehörenden Quittung freigegeben wird und die Quittierung Ende–zu–Ende erfolgt. Sie ist mindestens 150 *ms* groß.

Ein Beispiel für die Szenarioklasse 1:*n* ist der transparente Datenaustausch in beide Richtungen, angestoßen von der SINEC–Seite aus. Während dafür bei AP ein einziger quittierter

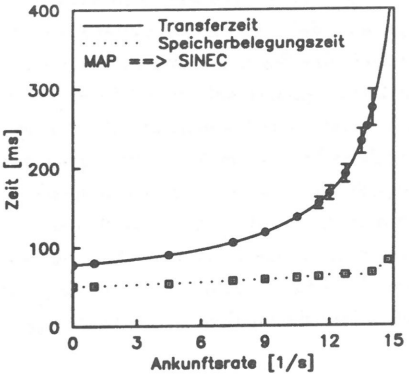


Bild 5.5: Stationäre Simulation des unquittierten Auftrags (1:1–Abblidung)

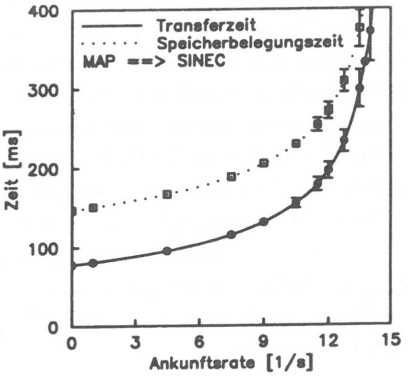


Bild 5.6: Stationäre Simulation des quittierten Auftrags (1:1–Abbildung)

Auftrag ausreicht, muß bei MMS die Sequenz der quittierten Aufträge *Write* und *Read* durchlaufen werden. Simulationsergebnisse dafür sind in Bild 5.7 enthalten. Durch den doppelten Verkehr auf der MAP–Seite sinkt die Stabilitätsgrenze drastisch, was dadurch noch verstärkt wird, daß die MAP–Seite sowieso schon eine größere Anzahl von Protokollinstanzen enthält. Der Sender darf hier ein belegtes Pufferspeichersegment erst freigeben, wenn das gesamte (komplexere) Szenario abgeschlossen ist, was sich negativ auf die mittlere Speicherbelegungszeit auswirkt.

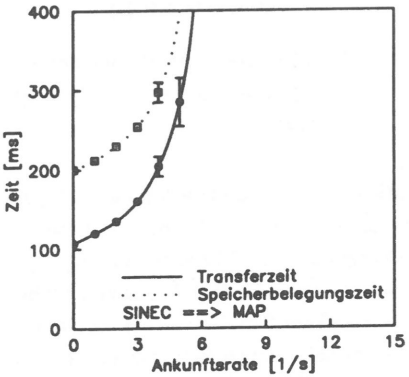


Bild 5.7: Stationäre Simulation des transparenten Datenaustauschs (1:2–Abbildung)

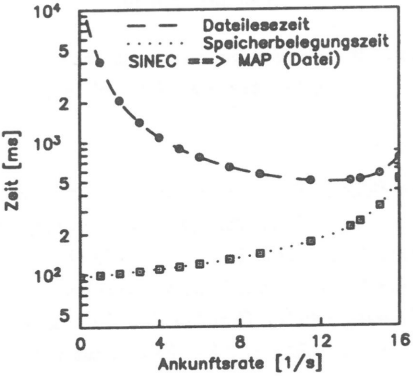


Bild 5.8: Stationäre Simulation des Dateitransfers (jeweils drei Dateisegmente)

Bei der letzten stationären Simulation, welche hier vorgestellt werden soll, wird das Szenario von Bild 5.4 untersucht. Es wird davon ausgegangen, daß Dateien konstanter Länge zu übertragen sind, welche für die Kommunikation jeweils in drei Segmente aufgespalten werden müssen. Die Fenstergröße zur Übertragung dieser Dateisegmente auf der SINEC–Seite sei zur Gewährleistung der richtigen Reihenfolge eins, so daß jedes Segment erst dann ausgesandt werden darf, wenn das MAP–Gateway den erfolgreichen Empfang des vorhergehenden bestätigt hat. Die mittlere Speicherbelegungszeit in Bild 5.8 liegt zwischen den entsprechenden Werten der bisherigen Simulationen, da es sich hier im wesentlichen um abschnittsweise quittierte Aufträge handelt. Die mittlere Dateilesezeit nimmt zunächst mit zunehmender Ankunftsrate ab, da die Abstände zusammengehörender *FileRead*–Aufträge kürzer werden. Bei hoher Ankunftsrate bauen sich Warteschlangen auf, welche die mittlere Dateilesezeit wieder erhöhen.

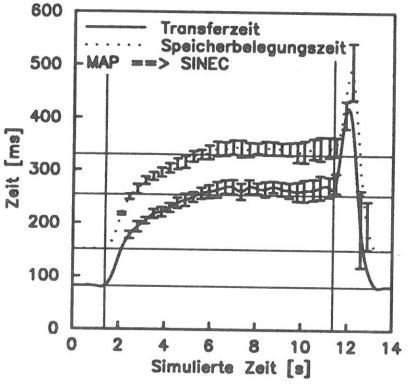


Bild 5.9: Instationäre Simulation des quittierten Auftrags (1:1–Abbildung)

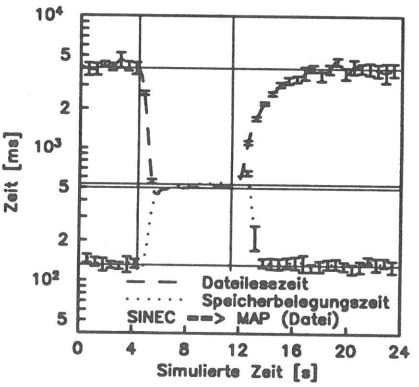


Bild 5.10: Instationäre Simulation des Dateitransfers (jeweils drei Dateisegmente)

Die Bilder 5.9 und 5.10 zeigen instationäre Simulationen zu den Bildern 5.6 und 5.8. Untersucht wird das Ein– und Ausschwingverhalten bei einer Grundlast von 1 Auftrag pro Sekunde und einer Hochlast von 13 (Bild 5.9) beziehungsweise 14 (Bild 5.10) Aufträgen pro Sekunde. Begin und Ende der Hochlastperiode sind durch senkrechte Striche gekennzeichnet. Die waagerechten Geraden sind die Ergebnisse der stationären Simulation. Sie repräsentieren die charakteristischen Zeiten, auf welche das System nach einiger Zeit einschwingt. Die simulierten Punkte werden aus Übersichtlichkeitsgründen nicht mit Symbolen markiert. Die gemessenen Zeiten werden hier über der simulierten Zeit aufgetragen und zwar in dem Augenblick, in welchem sie tatsächlich beobachtet werden. Dadurch ist der Beobachtungsort bei der mittleren Transferzeit der Empfänger und bei der mittleren Speicherbelegungszeit

der Sender. Diese Zeiten weisen nach dem Sprung zurück auf die Grundlast vorübergehend erhöhte Werte auf, was direkt aus dem Auftragen über der Beobachtungszeit am Beobachtungsort resultiert, siehe Abschnitt 4.1.2.3. Das bedeutet beispielsweise, daß die Pufferspeichersegmente, welche unmittelbar nach der Hochlastperiode beim Sender wieder freigegeben werden, überdurchschnittlich lange belegt waren.

Die mittlere Dateilesezeit ist kurz nach dem Sprung auf die hohe Ankunftsrate minimal, da dort die Ankunftsabstände zusammengehörender *FileRead*–Aufträge bereits sehr klein sind, die Warteschlangen sich aber noch nicht voll aufgebaut haben.

Die vorgestellten Simulationsergebnisse zeigen, daß die mittleren Transferzeiten über die Netzgrenze hinweg innerhalb einer flexiblen Fertigungszelle nicht akzeptabel wären, da dort oft im Millisekundenbereich reagiert werden muß. Aus diesem Grund müssen die Fertigungszellen selbst homogen sein, und nur die Kommunikation zwischen Zellen– oder Leitrechnern kann über das MAP–Gateway hinweg abgewickelt werden, wie dies auch in Bild 5.1 dargestellt ist.

5.4 Systemtechnische Realisierung

Nachdem die Architektur und eine Leistungsuntersuchung des MAP–Gateways zu SINEC vorgestellt sind, soll die Realisierbarkeit der Architekturüberlegungen und der erarbeiteten Szenarien anhand einer Pilotimplementierung nachgewiesen werden.

5.4.1 Entwicklungsumgebung und Voraussetzungen

Zur systemtechnischen Realisierung des MAP–Gateways wird ein Intel–310–Rechner mit Speichererweiterung und Anschaltung für die Protokolle des Transportsystems verwendet. Auf diesem Rechner sind für das Multitasking–Betriebssystem iRMX II die Protokolle der Schichten 5 bis 7 von SINEC und MAP verfügbar. Als Testumgebung steht ein LAN mit weiteren SINEC– und MAP–Stationen zur Verfügung.

