



**Universität Stuttgart**

Institut für Nachrichtenvermittlung und Datenverarbeitung

Prof. Dr.-Ing. P. Kühn

**48. Bericht über verkehrstheoretische Arbeiten**

**ANALYSE DATENFLUSSREGULIERTER VERBINDUNGS-  
KONZEPTE IN VERTEILTEN SYSTEMEN  
MIT MEHRSCICHTIGER PROTOKOLLARCHITEKTUR**

von

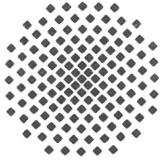
**Ottmar Gühr**

1990

© 1991 Institut für Nachrichtenvermittlung und Datenverarbeitung Universität Stuttgart

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

ISBN 3-922403-58-1



**University of Stuttgart**

Institute of Communications Switching and Data Technics

Prof. Dr.-Ing. P. Kühn

**48th Report on Studies in Congestion Theory**

**ANALYSIS OF FLOW CONTROLLED CONNECTIONS  
IN DISTRIBUTED SYSTEMS  
WITH MULTI-LAYERED PROTOCOL ARCHITECTURES**

by

**Ottmar Gühr**

1990



## Summary

This report deals with the modelling and performance analysis of flow control procedures for connection oriented and connectionless communication in distributed systems with multi-layered protocol architectures.

Flow control is a major problem in packet switching networks as the benefits of packet switching may be neutralized rapidly if congestion occurs. Traditionally window flow control procedures have been applied in data networks to control the number of data packets in the network from one connection. Recently rate control procedures are discussed for the ATM (Asynchronous Transfer Mode) technology used in Broadband ISDN. In this report mainly the window flow control procedures are considered on one layer for one or multiple connections and in a hierarchical multi-layer protocol architecture. The rate control procedures are considered briefly on one layer. These procedures are modelled using the syntax of queueing theory with new symbols, where necessary. The models are analysed by applying queueing theory. The analysis of queueing networks plays a key role as communication networks are modelled as queueing networks. Simple queueing networks are used to demonstrate the performance analysis and to show the accuracy of the approximate algorithms before a realistic multi-layer analysis of the MAP architecture is presented. Due to the complexity of the models the analysis is approximate for non trivial models. Therefore, the results obtained by the analytical analysis are validated by simulation results.

**Chapter 1** gives an introduction to the report and a brief outline of the contents of the report.

**Chapter 2** reviews the principles of data communication and their protocols used in open systems. Open systems are based on a layered architecture with well defined services offered from one layer to the next higher layer and protocols between entities on the same layer. The protocol functionalities include transformations of connections and data units, flow control and error control.

After introducing the basic elements used in the performance modelling the models of protocol mechanisms are described in **Chapter 3**. The modelling of rate control procedures is restricted to one layer whereas the window flow control procedures are modelled for one layer and a multi-layer protocol architecture.

**Chapter 4** describes the basic methods of performance analysis. These are the analytical methods of queueing theory and the event-by-event simulation method. Because of the importance of the analysis of queueing networks in the following chapters these methods are described in more detail. The class of the Product Form Queueing Networks is very important because exact algorithms exist to analyse these networks.

**Chapter 5** deals with the performance analysis on one layer. For the window flow control procedures several solution algorithms are described for a single connection and for multiple connections. If the network (without the flow control mechanism) belongs to the class of Product Form Queueing Networks then the results indicate that the solution algorithm using the aggregation of the network to a Flow Equivalent Service Centre gives a better accuracy than the solution algorithm using the Load Equivalent Service Centre. For general networks the analysis can be performed using Diffusion Approximation, Maximum Entropy methods or the Decomposition approach. But the accuracy is only acceptable if

the arrival process is not too far from a Poisson process and the service time distributions are not too far from the negative exponential distribution.

**Chapter 6** extends the analysis of window flow control procedures to multi-layered protocol architectures. The MAP architecture (Manufacturing Automation Protocols) is taken as an example. The network was again considered to belong to the class of the Product Form Queueing Networks. The results show good accuracy compared to simulation results.

**Chapter 7** concludes the report with a summary and an outlook.

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
1.1	Übersicht über die Arbeit . . . . .	8
<b>2</b>	<b>Grundlagen der Kommunikation</b>	<b>9</b>
2.1	Kommunikation in verteilten Systemen . . . . .	9
2.1.1	Definition von verteilten Systemen . . . . .	9
2.1.2	Kommunikation von lose gekoppelten verteilten Systemen . . . . .	9
2.1.3	Offene Kommunikation . . . . .	10
2.2	Grundlagen der offenen Kommunikation . . . . .	10
2.2.1	Architekturmodell für offene Kommunikation . . . . .	10
2.2.2	Verbindungskonzepte . . . . .	11
2.2.2.1	Verbindungsorientierte Kommunikation . . . . .	11
2.2.2.2	Verbindungslose Kommunikation . . . . .	11
2.3	Protokollmechanismen . . . . .	12
2.3.1	Verbindungsorientierte Protokollmechanismen . . . . .	12
2.3.1.1	Protokollmechanismen bezogen auf die Verbindung . . . . .	12
2.3.1.2	Protokollmechanismen bezogen auf Dateneinheiten . . . . .	12
2.3.1.3	Protokollmechanismen zur Fehlerbehebung . . . . .	14
2.3.2	Verbindungslose Protokollmechanismen . . . . .	15
2.4	Dienste und Protokolle . . . . .	15
2.4.1	Dienste von Schicht zu Schicht . . . . .	15
2.4.2	Protokolle von Instanz zu Instanz . . . . .	16
2.4.3	Zusammenwirken von Dienst und Protokoll . . . . .	18
2.5	Protokollarchitekturen . . . . .	18
2.6	Leistungsuntersuchung von Systemen . . . . .	19
<b>3</b>	<b>Modellierung</b>	<b>22</b>
3.1	Warteschlangenmodelle . . . . .	22
3.1.1	Elementare Warteschlangenmodelle . . . . .	22
3.1.2	Warteschlangennetze . . . . .	25
3.1.2.1	Produktform-Warteschlangennetze . . . . .	27
3.1.2.2	Allgemeine Warteschlangennetze . . . . .	28
3.1.2.3	Erweiterte Warteschlangennetze . . . . .	28
3.2	Protokollmechanismen . . . . .	28
3.2.1	Verbindungsorientierte Protokollmechanismen . . . . .	28
3.2.1.1	Flußregelung . . . . .	28
3.2.1.2	Fehlerbehebung . . . . .	30

3.2.1.3	Multiplexen/Demultiplexen von Verbindungen . . . . .	33
3.2.1.4	Aufspaltung/Sammlung von Verbindungen . . . . .	33
3.2.1.5	Aufteilen/Vereinigen von Dateneinheiten . . . . .	34
3.2.1.6	Verketten/Trennen von Dateneinheiten . . . . .	34
3.2.1.7	Blocken/Entblocken von Dateneinheiten . . . . .	35
3.2.2	Verbindungslose Protokollmechanismen . . . . .	35
3.2.2.1	Flußregelung . . . . .	35
3.2.2.2	Fehlerbehebung . . . . .	37
3.3	Protokolle . . . . .	37
3.3.1	Verbindungsorientierte Protokolle . . . . .	37
3.3.2	Verbindungslose Protokolle . . . . .	38
3.4	Protokollarchitekturen . . . . .	38
3.4.1	MAP-Architektur . . . . .	38
3.4.2	Mehrschichtige verbindungsorientierte Flußregelung . . . . .	39
3.4.3	Sequenzen von Verbindungen . . . . .	42
3.4.3.1	Unabhängige Sequenz von Verbindungen . . . . .	42
3.4.3.2	Abhängige Sequenz von Verbindungen . . . . .	42
3.5	Einbeziehung von Implementierungsaspekten . . . . .	43
3.5.1	Betriebssystem und Verarbeitungsinstanzen . . . . .	43
3.5.2	Ressourcenbeschränkungen . . . . .	44
3.5.3	Aufruf- und Übergabemechanismen . . . . .	45
3.5.4	MAP-Architektur als Beispiel . . . . .	45
<b>4</b>	<b>Grundlagen für die Leistungsuntersuchung</b> . . . . .	<b>46</b>
4.1	Analytische Methoden der Warteschlangentheorie . . . . .	46
4.1.1	Stochastischer Prozeß . . . . .	47
4.1.2	Elementare Warteschlangensysteme . . . . .	48
4.1.3	M/M/1- $\infty$ Warteschlangensystem . . . . .	48
4.1.4	M/G/1- $\infty$ und G/M/1- $\infty$ Warteschlangensysteme . . . . .	49
4.1.5	G/G/1- $\infty$ Warteschlangensystem . . . . .	49
4.1.6	Prioritäts-Warteschlangenmodelle . . . . .	50
4.1.7	Pollingsysteme . . . . .	51
4.1.8	Diffusions-Approximation . . . . .	52
4.1.9	Maximum-Entropie-Approximation . . . . .	53
4.1.10	Warteschlangennetze . . . . .	54
4.1.10.1	Produktform-Warteschlangennetze . . . . .	54
4.1.10.2	Allgemeine Netzwerke . . . . .	55
4.1.10.3	Erweiterte Netzwerke . . . . .	56
4.2	Simulation . . . . .	57
4.2.1	Ereignisgesteuerte Simulation . . . . .	58
4.2.1.1	Simulationsmodell . . . . .	58



4.2.1.2	Ereignisbearbeitung	58
4.2.1.3	Erfassung und Auswertung der Meßgrößen	59
4.2.1.4	Simulationsablauf	60
4.2.1.5	Implementierungsgesichtspunkte	61
4.2.1.6	Aufbau der Modul-Bibliothek	61
4.2.1.7	Softwarestruktur der Modul-Bibliothek	64
4.3	Messung	64
4.3.1	Passive Messung	64
4.3.2	Aktive Messung	65
4.3.3	Messung am konzentrierten System	65
4.3.4	Messung am verteilten System	66
<b>5</b>	<b>Analytische Leistungsuntersuchung von Flußregelungen</b>	<b>67</b>
5.1	'Handshake'-Mechanismus	67
5.2	Fenstermechanismus	69
5.2.1	Verfahren für eine Verbindung	69
5.2.1.1	Literaturübersicht	69
5.2.1.2	Flußäquivalentes Ersatznetzwerk	72
5.2.1.3	Lastäquivalentes Ersatznetzwerk	74
5.2.1.4	Maximum-Entropie-Verfahren	77
5.2.1.5	Bewertung der Verfahren	78
5.2.2	Verfahren für mehrere Verbindungen	81
5.2.2.1	Literaturübersicht	81
5.2.2.2	Lastäquivalentes Ersatznetzwerk	82
5.2.2.3	Flußäquivalentes Ersatznetzwerk	83
5.2.2.4	Bewertung der Verfahren	83
5.2.3	Schritthaltender Fenstermechanismus	85
5.3	Ratenregelung	85
5.3.1	Ratenregelung der Dateneinheiten	85
5.3.2	Ratenregelung der Arbeit	86
5.3.2.1	Zulassung bei Unterschreitung von $U_{max}$	86
5.3.2.2	Zulassung bei Unterschreitung von $U_{max}$ mit Auftrag	87
5.3.2.3	Beispiele	87
5.4	Optimierung	89
5.4.1	Verbindungsorientierte Kommunikation	89
5.4.1.1	Eine Verbindung	89
5.4.1.2	Mehrere Verbindungen	91
5.4.2	Verbindungslose Kommunikation	93
5.5	Einbeziehung der Fehlerbehebung	94
5.5.1	Einzelquittierung	94
5.5.2	Reihenfolgeüberwachung	94

<b>6 Analytische Leistungsuntersuchung von Flußregelungen in Protokollarchitekturen</b>	<b>98</b>
6.1 Literaturübersicht . . . . .	98
6.2 Hierarchisch flußregulierte Verbindungen . . . . .	99
6.2.1 Konstruktion des flußäquivalenten Ersatzknotens . . . . .	100
6.3 Analyse der MAP-Architektur . . . . .	101
6.3.1 Modell und Modellparameter . . . . .	101
6.3.2 Flußäquivalenter Ersatzknoten für Kanalzugriff . . . . .	103
6.3.3 Auslastung der Prozessoren . . . . .	104
6.3.4 Analyse der Schichten 2b-4 . . . . .	104
6.3.5 Separate Analyse der Schichten 5-7 . . . . .	106
6.3.6 Analyse der gesamten Protokollarchitektur . . . . .	106
6.3.7 Optimierung der Fenstergröße . . . . .	107
6.4 Unabhängige Sequenz von Verbindungen . . . . .	109
6.4.1 Analyse . . . . .	109
6.4.2 Optimierung der Fenstergröße . . . . .	109
6.5 Abhängige Sequenz von Verbindungen . . . . .	111
6.5.1 Analyse . . . . .	111
6.5.2 Optimierung der Fenstergröße . . . . .	112
<b>7 Zusammenfassung und Ausblick</b>	<b>115</b>
<b>A Berechnung der Wartezeitverteilungsfunktion für das GE/GE/1-<math>\infty</math> Warteschlangensystem</b>	<b>125</b>
<b>B Analyse des begrenzt zustandsabhängigen GE/GE/1-<math>\infty</math> Warteschlangensystems</b>	<b>129</b>

# Abkürzungen

ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DIN	Deutsches Institut für Normung
ECMA	European Computer Manufacturers Association
FIFO	First-In, First-Out
IC	Inactivity Control
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
LAN	Local Area Network
LBANC	Local Balance Algorithmen for Normalizing Constants
LIFO	Last-In, First-Out
LIFO-PR	First-In, First-Out Preemptive Resume
LLC	Logical Link Control
LLC1	Logical Link Control Procedure, Type 1
MAC	Media Access Control
MAP	Manufacturing Automation Protocols
MVA	Mean Value Analysis
NBS	National Bureau of Standards
RECAL	Recursion over Chains
TOP	Technical and Office Protocols

# Formelzeichen

$a(t)$	Verteilungsdichtefunktion des Ankunftsabstandes
$\mathcal{A}^*(s) \bullet \longrightarrow a(t)$	Laplace-Stieltjes-Transformierte der Verteilungsdichtefunktion der Ankunftsabstände $a(t)$
$A(t)$	Ankunftsabstandverteilung
$b(t)$	Verteilungsdichtefunktion der Bedienzeit
$\mathcal{B}^*(s) \bullet \longrightarrow b(t)$	Laplace-Stieltjes-Transformierte der Bedienzeitverteilungsdichtefunktion $b(t)$
$B(t)$	Bedienzeitverteilung
$B$	Übertragungsrate
$c$	Variationskoeffizient
$c_a$	Variationskoeffizient der Verteilung der Ankunftsabstände $A(t)$
$c_b$	Variationskoeffizient der Bedienzeitverteilung $B(t)$
$d$	Mittlere Rahmenlänge
$\delta(t)$	Diracimpuls
$E[X]$	Mittelwert der Zufallsvariablen $X$
$f_1(\mathbb{1}_1) \dots f_M(\mathbb{1}_M)$	Terme der Knoten

$G$	Normierungskonstante
$h$	Mittlere Bedienzeit
$H(m, n) = [p_{i,j}(m, n)]$	mehrschrittige Übergangsmatrix zur Zeit $m$
$\underline{H}(s, t) = [p_{i,j}(s, t)]$	Übergangswahrscheinlichkeitsmatrix
$\underline{1}$	Einheitsvektor
$\bar{n}$	Mittlere Anzahl von Aufträgen
$\bar{n}_w$	Mittlere Anzahl von Aufträgen in der Warteschlange
$\underline{n}$	Zustandsvektor des Netzwerkes
$\underline{n}_i = (n_{i,1} \dots n_{i,j} \dots n_{i,k})$	Zustandsvektor des Knotens $i$ für alle Ketten
$n_{i,j}$	Anzahl der Aufträge in Knoten $i$ der Kette $j$
$P(n) = [p_{i,j}(n, n+1)]$	einschrittige Übergangswahrscheinlichkeiten zur Zeit $n$
$p_F$	Fehlerwahrscheinlichkeit
$p_{i,j}(m, n) = P\{\xi_n = j   \xi_m = i\}$	Übergangswahrscheinlichkeit
$\pi^{(n)} = P\{\xi_n = i\}$	Zustandswahrscheinlichkeit $i$ zur Zeit $n$
$\underline{\pi}^{(n)}$	Vektor der Zustandswahrscheinlichkeiten zur Zeit $n$
$\underline{\pi}$	Vektor der stationären Zustandswahrscheinlichkeiten
$\underline{Q}(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t) - I}{\Delta t}$	Matrix der Übergangsraten
$\lambda$	Ankunftsrate
$\mu$	Enderate
$R(z)$	Erzeugende Funktion der Regenerationszeitpunkt-Zustandswahrscheinlichkeiten $r_k$
$U(t)$	Prozess der unerledigten Arbeit
$S$	'Slot' für Kollisionsauflösung
$T_A$	Zufallsvariable des Ankunftsabstandes
$T_B$	Zufallsvariable der Dauer der Bedienzeit
$T^R$	Zufallsvariable der Dauer der Restbedienzeit
$T_w$	Zufallsvariable der Wartezeit
$t_{\alpha/2; n-1}$	Student-t-Wert für den Vertrauensgrad $\alpha$ bei $n-1$ Freiheitsgraden
$t_{ad}$	Mittlere Zugangsverzögerung
$t_{ts}$	Mittlere Transit-Zeit
$t_{if} = t_{ad} + t_{ts}$	Mittlere Transferzeit
$t_{ak}$	Mittlere Quittierungs-Transit-Zeit
$t_{cy} = t_{if} + t_{ak}$	Mittlere Zykluszeit
$T_{ad}$	Zufallsvariable der Zugangsverzögerung
$T_{ts}$	Zufallsvariable der Transit-Zeit
$T_{if} = T_{ad} + T_{ts}$	Zufallsvariable der Transferzeit
$T_{ak}$	Zufallsvariable der Quittierungs-Transit-Zeit
$T_{cy} = T_{if} + T_{ak}$	Zufallsvariable der Zykluszeit
$w$	Fenstergröße
$w(t)$	Wartezeitverteilungsdichtefunktion
$W(t)$	Wartezeitverteilungsfunktion
$x_{i,j}$	$j$ -te Realisierung der Zufallsvariablen $X$ im Teilstest $i$
$\hat{X}_i$	Schätzwert für den Mittelwert von $X$ im Teilstest $i$

# Kapitel 1

## Einleitung

Durch die rasche Entwicklung der Elektronik in dem Bereich der Halbleiterintegration kommt es heute zu einem breiten Einsatz von Datenverarbeitungsanlagen in Wissenschaft, Verwaltung und Industrie. Während in der Anfangszeit hauptsächlich die Zahlenmanipulation, für die Berechnung bis dahin nicht lösbarer Probleme, im Vordergrund stand, haben sich heute die Hauptanwendungen geändert. Dokumentenbearbeitung, Datenarchivierungen, Steuerung technischer Prozesse und, allgemein, die Symbolmanipulation stehen im Vordergrund. Obwohl Datenverarbeitungsanlagen, nach Alan Mathison Turing [110], im technischen Sinn Universalmaschinen sind, können sie heute meist als Spezialmaschinen angesehen werden. Datenbanken, Transaktionssysteme, Dokumentenverarbeitungssysteme oder Rechensysteme sind Beispiele dieser Spezialisierung. Diese Spezialisierung, verbunden mit einer örtlichen Verteilung der Datenhaltung und Datenverarbeitung, führt zu einem wachsenden Bedürfnis der Kommunikation der Datenverarbeitungsanlagen zum Austausch von Meldungen, Daten und Programmen.

Durch das natürliche Bedürfnis der Menschen, Kommunikation zu betreiben, haben sich Post und Telefon entwickelt. Nun ergibt sich durch Datenverarbeitungsanlagen die Möglichkeit, Nachrichten, die bisher durch Telegramme, Briefe, Postkarten und Notizzettel übermittelt wurden, auf elektronischem Wege zu übermitteln.

Durch dieses einerseits technische und andererseits menschliche Bedürfnis erleben Kommunikationsnetze einen steilen Aufschwung. Bei der Kommunikation fällt eine Vielzahl von Aufgaben an, wie gesicherte zeitgerechte Übertragung, transparente Adressierung und Umgebungsunabhängigkeit durch Datentransformation. Diese Vielzahl an Aufgaben bedingt eine hohe Komplexität. Zum Betrieb des Kommunikationsnetzes sind weitere Funktionen, die für den Anwender transparent sind, notwendig. Sie sind unter dem Begriff Netzwerk-Management zusammengefaßt. Dazu gehören Fehlererkennung, -ortung und -diagnose, Ressourcenverwaltung, Verkehrslenkung, Datenflußregelung, Überlasterkennung und -abwehr, Gebührenabrechnung und Schutz vor mißbräuchlichem Zugriff.

Der Komplexität des Kommunikationsnetzes wird durch eine funktionale Schichtung mit spezifischen Aufgaben für jede Schicht begegnet. Diese Funktionen werden international durch Standardisierungsgremien festgelegt. Halten sich die Hersteller von Datenverarbeitungsanlagen an diese Festlegungen, so ist eine offene herstellerübergreifende Kommunikation möglich. Um einen effizienten Aufbau und Betrieb von Kommunikationsnetzen zu gewährleisten, werden während der Entwicklung Leistungsuntersuchungen und während des Betriebs Messungen durchgeführt. Die Leistungsuntersuchung basiert auf einer Modellierung des Verhaltens des Kommunikationsnetzes und dessen Benutzer. Sie wird durch mathematische Analyse oder durch Simulation durchgeführt. Die komplexen Systeme führen zu komplexen Modellen, die die Grenzen der Leistungsuntersuchung oft

überschreiten, so daß neue Wege, sowohl in der Analyse als auch der Simulation, gesucht werden, müssen dieser Komplexität zu begegnen.