5.4.2 Die Transformationssoftware

5.4.2.1 Umsetzungsszenarien

Der Kern der Transformationssoftware ist die Realisierung der Szenarien [164]. Aufgrund der völlig unterschiedlichen Protokolle auf der Verarbeitungsschicht und wegen den verfügbaren, abgeschlossenen Protokollimplementierungen auf dem verwendeten Rechner, bietet es sich

an, eine Transformation der Dienstprimitive vorzunehmen, welche die Verarbeitungsinstanzen an ihren Anwenderschnittstellen anbieten beziehungsweise benötigen. Ein Verlust an Funktionalität läßt sich allerdings nicht vermeiden, da es in beiden Verarbeitungsprotokollen Aufträge gibt, welche kein Analogon im jeweils anderen besitzen. Entsprechendes gilt für die einzelnen Parameter innerhalb eines Auftrages.

Eine detaillierte Spezifikation der implementierten Umsetzungsszenarien mit Hilfe der von CCITT standardisierten Functional Specification and Description Language (SDL) ist in [127] zu finden. Die einzelnen Szenarien sind jeweils in einem eigenen Programm–Modul enthalten und können deshalb leicht ergänzt, geändert oder ausgetauscht werden. In der Spezifkation ist jedes Szenario ein SDL–Prozeß, welcher als endlicher Zustandsautomat realisiert ist. Jedes Szenario kann prinzipiell beliebig oft parallel aktiv sein. Die Zustandsinformation wird für jeden aktiven Prozeß in einer eigenen Datenstruktur abgespeichert. Die Zuordnung ankommender Dienstprimitive zu einem speziellen Szenario und zum richtigen Prozeß nimmt ein spezieller Verteilprozeß wahr.

Als Beispiel für die Anpassung inkompatibler Paketgrößen kann folgendes implementierte Szenario betrachtet werden: Bei AP können Daten in Form eines segmentierten Auftrages gesandt werden, wobei eine relativ kleine maximale Paketgröße bei AP angenommen wird. Die einzelnen Segmente werden im MAP–Gateway aufgesammelt, in *ein* Pufferspeichersegment kopiert und nach dem Eintreffen der letzten Daten im MAP–Gateway als *Write*–Auftrag ausgesandt.

Die *Szenarien* sind zusammen mit dem dazugehörenden *Verteilprozeß* als eigene Task des Multitasking–Betriebssystems realisiert, welche über *eine* Warteschlange mit Dienstprimitiven und den benötigten Zusatzinformationen versorgt wird, so daß diese Schnittstelle sehr einfach ist.

5.4.2.2 Globale Aufgaben

Der in Abschnitt 5.4.2.1 beschriebene Kern der Transformationssoftware ist in Bild 5.11 grau unterlegt. Er wird in eine Schale eingebettet, welche globale, szenariounabhängige Aufgaben wahrnimmt und die Schnittstellen zu den Protokollinstanzen der Verarbeitungsschicht bereitstellt. Diese Schale läßt sich in verschiedene funktionelle Blöcke aufteilen, welche im folgenden beschrieben werden. Sie ist im wesentlichen als zwei Tasks des Multitasking–Betriebssystems realisiert, welche jeweils über *eine* Eingangswarteschlange für ankommende Dienstprimitive verfügen.

- Ankommenden Dienstprimitive erreichen zunächst einen *AP–* beziehungsweise *MMS–spezifischen Block*, welcher eine Vorverarbeitung durchführt. Dabei werden insbesondere Aufgaben erledigt, die bei allen AP– beziehungsweise MMS–Dienstprimitiven not-

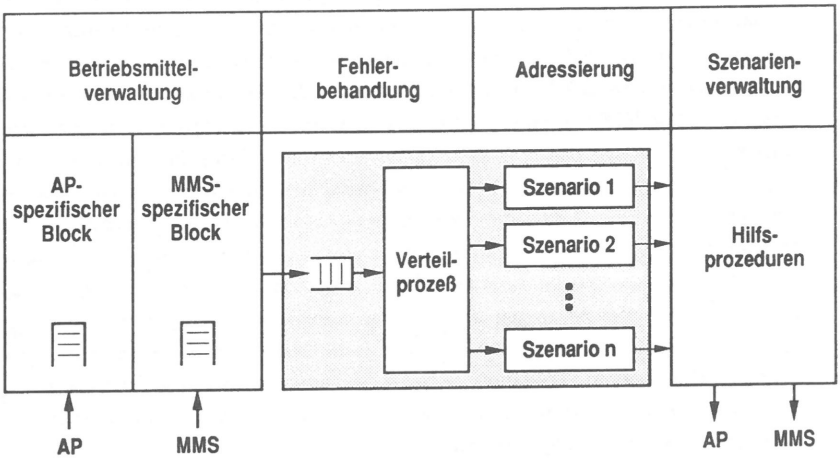


Bild 5.11: Funktionelle Blöcke der Transformationssoftware

wendig sind. Die Dienstprimitive werden mit Hilfe von Zeigern an interne Datenstrukturen gebunden, welche weitere für die Transformation benötigte Parameter enthalten und später an die Warteschlange vor dem Verteilprozeß weitergegeben werden.

- Die *Betriebsmittelverwaltung* überwacht, daß eine vorgebbare Anzahl von Verbindungen und der dafür benötigte reservierte Pufferspeicherplatz nicht überschritten wird. Ist die Maximalanzahl erreicht, so wird jeder weitere Verbindungsaufbauwunsch abgelehnt. Es wird hier also eine Überlastabwehrstrategie auf der Ebene von Verbindungen realisiert.
- Die *Fehlerbehandlung* ist beispielsweise dafür zuständig, eintreffende Duplikate, welche wegen abgelaufenen Zeitüberwachungen entstanden sind, zu erkennen und negativ zu quittieren, anstatt sie wie die Originale an den Verteilprozeß weiterzugeben. Weitere denkbare Fehlerfälle und ihre mögliche Behandlung sind in [11] dokumentiert.
- Bei der *Adressierung* werden die in den verwendeten Protokollimplementierungen sowieso vorhandenen Tabellen (Applikationsbeziehungstabelle bei AP und Nametable bei ACSE) ausgenützt, um beim Verbindungsaufbau mit Hilfe von Namen die Anwenderprozesse (Transformationssoftware im MAP–Gateway und Anwendersoftware beim Empfänger) zu adressieren. Diese Tabellen werden beim Verbindungsaufbau um Identifikationen ergänzt, welche während der Datentransferphase einen schnelleren Zugriff ermöglichen. Für die hier verwendete Adreßtransformation ist eine zusätzliche Abbildungstabelle in der Transformationssoftware nötig: die Kommunikationsbeziehungstabelle. Die Zuordnung der Namen ist für den Verbindungsaufbau statisch projektiert und die Identifikationen werden beim Verbindungsaufbau auch hier eingetragen und erlauben während der Datentransferphase eine schnelle Zuordnung.

- Die *Szenarienverwaltung* ist dafür zuständig, bei jedem ankommenden Dienstprimitiv das dazugehörende Szenario zu ermitteln, bei einem neuen Prozeß die benötigte Datenstruktur aufzubauen und ihm eine Prozeßnummer zur Identifikation zuzuordnen. Diese Prozeßnummer wird neben anderen Parametern in die interne Datenstruktur des AP– beziehungsweise MMS–spezifischen Blocks aufgenommen und später vom Verteilprozeß ausgewertet. Die Zuordnung eines Dienstprimitivs zum dazugehörenden Szenario erfolgt mit Hilfe einer Tabelle, wobei auf der SINEC–Seite die sowieso vorhandene Funktionsverteiltabelle von AP für diese Aufgabe ausgenützt wird. Die Prozeßnummer kann bei Quittungen direkt einem Parameter entnommen werden, welcher beim dazugehörenden Auftrag von der Transformationssoftware ausgefüllt und vom Empfänger lediglich gespiegelt wird. Bei Aufträgen ist dazu eine Tabelle notwendig, deren Eingang die Verbindungsidentifikation und das Szenario sind. Deshalb ist die Einschränkung notwendig, daß auf *einer* Verbindung dasselbe Szenario nur einmal gleichzeitig ablaufen darf. Ist ein Prozeß vollständig durchlaufen, so muß seine Datenstruktur gelöscht und die Nummer wieder freigegeben werden.
- Die *Hilfsprozeduren* dienen dazu, zu sendende Dienstprimitive auszufüllen und sie in die Warteschlange am Eingang des entsprechenden Verarbeitungsprotokolls einzutragen. Dadurch realisieren sie die Schnittstelle zu anderen Tasks in abgehender Richtung. Außerdem sind sie dafür zuständig, benötigte Pufferspeichersegmente anzufordern und ausgebrauchte zurückzugeben. Zu Demonstrationszwecken existieren auch Hilfsprozeduren, welche in eine Bildschirmmaske neben statistische Daten und Tabellen die bearbeiteten Aufträge des aktuellen Szenarios mit ihren Transformationen eintragen und ständig aktualisieren. Die Hilfsprozeduren sind in separaten Moduln enthalten, welche zu jeder der drei Tasks der Transformationssoftware dazugebunden werden, so daß sie auch von den Szenarien aus aufgerufen werden können.

5.4.3 Systemintegration und Funktionstest

Alle Tasks, welche die Protokolle beider Netze enthalten, sind so in das Betriebssystem integriert, daß sie beim Systemstart automatisch mitgestartet werden und im Hintergrund ablaufen.

Für den Start der Transformationssoftware ist eine weitere, bisher nicht erwähnte, temporäre Initialisierungstask zuständig, welche von der Konsole aus aufgerufen wird. Nach dem Bereitstellen der Warteschlangen für jede Task liest sie die Kommunikationsbeziehungstabelle von einer Datei ein und baut die als statisch gekennzeichneten Verbindungen auf. Neben logischen Namen und Verbindungsidentifikationen enthält die Kommunikationsbeziehungstabelle noch weitere Eigenschaften der projektierten Verbindungen, welche für den Verbindungsaufbau benötigt werden. Anschließend gibt sie die Bildschirmmaske mit ihren statischen Einträgen

aus und startet dann die drei im letzten Abschnitt beschriebenen Tasks der Transformationssoftware. Nachdem ihre Aufgabe erfüllt ist, vernichtet sich die Initialisierungstask selbst.

Ein Funktionstest der implementierten Szenarien ist mit Hilfe von Testanwenderprogrammen auf den MAP– und SINEC–Stationen möglich. Diese erlauben das Ausfüllen und Aussenden von Aufträgen und zeigen ankommende Aufträge, einschließlich ihrer relevanten Parameter, auf dem Bildschirm des jeweiligen Empfängers an. Der einwandfreie Ablauf des betrachteten Szenarios kann auch auf dem Bildschirm des MAP–Gateways verfolgt werden. Anhand der dargestellten Tabellen und statistischen Daten kann auch der richtige Zustand der Verbindungen und die korrekte Arbeitsweise der Pufferspeicherverwaltung überprüft werden.

5.5 Verifikationsaspekte

Der erfolgreiche Funktionstest am realisierten Prototyp verifiziert im Nachhinein das der Simulation zugrundeliegende Verkehrsmodell, welches zunächst auf der Basis von theoretisch überlegten Szenarien entstanden ist.

Um auch die durch Simulation ermittelten mittleren Transfer– und Speicherbelegungszeiten zu verifizieren, wäre ein verteiltes Meßsystem notwendig. Ein solches Meßsystem ist Gegenstand einer separaten Arbeit am Institut für Nachrichtenvermittlung und Datenverarbeitung der Universität Stuttgart und kann im Rahmen der vorliegenden Arbeit noch nicht eingesetzt werden. Eine Verkehrsquelle soll dabei Aufträge (hier: mit negativ exponentiell verteilten Ankunftsabständen und der gewünschten Ankunftsrate, analog zur Simulation) erzeugen und sie der Verarbeitungsschicht der sendenden Station anbieten. Mit Hilfe verschiedener Sensoren sollen dann Daten in jeder beteiligten Station aufgesammelt, mit Zeitstempel versehen und lokal zwischengespeichert werden. Nach der eigentlichen Messung sollen diese Daten zur Auswertung über das LAN an eine zentrale Station verschickt werden. Dabei sind auch die Probleme der Zuordnung zusammengehörender Daten von verschiedenen Stationen, sowie der Umrechnung von absoluten Zeiten jeder Station in eine globale Zeit auf der Auswertestation (mit Hilfe von bekannten Laufzeiten zwischen zwei Stationen) zu lösen [198]. Zur Erfassung von Speicherbelegungszeiten wäre neben den oben erwähnten Testanwenderprogrammen ein weiteres Anwenderprogramm notwendig, welches nach dem Empfang eines Auftrages diesen auswertet, eine passende Quittung dazu erzeugt und nach einer definierten Verzögerungszeit zurückschickt.

5.6 Ausblick auf Erweiterungen für das Netzmanagement

Bei der Koexistenz der Protokollprofile MAP und SINEC auf *einem* Netz ist zwar durch das beschriebene MAP–Gateway die Kommunikationsaufgabe als solche gelöst, es bleibt jedoch noch das Problem eines zentralen und komfortablen Netzmanagements offen.

Eine zentrale Manager–Station im MAP–Netz kann zwar sämtliche MAP–Stationen sowie das MAP–Gateway verwalten und von ihnen Alarmmeldungen entgegennehmen, sie ist aber zunächst nicht in der Lage, die Stationen im SINEC–Netz anzusprechen. Eine Abhilfe ist dadurch möglich, daß im MAP–Gateway neben seinem eigenen Agent–Prozeß ein zweiter Agent–Prozeß implementiert wird, welcher aus der Sicht der Manager–Station sämtliche Stationen des SINEC–Netzes repräsentiert. Veränderliche Daten muß sich dieser Agent–Prozeß periodisch direkt von den entsprechenden SINEC–Stationen mit Hilfe geeigneter herstellerspezifischer Protokolle beschaffen. Er muß kritische Situationen erkennen und gegebenenfalls selbständig Alarmmeldungen an die Manager–Station verschicken.

Um die Leistungsfähigkeit des MAP–Gateways durch das Netzmanagement nicht weiter zu reduzieren, kann dieser zweite Agent–Prozeß auch auf einer separaten Station, einem Netzmanagement–Gateway, installiert werden. Eine solche Realisierung wird in [34] beschrieben.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Die vorliegende Arbeit beleuchtet die *Kopplung von Kommunikationsnetzen* von verschiedenen Seiten her. Dabei werden grundlegende Architekturen und Zusammenhänge verdeutlicht und Werkzeuge zur Leistungsuntersuchung zur Verfügung gestellt. Sie wendet sich vor allem an die folgenden drei Zielgruppen:

- Der Architekturteil soll einem *Netzplaner* den momentanen Stand der Technik auf diesem Gebiet vermitteln. Dieser wird dadurch mit dem Aufbau von Netzkoppeleinheiten vertraut und lernt die Vor- und Nachteile unterschiedlicher Netzkoppeleinheiten gegeneinander abzuwägen. Dieses Wissen ist eine wesentliche Voraussetzung, um bei einer geplanten größeren Vernetzung oder Netzerweiterung die strategisch richtigen Entscheidungen treffen zu können.

  Die entwickelten simulativen und mathematischen Analyseprogramme erlauben die leistungsmäßige Untersuchung von vielen konkreten Kopplungsproblemen und den Vergleich unterschiedlicher Alternativen. Es können vor allem zu erwartende Transferzeiten, Stabilitätsgrenzen und der Pufferspeicherbedarf vorhergesagt werden, ohne daß sich der Anwender dieser Programme mit der zugrundeliegenden Theorie befassen muß.

- Die Leistungsuntersuchungen und einige daraus abgeleitete Grundregeln helfen einem *Netzbetreiber*, die veränderlichen Parameter optimal einzustellen. Für den Netzbetrieb werden Möglichkeiten aufgezeigt, wie auch ein gekoppeltes Netz von einer zentralen Stelle aus verwaltet werden kann.

- *Forscher* und *Entwickler* können aus den Verkehrsmodellen ein tieferes Verständnis der internen Vorgänge von Netzkoppeleinheiten und von einigen Details ableiten. Die

dazugehörenden Analyseprogramme sind so flexibel, daß auch unterschiedliche Realisierungsmöglichkeiten einer Netzkoppeleinheit bezüglich ihrer Leistungsfähigkeit miteinander verglichen werden können. Dies gilt insbesondere für Lage und Implementierungsvarianten von Protokollmechanismen und für die Aufteilung der einzelnen Aufgaben auf die zur Verfügung stehenden Prozessoren. Sie helfen ferner bei der verkehrsgerechten Dimensionierung von Netzkoppeleinheiten.

Bei der Implementierung einer Netzkoppeleinheit sollte insbesondere auf die richtige Priorisierung (Senden vor Empfangen oder Quittung vor Auftrag) der Aufgaben *eines* Prozessors geachtet werden. Für jedes Kopplungsproblem ist eine spezielle Netzkoppeleinheit zu entwickeln. Die prototypische Realisierung des MAP–Gateways und dessen Leistungsuntersuchung kann dafür als repräsentatives Beispiel angesehen werden.

Jetzt sollen nocheinmal die wichtigsten Ergebnisse dieser Arbeit zusammengefaßt werden:

- Im methodischen Teil wird gezeigt, daß bei der instationären Simulation der Verlauf der mittleren Transferzeiten abhängig von der Art ihrer Erfassung ist. Bei einer Beobachtung aus der Sicht des Empfängers kann aufgrund einer veränderten Grundgesamtheit die mittlere Transferzeit zunehmen, obwohl die Ankunftsrate reduziert wird. Dieses Phänomen kann so dominant sein, daß es die Auswirkungen der eigentlich zu untersuchenden Mechanismen vollständig überdeckt. Die Beobachtung aus der Sicht des Senders ist deshalb meist sinnvoller.
- Obwohl das Aufteilen einer SDU in mehrere PDUs zur Reduzierung der Auslastung beteiligter Prozessoren so spät wie möglich erfolgen sollte, kann es bei gekoppelten Netzen sinnvoll sein, dies bereits auf einer Schicht mit Ende–zu–Ende–Signifikanz zu tun, insbesondere wenn die Auslastung von Prozessoren und Netzen normalerweise noch nicht allzu groß ist.
- Protokollmechanismen, welche eine künstliche Verzögerung benötigen (beispielsweise Verketten oder Piggybacking) können sich nur dann positiv auf die Leistungsfähigkeit auswirken, wenn sie den Engpaß des Systems signifikant beeinflussen. Ansonsten vergrößern sie in der Regel beispielsweise die Transferzeiten.
- Durch eine adaptive Zeitbegrenzung beim Verketten und Blocken kann man erreichen, daß die mittlere Transferzeit für alle Ankunftsraten minimale Werte annimmt.
- Um den Datenfluß nicht unnötig zu bremsen, sollten die maximalen Fenstergrößen von Flußkontrollen so dimensioniert werden, daß sie dem Produkt aus Stabilitätsgrenze bei sehr großer maximaler Fenstergröße und gerade noch tolerierbarer Quittierungszeit entsprechen. Bei mehreren zulässigen Verbindungen und Richtungen ist dieser Wert entsprechend aufzuteilen.
- Zur Reduzierung des Pufferspeicherbedarfs einer Netzkoppeleinheit und zu ihrer Überlastabwehr sollten nach Möglichkeit vorhandene abschnittsweise Flußkontrollen gekop-

pelt werden. Ist dies nicht möglich, so kann sich die Netzkoppeleinheit durch das Zurückhalten von Quittungen vor einer Überlastung schützen.