In dieser Arbeit wird auf die Leistungsfähigkeit der Kommunikation durch mathematische Analyse und Simulation eingegangen. Die Darstellung und Untersuchung der Datenflußregelung steht dabei im Vordergrund. Als typische Leistungsgrößen werden der Durchsatz und die Transferzeit von Dateneinheiten herangezogen. Mit der Leistungsuntersuchung können der Einfluß und die Wechselwirkung der einzelnen Parameter studiert werden. Die frei wählbaren Parameter können über die Optimierung so dimensioniert werden, daß Randbedingungen, wie Effizienz und Wirtschaftlichkeit, berücksichtigt werden.

## 1.1 Übersicht über die Arbeit

In Kapitel 2 werden die grundlegenden Konzepte der Kommunikation von Datenverarbeitungsanlagen besprochen. Ausgehend von der Definition von verteilten Systemen wird die Schichtung der Funktionalitäten mit dem Dienst- und Protokollkonzept dargestellt. Nachdem verbindungsorientierte und verbindungslose Kommunikation eingeführt wurden, werden die jeweiligen Protokollmechanismen dargestellt und in das Dienst- und Protokollkonzept eingebettet. Die Protokolle fügen sich in ihrer Schichtung wiederum zu Protokollarchitekturen zusammen. Die Darstellung der Methoden der Leistungsuntersuchung beschließt Kapitel 2.

Die Modellierung ist Hauptgegenstand von Kapitel 3. Zu der Modellierung werden in dieser Arbeit Elemente der Warteschlangentheorie verwendet. Es werden in einzelnen Abschnitten die Modellierung der Protokollmechanismen, der Protokolle und der Protokollarchitekturen dargestellt. Während die Protokollmechanismen allgemein dargestellt sind, erfolgt bei der Modellierung von Protokollen und Protokollarchitekturen eine Einschränkung auf Datenflußregelung und Fehlerbehebung.

In Kapitel 4 sind die grundlegenden Methoden der Leistungsuntersuchung, analytische Berechnung, Simulation und Messung, dargestellt. Bei den analytischen Methoden werden die wichtigsten Warteschlangensysteme und -netze und den wichtigsten Berechnungsmethoden zusammenfaßt. In dem Abschnitt über die Simulation wird hauptsächlich auf die ereignisgesteuerte Simulationsmethode eingegangen. Es wird weiterhin der Aufbau einer Simulationsbibliothek gezeigt. Die grundlegenden Konzepte der Messung an konzentrierten und verteilten Systemen werden anschließend geschildert.

In Kapitel 5 wird hauptsächlich auf die Analyse von Flußregelungen in Protokollen innerhalb einer Schicht eingegangen. Dazu werden verschiedene Analysemethoden für die unterschiedlichen Arten der Flußregelung dargestellt. Äquivalente Methoden werden miteinander verglichen und durch Simulation validiert. Der optimalen Wahl der Parameter der Flußregelung ist ein weiterer Abschnitt gewidmet. Der letzte Abschnitt befaßt sich mit der Fehlerbehebung.

Die analytische Leistungsuntersuchung von Flußregelungen in mehrschichtigen Protokollarchitekturen wird in Kapitel 6 dargestellt und anhand eines Beispiels durchgeführt. Die Hauptschwierigkeit besteht in der geeigneten Aggregation von Subsystemen, da das Gesamtsystem keiner geschlossenen Analyse zugänglich ist. Dies wird durch das Konzept des flußäquivalenten Ersatzknotens erreicht.

Eine Zusammenfassung mit einem Ausblick beschließen die Arbeit.

## Kapitel 2

# Grundlagen der Kommunikation

### 2.1 Kommunikation in verteilten Systemen

#### 2.1.1 Definition von verteilten Systemen

Unter einem System soll in dieser Arbeit ein technisches System verstanden werden, das Funktionen oder Dienste dem Anwender oder anderen technischen Systemen zur Verfügung stellt. Unter einem verteilten System soll ein System verstanden werden, das durch Zusammenarbeit von technischen Systemen Funktionen und Dienste erbringt. Eine Klassifizierung verteilter Systeme kann hauptsächlich durch den Grad der Kopplung und die örtliche Verteilung der Systeme erfolgen. Je weiter Systeme voneinander entfernt sind, um so loser ist die Kopplung, und je enger die Systeme gekoppelt sind, um so näher sind sie zusammen. Eine enge Kopplung zeichnet sich durch starre häufige Synchronisation mit geringem Aufwand aus. Lose Kopplungen haben meist eine hohe Variabilität und Flexibilität mit einer schwachen seltenen Synchronisation mit hohem Aufwand. Datenflußregulierung, Fehlererkennung und Fehlerbehebung spielen hier eine wichtige Rolle. Als Beispiel eng gekoppelter Systeme kann ein Prozessorsystem mit Systemkomponenten wie Zentraleinheit, Speichereinheit oder Ein-/Ausgabereinheit dienen. Als lose gekoppelte Systeme werden Endgeräte an lokalen Netzen oder Weitverkehrsnetzen betrachtet. In dieser Arbeit werden in diesem Sinne die lose gekoppelten verteilten Systeme betrachtet.

#### 2.1.2 Kommunikation von lose gekoppelten verteilten Systemen

Während die eng gekoppelten Systeme meist über einen parallelen synchronen oder asynchronen Bus mit getrennten Steuer- und Datenleitungen verbunden sind, sind lose gekoppelte Systeme über serielle asynchrone Übertragungsmedien miteinander verbunden. Bei enger Kopplung ist der Mechanismus der Datenübertragung meist ein Zyklus von Übergabe der Dateneinheit und direkter Bestätigung. Die Regeln der Datenübertragung werden als Protokoll bezeichnet. Bei loser Kopplung laufen komplexere Protokolle ab, um die Effizienz des Übertragungskanales zu erhöhen, da durch die räumliche Trennung der Systeme eine direkte Quittierung oft ineffektiv ist. Kommt es zu Fehlern bei der Übertragung, so ist jedoch ein erheblicher Aufwand nötig, um die Kommunikation sinnvoll und geordnet weiterzuführen.

### 2.1.3 Offene Kommunikation

Durch die Vielzahl der Hersteller und die Unterschiede in den Realisierungen der einzelnen Datenverarbeitungsanlagen entstehen heterogene verteilte Systeme, die miteinander kommunizieren wollen und müssen. Dies geht von dem Anschluß an das Übertragungsmedium bis zu der Verwendung derselben Kommunikationsmechanismen. Sind die Anschlußmöglichkeiten und Kommunikationsmechanismen bekannt, und nicht ein Firmengeheimnis, so spricht man von offener Kommunikation. Um allgemeine herstellerübergreifende Festlegungen bemühen sich internationale, europäische und nationale Standardisierungsgremien. Dies sind vor allem:

- ISO ('International Organization for Standardization'),
- IEEE ('Institute of Electrical and Electronics Engineers'),
- CCITT ('Comité Consultatif International Télégraphique et Téléphonique'),
- IEC ('International Electrotechnical Commission'),
- ECMA ('European Computer Manufacturers Association') und
- DIN (Deutsches Institut für Normung).

Die Aufteilung der Übertragungsfunktionen in sieben Schichten ist heute allgemein anerkannt. Man bezeichnet dies als mehrschichtige Protokollarchitektur. Die Standards lassen oft noch weitläufige Parameter bei ihren Spezifikationen zu. Für Anwendungsgruppen, wovon augenblicklich MAP ('Manufacturing Automation Protocols') und TOP ('Technical and Office Protocols') am bekanntesten sind, werden diese Parameter durchgehend für alle Schichten festgelegt. Die so entstandene Protokollarchitektur, mit wohldefinierten Protokollen und genau festgelegten Parametern, nennt man Protokollprofil.

## 2.2 Grundlagen der offenen Kommunikation

Aus der Sicht eines Benutzers erbringt das Kommunikationsnetz einen Dienst, der über Dienstprimitive angesprochen wird. Dieser Dienst wird durch den Nachrichtenaustausch, gemäß den Regeln eines Protokolles, zwischen Instanzen, die als Prozesse aufgefaßt werden, erbracht.

### 2.2.1 Architekturmodell für offene Kommunikation

Durch die Vielzahl der Anforderungen an die Güte eines Kommunikationsdienstes wurde eine funktionale Aufteilung in Schichten vorgenommen, so daß eine höhere Schicht jeweils auf dem Dienst, und damit auch der Güte, der niedereren Schicht aufbaut. In dem heute allgemein anerkannten ISO-Referenzmodell [46] wurden sieben Schichten mit folgenden Hauptfunktionen definiert:

7	Anwendung	Unterstützung der Anwendung
6	Darstellung	Angleichung des Darstellungskontextes
5	Kommunikationssteuerung	Steuerung einer Kommunikationsbeziehung
4	Transport	Endgerät-zu-Endgerät Kommunikation
3	Vermittlung	Vermittlung der zu übertragenden Nachrichten
2	Sicherung	Sicherung der Übertragung gegenüber Fehler
1	Bitübertragung	Bitserielle Übertragung der Nachricht



Jede Schicht erbringt ihren Teil der Gesamtgüte, indem der zu übertragenden Nachricht Kontrollinformation beigefügt wird, die von der Partnerinstanz wieder ausgewertet wird. Die Dienstprimitiven werden über die Dienstzugangspunkte ausgetauscht (siehe Bild 2.1).

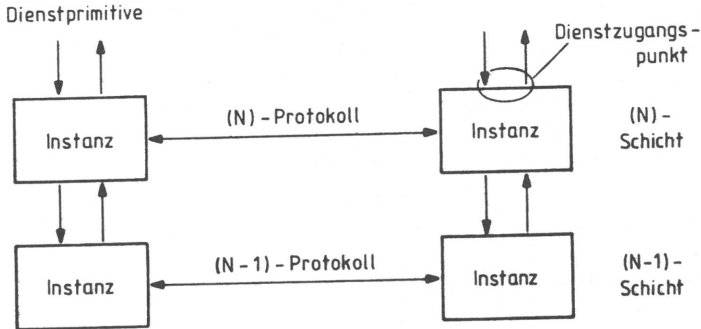


Bild 2.1: Prinzip der Schichtung

## 2.2.2 Verbindungskonzepte

Es werden in der Kommunikationstechnik zwei unterschiedliche Verbindungskonzepte, die verbindungsorientierte und die verbindungslose Kommunikation, angewendet.

### 2.2.2.1 Verbindungsorientierte Kommunikation

Bei der verbindungsorientierten Kommunikation wird, für eine bestimmte Zeit, eine Verbindung zum Austausch von Dateneinheiten zwischen den Kommunikationspartnern hergestellt. Die Verbindung durchläuft somit die drei Phasen

- Verbindungsaufbau,
- Datentransfer und
- Verbindungsabbau.

Die Verbindung kommt zustande, indem die Partner und die darunter liegende Schicht (Dienst-Erbringer) sich über die Parameter der Verbindung durch eine Aushandlungsprozedur einigen. Kann keine Übereinkunft gefunden werden, so kommt die Verbindung nicht zustande. Durch den Verbindungsaufbau werden in den einzelnen Instanzen Betriebsmittel reserviert und der Verbindung eindeutige Adressen zugewiesen ('Connection Identifiers'). Während der Datenübertragungsphase haben die auf einer Verbindung übertragenen Dateneinheiten eine Beziehung zueinander. Zum Beispiel kann die Reihenfolge der ausgesendeten Dateneinheiten bei Fehlverhalten der darunter liegenden Schicht beim Empfänger wieder sichergestellt werden.

### 2.2.2.2 Verbindungslose Kommunikation

Bei der verbindungslosen Kommunikation enthält jede Dateneinheit die vollständige Adressinformation des Empfängers. Diese Dateneinheit wird ohne Beziehung zu anderen Da-

teneinheiten dem Partner übermittelt, indem die Dateneinheit über eine Dienstprimitive der darunter liegenden Schicht übergeben wird. Es kann somit keine Aushandlung von Parametern geben.

In Tabelle 2.1 sind die Charakteristiken von verbindungsorientierter und verbindungsloser Kommunikation gegenübergestellt.

Verbindungslos	Verbindungsorientiert
Zwei-Parteien-Übereinkunft	Drei-Parteien-Übereinkunft
Ein einziger Zugriff	Lebensdauer der Verbindung
Unabhängigkeit der Dateneinheiten	Beziehung der Dateneinheiten zueinander
Selbst beschreibende Dateneinheiten	Verbindungs-Kennzeichnungen
Keine Aushandlung	Aushandlung

Tabelle 2.1: Gegenüberstellung von verbindungsloser und verbindungsorientierter Kommunikation

## 2.3 Protokollmechanismen

### 2.3.1 Verbindungsorientierte Protokollmechanismen

Die verbindungsorientierten Protokollmechanismen lassen sich in die drei Kategorien

- Protokollmechanismen bezogen auf die Verbindung,
- Protokollmechanismen bezogen auf Dateneinheiten und
- Protokollmechanismen zur Fehlerbehebung

einteilen, die in den nächsten Abschnitten besprochen werden.

#### 2.3.1.1 Protokollmechanismen bezogen auf die Verbindung

Es existieren die zwei Protokollmechanismen Multiplexen und Aufspalten, die sich auf die Verbindung beziehen. Diese Protokollmechanismen werden beim Empfänger jeweils wieder durch Demultiplexen und Zusammenführen rückgängig gemacht (siehe Bild 2.2).

#### 2.3.1.2 Protokollmechanismen bezogen auf Dateneinheiten

Wesentlich mehr Protokollmechanismen beziehen sich auf die Operationen mit Dateneinheiten innerhalb einer Verbindung. Drei Protokollmechanismen ändern die Zusammensetzung der Dateneinheiten und zwei weitere führen eine Flußregelung und eine Wiederherstellung der Reihenfolge der Dateneinheiten durch. Die beiden Protokollmechanismen Verketteten/Trennen und Blocken/Entblocken bilden aus jeweils mehreren Dateneinheiten eine Dateneinheit für die Übertragung. Dabei bezieht sich Verketteten/Trennen auf die Dateneinheiten der Dienstprimitiven und bildet daraus eine Dateneinheit des Protokolles, während Blocken/Entblocken aus Protokoll-Dateneinheiten eine Dateneinheit der Dienstprimitive bildet (siehe Bild 2.3).

Der Protokollmechanismus Aufteilen/Vereinigen führt nun das Gegenteil aus, indem er aus einer Dienst-Dateneinheit mehrere Protokoll-Dateneinheiten bildet. Dies wird zum Beispiel angewendet, um die Länge der Dateneinheiten an die Bedürfnisse des Dienstbringers anzupassen (siehe Bild 2.4).

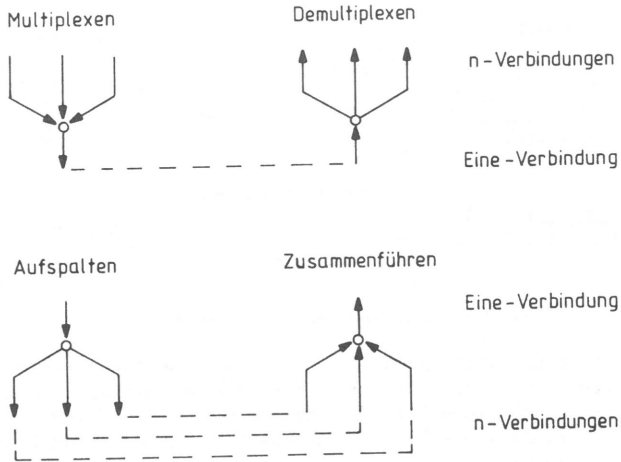


Bild 2.2: Protokollmechanismen Multiplexen/Demultiplexen und Aufspalten/Zusammenführen

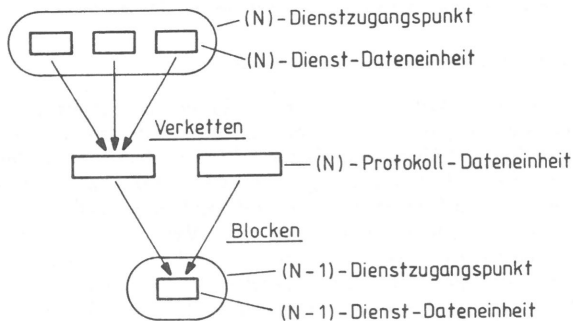


Bild 2.3: Protokollmechanismen Verketteten und Blocken

Die Datenflußregelung dient einerseits zur Anpassung der Leistungskapazitäten des Empfängers und des Diensterbringers an die Leistungsfähigkeit des Senders und andererseits zur Vermeidung von Netzwerk-Verstopfungen, indem nur eine bestimmte Anzahl von Dateneinheiten einer Verbindung sich im Netzwerk befinden können. Die Flußregelung wird meist mit dem Fenstermechanismus ausgeführt, von dem es zwei Ausprägungen, das gleitende Fenster ('Sliding Window') und das schritthaltende Fenster ('Pacing Window'), gibt. Bei dem gleitenden Fenster erhält der Sender ein Guthaben von  $w$  zu sendenden Dateneinheiten. Für jede gesendete Dateneinheit erniedrigt sich das Guthaben um 1. Das Guthaben wird durch den Empfänger mit Quittierungen wieder erhöht, so daß im Normalfall ein kontinuierliches Aussenden von Dateneinheiten erreicht wird. Im einfach-

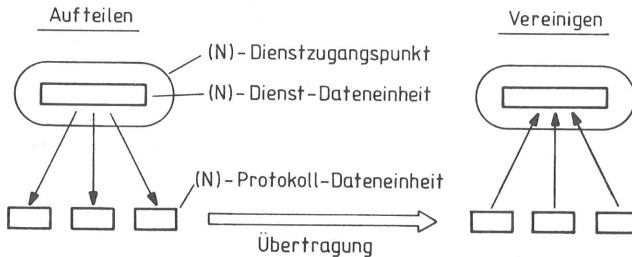


Bild 2.4: Protokollmechanismus Aufteilen/Vereinigen

sten Fall quittiert der Empfänger sofort jede erhaltene Dateneinheit einzeln. Bei dem schritthaltenden Fenstermechanismus erhält der Sender ein Guthaben von  $w$  zu sendenden Dateneinheiten. Unterschreitet sein Guthaben  $K$ , so fordert er vom Empfänger eine Erhöhung des Guthabens um  $K$  an. Sein Guthaben kann sich somit im Bereich von  $0..w = 2K - 1$  bewegen.

Bei der Übertragung bekommt jede Dateneinheit von dem Sender eine, innerhalb der Verbindung eindeutige Reihenfolgennummer zugeordnet, die als Zusatzinformation mit übertragen wird. Der Empfänger kann dadurch die Reihenfolge der Dateneinheiten wieder herstellen, falls sie durch den Diensterbringer durcheinander gebracht wurde.

### 2.3.1.3 Protokollmechanismen zur Fehlerbehebung

Aus der Menge der Möglichkeiten der Fehlererkennungs- und Fehlerbehebungsmethoden seien hier die wichtigsten angeführt. Einzelne Dateneinheiten werden bei der Übertragung gegen Fehler durch Prüfsummen gesichert ('Cyclic Redundancy Check'), die sich auf den unteren Schichten (Bitübertragungs-, Datensicherungs- und Vermittlungsschicht) auf die seriell übertragenen Bits beziehen und auf den höheren Schichten (hauptsächlich Transportschicht) auf Oktette aus 8 Bits. Von einer Verwendung von Prüfsummen, die eine Korrektur gestörter Dateneinheiten zulassen, wird im allgemeinen abgesehen und eine Wiederholung veranlaßt. Eine Ausnahme bildet die Kommunikation über Satelliten, bei denen die Übertragungsgeschwindigkeiten und Fehlerraten sehr groß und die Signallaufzeiten sehr lang sind.

Durch die Methode der Prüfsummen ist der Empfänger in der Lage, aufgetretene Fehler in Dateneinheiten zu entdecken und eine Behebung einzuleiten. Eine zusätzliche restriktive Möglichkeit besteht darin, daß der Sender für jede ausgesendete Dateneinheit eine einzelne Quittierung erwartet. Ist die Quittierung positiv, so ist der Vorgang abgeschlossen. Ist die Quittierung negativ, so wird eine Wiederholung der Dateneinheit vorgenommen. Das Eintreffen einer Quittierung wird zusätzlich mit einer Zeitschranke ('Timer') überwacht, und bei Ausbleiben einer Quittierung wird ebenfalls eine Wiederholung vorgenommen. Mit einer durchgehenden Numerierung der Dateneinheiten innerhalb einer Verbindung kann der Empfänger die korrekte Reihenfolge überwachen und bei lückenhaftem Empfang entweder die fehlenden Dateneinheiten separat anfordern ('Selective Repeat') oder eine Wiederholung ab der fehlenden Dateneinheit anfordern ('Go-Back-N').

Verwendet der Diensterbringer ein verbindungsorientiertes Protokoll, so kann bei lücken-

haftem Empfang von einem Fehler ausgegangen werden. Bei verbindungslosen Protokollen ist das nicht unbedingt der Fall. Bei dem Verlust von Quittierungen kann es zu Verklemmungszuständen im Protokoll kommen. Diese können mit dem sogenannten Kontrollpunktverfahren behoben werden. Dabei wird die Partnerinstanz gezielt über ihren Zustand abgefragt. Ein weiteres Verfahren, die Untätigkeits-Überwachung, überprüft eine Verbindung periodisch auf ihre Funktion, falls keine Datenübertragungen auf der Verbindung ablaufen.

### 2.3.2 Verbindungslose Protokollmechanismen

Bei der verbindungslosen Kommunikation haben sich zwei Ausprägungen herausgebildet. Einmal erfolgt die Übermittlung der Dateneinheit ohne Quittierung und das andere Mal mit Quittierung. Aber es wird weiterhin von einem Zugriff auf den Dienstbringer ausgegangen, da dieser Zugriff zwischen Dateneinheit und Quittierung nicht unterbrochen werden kann. Die Flußregelung der Dateneinheiten erfolgt ausschließlich beim Sender durch eine Regelung der Übertragungsrate in Bits oder Dateneinheiten. Die Quittierung wird nicht zur Flußregelung herangezogen. Die Flußregelung kann entweder auf den Dienstzugangspunkt oder den Empfänger (Instanz-zu-Instanz) bezogen sein (siehe Bild 2.5).

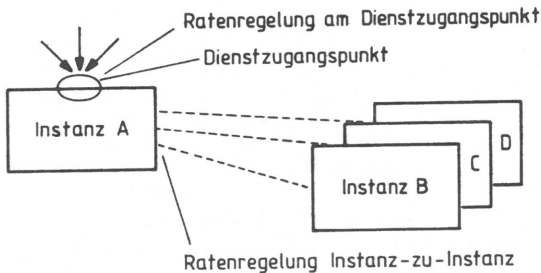


Bild 2.5: Flußregelung durch Ratenregelung bei verbindungsloser Kommunikation

Die Fehlerregelung beschränkt sich in dem Fall ohne Quittierung auf das Erzeugen einer Prüfsumme beim Sender und deren Auswertung beim Empfänger. Erkannte Fehler können nicht behoben werden. Der Fehler wird aber meist an die nächsthöhere Schicht gemeldet. Werden Quittierungen verwendet, so wiederholt der Sender bei Ausbleiben der Quittierung oder negativer Quittierung die Dateneinheit. Nach einigen Wiederholungen wird von einem nicht korrigierbaren Fehler ausgegangen.

## 2.4 Dienste und Protokolle

### 2.4.1 Dienste von Schicht zu Schicht

Die Dienste werden dem Dienst-Benutzer über den Dienst-Zugangspunkt vom Dienst-Erbringer bereitgestellt. Die Interaktion geschieht über Dienst-Primitive, denen Dienst-Dateneinheiten zugeordnet sind (siehe Bild 2.6).

Die elementaren Dienst-Primitive können zu Klassen mit unterschiedlicher Bedeutung zusammengefaßt werden. Diese Klassen sind

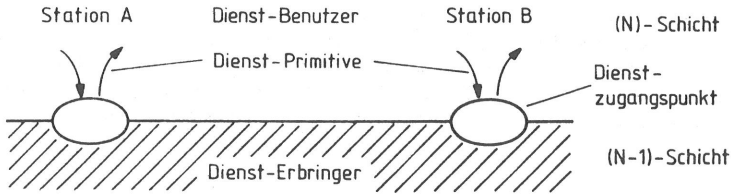


Bild 2.6: Dienst-Primitive und Dienst-Zugangspunkte

Anfragen ('Request'),  
 Anzeigen ('Indication'),  
 Antwort ('Response') und  
 Bestätigung ('Confirm').

Die Reihenfolge und grundsätzliche Bedeutung der Primitiven läßt sich im Ablauf-Zeitdiagramm (Bild 2.7) verdeutlichen.

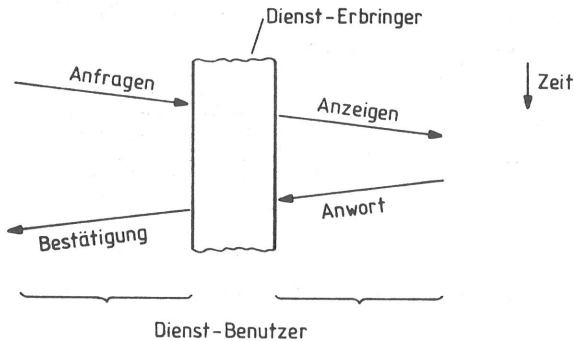


Bild 2.7: Ablauf-Zeitdiagramm der Dienst-Primitive

Neben dem vollen Ablauf-Zeitdiagramm, wie es in Bild 2.7 dargestellt ist, kann es auch verkürzte geben. Es erfolgt dann keine *Antwort* und somit auch keine *Bestätigung*. Die *Bestätigung* muß nicht unbedingt durch die *Antwort* angestoßen sein, sondern kann bei bestimmten Abläufen, wie zum Beispiel Verbindungsabbau ('Disconnect'), jederzeit nach *Anfragen* erfolgen. Verbindungsaufbau ('Connect') ist ein Beispiel für das volle Ablauf-Zeitdiagramm und der Datentransfer ('Data') ist ein Beispiel für den verkürzten Ablauf.

Der Dienst-Zugangspunkt wird als Zustandsautomat aufgefaßt. In Bild 2.8 sind seine Zustände und Zustandsübergänge dargestellt.

#### 2.4.2 Protokolle von Instanz zu Instanz

Die Instanzen tauschen Protokoll-Dateneinheiten zwischeneinander aus. Die Instanz wird als Prozeß betrachtet und durchläuft die vereinfachten Zustände wie sie in Bild 2.9

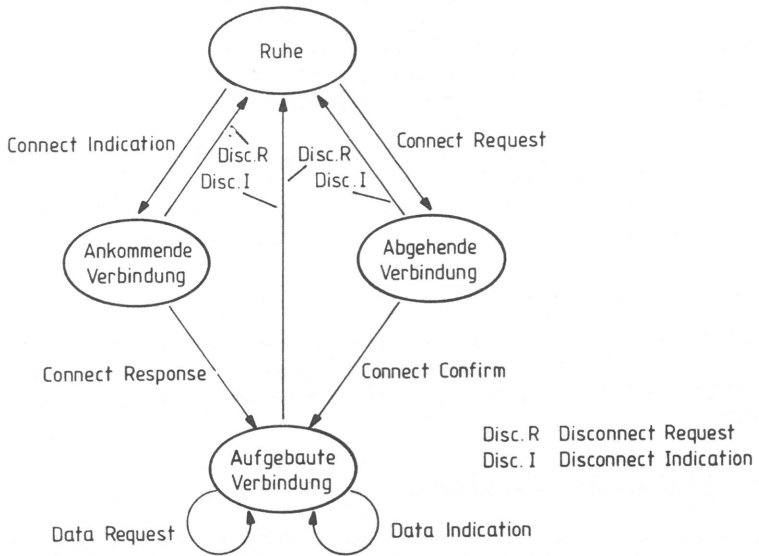


Bild 2.8: Zustände des Dienstzugangspunktes

dargestellt sind. Die Zustandsübergänge werden durch die entsprechenden Protokoll-Dateneinheiten ausgelöst.

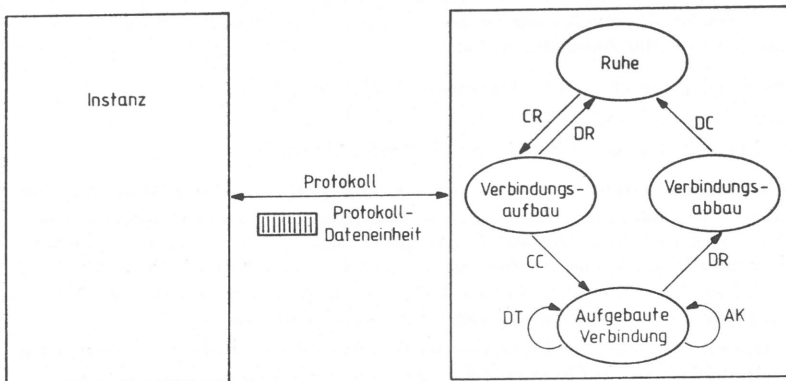


Bild 2.9: Protokolle von Instanz zu Instanz

Beispiele für Protokoll-Dateneinheiten sind:

CR - 'Connect Request'	Verbindungsaufbau	Anforderung
CC - 'Connect Confirm'	Verbindungsaufbau	Bestätigung
DR - 'Disconnect Request'	Verbindungsabbau	Anforderung
DC - 'Disconnect Confirm'	Verbindungsabbau	Bestätigung
DT - 'Data'	Datenübertragung	
AK - 'Acknowledge'	Quittierung	

Das Protokoll von Instanz-zu-Instanz führt als wesentliche Aufgabe die unter Abschnitt 2.3 erwähnten Protokollmechanismen aus.

Da die Instanzen nicht direkt miteinander verbunden sind, werden Protokoll-Dateneinheiten zwischen den Instanzen durch die Dienst-Primitiven transportiert.

### 2.4.3 Zusammenwirken von Dienst und Protokoll

Da der Dienst durch das Protokoll erbracht wird, besteht ein enger Zusammenhang zwischen ihnen. In Bild 2.10 ist das Ablauf-Zeitdiagramm für eine Verbindung mit Verbindungsaufbau, Datentransfer und Verbindungsabbau am Beispiel der Transportschicht gezeigt.

## 2.5 Protokollarchitekturen

Unter einer Protokollarchitektur versteht man den abstrakten Begriff einer geschichteten Aufteilung der Funktionalitäten. In jeder Schicht ist dann jeweils ein Protokoll definiert, das gewisse Funktionalitäten erbringt. Diese Funktionalitäten werden über die Dienstedefinition der nächsthöheren Schicht zugänglich gemacht. Nachdem sich das ISO-Referenzmodell [46] mit einer Funktionsaufteilung in sieben Schichten durchgesetzt hat und für jede Schicht separat Standards erarbeitet wurden, werden in nächster Zeit auch standardisierte Protokollprofile verfügbar sein. Da jede Anwendung ganz spezielle Anforderungen an Fehlersicherheit, Übertragungsbandbreite, Transferzeit oder Abhörsicherheit hat, wird es mehrere Protokollprofile für jeweils Klassen von Anwendungen geben. Eine grobe Klassifizierung der Anwendungen ist :

- Nachrichtenaustausch (Texte, Dokumente, Programme, Daten...),
- Steuern und Regeln von technischen Prozessen und
- gemeinsame Nutzung von Ressourcen ('Resource Sharing').

Die Protokollprofile entstehen durch die Auswahl eines geeigneten Protokolles bzw. einer Protokollalternative auf jeder Schicht und der Festlegung der Parameter, die durch den Standard noch offen gelassen wurden, auf geeignete Werte für die jeweilige Anwendung. Für die Fabrikautomatisierung hat General Motors die Initiative MAP ('Manufacturing Automation Protocols') gestartet. Für die Büroautomatisierung hat Boeing die entsprechende Initiative TOP ('Technical and Office Protocols') gegründet.

Erst durch die Festlegung der Parameter der Protokolle in Protokollprofilen kann es zur herstellerübergreifenden Kompatibilität der Kommunikationssysteme in Datenverarbeitungsanlagen kommen. Um diese Kompatibilität sicherzustellen, haben sich Prüfstellen, wie bei NBS ('National Bureau of Standards') für MAP, entwickelt, die die Produkte verschiedenen Tests unterziehen ('Conformance Test'). Besteht ein Produkt dieses Testverfahren, so erhält es ein Gütesiegel und kann mit allen anderen Produkten, die dieses Gütesiegel ebenfalls haben, kommunizieren.



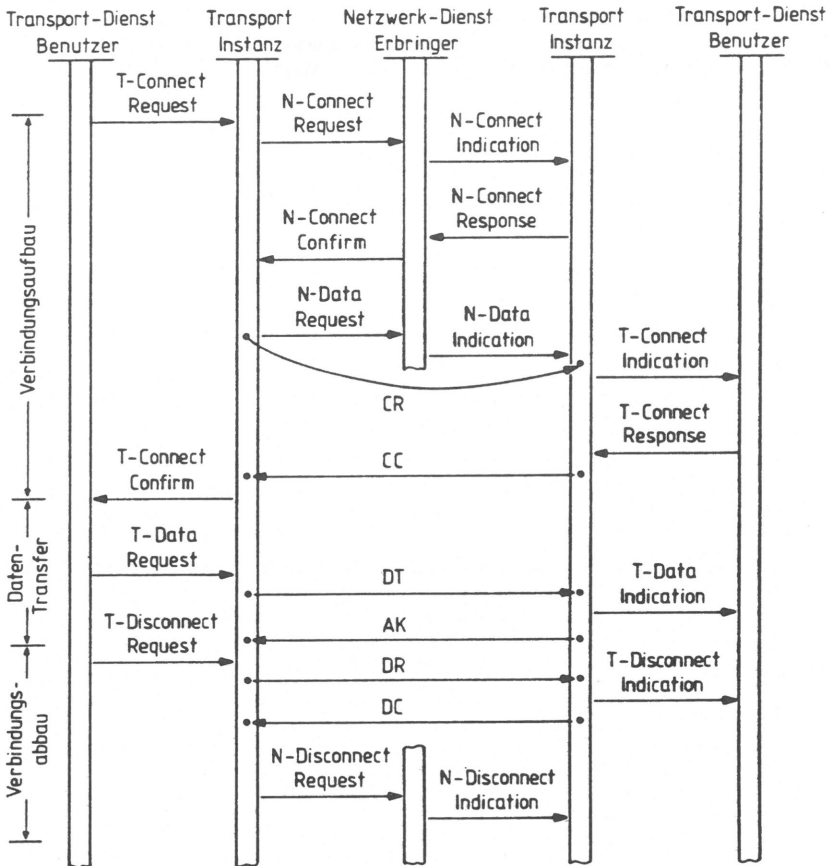


Bild 2.10: Ablauf-Zeitdiagramm für eine Transport-Verbindung

## 2.6 Leistungsuntersuchung von Systemen

Die Leistungsuntersuchung von technischen Systemen wird mit analytischen oder simulativen Methoden an einem Modell oder durch Messung an einem realen System durchgeführt. Bei der mathematischen Analyse und der Simulation wird jeweils ein Modell durch Abstraktion des Systems konstruiert. Dieses Modell wird anschließend durch analytische Methoden der Warteschlangentheorie untersucht oder durch einen Simulator in seinem Verhalten nachvollzogen. Durch Beobachtung der Modellgrößen werden die Ergebnisse extrahiert. Sowohl analytische wie auch simulative Ergebnisse bedürfen meist noch einer Interpretation, um die beobachteten Effekte richtig zu deuten (Bild 2.11). Bei der Messung wird das System direkt durch geeignete Meßeinrichtungen beobachtet. Das Sy-

stem kann dabei für die Messung stimuliert werden oder in seiner natürlichen Umgebung beobachtet werden. Neben diesen 'reinen' Methoden werden natürlich ebenfalls hybride Methoden mit der Ausnutzung der Vorteile der 'reinen' Methoden angewendet.

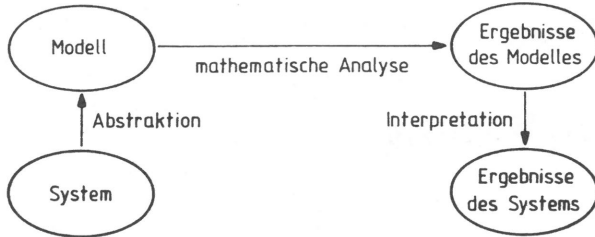
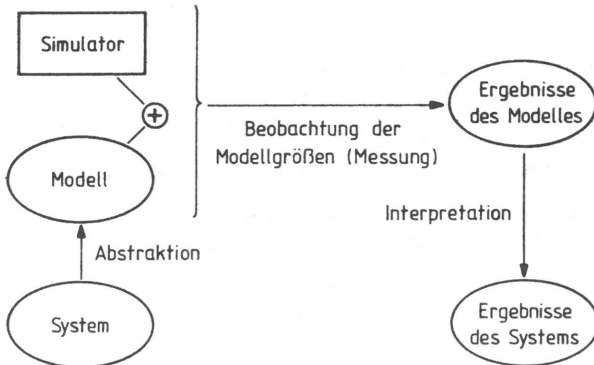
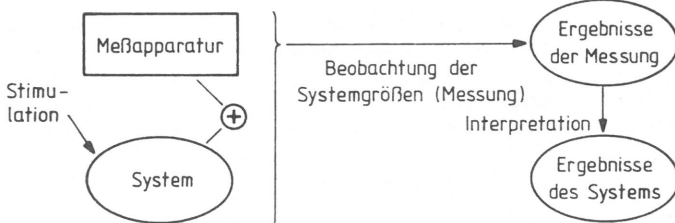
Mathematische AnalyseSimulationMessung

Bild 2.11: Methoden der Leistungsuntersuchung

## Kapitel 3

# Modellierung

Unter der Modellierung im technischen Sinn versteht man die Darstellung der allgemeinen und abstrakten Systemmerkmale. Die Modellierung abstrahiert daher die Realität und setzt sie in eine geeignete graphische, mathematische oder textliche Darstellung um. Sie wird eingesetzt zum tieferen Verständnis komplexer Systeme, und sie ermöglicht eine weitergehende Untersuchung von Systemeigenschaften.

### 3.1 Warteschlangenmodelle

Wenn sich das System durch Warteschlangen und Bedieneinheiten, die von abstrakten Aufträgen durchlaufen werden, beschreiben läßt, so kann zur Darstellung ein Warteschlangenmodell verwendet werden. Beispiele für solche Systeme sind:

- Abläufe in einer Datenverarbeitungsanlage,
- Kommunikation von Datenverarbeitungsanlagen oder
- Straßenverkehr.

In dieser Arbeit soll ausschließlich die Kommunikation von Datenverarbeitungsanlagen betrachtet werden.

#### 3.1.1 Elementare Warteschlangenmodelle

Das einfachste elementare Warteschlangenmodell besteht aus einer Warteschlange mit Warteraum und einer Bedieneinheit (siehe Bild 3.1).

Aufträge treffen bei der Warteschlange ein und warten dort, bis sie durch die Bedieneinheit bedient werden. Hat die Warteschlange einen endlichen Warteraum, so wird bei voller Warteschlange der ankommende Auftrag abgewiesen. Die Reihenfolge der Bedienung wartender Aufträge hängt von der Abfertigungsstrategie der Warteschlange ab. Die gebräuchlichsten Abfertigungsstrategien sind:

- FIFO ('First-In, First-Out')
- LIFO ('Last-In, First-Out') und
- zufällig (RANDOM).

Bei FIFO-Abfertigungsstrategie werden die Aufträge in der Reihenfolge ihres Eintreffens bedient. Bei LIFO-Abfertigungsstrategie wird die Reihenfolge umgedreht, so daß bei freiwerdender Bedieneinheit der zuletzt eingetroffene Auftrag bedient wird. Verdrängt ein

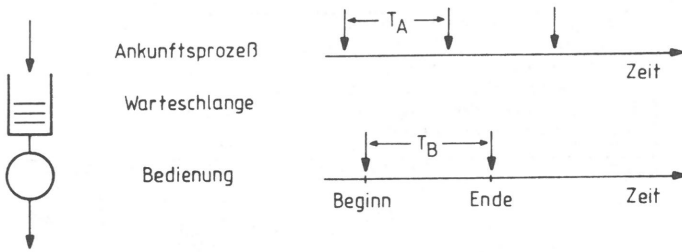


Bild 3.1: Elementares Warteschlangenmodell mit einer Bedieneinheit

ankommender Auftrag, bei der LIFO-Abfertigungsstrategie, den gerade bedienten Auftrag aus der Bedieneinheit in die Warteschlange und wird selbst bedient, so spricht man von der LIFO-PR-Abfertigungsstrategie (PR-‘Preemptive Resume’). Bei der Abfertigungsstrategie RANDOM wird bei freierwerdender Bedieneinheit ein beliebiger wartender Auftrag zur Bedienung ausgewählt.

Die Ankunft der Aufträge wird im Warteschlangenmodell als ein stochastischer Prozeß aufgefaßt und meist durch die Verteilungsfunktion der Ankunftsabstände  $A(t)$  beschrieben.

$$A(t) = P\{T_A \leq t\} \quad ; T_A\text{-Zufallsvariable des Ankunftsabstandes.} \quad (3.1)$$

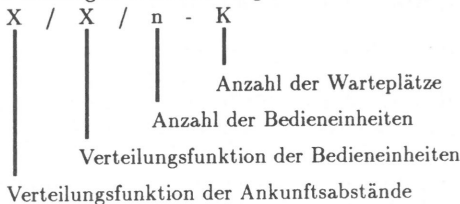
Die Bedienung eines Auftrages ist ebenfalls stochastisch und wird meist durch die Bedienzeitverteilungsfunktion  $B(t)$  beschrieben.

$$B(t) = P\{T_B \leq t\} \quad ; T_B\text{-Zufallsvariable der Bediendauer.} \quad (3.2)$$

Der wichtigste Ankunftsprozeß in der Warteschlangentheorie ist der Poisson-Prozeß, bei dem die Ankunftsabstände unabhängig voneinander sind und einer Negativ-Exponentiellen Verteilungsfunktion genügen.

$$A(t) = 1 - e^{-\lambda t} \quad ; \lambda\text{-Ankunftsrate} \quad (3.3)$$

Es hat sich in der Warteschlangentheorie die Kendallsche Symbolik zur Charakterisierung eines derartigen Warteschlangenmodells durchgesetzt.