- Mit Hilfe eines iterativen mathematischen Algorithmusses kann bei der Analyse einer Bridge ihr zwischen verschiedenen Warteschlangen aufgeteilter begrenzter Pufferspeicher berücksichtigt werden. Damit kann man beispielsweise bei gegebener Pufferspeichergröße die Verlustwahrscheinlichkeit ermitteln oder umgekehrt, bei vorgeschriebener maximaler Verlustwahrscheinlichkeit, den Pufferspeicher entsprechend dimensionieren.
- Anhand eines konkreten Beispiels werden die Schritte zur Leistungsuntersuchung, zum Entwurf und zur Realisierung von Netzkoppeleinheiten aufgezeigt.

6.2 Ausblick

Aufgrund der fortschreitenden Standardisierung werden die meisten heute noch herstellerspezifischen Protokollprofile im Laufe der Zeit zu standardisierten migrieren. Dadurch verändern sich die Schwerpunkte auf dem Gebiet der Netzkopplung. Während bisher das Hauptproblem darin bestand, heterogene Netze mit einem Minimum an Funktionalitätsverlust zu verbinden, wird man in Zukunft sich vorwiegend mit einer Kopplung von kompatiblen Netzen beschäftigen, wobei die Herausforderung das Erreichen möglichst hoher Bearbeitungsgeschwindigkeiten sein wird [208, 209, 210]. Insbesondere ist hier die Kopplung von MANs oder HSLANs über Breitband–ISDN zu nennen [1], wobei bereits bei der jeweiligen Standardisierung darauf geachtet wird, daß keine unnötigen Inkompatibilitäten auftreten werden. Diese Bemühungen zur Steigerung von Bearbeitungsgeschwindigkeiten sind nicht auf das Gebiet der Netzkopplung begrenzt, sondern im Zusammenhang mit dem Ziel effizienterer Implementierungen von Protokollen höherer Schichten zu sehen, um den Engpaß dort zu beseitigen und dem Anwender die Übertragungsgeschwindigkeiten von Hochgeschwindigkeitsnetzen tatsächlich zugänglich zu machen.

Im Bereich der Forschung gibt es seit einiger Zeit auch Bemühungen von Informatikern, Netzkopplungen mit Hilfe verschiedener formaler Methoden zu beschreiben oder Netzkoppeleinheiten damit zu entwerfen [44, 119, 165], was allerdings zur Lösung praktischer Probleme relativ wenig beiträgt.

Ein anderer Aspekt der Netzkopplung, welcher zunehmend an Bedeutung gewinnen wird, ist die zentrale Verwaltung eines gekoppelten Netzes. Trotz mancher Ansätze zur Lösung dieser Problematik sind viele Aspekte erst unbefriedigend umgesetzt. Dazu gehört beispielsweise die Definition von sinnvollen Tests zur Fehlerdiagnose oder die Definition typischer zu verwaltender Objekte mit ihren Attributen in Netzkoppeleinheiten, sowie deren gewinnbringende Ausnützung in einer Manager–Station.

Literaturverzeichnis

[1] S. AGRAWAL, A. KAYE, S. MAHMOUD: *Data–transfer Protocol for a High–Speed FDDI to ATM Bridge.* Proceedings **2nd International Workshop on Protocols for High–Speed Networks**, Palo Alto, USA (November 1990), Pages 1–10.

[2] M. AJMONE MARSAN, F. NERI: *Modelling and Analysis of Communication Protocols Using Petri Nets.* Proceedings **Modelling the Innovation: Communications, Automation and Information Systems**, Rome, Italy (March 1990), North–Holland, Amsterdam, The Netherlands, Pages 9–20.

[3] ANSI X3T9: *Fiber Distributed Data Interface (FDDI) — Hybrid Ring Control (HRC).* ANSI Draft Standard, Revision 6 (May 1990).

[4] J. ARETZ: *Verwirrung bei der Geschwindigkeitsangabe von FDDI–/Ethernet–Brücken.* **DATACOM**, 7. Jahrgang, Nummer 10 (Oktober 1990), Seiten 60–62.

[5] H. VAN AS: *Modellierung und Analyse von Überlast–Abwehrmechanismen in Paketvermittlungsnetzen.* **38. Bericht über verkehrstheoretische Arbeiten**, Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1984).

[6] M. ASAWA, A. KUMAR: *A New Algorithm for Adaptive Flow Control in Interconnected Local Area Networks.* Proceedings **10th International Conference on Computer Communication (ICCC)**, New Delhi, India (November 1990), Pages 484–491.

[7] F. BACKES: *Transparent Bridges for Interconnection of IEEE 802 LANs.* **IEEE Network**, Volume 2, Number 1 (January 1988), Pages 5–9.

[8] E. BALL, N. LINGE, P. KUMMER, R. TASKER: *Local Area Network Bridges.* **Computer Communications**, Volume 11, Number 3 (June 1988), Pages 115–117.

[9] A. BARATZ, J. JAFFE: *Establishing Virtual Circuits in Large Computer Networks.* **Computer Networks and ISDN Systems**, Volume 12, Number 1 (August 1986), Pages 27–37.

[10] F. BASKETT, M. CHANDY, R. MUNTZ, F. PALACIOS: *Open, Closed, and Mixed Networks of Queues with Different Classes of Customers.* **acm Journal of the Association for Computing Machinery**, Volume 22, Number 2 (April 1975), Pages 248–260.

[11] M. BAUDISCH: *Spezifikation und Implementierung des Managementsystems eines MAP–Gateways.* Studienarbeit Nummer 923, Diplomarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (April 1989).

[12] W. BAUERFELD: *A Tutorial on Network Gateways and Interworking of LANs and WANs.* **Computer Networks and ISDN Systems,** Volume 13, Number 3 (March 1987), Pages 187–193.

[13] W. BAUERFELD: *Interconnection Issues between Local Area and Wide Area Networks in DFN.* Proceedings **Eighth International Conference on Computer Communication (ICCC),** Munich (September 1986), Pages 717–722.

[14] W. BAUERFELD: *Zur Architektur von Kopplungen von "Local Area Networks" und "Wide Area Networks" im Deutschen Forschungsnetz DFN.* Studie, Hahn–Meitner–Institut für Kernforschung Berlin GmbH, Regionales Rechenzentrum für Niedersachsen, Technische Universität Berlin (Juli 1984).

[15] W. BAUERFELD, J. HEIGERT: *Gateways: Struktureller Überblick.* **DATACOM,** 4. Jahrgang, Nummer 10 (Oktober 1987), Seiten 100–106 und Nummern 11/12 (November/Dezember 1987), Seiten 103–107.

[16] S. BEDERMANN: *Source Routing.* **Data Communications,** Volume 15, Number 2 (February 1986), Pages 127–128.

[17] E. BEHNKE: *Die Gateway–Problematik in offenen Datennetzen.* **DATACOM,** 2. Jahrgang, Nummer 2 (Februar 1985), Seiten 48–55.

[18] E. BENHAMOU: *Integrating Bridges and Routers in a Large Internetwork.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 65–71.

[19] E. BENHAMOU, J. ESTRIN: *Multilevel Internetworking Gateways: Architecture and Applications.* **IEEE Computer,** Volume 16, Number 9 (September 1983), Pages 27–34.

[20] J. BERNTSEN, J. DAVIN, D. PITT, N. SULLIVAN: *MAC Layer Interconnection of IEEE 802 Local Area Networks.* **Computer Networks and ISDN Systems,** Volume 10, Number 5 (1985), Pages 259–273.

[21] E. BIERSACK: *Annotated Bibliography on Network Interconnection.* **IEEE Journal on Selected Areas in Communications,** Volume 8, Number 1 (January 1990), Pages 22–41.

[22] E. BIERSACK: *A Systematic Approach for Constructing Gateways.* **Computer Networks and ISDN Systems,** Volume 18, Number 2 (February 1990), Pages 79–95.

[23] E. BIERSACK: *Principles of Network Interconnection.* Proceedings **Seventh European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN),** Amsterdam, The Netherlands (June 1989), Pages 37–43.

[24] E. BIERSACK: *Techniken zum Zusammenschluß von Rechnernetzen und deren Anwendung auf Protokolle des Transportsystems.* Dissertation, Technische Universität München, Institut für Informatik (Februar 1988).

[25] G. BOCHMANN, P. MONDAIN–MONVAL: *Design Principles for Communication Gateways.* **IEEE Journal on Selected Areas in Communications,** Volume 8, Number 1 (January 1990), Pages 12–21.

[26] P. BOCKER: *ISDN Das diensteintegrierende digitale Nachrichtennetz.* Dritte Auflage, Springer–Verlag, Berlin (1990).

[27] D. BOGGS, J. SHOCH, E. TAFT, R. METCALFE: *Pup: An Internetwork Architecture.* **IEEE Transactions on Communications,** Volume 28, Number 4 (April 1980), Pages 612–624.