In Tabelle 3.1 sind die gebräuchlichsten Verteilungsfunktionen aufgeführt.

Einige wichtige Warteschlangenmodelle sind:

- M/M/1- $\infty$ ,
- M/G/1- $\infty$ ,
- G/M/1- $\infty$ ,
- G/G/1- $\infty$  und
- G/G/ $\infty$  (wird auch als ‘Infinite Server’ bezeichnet).

Abkürzung	Name
D	Deterministische Verteilungsfunktion
$E_k$	Erlang-Verteilungsfunktion $k$ -ter Ordnung
M	Markov (Negativ Exponentiell)
GE	Verallgemeinerte Markov Verteilungsfunktion
$H_k$	Hyperexponentielle Verteilungsfunktion $k$ -ter Ordnung
G	Allgemeine Verteilungsfunktion

	Verteilungsfunktion	Mittelwert $E[T]$	Variationskoeffizient $c = \sqrt{\frac{E[T^2]}{E^2[T]} - 1}$	Laplace-Stieltjes Transformierte
D	$F_D(t) = \begin{cases} 0 & ; t < d \\ 1 & ; t \geq d \end{cases}$	d	0	$\frac{1}{s} e^{-sd}$
$E_k$	$F_{E_k}(t) = 1 - e^{-kx} \sum_{i=0}^{k-1} \frac{(kx)^i}{i!}$	$1/x$	$1/\sqrt{k}$	$\frac{1}{s} \left( \frac{x}{s+x} \right)^k$
M	$F_M(t) = 1 - e^{-xt}$	$1/x$	1	$\frac{1}{s} \frac{x}{s+x}$
GE	$F_{GE}(t) = 1 - re^{-rx}$ $0 < r \leq 1$	$1/x$	$\sqrt{2/r-1}$	$\frac{1}{s} \frac{rx + s(1-r)}{s+rx}$
$H_k$	$F_{H_k}(t) = 1 - \sum_{i=1}^k p_i e^{-x_i t}$ $\sum_{i=1}^k p_i = 1$	$\sum_{i=1}^k \frac{p_i}{x_i}$	$\sqrt{\frac{2 \sum_{i=1}^k p_i / x_i}{\left( \sum_{i=1}^k p_i / x_i \right)^2} - 1}$	$\frac{1}{s} \sum_{i=1}^k \frac{p_i x_i}{s+x_i}$

Tabelle 3.1: Verteilungsfunktionen und ihre Parameter

Treffen mehrere Ströme von Aufträgen, mit eventuell unterschiedlichen Bedienzeitforderungen, bei dem Warteschlangenmodell ein, so durchlaufen die Aufträge als unterscheidbare Klassen das Warteschlangenmodell. Werden den Klassen bei der Bedienung Prioritäten zugeordnet, so entsteht ein Prioritäts-Warteschlangenmodell nach Bild 3.2.

Jeder Klasse wird entsprechend ihrer Priorität eine Warteschlange und eine Bedienphase, die die Bedienzeitforderung dieser Klasse repräsentiert, zugeordnet. Bei der Bedienung werden nun unterbrechbare und nichtunterbrechbare Prioritäten unterschieden. Bei den ankommenden Aufträgen wird je Klasse zwischen unterbrechend und nichtunterbrechend unterschieden.

Eine weitere Abfertigungsstrategie für Klassen von Aufträgen ist das Absuchen der Warteschlangen nach zu bedienenden Aufträgen ('Polling'), siehe Bild 3.3. Eine Klassifizierung der Polling-Warteschlangenmodelle kann nach folgenden Kriterien erfolgen (Hideaki Takagi [106]):

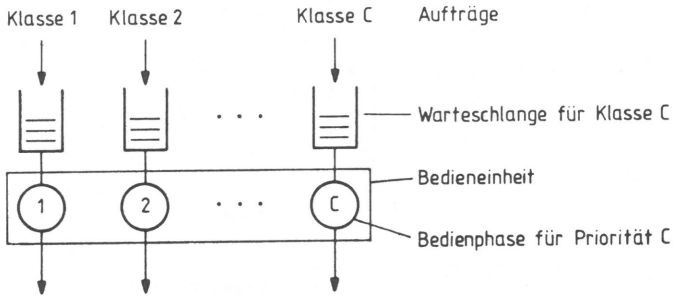


Bild 3.2: Prioritäts-Warteschlangenmodell

- ein Warteplatz oder unendlich viele Warteplätze,
- Bedienstrategie ('Exhaustive', 'Gated' oder 'Limited'),
- mit oder ohne Umschaltzeit zwischen den Warteschlangen,
- symmetrische oder asymmetrische Anordnung,
- zyklische oder nichtzyklische Reihenfolge der Bedienung und
- kontinuierliche oder nichtkontinuierliche Zeit.

Bei der 'Exhaustive'-Bedienstrategie werden alle Aufträge einer Warteschlange bedient, bis sie leer ist. Neue, während einer Bedienung, ankommende Aufträge sind darin eingeschlossen. Bei der 'Gated'-Bedienstrategie werden die Aufträge einer Warteschlange bedient, die sich darin befinden, wenn die Bedieneinheit beginnt, den ersten Auftrag zu bedienen. Es werden neu ankommende Aufträge nicht bedient. Bei der 'Limited'-Bedienstrategie wird höchstens eine bestimmte Anzahl von Aufträgen einer Warteschlange bedient.

Elementare Warteschlangenmodelle werden im Kontext von Warteschlangennetzen einfach Warteschlange oder auch Knoten genannt.

### 3.1.2 Warteschlangennetze

Warteschlangennetze bestehen aus den elementaren Warteschlangenmodellen mit eventuellen zusätzlichen Elementen zur Darstellung bestimmter Mechanismen. Die Aufträge werden nach der Bedienung in einer Warteschlange zur nächsten weitergegeben. Es entsteht somit ein Fluß von Aufträgen im Netzwerk. Es werden nach der Topologie drei Arten von Warteschlangennetzen unterschieden:

- geschlossene,
- offene und
- gemischte Warteschlangennetze.

Bei geschlossenen Warteschlangennetzen ist immer eine feste Anzahl von Aufträgen im Netzwerk, ohne daß Aufträge von außerhalb ankommen oder das Netzwerk verlassen. Bei offenen Netzwerken kommen die Aufträge von außerhalb an und verlassen das Netzwerk nach endlicher Zeit wieder. Existieren sowohl Aufträge, die im Netzwerk bleiben, wie auch

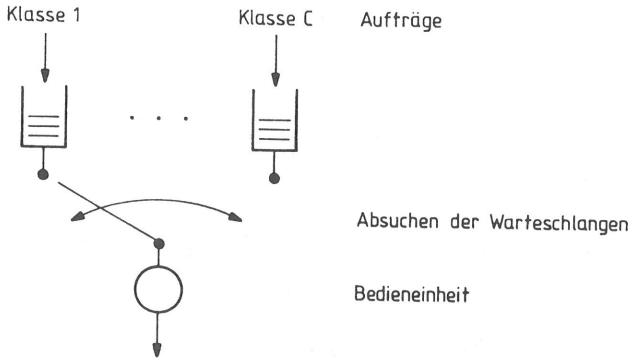


Bild 3.3: Polling-Warteschlangenmodell

Aufträge die von Außerhalb ankommen, so spricht man von gemischten Warteschlangennetzen. In Bild 3.4 sind die drei Typen schematisch dargestellt.

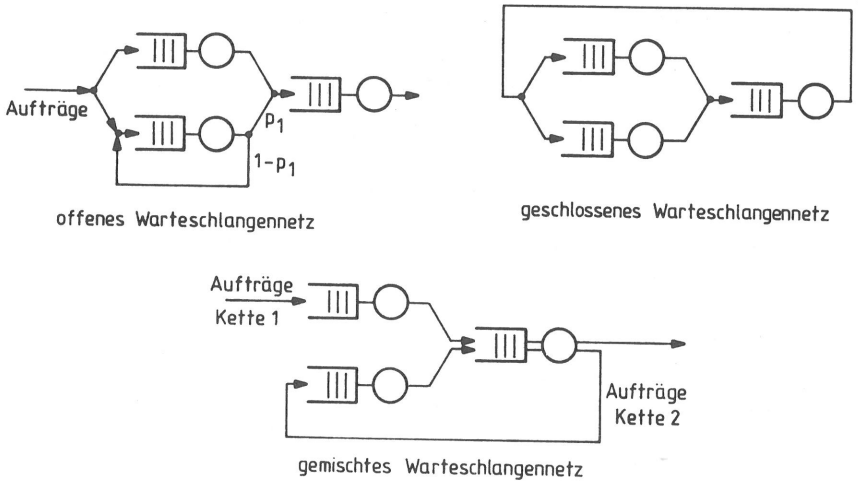
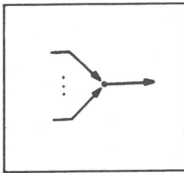


Bild 3.4: Typen der Warteschlangennetze

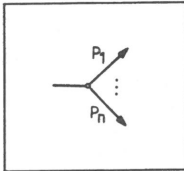
Der Weg von Aufträgen gleichen Typs mit allen möglichen Verzweigungen im Netzwerk wird als Kette bezeichnet. So sind im Bild 3.4 für das gemischte Netzwerk zwei Ketten enthalten. Für die Aufträge, die im Netzwerk bleiben, ist die geschlossene Kette 2 enthalten und für Aufträge, die von Außerhalb ankommen ist die offene Kette 1 enthalten. Ein Auftrag, der einer Kette angehört, kann bestimmte Klassen annehmen, die sich in ihrem Verhalten unterscheiden. Ein Klassenwechsel kann nur bei dem Wechsel von einer



Warteschlange zur nächsten Warteschlange geschehen. Außer dem Warteschlangensystem treten noch die folgenden beiden Elemente auf:



Zusammenführen von Aufträgen und



Verzeigen von Aufträgen der gleichen Kette gemäß den Wahrscheinlichkeiten  $p_1, \dots, p_n$ .

### 3.1.2.1 Produktform-Warteschlangennetze

Diese ausgezeichnete Klasse von Netzwerken hat sich in der Warteschlangentheorie als überaus wichtig herausgestellt, da für diese Klasse exakte geschlossene Lösungen existieren. Die Einschränkungen bestehen in den erlaubten Warteschlangensystemen und den Ankunftsprozessen [5]. Die Warteschlangen werden in diesem Zusammenhang auch Knoten genannt. Die Warteschlangen müssen die Eigenschaft der 'Local Balance' [15] aufweisen. Die gebräuchlichsten Warteschlangen sind:

- M/M/1- $\infty$  mit FIFO-Abfertigungsstrategie
- M/G/1- $\infty$  mit LIFO-PR-Abfertigungsstrategie
- M/G/ $\infty$  ('Infinite Server') und
- 'Processor Sharing'.

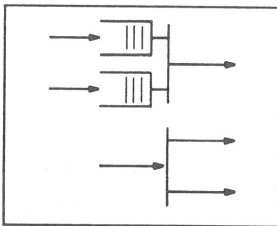
Bei 'Processor Sharing' wird die Bedieneinheit virtuell auf alle in der Warteschlange vorhandenen Aufträge gleichmäßig aufgeteilt. Datenverarbeitungsanlagen, die nach dem Zeitschlitzverfahren arbeiten, können damit modelliert werden. Sowohl offene, geschlossene wie auch gemischte Netzwerke mit mehreren Ketten und mehreren Klassen pro Kette können exakt berechnet werden. (Bei M/M/1- $\infty$  FIFO Knoten müssen allerdings alle Klassen pro Knoten die gleiche Bedienzeitanforderung haben!) Die Bedienzeitverteilungsfunktion kann außerdem abhängig von der Population in dem Warteschlangenmodell sein [98]. Als externe Ankunftsprozesse sind ausschließlich Poisson-Prozesse erlaubt, die allerdings in der Ankunftsrate von der momentanen Population des Gesamtnetzwerkes oder der Population ihrer Kette abhängig sein können. Der Name Produktform-Warteschlangennetze kommt von der Art der Lösung. Die Zustandswahrscheinlichkeit des Gesamtnetzwerkes wird als Produkt von Termen, die jeweils lediglich von einem Knoten abhängen, berechnet. Bei der Untermenge der offenen Netzwerke mit konstanten Ankunftsrate sind die Terme die Zustandswahrscheinlichkeiten der einzelnen Knoten mit einer geeigneten Normierungskonstanten. Bei geschlossenen Netzwerken werden die Zustandswahrscheinlichkeiten des offenen Netzwerkes für die möglichen Zustände des geschlossenen Netzwerkes durch die totale Wahrscheinlichkeit geeignet normiert. Die Berechnung der Normierungskonstanten ist dabei die Hauptschwierigkeit.

### 3.1.2.2 Allgemeine Warteschlangennetze

Läßt man die einschränkenden Bedingungen der Produktform-Warteschlangennetze fallen, so kommt man zu der Klasse der allgemeinen Warteschlangennetze. Vor allem die Erweiterung auf  $G/G/1-\infty$  FIFO-Warteschlangen ist von großer Bedeutung, da viele Systeme darauf abgebildet werden können. Leider gibt es im Rahmen der Warteschlangentheorie dafür keine exakten Lösungen. Es existieren jedoch für Teilbereiche Approximationen.

### 3.1.2.3 Erweiterte Warteschlangennetze

Um bestimmte Mechanismen in Systemen modellieren zu können, müssen oft weitere Elemente eingeführt werden. Auf der Ebene der Warteschlangen sind dies zum Beispiel das Prioritäts-Warteschlangenmodell oder das Polling-Warteschlangenmodell.



Ein weiteres Paar von Elementen ist die *Synchronisation* von zwei Aufträgen, die als ein Auftrag weiterlaufen, bis sie durch *Aufteilung* getrennt werden und dann wieder selbständige Wege laufen. Diese Mechanismen werden durch Symbole dargestellt, die aus der Petri-Netz-Theorie entlehnt sind.

## 3.2 Protokollmechanismen

### 3.2.1 Verbindungsorientierte Protokollmechanismen

#### 3.2.1.1 Flußregelung

**'Handshake'-Mechanismus** Der 'Handshake'-Mechanismus ist ein Sonderfall des Fenstermechanismus, wobei die Fenstergröße  $w = 1$  gewählt ist. Er wird hier separat behandelt, da er, im Vergleich zum allgemeinen Fenstermechanismus, einfach zu modellieren ist.

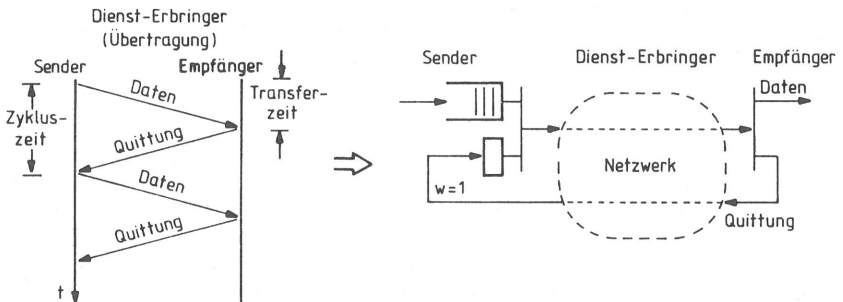


Bild 3.5: Darstellung des 'Handshake'-Flußregelmechanismus

In Bild 3.5 ist das Ablauf-Zeitdiagramm für den 'Handshake'-Mechanismus und das entsprechende Modell für eine Verbindung dargestellt. Für den Sender besteht eine Übertragung aus dem Zyklus vom Senden einer Dateneinheit bis zu dem Empfang der entsprechenden Quittierung. Nach dem Empfang der Quittierung kann die nächste Dateneinheit gesendet werden. Dies wird modelliert, indem bei dem Sender ein Synchronisationsmechanismus von Dateneinheit und 'Credit' (Erlaubnis zum Senden) eingesetzt wird. Dateneinheit und 'Credit' werden nun gemeinsam durch das Netzwerk, das den Dienst-Erbringer repräsentiert, geleitet. Beim Empfänger werden Dateneinheit und 'Credit' wieder getrennt, und das 'Credit' läuft als Quittierung wieder über das Netzwerk zum Sender zurück. Sind im Netzwerk keine weiteren Aufträge vorhanden, so kann das Netzwerk zu zwei Bedienphasen, entsprechend Bild 3.6, aggregiert werden.

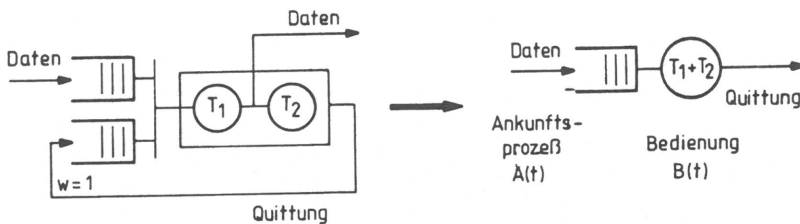


Bild 3.6: Vereinfachtes Modell für den 'Handshake'-Mechanismus

Damit läßt sich der 'Handshake'-Mechanismus im einfachsten Fall auf ein G/G/1- $\infty$  Warteschlangenmodell zurückführen. Werden mehrere Verbindungen über das Netzwerk geleitet, die sich gegenseitig beeinflussen, so kann diese Vereinfachung nicht vorgenommen werden, da die Durchlaufzeiten in jeder Warteschlange in komplexer Weise von den anderen Verbindungen abhängen. Die Modellierung muß dann entsprechend dem nachfolgenden Abschnitt durchgeführt werden.

Die Aggregation in obigem Modell ist für allgemeine Warteschlangennetze exakt durchführbar.

**Fenstermechanismus** In diesen Abschnitt sollen die Modelle für den gleitenden Fenstermechanismus ('Sliding Window') und den schritthaltenden Fenstermechanismus ('Pacing') dargestellt werden. Während das gleitende Fenster für unterschiedliche Umgebungsbedingungen dargestellt wird, wird für das schritthaltende Fenster das Grundmodell dargestellt, das entsprechend auf andere Umgebungsbedingungen übertragen werden könnte.

In Bild 3.7 ist ein Ablauf-Zeitdiagramm für die Fenstergröße  $w = 3$  und das Grundmodell einer flußregulierten Verbindung dargestellt. Dieses Modell kann nur für bestimmte Fälle weiter vereinfacht werden. Für die Fenstergröße  $w = 1$  entsteht das 'Handshake'-Protokoll. Für Netzwerke mit einer Warteschlange können ebenfalls Vereinfachungen durchgeführt werden.

In Bild 3.8 ist der Fenster-Flußregelmechanismus an den Modellen zweier einfacher Netzwerke dargestellt.

Werden mehrere flußregulierte Verbindungen, die sich gegenseitig beeinflussen, über das gleiche Netzwerk geleitet, so entsteht ein Modell nach Bild 3.9. Jede uni-direktionale

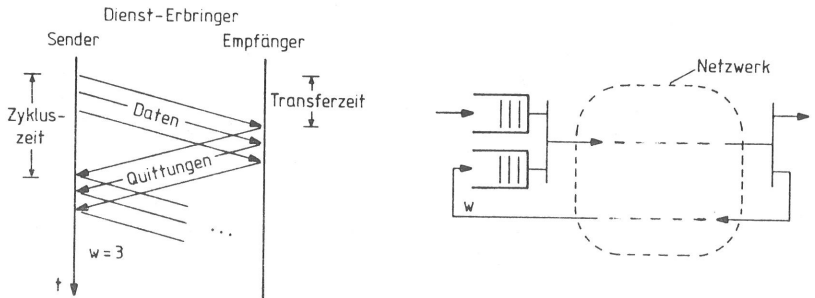


Bild 3.7: Grundmodell des gleitenden Fenstermechanismus ('Sliding Window')

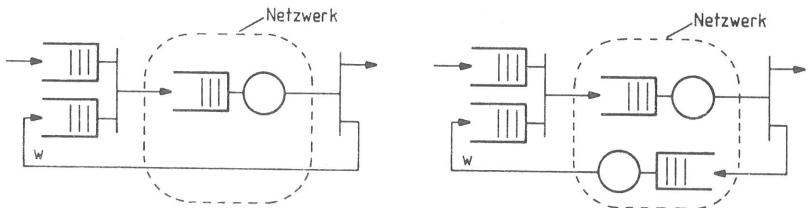


Bild 3.8: Grundmodell für zwei einfache Netzwerke

Verbindung (Halb-Duplex) ist separat mit einem eigenen Fenster  $w_i$  flußreguliert. Zwei uni-direktionale Verbindungen ergeben insbesondere wieder eine Voll-Duplex-Verbindung.

Für den schritthaltenden Fenstermechanismus ist im Bild 3.10 das Grundmodell dargestellt. Der Synchronisationsmechanismus befindet sich wiederum beim Sender. Beim Empfänger wird Auftrag und 'Credit' wieder getrennt und mit dem 'Credit' ein Zähler erhöht. Erreicht dieser Zähler den Wert  $K$ , so werden dem Sender weitere  $K$  'Credits' zur Verfügung gestellt. Dazu wird eine Quittierung über das Netzwerk geschickt, die der Sender dementsprechend interpretiert. Der Zähler wird beim Empfänger auf  $K - 1$  initialisiert und die Fenstergröße ist damit  $w = 2K - 1$ .

### 3.2.1.2 Fehlerbehebung

Die Modellierung der Fehlerbehebung kann oft vernachlässigt werden, wenn das Auftreten eines Fehlers bei der Übertragung selten ist. Fehler können sein:

- Verfälschung von Dateneinheiten,
- Verlust von Dateneinheiten,
- Anzeigen von Dateneinheiten in der falschen Reihenfolge und
- unbemerkter Abbau einer Verbindung.

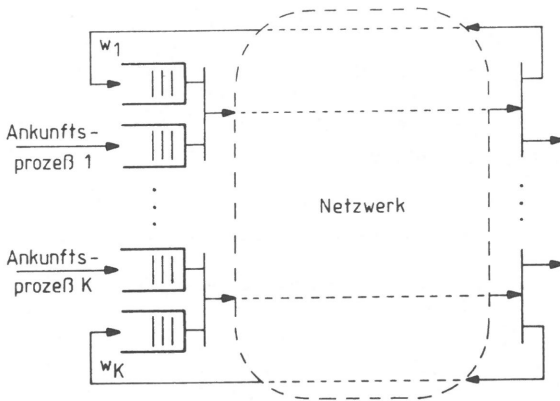


Bild 3.9: Modell für mehrere Verbindungen über das gleiche Netzwerk

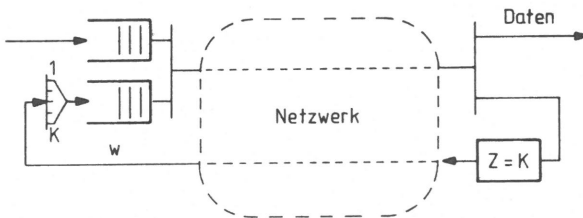


Bild 3.10: Grundmodell für den schritthaltenden Fenstermechanismus ('Pacing')

In den nächsten Abschnitten werden die wichtigsten Mechanismen zur Fehlerentdeckung und Fehlerbehebung modelliert.

**Verbindungsüberwachung** Die Verbindungsüberwachung (IC-'Inactivity Control') überwacht die Existenz einer Verbindung durch periodische Übertragungen, wenn keine anderen Dateneinheiten übertragen werden. Nach einer fest vorgegebenen Zeit  $d_w$ , die durch einen 'Timer' überwacht wird, wird eine redundante Dateneinheit übertragen (Bild 3.11). Abhängig von der Intensität der übertragenen Dateneinheiten und der Zeit  $d_w$  kann der Anteil der redundanten Dateneinheiten vernachlässigt werden, oder er bringt eine Grundlast bei ansonsten schwachem Verkehr auf die Verbindung.

**Einzelquittierung** Jede übertragene Dateneinheit wird einzeln positiv oder negativ quittiert. Bei einer negativen Quittierung wird die entsprechende Dateneinheit sofort quittiert. Fehler werden bei der Übertragung von Dateneinheiten zum Beispiel durch die Auswertung der Prüfsumme erkannt und durch eine negative Quittierung behoben. Fehler bei der Quittierung können erkannt aber so nicht behoben werden. In Bild 3.12 ist

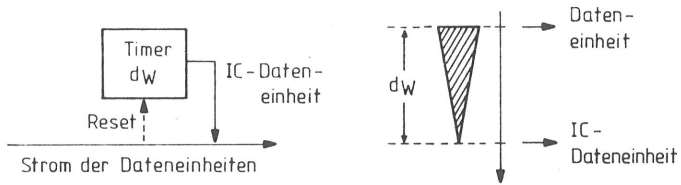


Bild 3.11: Modellierung des Verbindungsüberwachung

das Ablaufzeit-Diagramm und die Modellierung dargestellt.

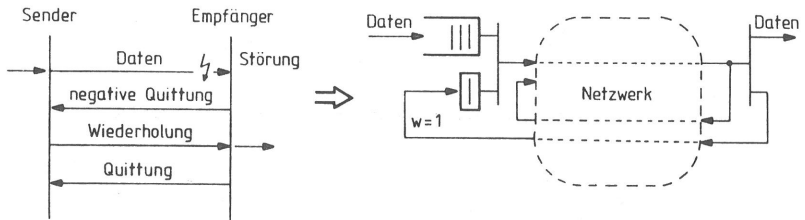


Bild 3.12: Modellierung der Einzelquittierung als Mechanismus zur Fehlerbehebung

**Reihenfolgeüberwachung** Alle übertragenen Dateneinheiten werden dazu beim Sender durchnummeriert. Der Empfänger kontrolliert die Reihenfolge und liefert nur Dateneinheiten in der richtigen Reihenfolge an die nächsthöhere Schicht. Trifft eine Dateneinheit in der falschen Reihenfolge ein, so wird von einem Verlust der fehlenden Dateneinheit ausgegangen. Der Fehler kann auf zwei Arten behoben werden.

1. Der Sender wird davon benachrichtigt und alle Dateneinheiten werden verworfen, bis die als nächste erwartete eintrifft ('Go-Back-N').
2. Der Sender wird benachrichtigt und alle eintreffenden Dateneinheiten werden zwischengespeichert, bis die fehlende Dateneinheit eintrifft (Bild 3.13, 'Selective Repeat').

Die erste Art kann effizient eingesetzt werden, wenn die Transferzeit der Aufträge und die Quittierungszeit sehr kurz sind, so daß in der Zwischenzeit nur wenige neue Aufträge gesendet werden. Dies tritt vor allem bei schmalbandigen terrestrischen Übertragungsstrecken oder bei den Vielfachzugriffsverfahren lokaler Netze mittlerer Bandbreite (10 MBit/s) auf. Verlängern sich die Transferzeiten, so verschlechtert sich die Effizienz dramatisch. Deshalb wird dann der 'Selective Repeat'-Mechanismus angewendet. Dies ist vor allem bei Satellitenübertragungskanälen oder breitbandigen terrestrischen Übertragungsstrecken der Fall. Es muß allerdings beim Empfänger genügend Pufferspeicher bereitgestellt werden, um Aufträge zwischenspeichern, bis der fehlerhafte Auftrag erfolgreich übertragen ist.

Sowohl Einzelquittierung als auch Reihenfolgeüberwachung mit 'Go-Back-N' oder 'Selective Repeat' werden zusammengefaßt unter dem Begriff ARQ-Protokolle ('Automatic Repeat Request') [100]. Im realen Einsatz kann es jeweils noch verschiedene Varianten von diesen Mechanismen geben.

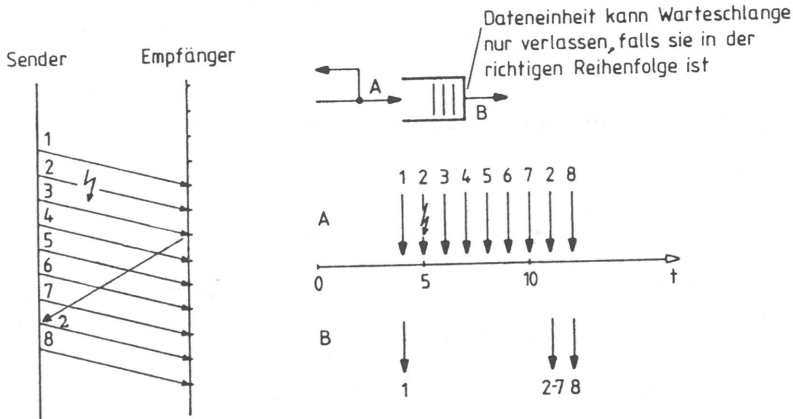


Bild 3.13: Modellierung der Reihenfolgeüberwachung beim Empfänger (z.B. 'Selective Repeat')

**Zeitüberwachung** Viele Aktionen während des Bestehens einer Verbindung bedürfen einer Zeitüberwachung, um zusätzliche Fehler bei der Übertragung oder bei Verklemmungs-Zuständen ('Deadlok') im Protokoll zu erkennen und zu beheben. Die Störung einer Quittierung ist ein solcher Fall, der unter dem Begriff Kontrollpunkt-Verfahren (HDLC [41]) bekannt geworden ist. Im allgemeinen wird dieser Mechanismus nicht modelliert, da er einen vernachlässigbaren Beitrag zum Verkehrsaufkommen ergibt und im Normalbetrieb selten auftreten sollte.

### 3.2.1.3 Multiplexen/Demultiplexen von Verbindungen

Durch das Multiplexen werden Verbindungen zusammengeführt. Dies entspricht in der Modellierung dem Zusammenführen von Verkehrsströmen und kann direkt abgebildet werden (Bild 3.14).

Falls die Aufträge im Netzwerk gleiche Bedienzeitanforderungen haben, kann die gemultiplexte Verbindung als eine Kette betrachtet werden, ansonsten bildet jede Verbindung eine separate Kette mit dem gleichen Weg durch das Netzwerk.

### 3.2.1.4 Aufspaltung/Sammlung von Verbindungen

Die Aufspaltung von Verbindungen kann entsprechend dem Multiplexen erfolgen, indem ein Verkehrsstrom auf mehrere Verkehrsströme aufgeteilt wird (Bild 3.15).

Haben die Aufträge nach der Aufspaltung im Netzwerk das gleiche Verhalten (Weg und Bedienzeitanforderung), so können die Verbindungen als eine Kette betrachtet werden. Ansonsten kann jede Verbindung im Netzwerk durch eine Kette repräsentiert werden.

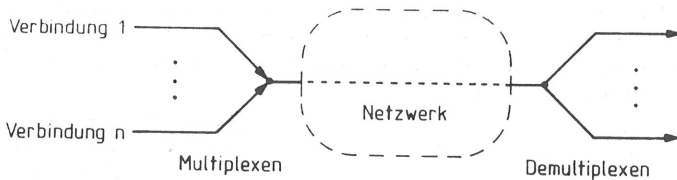


Bild 3.14: Modellierung von Multiplexen/Demultiplexen von Verbindungen

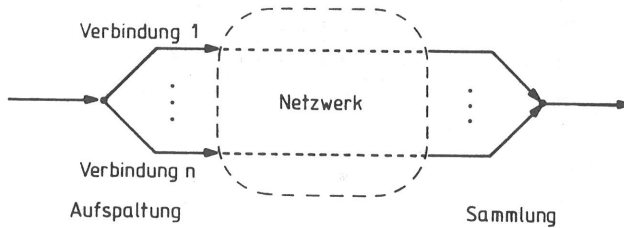


Bild 3.15: Modellierung von Aufspaltung/Sammlung von Verbindungen

### 3.2.1.5 Aufteilen/Vereinigen von Dateneinheiten

Die Aufteilung einer Protokoll-Dateneinheit in mehrere Dienst-Dateneinheiten hat eine Entsprechung in der Modellierung nach Bild 3.16.

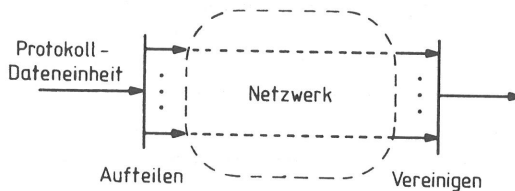


Bild 3.16: Modellierung von Aufteilen/Vereinigen von Dateneinheiten

### 3.2.1.6 Verketteten/Trennen von Dateneinheiten

Bei dem Verketteten von Dateneinheiten können mehrere Protokoll-Dateneinheiten zu einer Dienst-Dateneinheit zusammengefaßt werden, falls sie während einer Zeitspanne  $d_{r,k}$  eintreffen (Bild 3.17).



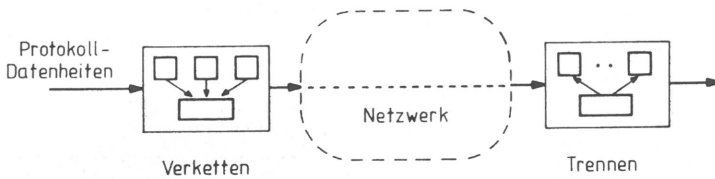


Bild 3.17: Modellierung von Verkettung/Trennen von Dateneinheiten

### 3.2.1.7 Blocken/Entblocken von Dateneinheiten

Dieser Protokollmechanismus ist von der Modellierung äquivalent mit Verkettung/Trennen. Protokoll- und Dienst-Dateneinheiten vertauschen lediglich ihre Rollen (siehe Bild 3.17).

## 3.2.2 Verbindungslose Protokollmechanismen

### 3.2.2.1 Flußregelung

**'Handshake'-Flußregelung** Die Modellierung eines 'Handshake'-Mechanismus bei verbindungsloser Kommunikation ist identisch mit der verbindungsorientierten Kommunikation (siehe Bild 3.5). Die Unterschiede liegen in der Interpretation. Während bei der verbindungsorientierten Kommunikation die Verbindungen nach dem Verbindungsaufbau als statisch angesehen werden, kann es bei verbindungsloser Kommunikation jederzeit zwischen beliebigen Stationen Übertragungen geben. Falls jedoch einschränkende Annahmen gemacht werden können (z.B. durch verbindungsorientierte Kommunikation auf den höheren Schichten), ist die Modellierung äquivalent.

**Flußregelung durch Ratenregelung** Die verbindungslose Flußregelung wird an dem Dienst-Zugangspunkt oder bezüglich der empfangenden Instanz jeweils beim Sender durchgeführt. Sie orientiert sich an

- dem Fluß der Dateneinheiten (Anzahl) oder
- der abstrakten Arbeit, die Dateneinheiten bei der Übertragung verursachen.

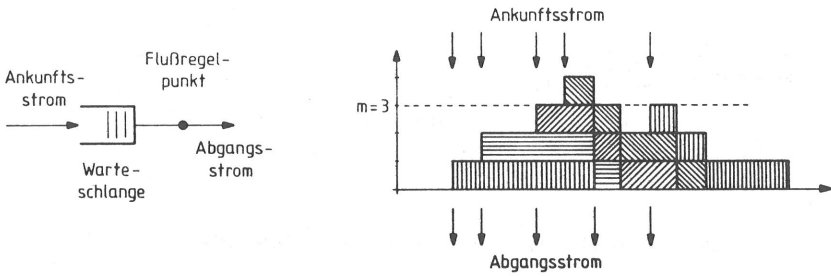
Wird die Flußregelung über den Fluß der Dateneinheiten durchgeführt, so können pro Zeitabschnitt  $d$  jeweils  $m$  Dateneinheiten den Flußregel-Punkt passieren (Bild 3.18).

Die Wartezeiten in der Warteschlange können durch ein  $G/D/m-\infty$  Warteschlangensystem modelliert werden (Bild 3.19).

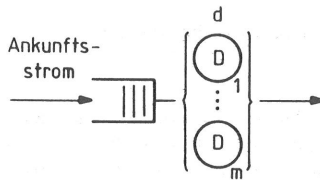
Bei der Ratenregelung über die Arbeit, die die Dateneinheiten in das System hineintragen, wird zum Zeitpunkt des Passierens eine Meßgröße für die Arbeit erhöht und mit der Zeit wieder kontinuierlich erniedrigt (Bild 3.20).

Das Passieren des Kontrollpunktes kann auf zweierlei Arten durchgeführt werden.

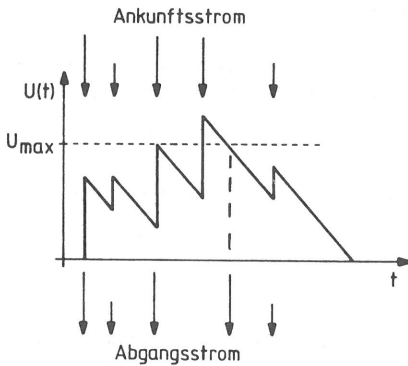
1. Der Auftrag darf den Kontrollpunkt passieren, wenn die unerledigte Arbeit vor ihm  $U_{max}$  unterschritten hat.
2. Der Auftrag darf den Kontrollpunkt passieren, sobald die unerledigte Arbeit, mit seiner eigenen Arbeitsanforderung,  $U_{max}$  unterschreitet. Es wird dabei vorausgesetzt, daß die Arbeitsanforderung des Auftrages  $U_{max}$  nicht überschreitet.



**Bild 3.18: Verbindungslose Flußregelung über den Fluß der Dateneinheiten**



**Bild 3.19: Modellierung der Ratenregelung über den Fluß der Dateneinheiten**



$U(t)$  Arbeit, die den Flußregel-punkt passiert hat oder in der Warteschlange steht

$U_{max}$  Maximale Arbeit im System



**Bild 3.20: Modellierung der Ratenregelung über die Arbeit (Fall 2)**

Die Modellierung erfolgt mit Hilfe eines  $G/G/1-\infty$  Warteschlangenmodelles, bei dem die Wartezeiten über einer gewissen Schranke  $U_{max}$  betrachtet werden.

### 3.2.2.2 Fehlerbehebung

Fehler können bei verbindungsloser Kommunikation lediglich an einzelnen Dateneinheiten festgestellt werden, da keine Beziehung zu anderen Dateneinheiten besteht. Wird eine fehlerhafte Dateneinheit beim Empfänger entdeckt, so kann eine Wiederholung veranlaßt werden. Dies setzt voraus, daß jede Dateneinheit eindeutig bestimmbar ist. Für die Fehlerbehebung wird ausschließlich die Einzelquittierung eingesetzt. Das Zugriffsverfahren 'Token-Bus' bei lokalen Netzen ist ein Beispiel hierfür.

Meist wird von einer Fehlerbehebung abgesehen, die Dateneinheit vernichtet, und der Fehler eventuell an die höhere Schicht weitergemeldet. Höhere Schichten verwenden oft verbindungsorientierte Protokolle. Sie entdecken dadurch die fehlende Dateneinheit und beheben den Fehler.

## 3.3 Protokolle

### 3.3.1 Verbindungsorientierte Protokolle

Das Grundmodell eines verbindungsorientierten Protokolles ist in Bild 3.21 dargestellt. Das Protokoll wird in den Instanzen abgewickelt und der Dienst-Erbringer stellt den bidirektionalen transparenten Transport von Dateneinheiten bereit.

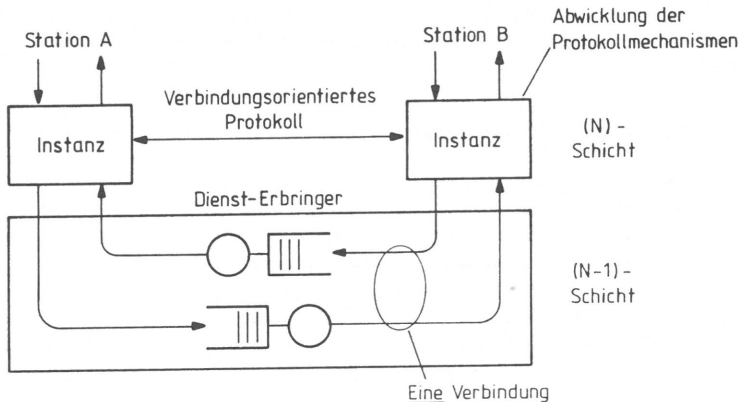


Bild 3.21: Grundmodell einer Verbindung

Meist ist es jedoch nicht möglich, eine Verbindung separat zu modellieren, da mehrere Verbindungen bestehen und sich gegenseitig beeinflussen. Das Warteschlangennetzwerk für den Dienst-Erbringer weist ebenfalls oft komplexere Strukturen auf.

### 3.3.2 Verbindungslose Protokolle

Bei verbindungslosen Protokollen besteht keine Vereinbarung zwischen Sender und Empfänger. Die Dienst-Zugangspunkte sind 'a priori' bekannt. Der Dienst-Erbringer kann wiederum als ein Paar von Warteschlangen von Instanz zu Instanz modelliert werden, die jedoch für alle übertragenen Dateneinheiten bereitstehen, während sie bei verbindungsorientierter Kommunikation für eine Verbindung reserviert sind (Bild 3.22).

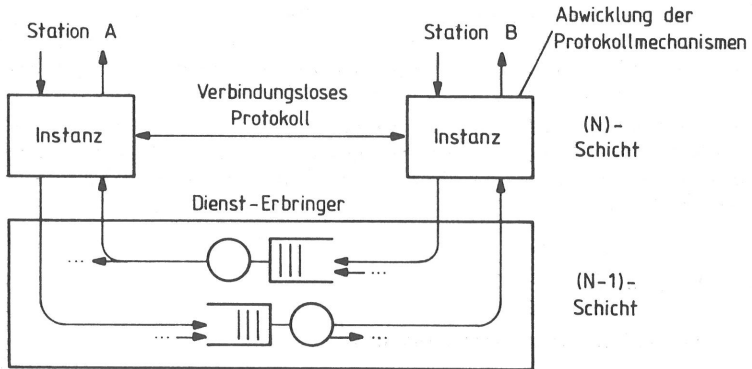


Bild 3.22: Grundmodell für verbindungslose Kommunikation

Bei der verbindungslosen Kommunikation werden alle Protokollmechanismen, die bei der sendenden Instanz ausgeführt wurden, bei der empfangenden Instanz wieder rückgängig gemacht, so daß immer unabhängige Sequenzen von Kommunikationsabschnitten entstehen.

## 3.4 Protokollarchitekturen

Nachdem in den vorhergehenden Kapiteln einzelne Protokolle modelliert wurden, sollen jetzt mehrschichtige Protokollarchitekturen betrachtet werden. Da die Modellierung eines Protokolles mit all seinen Protokollmechanismen im allgemeinen schon nicht mehr möglich ist, ist eine weitere Abstraktion bei Protokollarchitekturen notwendig.

In den nächsten Abschnitten werden das Beispiel der MAP-Protokollarchitektur diskutiert, eine zweischichtige flußregulierte Verbindung auf der (N)- und (N-1)-Schicht modelliert, unabhängige und abhängige Verbindungsabschnitte modelliert und zusätzliche Implementierungsgesichtspunkte angesprochen.