[28] P. BOROWKA: *Netzstrukturierung — Brücken versus Router.* **DATACOM,** 6. Jahrgang, Nummer 6 (Juni 1989), Seiten 76–80 und Nummer 8 (August 1989), Seiten 102–113.

[29] L. BOSACK, C. HEDRICK: *Problems in Large LANs.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 49–56.

[30] M. BOSCH: *Design, Implementation, Modelling and Simulation of a MAP–Gateway for Flexible Manufacturing.* Proceedings **Modelling the Innovation: Communications, Automation and Information Systems,** Rome, Italy (March 1990), North–Holland, Amsterdam, The Netherlands, Pages 259–270.

[31] M. BOSCH: *Konzepte und Komponenten für komplexe Strukturen aus lokalen Netzen.* **Wissenschaftliche Beiträge zur INFORMATIK,** Informatik–Zentrum an der Technische Universität Dresden — 5, Heft 2/1991 (Januar 1991), Seiten 121–130.

[32] M. BOSCH: *Performance Evaluation of Protocol Mechanisms in an Internetworking Environment.* Proceedings **5th International Workshop on Telematics (IWT),** Denver, USA (September 1989), Pages 1–12.

[33] M. BOSCH, O. GIHR, W. KIESEL: *Modulare Simulationstechnik für komplexe Anwendungsprotokolle in der Fertigungsautomatisierung.* Tagungsband **Prozeßrechensysteme,** Stuttgart (März 1988), Informatik–Fachberichte 167, Springer–Verlag, Berlin, Seiten 193–204.

[34] M. BOSCH, G. RÖSSLER, W. SCHOLLENBERGER: *Network Management in Heterogeneous Networks for Factory Automation.* Proceedings **Information Network and Data Communication, III (INDC),** Lillehammer, Norway (March 1990), North–Holland, Amsterdam, The Netherlands, Pages 67–79.

[35] M. BOSCH, G. SCHMID: *Generic Petri Net Models of Protocol Mechanisms in Communication Systems.* **Computer Communications,** Volume 14, Number 3 (April 1991), Pages 143–156.

[36] R. Boulé, J. Moy: *Inside Routers: A Technology Guide for Network Builders.* **Data Communications,** Volume 18, Number 12 (September 1989), Pages 53–66.

[37] D. Brunn: *OSI–Anwendungen auf ISDN.* Tagungsband **Das ISDN in der Einführung,** Berlin (Februar 1988), ITG–Fachbericht 100, VDE–Verlag, Berlin, Seiten 409–433.

[38] F. Burg, N. Iorio: *Networking of Networks: Interworking According to OSI.* **IEEE Journal on Selected Areas in Communications,** Volume 7, Number 7 (September 1989), Pages 1131–1142.

[39] P. Burke: *The Output of a Queuing System.* **Journal of the Operations Research Society of America,** Volume 4, Number 6 (December 1956), Pages 699–704.

[40] W. Bux: *Modeling Token Ring Networks — A Survey.* Tagungsband **Messung, Modellierung und Bewertung von Rechensystemen,** Erlangen (September/Oktober 1987), Informatik–Fachberichte 154, Springer–Verlag, Berlin, Seiten 192–221.

[41] W. Bux, D. Grillo: *Flow Control in Local–Area Networks of Interconnected Token Rings.* **IEEE Transactions on Communications,** Volume 33, Number 10 (October 1985), Pages 1058–1066.

[42] L. Caffrey: *EPHOS: Towards a European GOSIP.* **Computer Networks and ISDN Systems,** Volume 19, Numbers 3–5 (November 1990), Pages 265–269.

[43] P. Callahan, B. Bradley: *New Token Ring versus Ethernet: Counterpoint.* **Data Communications,** Volume 18, Number 1 (January 1989), Pages 127–134.

[44] K. Calvert, S. Lam: *Formal Methods for Protocol Conversion.* **IEEE Journal on Selected Areas in Communications,** Volume 8, Number 1 (January 1990), Pages 127–142.

[45] V. Catania, A. Puliafito, L. Vita: *Availability and Performability Assessment in LAN Interconnection.* Proceedings **Ninth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),** San Francisco, USA (June 1990), Pages 1181–1187.

[46] CCITT: *Integrated Services Digital Network (ISDN) — Internetwork Interfaces.* **Blue Book,** Volume III, Fascicle III.9, Recommendations I.500–I.560 (November 1988), Pages 1–58.

[47] CCITT: *Terminal Adaptor Functionalities.* **Blue Book,** Volume VIII, Fascicle VIII.2, Recommendation X.31 (November 1988), Pages 449–480.

[48] CCITT: *Data Communication Networks — Transmission, Signalling and Switching.* **Blue Book,** Volume VIII, Fascicle VIII.3, Recommendations X.75–X.82 (November 1988), Pages 152–294.

[49] CCITT: *Data Communication Networks — Interworking between Networks, Mobile Data Transmission Systems, Internetwork Management.* **Blue Book,** Volume VIII, Fascicle VIII.6, Recommendations X.300–X.370 (November 1988).

[50] Y. CHENG, T. ROBERTAZZI: *Annotated Bibliography of Local Communication System Interconnection.* **IEEE Journal on Selected Areas in Communications,** Volume 5, Number 9 (December 1987), Pages 1492–1499.

[51] G. CHIOLA: *GreatSPN Users' Manual.* Version 1.3, Università di Torino, Dipartimento di Informatica, Italy (August 1987).

[52] L. CHISVIN, J. DUCKWORTH: *Content-Addresseable and Associative Memory: Alternatives to the Ubiquitous RAM.* **IEEE Computer,** Volume 22, Number 7 (July 1989), Pages 51–64.

[53] I. CHLAMTAC: *A Programmable VLSI Controller for Standard and Prioritized Ethernet Local Networks.* **Local Area & Multiple Access Networks,** R. Pickholtz, Computer Science Press, Rockville, USA (1986), Chapter 4, Pages 69–91.

[54] P. CHRIST, B. LORTZ: *The Baden-Württemberg Research Network: Interconnecting HSLANs by 140 Mbps PTT Links.* Proceedings **High Speed Local Area Networks, II,** Liège, Belgium (April 1988), Pages 267–271.

[55] P. CHYLLA, H.-G. HEGERING: *LAN-Gateways* und *Kopplungen lokaler Netze.* **Ethernet-LANs,** DATACOM-Verlag, Bergheim (1987), Kapitel IV, Seiten 100–111 und Kapitel V, Abschnitt 5, Seiten 133–143.

[56] L. CLYNE: *LAN/WAN Interworking.* **Computer Networks and ISDN Systems,** Volume 16, Numbers 1/2 (1988), Pages 34–39.

[57] R. COLE: *Experience and Analysis of Network Interconnection.* **IEEE Journal on Selected Areas in Communications,** Volume 8, Number 1 (January 1990), Pages 49–56.

[58] S. DEERING, D. CHERITON: *Multicast Routing in Datagram Internetworks and Extended LANs.* **acm Transactions on Computer Systems,** Volume 8, Number 2 (May 1990), Pages 85–110.

[59] R. DIXON, D. PITT: *Addressing, Bridging, and Source Routing.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 25–32.

[60] B. DRESCHER: *Schnelle Gateways zur Verbindung heterogener Netze.* **DATACOM,** 5. Jahrgang, Nummer 11 (November 1988), Seiten 80–88.

[61] C. ERSOY, S. PANWAR, R. DALIAS, D. SEGAL: *Transient Phenomena in Bridged Local Area Networks.* Proceedings **IEEE Global Telecommunications Conference & Exhibition (GLOBECOM),** San Diego, USA (December 1990), Pages 1405–1409.

[62] D. ESTRIN: *Interconnection Protocols for Interorganization Networks.* **IEEE Journal on Selected Areas in Communications,** Volume 5, Number 9 (December 1987), Pages 1480–1491.

[63] FERNMELDETECHNISCHES ZENTRALAMT: *ATM — Ein universelles Übermittlungsverfahren.* **FTZ–Nachrichten,** Sonderheft ATM (Juli 1988).

[64] W. FISCHER: *Modellierung und Analyse des Netzzugangssystems für das dienstintegrierende Digitalnetz ISDN.* **45. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1989).

[65] D. GANTENBEIN, W. STOLL, M. ZIEHER: *OSI Internetworking in a Heterogeneous LAN and WAN Environment.* Proceedings **Fifth European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN),** Basel, Switzerland (June 1987), Pages 300–306.

[66] K. GEE: *Local Area Network Gateways.* National Computing Centre (NCC) Publications, Manchester, England (1983).

[67] M. GERLA, L. KLEINROCK: *Congestion Control in Interconnected LANs.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 72–76.

[68] O. GIHR: *Analyse datenflußregulierter Verbindungskonzepte in verteilten Systemen mit mehrschichtiger Protokollarchitektur.* **48. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1990).

[69] E.–H. GÖLDNER: *Ein Ringsystem mit integrierter Durchschalte– und Paketvermittlung.* **44. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1988).

[70] G. GOLDACKER, T. LUCKENBACH, R. RUPPELT, R. SCHMIDT, M. VOGELSÄNGER: *BERGATE: Ein Transitsystem zur Kopplung Lokaler Netze über Breitband–ISDN.* Tagungsband **Kommunikation in verteilten Systemen,** Stuttgart (Februar 1989), Informatik–Fachberichte 205, Springer–Verlag, Berlin, Seiten 775–788.