### 3.4.1 MAP-Architektur

Für die offene Fabrikkommunikation wird sich in absehbarer Zeit eine Protokollarchitektur nach Bild 3.23 durchsetzen. Firmenspezifische Lösungen werden sich diesem Standard anpassen. Für realzeitkritische oder kostengünstige Anwendungen könnte es eine abgemagerte Version mit lediglich den Schichten 2a (Kanalzugriff) und 7 (Anwendung) geben.

Ausgehend von der physikalischen Verbindung der Stationen über lokale Netze mit den Zugriffsverfahren Token-Bus [45], CSMA/CD [44] oder noch zu standardisierenden Verfahren mit niedrigeren Übertragungsraten unter dem Oberbegriff Feldbus, wird auf der Schicht 2b das verbindungslose Protokoll LLC1 [43] ('Logical Link Control Procedure, Type 1') eingesetzt (Bild 3.23). Da keine Vermittlungsfunktion innerhalb eines lokalen Netzes notwendig ist, wird auf der Schicht 3 das Netzwerkprotokoll [47] mit der transparenten Untermenge verwendet. Auf der Schicht 4 wird das komplexe verbindungsorientierte Transport-Protokoll [48] Klasse 4 angewendet. Auf der Schicht 5 werden die Grundfunktionen der Verbindungssteuerung eingesetzt [49], und auf der Schicht 6 findet eine Transformation der lokalen abstrakten Darstellung der Dateneinheiten auf eine einheitliche Transfersyntax [50, 51] statt. Die Anwendungsschicht ist aufgeteilt in zwei Unterschichten. In Schicht 7a werden gemeinsame, für unterschiedliche Schicht 7b-Protokolle notwendige Funktionen zusammengefaßt [54]. In der Schicht 7b bestehen nun mehrere Protokolle für unterschiedliche Anwendungen, wie 'File Transfer' [53], oder Prozeß-Steuerung [52].

Schicht 7b	FTAM, RS511 ...
Schicht 7a	Gemeinsame Funktionen (ACSE)
Schicht 6	Darstellung (Enkoder/Dekoder)
Schicht 5	Kommunikationssteuerung
Schicht 4	Transport, Klasse 4
Schicht 3	Netzwerk, 'Inactivity Set'
Schicht 2b	LLC 1
Schicht 2a	CSMA/CD, Token-Bus, Feldbus
Schicht 1	- "

Bild 3.23: Protokollarchitektur für Fabrikkommunikation

Betrachtet man die Protokoll-Architektur von dem Standpunkt der Protokollfunktionalitäten, so sind sie in den verbindungsorientierten Protokollen der Schichten 7b und 4, und dem verbindungslosen Kanalzugriffsprotokoll der Schicht 2a konzentriert. Werden alle weiteren Protokollmechanismen außer acht gelassen, so ergibt sich ein Modell nach Bild 3.24 mit CSMA/CD als Kanalzugriffsverfahren.

### 3.4.2 Mehrschichtige verbindungsorientierte Flußregelung

Als Beispiel für eine Modellierung mehrschichtiger Protokollarchitekturen soll in diesem Abschnitt die Flußregelung auf der (N)-Schicht mit dem verbindungsorientierten (N)-Protokoll und auf der (N-1)-Schicht mit dem verbindungsorientierten (N-1)-Protokoll dargestellt werden.

Auf der (N)-Schicht ist eine uni-direktionale Übertragung von Dateneinheiten und auf der (N-1)-Schicht die bidirektionale Übertragung von Dateneinheiten und Quittierungen der (N)-Schicht dargestellt. Die Fenstergrößen können unabhängig voneinander gewählt werden, und es ist das Ziel der Optimierung, den Parametern günstige Werte zuzuweisen.

In Bild 3.25 ist in der oberen Hälfte das abstrakte Modell für ein mehrschichtiges Verbindungskonzept dargestellt. In der unteren Hälfte ist ein detaillierteres Modell der mehrschichtigen verbindungsorientierten Flußregelung dargestellt.

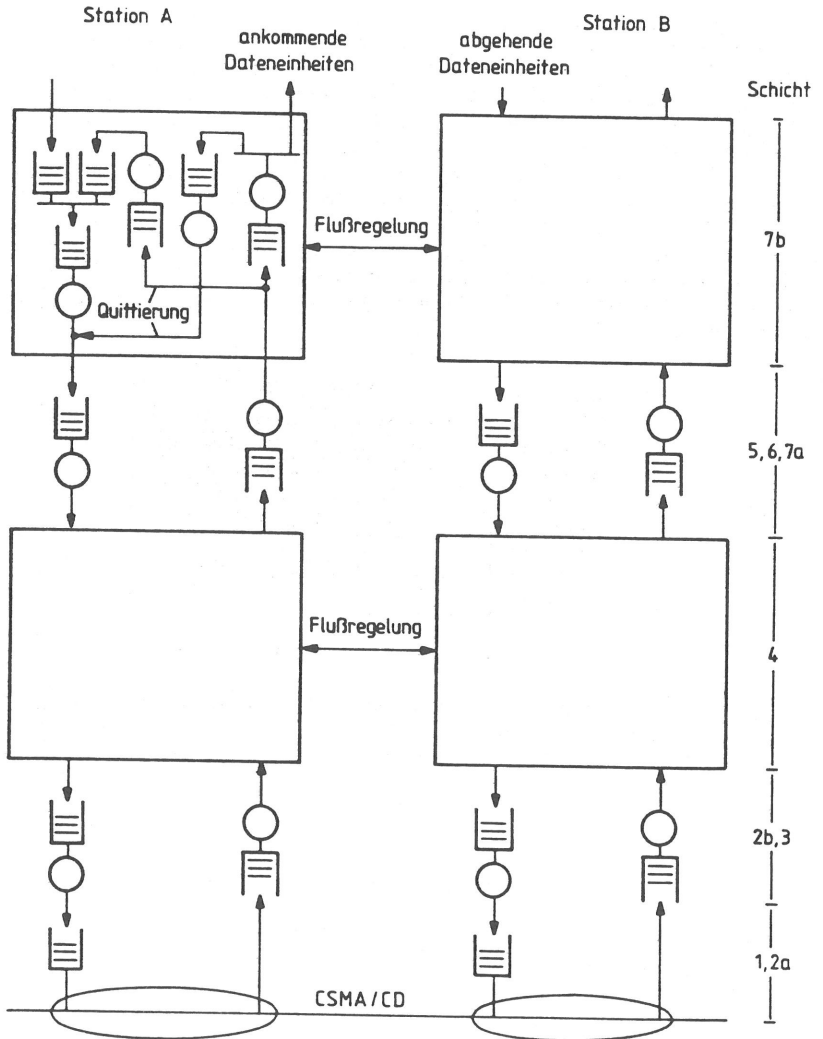


Bild 3.24: Modellierung der MAP-Architektur für die Flußregelung

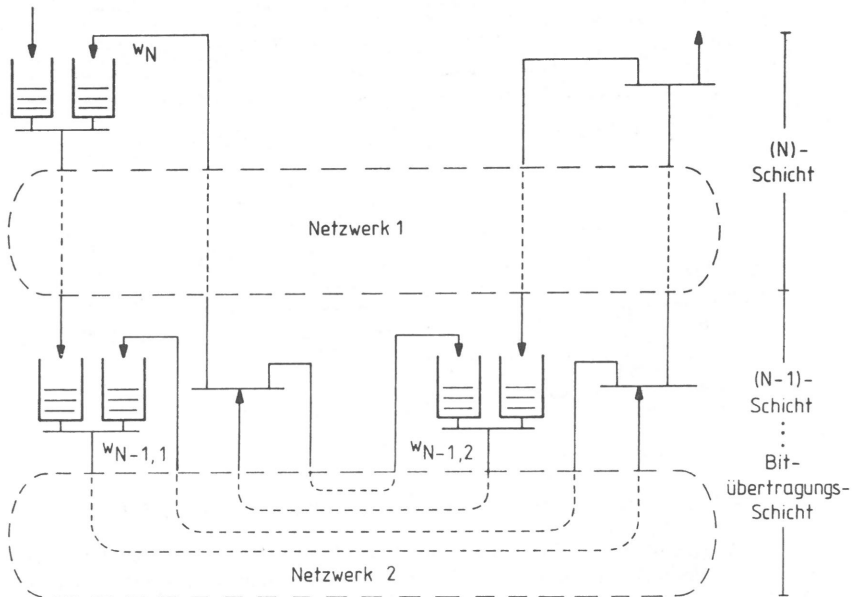
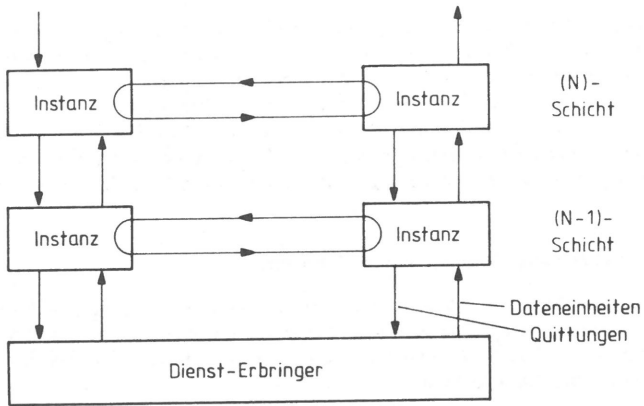


Bild 3.25: Modellierung einer zweischichtigen Flußregelung

### 3.4.3 Sequenzen von Verbindungen

Sequenzen von Verbindungen treten auf, wenn auf dem Übertragungsweg vom Sender zum Empfänger Vermittlungen stattfinden. Vermittlungen können innerhalb einer Protokollarchitektur erfolgen (Schicht 3) oder den Umstieg von einer Protokollarchitektur in eine andere bedeuten. Dieser Umstieg ist mit einer Protokollanpassung durch den Protokollkonverter ('Gateway') verbunden. In den nächsten beiden Abschnitten sollen Sequenzen von Verbindungen betrachtet werden. Die Verbindungen können dabei abhängig oder unabhängig voneinander sein. Dargestellt und modelliert werden sie anhand der Flußregelung.

#### 3.4.3.1 Unabhängige Sequenz von Verbindungen

Bei einer unabhängigen Sequenz von Verbindungen werden jegliche Protokolltransformationen wieder rückgängig gemacht, bevor die Dateneinheit der nächsten Verbindung übergeben wird. Auf die Flußregelung angewendet bedeutet dies, daß das 'Credit' sofort zum Sender zurückgeschickt wird.

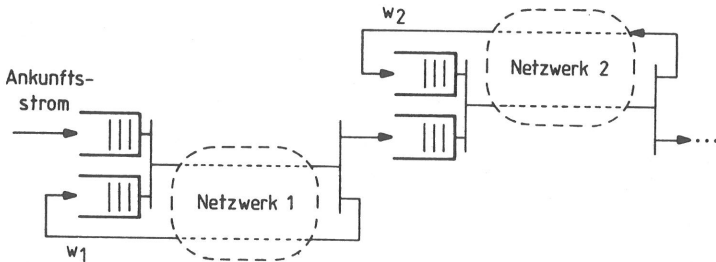
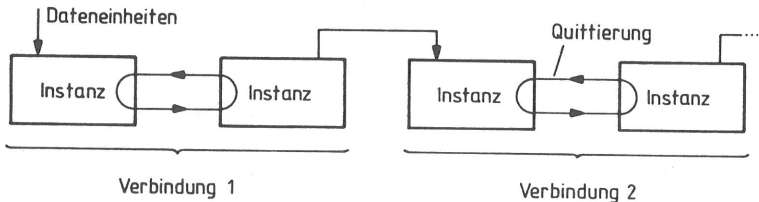


Bild 3.26: Modellierung der Flußregelung einer unabhängigen Sequenz von Verbindungen

In Bild 3.26 ist in der oberen Hälfte ein vereinfachtes Modell einer unabhängigen Sequenz von Verbindungen dargestellt. In der unteren Hälfte ist es für die Flußregelung detaillierter modelliert.

#### 3.4.3.2 Abhängige Sequenz von Verbindungen

Bei einer abhängigen Sequenz von Verbindungen werden bestimmte Protokolltransformationen erst bei der Ziel-Instanz rückgängig gemacht, beziehungsweise Zustandsübergänge



in der Instanz für die ankommende Verbindung sind abhängig vom Zustand der nächsten Verbindung. In Bild 3.27 ist für die Flußregelung der Fall dargestellt, daß die Quittierung erst zum Sender zurückgeschickt wird, wenn der Auftrag die Flußregelung der nächsten Verbindung passiert hat.

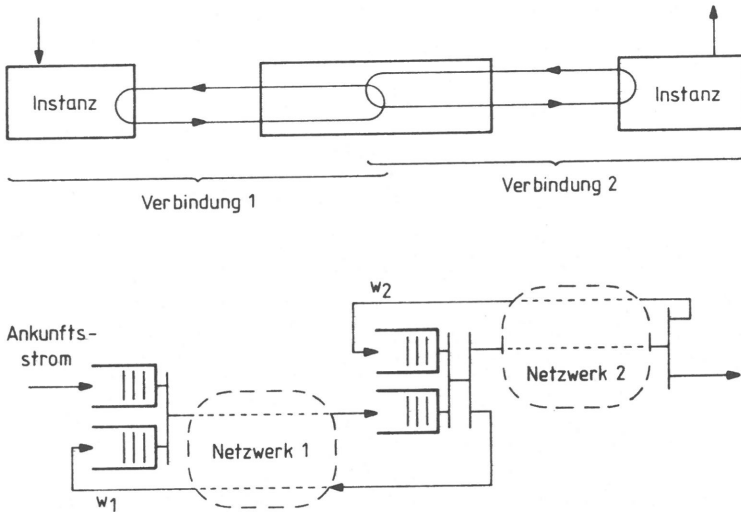


Bild 3.27: Modellierung der Flußregelung einer abhängigen Sequenz von Verbindungen

### 3.5 Einbeziehung von Implementierungsaspekten

Die bisher vorgestellten Modelle mehrschichtiger Protokollarchitekturen nahmen noch ungenügende Rücksicht auf die 'Hardware' und 'Software', in der die Protokolle implementiert sind. In den nächsten Abschnitten sollen einige dieser Gesichtspunkte näher betrachtet werden.

#### 3.5.1 Betriebssystem und Verarbeitungsinstanzen

Heutige Datenverarbeitungsanlagen bestehen meist aus mehreren Verarbeitungsinstanzen. Die Bearbeitung der Anwendungsprogramme erfolgt häufig in separaten Verarbeitungsinstanzen. Spezielle Aufgaben werden an dedizierte Instanzen, wie Ein- und Ausgabe zu Peripheriegeräten oder Kommunikation, weitergegeben. In Bild 3.28 ist die typische Konfiguration einer Datenverarbeitungsanlage für die Fabrikkommunikation oder eines 'Personal Computer' dargestellt.

Es stehen mehrere Verarbeitungsinstanzen für die Kommunikation zur Verfügung, und es kommt auf die Systemauslegung an, wie die einzelnen Aufgaben auf die Verarbeitungsinstanzen verteilt werden. Eine mögliche Aufteilung ist:

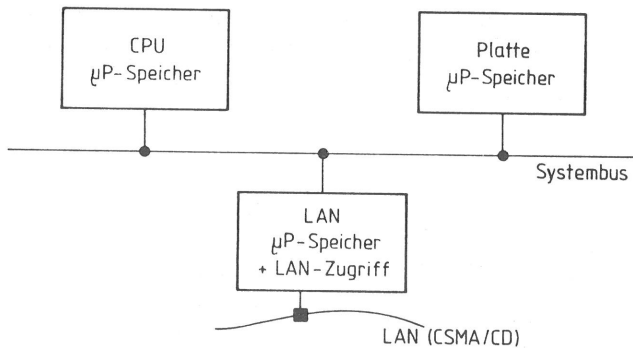


Bild 3.28: Typische Konfiguration einer kleinen Datenverarbeitungsanlage

- Schicht 2a auf einem Spezialprozessor für den Kanalzugriff,
- Schichten 2b-4 auf dem Mikroprozessor auf der LAN-Platine und
- Schichten 5-7 auf dem Prozessor der CPU-Platine zu bearbeiten.

Bei den Mikroprozessoren organisiert ein Betriebssystem den Ablauf der einzelnen Programme. Es sind im wesentlichen 'Time Sharing'- und Realzeit-Betriebssysteme zu unterscheiden. Bei den 'Time Sharing'-Betriebssystemen wird die zur Verfügung stehende Zeit in Zeitschlitze unterteilt, die den ablaufähigen Prozessen zugeordnet werden. Werden alle Aufgaben in einem gemeinsamen Prozeß implementiert, so unterliegt die weitere Organisation der Abarbeitungsreihenfolge dieses Prozesses. Verwendet werden zwei Strategien:

- Abarbeitung nach FIFO-Prinzip über alle Anforderungen oder
- Abarbeitung nach Prioritäten.

Bei dem Realzeit-Betriebssystem werden die zu bearbeitenden Aufgaben in einer strengen Prioritätsreihenfolge abgearbeitet.

Für die Bedieneinheiten in Bild 3.24 bedeutet dies, daß sie den Mikroprozessoren zugeordnet werden müssen und dort als Prozesse ablaufen oder durch Prioritäten gesteuert werden.

### 3.5.2 Ressourcenbeschränkungen

Zusätzliche Beschränkungen können auftreten, wenn Stationen ein hohes Verkehrsaufkommen haben und Ressourcen, wie Speicher oder 'Timer', nicht ausreichen. Diese Situationen sollten jedoch bei der Systemauslegung oder durch rechtzeitig wirkende Überlastabwehr verhindert werden. Sie werden hier, bei der Modellierung des Normalbetriebes, nicht berücksichtigt.

### 3.5.3 Aufruf- und Übergabemechanismen

Die Aufruf- und Übergabemechanismen sind abhängig vom Betriebssystem und man unterscheidet:

- Prozeß-zu-Prozeß Kommunikation über Briefkästen ('Mailbox') und
- Prozeduraufrufe mit Übergabe der Daten als Parameter.

Zusätzliche Aufwendungen sind notwendig, wenn die weitere Bearbeitung der Dateneinheiten bei einer anderen Verarbeitungsinstanz (Platine) ausgeführt wird.

### 3.5.4 MAP-Architektur als Beispiel

Wendet man diese Einschränkungen auf das MAP- Modell von Bild 3.24 an, so können die Bedieneinheiten nicht als unabhängig voneinander angesehen werden. Zu der weiteren Spezifizierung müssen noch die Fenstergrößen festgelegt werden und den Bedieneinheiten entsprechende Bedienzeitverteilungen zugeordnet werden. Es wird meist von einer M- oder D- Bedienzeitverteilung, mit dazu gehörigem Mittelwert, ausgegangen. Auf dem Übertragungsmedium überlagern sich die Verkehre aller Stationen, so daß die Gesamtkonfiguration für die genaue Modellierung des Kanalzugriffes betrachtet werden muß.

## Kapitel 4

# Grundlagen für die Leistungsuntersuchung

In diesem Kapitel sollen grundlegende Methoden für die Leistungsuntersuchung von technischen Kommunikationssystemen dargestellt werden. Für eine erfolgreiche Leistungsuntersuchung eines Systems ist das tiefe Verständnis für seine Funktionsweise unablässig. Dieses Verständnis wird dann in ein Modell umgesetzt, das je nach angewandeter Methode für die Leistungsuntersuchung im Detaillierungsgrad und den interessierenden Leistungsparametern unterschiedlich sein kann. Analytische Methoden der Warteschlangentheorie erlauben eine meist schnelle Antwort, wobei aber oft idealisierende Annahmen über das System und die Umwelt des Systems gemacht werden müssen. Durch die Simulation des Systemverhaltens und der Umwelt anhand ihrer Modelle können meist mehr Systemeigenschaften gleichzeitig in Betracht gezogen werden. Diese Simulationen werden auf Datenverarbeitungsanlagen durchgeführt, indem das Verhalten der Modelle durch ein Programm nachvollzogen wird. Die Leistungsgrößen werden durch Beobachtung der Modellzustände in Form von Meß- und Zählgrößen bestimmt und statistisch aufbereitet. Die Implementierung dieser Programme dauert meist Wochen bis Monate. Für bestimmte Klassen von Anwendungsproblemen werden zunehmend Simulationswerkzeuge entwickelt, mit denen eine schnellere Beantwortung der Fragen ermöglicht wird.

Als abschließende Methode bietet die Messung an dem realen System, unter simulierter oder realer Umwelt, die wirklichkeitstreuesten Leistungsuntersuchungen. Für Messungen ist aber ein hoher Aufwand an Meßausrüstung und Zeit notwendig. Durch die Einsetzbarkeit und die Aufwände ergibt sich, daß analytische Methoden meist im Vorfeld, simulative Methoden entwicklungsbegleitend und Messungen nach Fertigstellung eines Prototypen oder des endgültigen Systems eingesetzt werden.

### 4.1 Analytische Methoden der Warteschlangentheorie

Die Warteschlangentheorie wird als Teilgebiet der Analyse stochastischer Prozesse angesehen, die in physikalischen, technischen und soziologischen Systemen große Bedeutung haben. In den nächsten Abschnitten sollen, ausgehend von den stochastischen Prozessen, elementare Warteschlangenmodelle und Warteschlangennetze, sowie einige zusätzliche Methoden betrachtet werden.

### 4.1.1 Stochastischer Prozeß

Unter einem stochastischen Prozeß  $\xi_t$  versteht man in der Warteschlangentheorie eine Familie von Zufallsvariablen über einem gemeinsamen Wahrscheinlichkeitsraum. Durch  $\xi_t$  wird der Zustand des Systems zur Zeit  $t$  in einer vorgegebenen Menge von Zuständen charakterisiert. Für die Warteschlangentheorie hat die Menge der homogenen markovschen Prozesse eine besondere Bedeutung. Sie zeichnet sich dadurch aus, daß das zukünftige Verhalten des Prozesses nur von dem augenblicklichen Zustand und nicht von der Vergangenheit abhängt. Dies wird allgemein als Gedächtnislosigkeit bezeichnet. Der homogene Charakter bewirkt zusätzlich, daß die Zustandsübergangsfunktion nicht von der Zeit, sondern nur vom momentanen Zustand und einer Zeitdifferenz abhängt. Sowohl der Zustandsraum als auch der Parameter Zeit ( $t$ ) können diskret oder kontinuierlich sein. Die häufigsten Anwendungen betrachten einen diskreten Zustandsraum bei kontinuierlicher Zeit. Wenige Anwendungen betrachten diskrete Zeit oder kontinuierlichen Zustandsraum. Für weiterführende Literatur siehe Dieter König, Dietrich Stoyan [70]. Die allgemeine Analyse wird über Differenzen-Gleichungen ausgeführt und ergibt die Chapman-Kolmogorov-Gleichungen sowohl für den zeitdiskreten als auch den zeitkontinuierlichen Bereich. Man unterscheidet weiterhin zwischen den Rückwärts- und Vorwärts-Gleichungen, die zu der Berechnung der Differenzen das letzte oder nächste Zeitintervall verwenden (Leonard Kleinrock [71]). In Gleichung 4.1 und 4.2 sind für inhomogene markovsche Prozesse die Chapman-Kolmogorov-Gleichungen angegeben.

#### Zeitdiskret

$$\begin{array}{cc} \text{Vorwärts} & \text{Rückwärts} \\ \underline{H}(m, n) = \underline{H}(m, n-1)P(n-1) & \underline{H}(m, n) = P(m)\underline{H}(m+1, n) \end{array} \quad (4.1)$$

$\underline{H}(m, n) = [p_{i,j}(m, n)]$  - mehrschrittige Übergangsmatrix zur Zeit  $m$   
 $P(n) = [p_{i,j}(n, n+1)]$  - einschrittige Übergangswahrscheinlichkeiten zur Zeit  $n$   
 $p_{i,j}(m, n) = P\{\xi_n = j | \xi_m = i\}$  - Übergangswahrscheinlichkeit

#### Zeitkontinuierlich

$$\begin{array}{cc} \text{Vorwärts} & \text{Rückwärts} \\ \frac{\partial \underline{H}(s, t)}{\partial t} = \underline{H}(s, t)\underline{Q}(t) & \frac{\partial \underline{H}(s, t)}{\partial s} = -\underline{Q}(s)\underline{H}(s, t) \end{array} \quad (4.2)$$

$\underline{H}(s, t) = [p_{i,j}(s, t)]$  - Übergangs-Wahrscheinlichkeitsmatrix  
 $\underline{Q}(t) = \lim_{\Delta t \rightarrow 0} \frac{P(t) - I}{\Delta t}$  - Matrix der Übergangsraten

Daraus lassen sich sowohl die zeitabhängigen wie auch die stationären Zustandswahrscheinlichkeiten angeben.

#### Zeitdiskret

$$\begin{array}{cc} \text{Zeitabhängig} & \text{Stationär} \\ \underline{\pi}^{(n+1)} = \underline{\pi}^{(n)}P(n) & \underline{\pi}(I - P) = 0 \end{array} \quad (4.3)$$

$\pi^{(n)} = P\{\xi_n = i\}$  - Zustandswahrscheinlichkeit  $i$  zur Zeit  $n$   
 $\underline{\pi}^{(n)}$  - Vektor der Zustandswahrscheinlichkeiten zur Zeit  $n$   
 $\underline{I}$  - Einheitsvektor

### Zeitkontinuierlich

Zeitabhängig

$$\frac{d\underline{\pi}}{dt} = \underline{\pi} \underline{Q}$$

Stationär

$$\underline{\pi} \underline{Q} = 0$$

(4.4)

$\underline{\pi}$  - Vektor der stationären Zustandswahrscheinlichkeiten

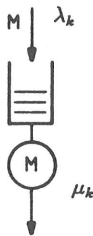
Die Auflösung dieser Gleichungssysteme wird vor allem bei abzählbar unendlich vielen Zuständen oft unmöglich, falls keine geeigneten Rekursionsalgorithmen gefunden werden können, die eine geschlossene Lösung ermöglichen.

#### 4.1.2 Elementare Warteschlangensysteme

Für einige stationäre Warteschlangenmodelle können anhand der Chapman-Kolmogorov-Gleichungen exakte Lösungen abgeleitet werden. Für zeitabhängige Systeme können nur in wenigen Fällen exakte Lösungen angegeben werden. Die Arbeit beschränkt sich auf stationäre Systeme, deren Analyse wesentlich einfacher ist und auf viele interessante Fragen Antworten liefern kann.

#### 4.1.3 M/M/1- $\infty$ Warteschlangensystem

Die zeitkontinuierlichen Chapman-Kolmogorov-Gleichungen können auf die M/M/1- $\infty$  Warteschlange (Bild 4.1) angewendet werden und ergeben für die Zustandswahrscheinlichkeiten die rekursive Beziehung:



$$p_{k+1} = \frac{\lambda_k}{\mu_{k+1}} p_k \quad (4.5)$$

$$p_k = p_0 \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}$$

$$p_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}}$$

$\lambda_k$  - Ankunftsrate im Zustand  $k$   
 $\mu_k$  - Enderate im Zustand  $k$

Bild 4.1: M/M/1- $\infty$  Warteschlange

#### 4.1.4 M/G/1- $\infty$ und G/M/1- $\infty$ Warteschlangensysteme

Die zeitkontinuierlichen Prozesse dieser Warteschlangen besitzen nicht die Markov-Eigenschaft. Der Prozeß besitzt beim M/G/1- $\infty$  System am Ende der Bedienung und beim G/M/1- $\infty$  System kurz vor der Ankunft des nächsten Auftrages Zeitpunkte, an denen das Gedächtnis sich auf den momentanen Zustand beschränkt. An diesen Punkten können ein zeitdiskreter Markov-Prozeß eingebettet und die Zustandswahrscheinlichkeiten  $r_k$  berechnet werden (Leonard Kleinrock [71]).

M/G/1- $\infty$

$$R(z) = \sum_{k=0}^{\infty} r_k z^k = (1 - \rho)(z - 1) \frac{K(z)}{z - K(z)}$$

$$K(z) = \sum_{k=0}^{\infty} k_k z^k = \sum_{k=0}^{\infty} \int_0^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} dB(t) z^k = \mathcal{B}^*(\lambda - \lambda z)$$

$R(z)$  - Erzeugende Funktion der Regenerationszeitpunkt-Zustandswahrscheinlichkeiten  $r_k$   
 $\mathcal{B}^*(s) \longleftrightarrow b(t)$  - Laplace-Stieltjes-Transformierte der Bedienzeitverteilungsdichtefunktion  $b(t)$

G/M/1- $\infty$

$$R(z) = \sum_{k=0}^{\infty} r_k z^k = \frac{1 - \sigma}{1 - \sigma z}$$

$$r_k = K * \sigma^k = (1 - \sigma)\sigma^k$$

$$\sigma = \mathcal{A}^*(\mu - \mu\sigma)$$

$\mathcal{A}^*(s) \longleftrightarrow a(t)$  - Laplace-Stieltjes-Transformierte der Verteilungsdichtefunktion der Ankunftsabstände

Die Größe  $\sigma$  muß aus Gleichung 4.7 (eventuell iterativ) bestimmt werden.

Aus den Zustandswahrscheinlichkeiten  $r_k$  zu den eingebetteten Punkten lassen sich die Zustandswahrscheinlichkeiten  $p_k$  zu beliebigen Zeitpunkten berechnen [20].

M/G/1- $\infty$

G/M/1- $\infty$

$$p_k = r_k \qquad p_k = \begin{cases} 1 - \rho & ; k = 0 \\ \rho r_{k-1} & ; k > 0 \end{cases} \quad ; \rho = \lambda/\mu$$

Die interessierenden Leistungsgrößen können dann aus den Zustandswahrscheinlichkeiten berechnet werden.

#### 4.1.5 G/G/1- $\infty$ Warteschlangensystem

Für die Analyse des G/G/1- $\infty$  Warteschlangensystems wird der Prozeß der unerledigten Arbeit  $U(t)$ , der an den Ankunftszeitpunkten einer neuen Anforderung einen eingebetteten Markov-Prozeß enthält, betrachtet. Die Wartezeitverteilung kann damit durch die Lindley-Integral-Gleichung (D.V. Lindley [84]), die vom Wiener-Hopf-Typ ist, angegeben werden.

$$\begin{aligned}
 W(t) &= \begin{cases} 0 & ; t < 0 \\ \int_0^\infty C(t-w)dW(w) & ; t \geq 0 \end{cases} \\
 C(u) &= \int_{t=0}^\infty B(u+t)dA(t)
 \end{aligned}
 \tag{4.7}$$

Diese Gleichung kann, bei rationalen Funktionen im Spektralbereich, über die Faktorisierung im Spektralbereich gelöst werden (Leonard Kleinrock [71]).

#### 4.1.6 Prioritäts-Warteschlangenmodelle

Aus der Menge der Prioritäts-Warteschlangenmodelle ist für die Modellierung von Verbindungskonzepten die Untermenge der  $M/G/1-\infty$  Modelle mit mehreren Klassen von Anforderungen und unterbrechenden oder nichtunterbrechenden Prioritäten pro Klasse von besonderer Bedeutung. Die Abfertigungsstrategie pro Klasse ist FIFO (Bild 4.2).

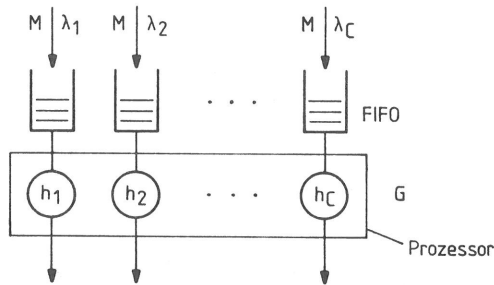


Bild 4.2: Prioritätsprozessormodell

Für diese wichtigen Modelle können die mittleren Wartezeiten pro Klasse  $w_p$  exakt angegeben werden [19, 63].

#### Unterbrechende Prioritäten:

$$\begin{aligned}
 w_{I,p} &= \underbrace{\sum_{i=1}^p \rho_i E[T_i^R]}_{\text{Restbedienzeit des gerade bedienten Auftrages}} + \underbrace{\sum_{i=1}^p n_{I,i} E[T_i^R]}_{\text{Restbedienzeit von unterbrochenen Aufträgen}} + \underbrace{\sum_{i=1}^p (n_i - n_{I,i}) h_i}_{\text{Bedienung wartender Aufträge}} + \underbrace{\sum_{i=1}^{p-1} (\lambda_i w_{I,p}) h_i}_{\text{Bedienung neu ankommender Aufträge}}
 \end{aligned}
 \tag{4.8}$$

- $w_{I,p}$  - Mittlere Wartezeit der Priorität  $p$  vor der 1. Bedienung  
 $h_i$  - Mittlere Bedienzeit der Priorität  $i$



- $n_{I,i}$  - Mittlere Anzahl unterbrochener Aufträge der Priorität  $i$   
 $\lambda_i$  - Ankunftsrate der Priorität  $i$   
 $E[T_i^R]$  - Mittlere Restbedienzeit der Priorität  $i$   
 $\rho_i = \lambda_i h_i$  - Auslastung der Priorität  $i$   
 $n_i$  - Mittlere Anzahl wartender Aufträge der Priorität  $i$

$$w_{I,p} = \frac{\sum_{i=1}^p \rho_i E[T_i^R]}{(1 - \rho_{(p)})(1 - \rho_{(p-1)})} \quad (4.9)$$

$$\rho_{(i)} = \sum_{j=1}^i \rho_j$$

$$w_p = w_{I,p} + \frac{\rho_{(p-1)}}{1 - \rho_{(p-1)}} h_p \quad (4.10)$$

#### Nichtunterbrechende Prioritäten:

$$w_p = \underbrace{\sum_{i=1}^C \rho_i E[T_i^R]}_{\text{Restbedienzeit des gerade bedienten Auftrages}} + \underbrace{\sum_{i=1}^p n_i h_i}_{\text{Bedienung wartender Aufträge}} + \underbrace{\sum_{i=1}^{p-1} (\lambda_i w_p) h_i}_{\text{Bedienung neu ankommender Aufträge}} \quad (4.11)$$

$$w_p = \frac{\sum_{i=1}^C \rho_i E[T_i^R]}{(1 - \rho_{(p)})(1 - \rho_{(p-1)})} \quad (4.12)$$

Für eine ausführliche Diskussion der Prioritätsprozessormodelle sei auf N.K. Jaiswal [63] verwiesen. Verallgemeinerte Modelle mit Unterbrechungs-Distanz-Prioritäten sind von Ulrich Herzog [42] und Wolfgang Krämer [75] angegeben worden.

#### 4.1.7 Pollingsysteme

Die Analyse der Polling-Warteschlangensysteme hat eine lange Tradition in der Warteschlangentheorie, die bis in die 50er Jahre zurückreicht. Dementsprechend gibt es eine Fülle an Literatur über dieses Gebiet. Bei Hideaki Takagi [106] ist eine hervorragende Übersicht über diese Literatur und den Stand der Forschung dargestellt.

Man kann die Art der Lösung von Polling-Warteschlangensystemen einteilen nach

- kontinuierliche oder diskrete Zeit und
- exakte oder approximative Lösung.

Viele Fragen, vor allem für diskrete Zeit, sind noch offen, so daß in den nächsten Jahren weitere Arbeiten auf diesem Forschungsgebiet notwendig sein werden. In dieser Arbeit sollen Pollingsysteme nicht weiter behandelt werden.

#### 4.1.8 Diffusions-Approximation

In den zwei Abschnitten 4.1.8 und 4.1.9 werden approximative Methoden der Analyse von Warteschlangensystemen dargestellt, die durch Verfahren aus der Physik angeregt wurden.

Bei der Diffusions-Approximation wird der Prozeß  $N(t)$  der im System befindlichen Aufträge betrachtet. Sowohl der Ankunftsprozeß  $\alpha(t)$  wie auch der Abgangsprozeß  $\delta(t)$  werden als zeitkontinuierliche Prozesse betrachtet, die zur Zeit  $t$  normalverteilt sind. Wird  $N(t)$  groß, so ist der Abgangsprozeß approximativ unabhängig von dem Ankunftsprozeß ([71, 32]). Die Chapman-Kolmogorov-Gleichung für den zeitkontinuierlichen und wertkontinuierlichen Bereich lautet:

$$F(x, t; y, \tau) = \int_{-\infty}^{\infty} F(w, u; y, \tau) d_w F(x, t; w, u). \quad (4.13)$$

$F(x, t; y, \tau)$  gibt die Wahrscheinlichkeit an, daß der Prozeß einen Wert  $\leq y$  zur Zeit  $\tau$  annimmt, wenn er zur Zeit  $t$  den Wert  $x$  annahm. Die Vorwärts- und Rückwärts-Kolmogorov-Gleichungen ergeben sich für zeitunabhängige Mittelwerte und Varianzen zu :

$$\begin{aligned} -\frac{\partial F}{\partial t} &= m \frac{\partial F}{\partial x} + \frac{1}{2} \sigma^2 \frac{\partial^2 F}{\partial x^2} & ; m &= \rho - 1 \\ -\frac{\partial f}{\partial t} &= m \frac{\partial f}{\partial w} + \frac{1}{2} \sigma^2 \frac{\partial^2 f}{\partial w^2} & ; \sigma^2 &= \lambda E[T_A^2]. \end{aligned}$$

Die Zustandswahrscheinlichkeiten für das G/G/1- $\infty$  System lassen sich durch Diskretisierung approximieren.

$$p_k = (1 - \hat{\rho}) \hat{\rho}^k \quad ; \hat{\rho} = e^{-2(1-\rho)/(\rho c_a^2 + c_b^2)} \quad (4.14)$$

$c_a^2, c_b^2$  - Quadrierter Variationskoeffizient der Verteilung der Ankunftsabstände und der Bedienzeit.

Hisashi Kobayashi [67] hat eine Verbesserung durch

$$p_k = \begin{cases} 1 - \rho & ; k = 0 \\ \rho(1 - \hat{\rho}) \hat{\rho}^k & ; k > 0 \end{cases} \quad (4.15)$$

vorgeschlagen, da  $p_0$  durch die Auslastung  $\rho = \lambda h$  bekannt ist. Die mittlere Anzahl von Aufträgen im System  $\bar{n}$  ergibt sich zu:

$$\bar{n} = \sum_{k=1}^{\infty} k p_k = \frac{\rho}{1 - \hat{\rho}} \approx \rho \left( 1 + \frac{\rho c_a^2 + c_b^2}{2(1 - \rho)} \right). \quad (4.16)$$

Der Fehler der Diffusions-Approximation strebt für  $\rho \rightarrow 1$  oder für Ankunftsabstand und Bedienzeit mit negativ exponentieller Verteilung (M/M/1- $\infty$ ) gegen 0.

#### 4.1.9 Maximum-Entropie-Approximation

Durch die Methode der maximalen Entropie kann diejenige Wahrscheinlichkeitsverteilung einer Zufallsvariablen  $X$  gefunden werden, die gewissen bekannten Einschränkungen genügt, aber ansonsten keine weiteren Annahmen enthält. Für eine diskrete Zufallsvariable  $X$  enthält der Ereignisraum endlich oder abzählbar unendlich viele Elemente  $\{x_0 \dots x_n\}$ . Die bekannten Informationen über die Zufallsvariable  $X$  sind Einschränkungen der Wahrscheinlichkeitsverteilung  $p_i = p(x_i)$  durch die Mittelwerte  $\{F_1 \dots F_m\}$  der Funktionen  $\{f_1 \dots f_m\}$ . (Die Beschränkung auf Mittelwerte von Funktionen ist nicht unbedingt notwendig [64].) Die Wahrscheinlichkeitsverteilung, die die Einschränkungen

$$\sum_{i=0}^n p_i = 1 \quad (4.17)$$

und

$$\sum_{i=0}^n p_i f_k(x_i) = F_k \quad ; k = 1 \dots m \quad (4.18)$$

berücksichtigt, aber von sonstigen Einschränkungen frei ist, ist durch Maximierung der Entropie-Gleichung

$$H = - \sum_{i=0}^n p_i \ln p_i \quad (4.19)$$

gegeben. Die Lösung dieser Gleichung ist durch

$$p_i = \frac{1}{Z(\lambda_1 \dots \lambda_m)} \exp(\lambda_1 f_1(x_i) + \dots + \lambda_m f_m(x_i)) \quad (4.20)$$

mit der Normierungskonstanten

$$Z(\lambda_1 \dots \lambda_m) = \sum_{i=0}^n \exp(\lambda_1 f_1(x_i) + \dots + \lambda_m f_m(x_i)) \quad (4.21)$$

prinzipiell bekannt. Die Konstanten  $\lambda_k$  sind durch die Einschränkungen

$$F_k = \frac{\partial}{\partial \lambda_k} \ln Z(\lambda_1 \dots \lambda_m) \quad (4.22)$$

festgelegt. Das Verfahren wurde inzwischen auf verschiedene Warteschlangensysteme von John E. Shore [101, 102], M. A. El-Affendi, Demetres D. Kouvatso [26], Demetres D. Kouvatso [72, 73] und Johann Christoph Strelen [104] angewendet. Zum Beispiel ergibt sich die Zustandsverteilung des G/G/1- $\infty$  Warteschlangenmodelles zu

$$p_i = \begin{cases} 1 - \rho & ; i = 0 \\ \frac{\rho^2}{\bar{n}} \left( \frac{\bar{n} - \rho}{\bar{n}} \right)^{i-1} & ; i > 0 \end{cases} \quad (4.23)$$

für die Einschränkungen  $\sum_{i=0}^{\infty} p_i = 1$ ,  $p_0 = 1 - \rho$  und  $\sum_{i=0}^{\infty} i p_i = \bar{n} = E[X]$ .

Obige Zustandsverteilung kann als Lösung des GE/GE/1- $\infty$  Warteschlangensystems, mit der verallgemeinerten negativ exponentiellen Verteilung, angesehen werden (siehe Anhang B).

$$\begin{aligned}
 \text{GE: } f(t) &= (1-r)\delta(t) + r^2 x \exp(-rxt) \\
 F(t) &= 1 - r \exp(-rxt) \\
 \delta(t) &= \begin{cases} \infty & ; t = 0 \\ 0 & ; t \neq 0 \end{cases} \text{ so daß } \int_{-\infty}^{+\infty} \delta(t) = 1
 \end{aligned}$$

Für diese Verteilungsfunktion kann über die Faktorisierung im Spektralbereich die Wartezeitverteilungsfunktion und damit die mittlere Anzahl von Aufträgen im System  $\bar{n}$  berechnet werden (Anhang A).

$$\bar{n} = \frac{\rho}{2} \left( 1 + \frac{c_a^2 + \rho c_b^2}{1 - \rho} \right) \quad ; \quad \frac{1 - c_a^2}{1 + c_b^2} \leq \rho < 1 \quad (4.24)$$

Aus informationstheoretischer Sicht stellt diese Lösung die beste Annahme dar, wenn von der Ankunftsabstandsverteilung und der Bedienzeitverteilung jeweils genau 2 Momente der Verteilungsfunktion bekannt sind. Demetres D. Kouvatso [73] hat eine Validierung dieser Lösung gegenüber einigen Verteilungsfunktionen durchgeführt. Er stellte eine gewisse Robustheit der Lösung, auch für die Fälle von  $\rho < (1 - c_a^2)/(1 + c_b^2)$ , fest. Die GE-Verteilungsfunktion existiert lediglich für  $c \geq 1$ .