[71] W. GORA: *Who is Who in der Kommunikationswelt.* **DATACOM,** 7. Jahrgang, Nummer 10 (1990), Seiten 132–142.

[72] W. GORA, R. SPEYERER: *ASN.1.* **Informatik Spektrum,** Band 11 (1988), Seiten 207–209.

[73] P. GREEN: *Protocol Conversion.* **IEEE Transactions on Communications,** Volume 34, Number 3 (March 1986), Pages 257–268.

[74] D. GREENFIELD: *An End to a Bridging Feud?.* **Data Communications International,** Volume 19, Number 6 (May 1990), Pages 33–34.

[75] D. GROSS, C. HARRIS: *Some Additional Results.* **Fundamentals of Queueing Theory,** John Wiley & Sons, New York, USA (1974), Section 5.1.8, Pages 251–254.

[76] C. HAMNER: *Source Routing Bridge Implementation.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 33–36.

[77] P. HARRISON: *Performance Prediction of a Flow Control System Using an Analytic Model.* Proceedings **Local Computer Networks,** Florence, Italy (April 1982), Pages 439–458.

[78] J. HART: *Extending the IEEE 802.1 MAC Bridge Standard to Remote Bridges.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 10–15.

[79] W. HAWE, M. KEMPF, A. KIRBY: *The Extended Local Area Network Architecture and LANBridge 100.* **Digital Technical Journal,** Number 3 (September 1986), Pages 54–72.

[80] M. HEIN, A. RENDEL: *Auswahlkriterien von Bridges.* **DATACOM,** 5. Jahrgang, Nummer 10 (Oktober 1988), Seiten 100–104.

[81] P. HENQUET: *Design of Gateways Interconnecting HSLANs: Performance Issues in Internal Gateway Communication.* Proceedings **High Speed Local Area Networks, II,** Liège, Belgium (April 1988), Pages 139–166.

[82] E. HINDIN: *IBM 8209 LAN Bridge Links Ethernet to Token Ring.* **Data Communications,** Volume 19, Number 4 (March 1990), Pages 75–81.

[83] E. HINDIN: *Joining LANs through the Mainframe Connection.* **Data Communications,** Volume 19, Number 8 (June 1990), Pages 18–24.

[84] HIRATA, S. MATSUI, T. YOKOYAMA, M. MIZUTANI, M. TERADA: *A High Speed Protocol Processor to Boost Gateway Performance.* Proceedings **IEEE Global Telecommunications Conference & Exhibition (GLOBECOM),** San Diego, USA (December 1990), Pages 1426–1430.

[85] K. HORN, F. KAUFFELS: *LAN–LAN–Kopplung über Satelliten.* **DATACOM,** 2. Jahrgang, Nummer 2 (Februar 1985), Seiten 44–47.

[86] M. HUBER: *Ein Netzknotenkonzept für integrierte Durchschalte– und Paketvermittlung.* **49. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1990).

[87] O. IBE, X. CHENG: *Analysis of Interconnected Systems of Token Ring Networks.* **Computer Communications,** Volume 13, Number 3 (April 1990), Pages 136–142.

[88] IEEE 802.1: *MAC Bridges.* IEEE Draft Standard, Part D, IEEE Computer Society (July 1989).

[89] IEEE 802.3c: *Repeater Unit for 10 Mb/s Baseband Networks.* ANSI/IEEE Standard, Supplement to ANSI/IEEE Standard 802.3, Section 9, IEEE Computer Society (1988).

[90] IEEE 802.5: *Enhancement for Multi–Ring Networks.* Draft Addendum to the ANSI/-IEEE Standard, IEEE Computer Society (September 1987).

[91] IEEE 802.6: *Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN).* IEEE Draft Standard, IEEE Computer Society (October 1990).

[92] M. ILYAS, H. MOUFTAH: *End–to–End Flow Control in Interconnected Local Area Ring Networks.* Proceedings **IEEE International Conference on Communications (ICC)**, Boston, USA (June 1989), Pages 1485–1489.

[93] Y. IP: *A Distributed–End–System Interworking Unit.* Proceedings **Fifth European Fibre Optic Communications and Local Area Networks Exposition (EFOC/-LAN)**, Basel, Switzerland (June 1987), Pages 307–311.

[94] Y. IP: *General Working Principles of a Connectionless LAN/WAN Router.* Proceedings **Seventh European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN)**, Amsterdam, The Netherlands (June 1989), Pages 50–54.

[95] D. IRVIN: *A Queueing Model for Local Area Network Bridges.* **Performance Evaluation Review**, Volume 16, Numbers 2–4 (February 1989), Pages 48–56.

[96] ISO 7498: *Information Processing Systems — Open Systems Interconnection — Basic Reference Model.* International Standard (1984).

[97] ISO 7498–3: *Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 3: Naming and Addressing.* International Standard (1989).

[98] ISO 7498–4: *Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 4: OSI Management Framework.* International Standard (1989).

[99] ISO 8348: *Information Processing Systems — Data Communications — Network Service Definition — Addendum 2: Network Layer Addressing.* International Standard (1988).

[100] ISO 8571: *Information Processing Systems — Open Systems Interconnection — File Transfer, Access and Management.* International Standard (1988).

[101] ISO 8648: *Information Processing Systems — Open Systems Interconnection — Internal Organization of the Network Layer.* International Standard (February 1988).

[102] ISO 8649: *Information Processing Systems — Open Systems Interconnection — Service Definition for the Association Control Service Element.* Draft International Standard (April 1988).

[103] ISO 8650: *Information Processing Systems — Open Systems Interconnection — Protocol Specification for the Association Control Service Element.* Draft International Standard (April 1988).

[104] ISO DIS 8802–2: *Information Processing Systems — Data Communications — Local Area Networks — Part 2: Logical Link Control.* International Standard (1988).

[105] ISO 8802–2: *Logical Link Control (LLC) Flow Control Techniques for Multi–Segment Networks.* Proposal for a Draft Addendum (September 1986).

[106] ISO 8802–3: *Information Processing Systems — Data Communications — Local Area Networks — Part 3: CSMA/CD Access Method and Physical Layer Specifications.* International Standard (1989).

[107] ISO 8802–4: *Information Processing Systems — Data Communications — Local Area Networks — Part 4: Token–Passing Bus Access Method and Physical Layer Specifications.* Draft International Standard (1988).

[108] ISO 8802–5: *Information Processing Systems — Data Communications — Local Area Networks — Part 5: Token Ring Access Method and Physical Layer Specifications.* Draft Proposal (1985).

[109] ISO 8824: *Information Processing Systems — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).* International Standard (December 1987).

[110] ISO 8825: *Information Processing Systems — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).* International Standard (November 1987).

[111] ISO 9506: *Information Processing Systems — Open Systems Interconnection — Manufacturing Message Specification.* International Standard (January 1989).

[112] ISO 9542: *Information Processing Systems — Telecommmunications and Information Exchange between Systems — End System to Intermediate System Routeing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless–Mode Network Service (ISO 8473).* International Standard (August 1988).

[113] ISO 9595–2: *Information Processing Systems — Open Systems Interconnection — Management Information Service Definition — Part 2: Common Management Information Service.* Draft International Standard (June 1988).

[114] ISO 9596–2: *Information Processing Systems — Open Systems Interconnection — Management Information Protocol Specification — Part 2: Common Management Information Protocol.* Draft International Standard (September 1988).

[115] ISO 10028.2: *Information Processing Systems — Data Communications — Definition of the Relaying Functions of a Network Layer Intermediate System.* Draft Proposal (August 1989).

[116] ISO 10029: *Information Technology — Telecommunications and Information Exchange between Systems — Operation of an X.25 Interworking Unit.* Technical Report

(March 1989).

[117] ISO 10589: *Intermediate System to Intermediate System Intra–Domain Routeing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionless–Mode Network Service (ISO 8473).* Draft Proposal (February 1990).

[118] J. ISRAEL, A. WEISSBERGER: *Communicating between Heterogeneous Networks.* **Data Communications,** Volume 16, Number 3 (March 1987), Pages 215–235.

[119] G. JUANOLE, C. FAURE: *On Gateway for Internetworking through ISDN: Architecture and Formal Modelling with Petri Nets.* Proceedings **Eighth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),** Ottawa, Canada (April 1989), Pages 458–467.

[120] M. KATZ, G. BIWER, K. BENDER: *Die PROFIBUS–Anwendungsschicht.* **Automatisierungstechnische Praxis (atp),** Volume 31, Nummer 12 (Dezember 1989), Seiten 588–597.

[121] F. KAUFFELS: *IBM–Bridge zwischen Token Ring und Ethernet.* **DATACOM,** 6. Jahrgang, Nummer 11 (November 1989), Seiten 22–23.

[122] F. KAUFFELS: *Kompatibilität von Token–Ring–Netzen: Anspruch und Wahrheit.* **DATACOM,** 7. Jahrgang, Nummer 3 (März 1990), Seiten 134–141 und Nummer 4 (April 1990), Seiten 98–104.

[123] C. KAWA, G. BOCHMANN: *Hierarchical Multi–Network Interconnection Using Public Data Networks.* Proceedings **Sixth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),** San Francisco, USA (March/-April 1987), Pages 426–435.

[124] D. KENDALL: *Stochastic Processes Occuring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain.* **The Annals of Mathematical Statistics,** Volume 24, Number 2 (June 1953), Pages 338–354.

[125] S. KING: *Multiport Bridges.* **Data Communications International,** Volume 19, Number 10 (August 1990), Pages 58–65.

[126] U. KÖRNER, S. FDIDA, H. PERROS, G. SHAPIRO: *End to End Delays in a Catenet Environment.* Proceedings **Third International Conference on Data Communication Systems and their Performance,** Rio de Janeiro, Brazil (June 1987), Pages 453–464.

[127] R. KOGEL: *Implementierung von Umsetzungsszenarien eines MAP–Gateways zwischen MMS und SINEC AP.* Studienarbeit Nummer 871, 2. Semesterarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (Juli 1988).

[128] G. KOSHY: *Understanding Multiple LANs: The Why and How of Linking Up.* **Data Communications,** Volume 15, Number 5 (May 1986), Pages 221–227.

[129] P. KÜHN: *Analyse zufallsabhängiger Prozesse in Systemen zur Nachrichtenvermittlung und Nachrichtenverarbeitung.* **30. Bericht über verkehrstheoretische Arbeiten,** Habilitationsschrift, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1981).

[130] P. KÜHN: *Multiqueue Systems with Nonexhaustive Cyclic Service.* **Bell Systems Technical Journal,** Volume 58, Number 3 (1979), Pages 671–699.

[131] P. KÜHN: *Nachrichtenvermittlung I/II.* Vorlesung an der Universität Stuttgart.

[132] P. KÜHN: *Wartezeitprobleme der Daten- und Nachrichtenverkehrstheorie.* Vorlesung an der Universität Stutgart.

[133] P. KUMMER, R. TASKER, N. LINGE, E. BALL: *A Protocol-less Scheme for Bridging between IEEE 802 Local Area Networks.* **Computer Networks and ISDN Systems,** Volume 12, Number 5 (May 1987), Pages 81–87.

[134] R. KURUPPILLAI, N. BENGTSON: *Performance Analysis in Local Area Networks of Interconnected Token Rings.* **Computer Communications,** Volume 11, Number 2 (April 1988), Pages 59–64.

[135] C. KWOK, B. MUKHERJEE: *Cut-Through Bridging for CSMA/CD Local Area Networks.* **IEEE Transactions on Communications,** Volume 38, Number 7 (July 1990), Pages 938–942.

[136] C. KWOK, B. MUKHERJEE: *On Transparent Bridging of CSMA/CD Networks.* Proceedings **IEEE Global Telecommunications Conference & Exhibition (GLOBECOM),** Dallas, USA (November 1989), Pages 185–190.

[137] H. LACKNER: *LANs in den Flegeljahren oder NESH zähmt LANs.* **Praxis der Informationsverarbeitung und Kommunikation (PIK),** 10. Jahrgang, Nummer 4 (Oktober–Dezember 1987), Seiten 255–259.

[138] W. LAI: *Packet Mode Services: From X.25 to Frame Relaying.* **Computer Communications,** Volume 12, Number 1 (February 1989), Pages 10–16.

[139] J. LAMONT, M. HUI: *Some Experience with LAN Interconnection via Frame Relaying.* **IEEE Network Magazine,** Volume 3, Number 5 (September 1989), Pages 21–24.

[140] M. LANG: *Analyse des Pufferspeichers in einer Netzkoppeleinheit unter Berücksichtigung unterschiedlicher Randbedingungen.* Studienarbeit Nummer 963, Diplomarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (Dezember 1989).

[141] M. LANG, M. BOSCH: *Performance Analysis of Finite Capacity Polling Systems with Limited-M Service.* Proceedings **13th International Teletraffic Congress (ITC),** Copenhagen, Denmark (June 1991), Volume 14, Pages 731–735.

[142] A. LATURNER: *Verbindungssteuerung in einer Netzkoppeleinheit bei der Kopplung unterschiedlicher Netztypen.* Studienarbeit Nummer 986, 1. Semesterarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (März 1990).

[143] T. LEE: *M/G/1/N Queue with Vacation Time and Exhaustive Service Discipline.* **Operations Research,** Volume 32, Number 4 (July/August 1984), Pages 774–784.

[144] T. LEE: *M/G/1/N Queue with Vacation Time and Limited Service Discipline.* **Performance Evaluation,** Volume 9 (1988/89), Pages 181–190.

[145] N. LINGE, E. BALL, R. TASKER, P. KUMMER: *A Bridge Protocol for Creating a Spanning Tree Topology within an IEEE 802 Extended LAN Environment.* **Computer Networks and ISDN Systems,** Volume 13 (1987), Pages 323–332.

[146] J. LITTLE: *A Proof for the Queuing Formula: $L = \lambda W$.* **Operations Research,** Volume 9, Number 3 (May/June 1961), Pages 383–387.

[147] T. LUCKENBACH: *BERGATE — Connecting VMEbus Systems to MAP, TOP, and Broadband ISDN.* Proceedings **2nd International Workshop on Protocols for High–Speed Networks,** Palo Alto, USA (November 1990), Pages 1–15.

[148] S. LÜSCHOW: *Bridges im Forschungsnetz von Baden–Württemberg.* **DATACOM,** 6. Jahrgang, Nummer 2 (Februar 1989), Seiten 42–44.

[149] MAP: *MAP Network Architecture* and *General Motors Multi–Vendor Gateway Specification.* **MAP 3.0 Implementation Release,** General Motors Technical Center (April 1987), Chapter 3 and Appendix 10.

[150] P. MARTINI: *High Speed Bridges for High Speed Local Area Networks, Packets per Second vs. Bits per Second.* Proceedings **Eighth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),** Ottawa, Canada (April 1989), Pages 474–483.

[151] P. MARTINI, T. WELZEL: *LAN Interconnection by Token Rings: A Performance Analysis.* Proceedings **Fifth European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN),** Basel, Switzerland (June 1987), Pages 281–285.

[152] J. MCQUILLAN: *Routers as Building Blocks for Robust Internetworks.* **Data Communications,** Volume 18, Number 12 (September 1989), Pages 28–33.

[153] M. MEHMET–ALI, B. GRELA–M'POKO, J. HAYES: *The Performance of Interconnected Ring Networks with Priority.* Proceedings **IEEE Global Telecommunications Conference & Exhibition (GLOBECOM),** Hollywood, USA (November/December 1988), Pages 1803–1807.

[154] L. MERAKOS, G. EXLEY, C. BISDIKIAN: *Interconnection of CSMA Local Area Networks: The Frequency Division Approach.* **IEEE Transactions on Communica-**

tions, Volume 35, Number 7 (July 1987), Pages 730–738.

[155] L. MERAKOS, H. XIE: *Interconnection of CSMA/CD LANs via an N–Port Bridge.* Proceedings **Eighth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)**, Ottawa, Canada (April 1989), Pages 28–37.

[156] E. MIER: *Adding to Your Net Worth with T1–to–LAN Devices.* **Data Communications International**, Volume 18, Number 11 (September 1989), Pages 89–108.

[157] D. VAN MIEROP: *System Finex: A First Experience with FDDI Technology.* Proceedings **Sixth European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN)**, Amsterdam, The Netherlands (June/July 1988), Pages 301–305.

[158] M. MURATA, H. TAKAGI: *Performance of Token Ring Networks with a Finite Capacity Bridge.* **TRL Research Report**, IBM Tokyo Research Laboratory (May 1987).

[159] M. NASSEHI: *Window Flow Control in Frame–Relay Networks.* Proceedings **IEEE Global Telecommunications Conference & Exhibition (GLOBECOM)**, Hollywood, USA (November/December 1988), Pages 1784–1790.

[160] I. NIEMEGEERS, M. ZAFIROVIC–VUKOTIC: *HSLANs for Communication within a Gateway: A Performance Evaluation.* Proceedings **High Speed Local Area Networks, II**, Liège, Belgium (April 1988), Pages 167–185.

[161] T. NISHIDA, M. MURATA, H. MIYAHARA, K.TAKASHIMA: *Congestion Control in Interconnected Local Area Networks.* **Local Area & Multiple Access Networks**, R. Pickholtz, Computer Science Press, Rockville, USA (1986), Chapter 6, Pages 107–136.

[162] T. NISHIDA, M. MURATA, H. MIYAHARA, K. TAKASHIMA: *Dynamic Congestion Control in Interconnected Local Area Networks.* Proceedings **11th International Teletraffic Congress (ITC)**, Kyoto, Japan (September 1985), Pages 4.1A-3-1–4.1A-3-7.

[163] R. NITZAN, P. GROSS: *The Role of the U.S. GOSIP.* **Computer Networks and ISDN Systems**, Volume 19, Numbers 3–5 (November 1990), Pages 270–274.

[164] J. NONNAST: *Untersuchung von Gateways zwischen verschiedenen Protokollarchitekturen in der Fertigungsautomatisierung.* Studienarbeit Nummer 850, 2. Semesterarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (März 1988).

[165] K. OKUMURA: *A Formal Protocol Conversion Method.* Proceedings **acm Special Interest Group Data Communication (SIGCOMM), Symposium on Communications Architectures and Protocols**, Stowe, USA (August 1986), Pages 30–37.

[166] G. PARULKAR, J. TURNER: *Towards a Framework for High–Speed Communication in a Heterogeneous Networking Environment.* **IEEE Network**, Volume 4, Number 2

(March 1990), Pages 19–27.

[167] A. PATEL, V. RYAN: *Introduction to Names, Addresses and Routes in an OSI Environment.* **Computer Communications,** Volume 13, Number 1 (February 1990), Pages 27–36.

[168] R. PERLMAN, A. HARVEY, G. VARGHESE: *Choosing the Appropriate ISO Layer for LAN Interconnection.* **IEEE Network,** Volume 2, Number 1 (January 1988), Pages 81–86.

[169] K. PETER: *Kommunikation über lokale Netze.* **Siemens Components,** 27. Jahrgang, Nummer 4 (Juli/August 1989), Seiten 135–140 und Nummer 5 (September/Oktober 1989), Seiten 194–198.

[170] D. PITT, J. WINKLER: *Table–Free Bridging.* **IEEE Journal on Selected Areas in Communications,** Volume 5, Number 9 (December 1987), Pages 1454–1462.

[171] G. POO, W. ANG: *OSI Addressing Strategies for Interconnected LANs.* **Computer Communications,** Volume 13, Number 5 (June 1990), Pages 290–297.

[172] G. POO, W. ANG: *OSI Protocol Choices for LAN Environments.* **Computer Communications,** Volume 13, Number 1 (January/February 1990), Pages 17–26.

[173] G. POO: *Performance Measurement of Interconnected CSMA/CD LANs.* **Computer Communications,** Volume 12, Number 1 (February 1989), Pages 3–9.

[174] S. RADACK: *US Government Moves toward Implementing OSI Standards.* **IEEE Computer,** Volume 21, Number 6 (June 1988), Pages 82–83.

[175] T. RAITH: *Leistungsuntersuchung von Multi–Bus–Verbindungsnetzwerken in lose gekoppelten Systemen.* **43. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1987).

[176] H. REBMANN: *Simulation von Kopplungen zwischen Netzen mit unterschiedlichen Medienzugangsverfahren.* Studienarbeit Nummer 998, 2. Semesterarbeit, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (Juni 1990).

[177] J. RICKERT: *Evaluating MAC–Layer Bridges — Beyond Filtering and Forwarding.* **Data Communications International,** Volume 19, Number 6 (May 1990), Pages 69–72.

[178] K. SATO: *Address Filtering Performance of Slotted Ring Bridge.* Proceedings **Seventh European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN),** Amsterdam, The Netherlands (June 1989), Pages 55–59.

[179] K. SAUER: *Integration von Sprach– und Datenkommunikation in Lokalen Netzen.* **50. Bericht über verkehrstheoretische Arbeiten,** Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1990).

[180] A. SCHRADER: *Brouter mit Hochschulerfahrung.* **DATACOM**, 7. Jahrgang, Nummer 11 (November 1990), Seiten 4–10.

[181] R. SEIFERT: *Have Remote Bridge Vendors Made a Big Blunder?.* **Data Communications International**, Volume 19, Number 5 (April 1990), Pages 27–28.

[182] W. SEIFERT: *Bridges and Routers.* **IEEE Network**, Volume 2, Number 1 (January 1988), Pages 57–64.

[183] R. SHANI: *Fibre Optic IEEE 802.3 CSMA/CD Networks.* Proceedings **Ninth International Conference on Computer Communication (ICCC)**, Tel Aviv, Israel (October/November 1988), Pages 408–414.

[184] J. SHOCH: *Inter–Network Naming, Addressing, and Routing.* Proceedings **IEEE Computer Conference (COMPCON)**, Washington, USA (Fall 1978), Pages 72–79.

[185] J. SHOCH: *Packet Fragmentation in Inter–Network Protocols.* **Computer Networks**, Volume 3, Number 1 (1979), Pages 3–8.

[186] M. SOHA, R. PERLMAN: *Comparison of Two LAN Bridge Approaches.* **IEEE Network**, Volume 2, Number 1 (January 1988), Pages 37–43.

[187] W. STALLINGS: *Internetworking.* **Local Networks**, Mcmillan, New York, USA (1987), Chapter 11, Pages 337–367.

[188] C. SUNSHINE: *Interconnection of Computer Networks.* **Computer Networks**, Volume 1, Number 3 (1977), Pages 175–195.

[189] C. SUNSHINE: *Network Interconnection and Gateways.* **IEEE Journal on Selected Areas in Communications**, Volume 8, Number 1 (January 1990), Pages 4–11.

[190] L. SVOBODOVA: *Implementing OSI Systems.* **IEEE Journal on Selected Areas in Communications**, Volume 7, Number 7 (September 1989), Pages 1115–1130.

[191] L. SVOBODOVA, P. JANSON, E. MUMPRECHT: *Heterogeneity and OSI.* **IEEE Journal on Selected Areas in Communications**, Volume 8, Number 1 (January 1990), Pages 67–79.

[192] A. TANENBAUM: *Internetworking.* **Computer Networks**, Second Edition, Prentice–Hall International, Hertfordshire, England (1990), Section 5.4, Pages 320–349.

[193] P. TOMSU: *Transparent Wide Area 802.3 Access via an ISDN PABX.* Proceedings **Seventh European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN)**, Amsterdam, The Netherlands (June 1989), Pages 26–31.

[194] P. TRAN–GIA: *Überlastprobleme in rechnergesteuerten Fernsprechvermittlungssystemen — Modellbildung und Analyse.* **36. Bericht über verkehrstheoretische Arbeiten**, Dissertation, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung (1982).

[195] O. ULUSOY, M. BARAY: *Window-based Congestion Control Mechanism in Interconnected Computer Networks: A Simulation Study.* Proceedings **10th International Conference on Computer Communication (ICCC)**, New Delhi, India (November 1990), Pages 472–477.

[196] G. VARGHESE, R. PERLMAN: *Transparent Interconnection of Incompatible Local Area Networks Using Bridges.* **IEEE Journal on Selected Areas in Communications**, Volume 8, Number 1 (January 1990), Pages 42–48.

[197] U. WARRIER, C. SUNSHINE: *A Platform for Heterogeneous Interconnection Network Management.* **IEEE Journal on Selected Areas in Communications**, Volume 8, Number 1 (January 1990), Pages 119–126.

[198] M. WEIXLER: *Distributed Measurement System for Protocols and Applications in ISO 8802/3 LANs.* Proceedings **Fourth International Conference on Data Communication Systems and their Performance**, Barcelona, Spain (June 1990), Pages 448–455.

[199] T. WELZEL: *Performance Analysis of Token Rings as High Speed Backbone Networks.* Proceedings **Ninth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)**, San Francisco, USA (June 1990), Pages 23–29.

[200] R. WOLFF: *Poisson Arrivals See Time Averages.* **Operations Research**, Volume 30, Number 2 (March/April 1982), Pages 223–231.

[201] G. WOOD: *Current Fieldbus Activities.* **Computer Communications**, Volume 11, Number 3 (June 1988), Pages 118–123.

[202] H. WORTMANN: *Sachstand der Festlegung "Einheitlicher Höherer Kommunikationsprotokolle" (EHKP) als nationale Zwischenlösung.* Tagungsband **Kommunikation in verteilten Systemen**, Berlin (Januar 1981), Informatik–Fachberichte 40, Springer–Verlag, Berlin, Seiten 126–140.

[203] H. XIE, L. MERAKOS: *Performance Evaluation of a System of Interconnected CSMA/CD LANs via an N-Port Bridge.* Proceedings **IEEE International Conference on Communications (ICC)**, Boston, USA (June 1989), Pages 640–645.

[204] M. YAMAMOTO, I. AKIYOSHI, H. NAKANISHI, H. SANADA, Y. TEZUKA: *Performance of Window Flow Control Scheme for Interconnected Packet Networks.* Proceedings **IEEE International Conference on Communications (ICC)**, Philadelphia, USA (June 1988), Pages 1162–1166.

[205] R. ZAMBRE: *Design Considerations for Extended Local Area Networks.* Proceedings **10th International Conference on Computer Communication (ICCC)**, New Delhi, India (November 1990), Pages 432–441.

[206] L. ZHANG: *Comparison of Two Bridge Routing Approaches.* **IEEE Network**, Volume 2, Number 1 (January 1988), Pages 44–48.

[207] X. ZHANG, R. DENG: *Gateway Design for LAN Interconnection via ISDN.* **Computer Networks and ISDN Systems**, Volume 19, Number 9 (1990), Pages 43–51.

[208] M. ZITTERBART: *A Multiprocessor Architecture for High Speed Network Interconnections.* Proceedings **Eighth Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)**, Ottawa, Canada (April 1989), Pages 212–218.

[209] M. ZITTERBART: *A Parallel Architecture for Transport Systems and Gateways.* Tagungsband **Kommunikation in verteilten Systemen**, Stuttgart (Februar 1989), Informatik–Fachberichte 205, Springer–Verlag, Berlin, Seiten 744–757.

[210] M. ZITTERBART: *Distributed Gateways based on a Transputer Network.* Proceedings **Seventh European Fibre Optic Communications and Local Area Networks Exposition (EFOC/LAN)**, Amsterdam, The Netherlands (June 1989), Pages 44–49.

[211] —: *Bridges and Routers.* **IEEE Network**, M .Gerla, L. Green, R. Rutledge, Volume 2, Number 1 (January 1988).

[212] — : *Heterogeneous Computer Networks Interconnection.* **IEEE Journal on Selected Areas in Communications**, P. Green, K. Naemura, R. Williamson, Volume 8, Number 1 (January 1990).

[213] — : *Interconnection of Local Area Networks.* **IEEE Journal on Selected Areas in Communications**, W. Bux, D. Grillo, N. Maxemchuk, Volume 5, Number 9 (December 1987).

[214] — : *Network Interconnection and Protocol Conversion.* P. Green, IEEE Press, New York, USA (1988).