#### 4.1.10 Warteschlangennetze

##### 4.1.10.1 Produktform-Warteschlangennetze

Die Zustandswahrscheinlichkeiten für Produktform-Warteschlangennetze lassen sich als ein Produkt über Terme darstellen:

$$P(\underline{n}) = \frac{1}{G} d(\underline{n}) f_1(\underline{n}_1) \dots f_M(\underline{n}_M). \quad (4.25)$$

$f_1(\underline{n}_1) \dots f_M(\underline{n}_M)$  - Terme der Knoten

$d(\underline{n})$  - Term abhängig vom Ankunftsprozeß

$G$  - Normierungskonstante

$\underline{n}$  - Zustandsvektor des Netzwerkes

$\underline{n}_i = (n_{i,1} \dots n_{i,j} \dots n_{i,k})$  - Zustandsvektor des Knotens  $i$  für alle Ketten

$n_{i,j}$  - Anzahl der Aufträge in Knoten  $i$  der Kette  $j$

Aus der prinzipiellen Kenntnis der Zustandswahrscheinlichkeiten wurden effiziente Algorithmen für komplexe offene, geschlossene oder gemischte Netzwerke entwickelt, mit denen die Leistungsgrößen berechnet werden können. Die Leistungsgrößen des offenen Netzwerkes mit konstanten Ankunftsraten lassen sich aus den Leistungsgrößen der einzelnen Knoten berechnen, nachdem der Fluß der Aufträge durch die Knoten bekannt ist (J.R. Jackson [55, 56]).

Für geschlossene Netzwerke ist der Term  $d(\underline{n}) = 1$ . Die Zustände der einzelnen Knoten sind voneinander abhängig und können nicht separat betrachtet werden. Die wichtigsten exakten Berechnungsalgorithmen sind

- Faltungsalgorithmus [12, 98],

- MVA ('Mean Value Analysis') [95] und
- LBANC ('Local Balance Algorithms for Normalizing Constants') [16].

Neben den exakten Algorithmen existieren sehr effiziente approximative Algorithmen, wie

- MVA-Iteration [95],
- LINEARIZER [17] und
- RECAL ('Recursion over chains') [21].

Gemischte Netzwerke lassen sich berechnen, indem ein äquivalentes geschlossenes Netzwerk konstruiert wird und die Lösung für das gemischte Netzwerk interpretiert wird [94]. Offene Netzwerke mit zustandsabhängigen Ankunftsraten lassen sich in ähnlicher Weise analysieren.

#### 4.1.10.2 Allgemeine Netzwerke

Bei der Analyse allgemeiner Netzwerke gibt es im wesentlichen zwei Ansätze für offene Netzwerke. Ein Ansatz wurde von Paul J. Kühn [78] entwickelt und ist als Dekomposition bekannt. Der zweite Ansatz wurde von E. Gelenbe [31] vorgestellt. Er verwendet die Diffusions-Approximation. Ein weiterer Ansatz verwendet die Maximum-Entropie-Approximation und kann sowohl offene als auch geschlossene Netzwerke lösen.

Bei offenen Netzwerken wird davon ausgegangen, daß jede Warteschlange als G/G/1- $\infty$  System, unabhängig von dem Rest des Netzwerkes, betrachtet werden kann. Ist der Ankunftsstrom und die Bedienzeitverteilung bei einem G/G/1- $\infty$  System bekannt, so kann die Abstandsverteilung des Abgangsstromes berechnet werden [88]. Wird der Ankunftsstrom und der Abgangsstrom mit 2 Momenten approximiert, so entsteht ein nichtlineares Gleichungssystem von Ankunftsströmen bei den G/G/1- $\infty$  Systemen, das iterativ gelöst werden kann. Die Berechnung der Leistungsgrößen der einzelnen G/G/1- $\infty$  Warteschlangen wird bei der Dekompositions-Approximation durch eine heuristische Formel von W. Kraemer und M. Langenbach-Belz [74] durchgeführt. Bei der Diffusions- und der Maximum-Entropie-Approximation [72, 73] werden die entsprechenden Zustandswahrscheinlichkeiten der einzelnen G/G/1- $\infty$  Systeme verwendet, um die Leistungsgrößen zu berechnen.

Für allgemeine geschlossene Warteschlangennetze hat Kouvatsos [72], mit der Maximum-Entropie-Methode, die Zustandswahrscheinlichkeit des Gesamtnetzwerkes als ein Produkt über geeignete Terme der Stationen angegeben.

$$P(\underline{n}) = \frac{1}{Z(N)} \prod_{i=1}^M g_i(n_i) x_i^{n_i} \quad (4.26)$$

$$g_i(n_i) = \begin{cases} 1 & ; n_i = 0 \\ \frac{\rho_i(1-x_i)}{(1-\rho_i)x_i} & ; n_i > 0 \end{cases} \quad (4.27)$$

$$x_i = \frac{\bar{n}_i - \rho_i}{\bar{n}_i} \quad (4.28)$$

$$Z(N) = \sum_{\text{alle } \underline{n}} \prod_{i=1}^M g_i(n_i) x_i^{n_i} \quad (4.29)$$

$\rho_i$  - Auslastung des Knotens  $i$

$\bar{n}_i$  - Mittlere Anzahl von Aufträgen in Knoten  $i$

Die mittlere Anzahl von Aufträgen  $n_i$  in den Warteschlangensystemen approximiert Demetres D. Kouvatso [72] durch ein GE/GE/1-K System. Für einige Beispielnetzwerke hat sich jedoch die GE/GE/1- $\infty$  Approximation als vorteilhafter erwiesen. Sie liefert für die Untermenge der Netzwerke, die auch der Produktform-Klasse angehören, exakte Ergebnisse. Die Normierungskonstante kann über einen ähnlichen Algorithmus berechnet werden, wie er für Produktformnetzwerke von J.P. Buzen [12] für zustandsabhängige Bedienraten verwendet wurde.

#### 4.1.10.3 Erweiterte Netzwerke

Für die Analyse erweiterter Netzwerke gibt es keine generellen Methoden. Die erweiterten Netzwerke werden durch Produktform- oder allgemeine Netzwerke approximiert. In den nächsten beiden Abschnitten werden Verfahren dargestellt, die bei Produktform-Netzwerken exakt sind und dort zur Verringerung der Komplexität oder zu Parameterstudien verwendet werden.

**Dekomposition / Aggregation** Diese Methode wurde von Chandy, Herzog und Woo [13] eingeführt und zur Reduzierung von allgemeinen Netzwerken auf geschlossene Netzwerke mit  $M/M/1-\infty$  Knoten angewendet [14]. Bei dieser Methode wird ein Teilnetzwerk durch einen flußäquivalenten Ersatzknoten mit zustandsabhängiger Bedienrate ersetzt (Bild 4.3). Die Autoren haben die Methode mit der Ersatzspannungsquelle (Norton-Theorem) aus dem Bereich elektrischer Schaltkreise verglichen. Hendrik Vantilborgh [111] hat für offene und geschlossene Netzwerke mit  $M/M/1-\infty$  Knoten und P.S. Kritzinger, S. van Wyk und A.E. Krzesinski [76] haben generell für Produktform-Netzwerke die Exaktheit der Methode nachgewiesen. Die Knoten, die zusammengefaßt werden, müssen nicht unbedingt einem Teilnetzwerk entstammen. Für den abstrakten Ersatzknoten existiert jedoch dann keine Interpretation mehr.

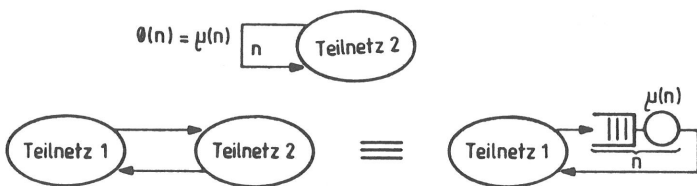


Bild 4.3: Dekomposition / Aggregation bei Warteschlangennetzen

P.J. Courtois [22] hat die Dekomposition, nach Vorbildern aus der Physik, auf sogenannte fast dekomponierbare Systeme ('Near Decomposable Systems') angewendet. Er hat hinreichende und notwendige Bedingungen für deren Anwendung angegeben. Exakte Dekomposition tritt nach Courtois bei Systemen auf :

- wenn keine Interaktion zwischen Teilsystemen stattfindet oder

- wenn die Interaktion zwischen Teilsystemen unabhängig von den Abläufen in den Teilsystemen sind.

Approximative Dekomposition, mit geringem Fehler, kann angewendet werden, wenn die Interaktionen in Teilsystemen wesentlich häufiger sind als die Interaktionen zwischen Teilsystemen. Diese Art der Systemabhängigkeit tritt oft bei hierarchischen Systemen auf.

**Erhaltung der Last ('Load Concealment')** Die Methode der Erhaltung der Last geht auf Martin Reiser und H. Kobayashi [94] zurück, die sie zur Analyse gemischter Netzwerke eingesetzt haben. Einzelne Ketten werden aus dem Netzwerk eliminiert, indem ein äquivalentes Netzwerk mit modifizierten Bedienraten für die Knoten konstruiert wird. Bei den Warteschlangen, die Produktform erfüllen, ändert sich bei einem 'Infinite Server' nichts und bei den anderen Knoten wird, bei nicht zustandsabhängiger Bedienrate, eine Anpassung nach Bild 4.4 vorgenommen. Für zustandsabhängige Bedienraten müssen spezielle Faltungsoptionen ausgeführt werden [94].

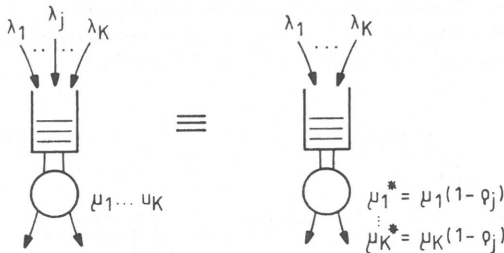


Bild 4.4: Erhaltung der Last

## 4.2 Simulation

Nachdem das zu untersuchende System in ein Modell abgebildet wurde, werden das Modell durch einen Simulator betrieben und die interessierenden Größen extrahiert. Für die in diesem Zusammenhang interessierenden technischen Kommunikationssysteme kann das Verhalten des Modells durch einen wertdiskreten, zeitkontinuierlichen stochastischen Prozeß dargestellt werden. Der Simulator verändert daher den Prozeßzustand anhand der Modellparameter und den stochastischen Variablen. Die Werte der interessierenden Größen werden einer statistischen Auswertung unterzogen. Die Simulation ist daher ein physikalisches Experiment an dem Modell.

Als Simulatoren werden heute fast ausschließlich Programme auf allgemeinen Datenverarbeitungsanlagen eingesetzt. Für gedächtnislose Prozesse wird vereinzelt die Monte-Carlo-Methode eingesetzt, bei der die weiteren Zustandsänderungen von der bisherigen Geschichte des Prozesses unabhängig sind und somit in jedem Zustand unabhängig ausgeführt werden. Die Zustände des Prozesses werden somit auf zufällige Weise ('Random Walk'), meist ohne Berücksichtigung der Zeit, durchlaufen. Im allgemeinen wird jedoch die zeitreue ereignisgesteuerte Simulationsmethode verwendet. Dabei wird eine mögliche Realisierung des Prozesses durchlaufen. Das Gedächtnis des Prozesses wird durch

den Simulator auf einen sogenannten Kalender abgebildet, der alle zum betreffenden Systemzeitpunkt bekannten zukünftigen Ereignisse in zeitlicher Ordnung enthält. Die bei der Simulation entnommenen und statistisch aufbereiteten Meßgrößen stellen Schätzwerte für die entsprechenden Prozeßgrößen dar. Bei George S. Fishman [28] findet man eine hervorragende Einführung in die Problematik der Systemsimulation, während Hisashi Kobayashi [68] spezieller auf Implementierungsaspekte eingeht. In den nächsten beiden Unterabschnitten wird vertieft auf die ereignisgesteuerte Simulationsmethode und die Implementierungsgesichtspunkte des Simulators eingegangen.

#### 4.2.1 Ereignisgesteuerte Simulation

##### 4.2.1.1 Simulationsmodell

Die Simulationsmodelle sind als erweiterte Warteschlangennetzwerke darstellbar. Der Abstraktionsgrad spiegelt sich in der Komplexität und den verwendeten Elementen des Modells wieder. Das Modell wird vom Simulator als stochastischer Prozeß aufgefaßt, der durch eine Menge von Zustandsvariablen beschrieben wird (Bild 4.5). Zu den Variablen, die den Prozeß beschreiben, kommen meist noch redundante Variablen, die die Auswertung bestimmter Systemgrößen erleichtern oder erst ermöglichen.

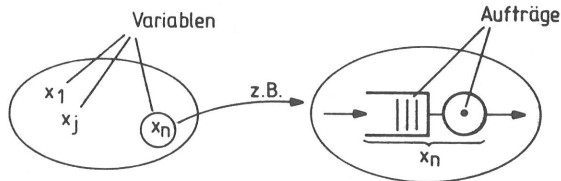


Bild 4.5: Simulationsmodell als stochastischer Prozeß

Meist ist der Detaillierungsgrad des Modells so groß, daß die Aufträge ebenfalls als Variablen mit oft komplexer Struktur beschrieben werden müssen. Werden bestimmte Meßgrößen, wie z.B. Lebensdauer des Auftrages, benötigt, so wird diese Repräsentation ebenfalls notwendig. Die Aufträge durchlaufen das Warteschlangennetzwerk und rufen komplexe Zustandsübergänge hervor.

##### 4.2.1.2 Ereignisbearbeitung

Das Simulationsprogramm vollzieht die Zustandsübergänge des stochastischen Prozesses zu den betreffenden Zeitpunkten. Tritt ein Zustandsübergang des stochastischen Prozesses auf, so werden alle betroffenen Variablen im Simulator geändert. Diese Zustandsübergänge entsprechen den Ereignissen im realen System. Insbesondere können Aufträge nach Bedienungsende in einer Warteschlange zur folgenden Warteschlange geleitet werden. Werden durch die momentane Ereignisbearbeitung Zustandsänderungen zu späteren Zeitpunkten ausgelöst, so werden diese Ereignisse in den Kalender, das Gedächtnis des Simulators, eingetragen. Im Kalender stehen somit alle zukünftigen Zustandsänderungen, die zum momentanen Zeitpunkt bekannt sind. Ist das Ereignis vollständig bearbeitet, so wird aus dem Kalender das zeitlich nächste Ereignis geholt, die simulierte



Systemzeit auf den neuen Zeitpunkt gesetzt und das Ereignis ebenfalls vollständig bearbeitet. In Bild 4.6 ist der Zusammenhang zwischen simulierter Systemzeit und von der Datenverarbeitungsanlage aufgewendeter Rechenzeit dargestellt.

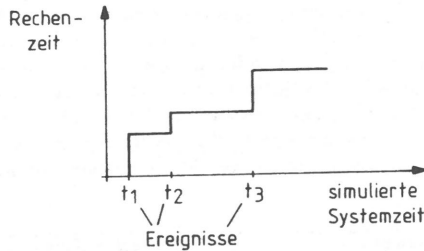


Bild 4.6: Zusammenhang zwischen Rechenzeit und simulierter Systemzeit

#### 4.2.1.3 Erfassung und Auswertung der Meßgrößen

Aus der Beobachtung der Zustandsvariablen oder der zusätzlichen Meßvariablen entsteht jeweils eine Sequenz von Meßwerten, die statistisch ausgewertet werden müssen, da die beobachteten Variablen stochastischen Charakter haben. Die Meßgrößen können in 4 Klassen eingeteilt werden:

- Messung der Momente einer Zufallsvariablen (meistens beschränkt auf Mittelwert und Variationskoeffizient),
- Zählung von ganzzahligen oder reellen Größen (Wahrscheinlichkeiten können insbesondere als Division zweier Zählwerte aufgefaßt werden),
- Integralmessung über der Zeit, wie zum Beispiel die Auslastung einer Bedieneinheit oder die mittlere Warteschlangenlänge und
- Häufigkeitsverteilung einer Zufallsvariablen.

Da statistische Messungen an Zufallsvariablen immer nur Schätzungen sein können, erhebt sich die Frage nach der Glaubwürdigkeit der Ergebnisse. Die gebräuchlichste Methode ermittelt Vertrauensintervalle zu den geschätzten Meßgrößen. Für eine ausführliche Diskussion der verwendeten Methoden siehe George S. Fishman [28].

Die Theorie der Vertrauensintervalle wurde von Gosset [105] entwickelt und beruht auf der Beobachtung von  $n$  unabhängigen Realisierungen einer normalverteilten Zufallsvariablen. Das Vertrauensintervall des Mittelwertes kann dann, bei unbekannter Varianz, über die Student-t-Verteilung mit  $n-1$  Freiheitsgraden abgeschätzt werden. Da die zu beobachtenden Zufallsvariablen meist keine Normalverteilung aufweisen, im allgemeinen die Verteilungsfunktion an sich unbekannt ist, wird eine Messung  $n$  mal durchgeführt. Die einzelnen Schätzwerte  $\hat{X}_i$  jeder Messung werden ihrerseits als Zufallsvariablen aufgefaßt.

$$\hat{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{i,j} \quad (4.30)$$

$x_{i,j}$  - j-te Realisierung der Zufallsvariablen X im Teilttest i  
 $\hat{X}_i$  - Schätzwert für den Mittelwert von X im Teilttest i  
 $n_i$  - Anzahl der berücksichtigten Realisierungen von X im Teilttest i

Da  $\hat{X}_i$  als Summe von Zufallsvariablen angesehen werden kann, ist  $\hat{X}_i$  nach dem zentralen Grenzwertsatz der Wahrscheinlichkeitsrechnung normalverteilt. Damit kann die Theorie der Vertrauensintervalle auf  $\hat{X}_i$  angewendet werden. Für die Berechnung des Vertrauensintervalles für den Variationskoeffizienten (oder die Varianz) müssen der Schätzwert für die Varianz des geschätzten Variationskoeffizienten  $\hat{\sigma}_i^2$  und der geschätzte Variationskoeffizient unabhängig voneinander sein (siehe George S. Fishman [28]). In Tabelle 4.1 sind Schätzfunktionen, mit den dazugehörigen Vertrauensintervallen, für einige Meßgrößen angegeben.

X	Schätzwerte der Parameter	Teilttestschätzwerte	Vertrauensintervall (VI)
$E[X]$  (Mittelwert)	$\hat{E}[X] = \hat{X} = \frac{1}{n} \sum_{i=1}^n \hat{X}_i$  $S_{\hat{X}}^2 = \frac{1}{n-1} \sum_{i=1}^n (\hat{X} - \hat{X}_i)^2$	$\hat{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{i,j}$	$VI = t_{\alpha/2; n-1} \frac{S_{\hat{X}}}{\sqrt{n}}$
$VAR[X] = \sigma^2$  (Varianz)	$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n \hat{\sigma}_i^2$  $S_{\hat{\sigma}^2}^2 = \frac{1}{n-1} \sum_{i=1}^n (\hat{\sigma}_i^2 - \hat{\sigma}^2)^2$	$\hat{\sigma}_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (x_{i,j} - \hat{X}_i)^2$	$VI = t_{\alpha/2; n-1} \frac{S_{\hat{\sigma}^2}}{\sqrt{n}}$

$t_{\alpha/2; n-1}$  - Student-t-Wert für des Vertrauensgrad  $\alpha$  bei  $n - 1$  Freiheitsgraden  
 $\hat{X}_i^*$  muß unabhängig von  $\hat{X}_i$  sein! (Z.B.  $\hat{X}_i^* = \hat{X}_{i-1}$ )

Tabelle 4.1: Schätzwerte für Mittelwert und Varianz einer Zufallsvariablen

#### 4.2.1.4 Simulationsablauf

Bei der in 4.2.1.3 geschilderten Methode zur Bestimmung der Vertrauensintervalle, wird der Simulationslauf in einzelne Teilttests eingeteilt. Jeder Teilttest liefert einen Schätzwert pro Meßgröße, die zu den endgültigen Ergebnissen zusammengefaßt werden. Soll die Simulation im stationären Bereich des Prozesses durchgeführt werden, so muß dieser Bereich erreicht werden, bevor der erste Schätzwert entnommen wird. Man bezeichnet dies auch als Vorlauf. Die Abschätzung, wann der Prozeß im stationären Bereich ist, kann in der Praxis oft problematisch sein. Ist der Prozeß im stationären Bereich, so können die Teilttests nacheinander ohne zusätzliche Vorläufe durchgeführt werden (siehe Bild 4.7). Diese Methode wird als 'Single Run Replication' bezeichnet.

Eine weitere Methode verwendet die gedächtnislosen Punkte in einem Prozeß zur Einteilung der Teiltests. Die gesammelten Meßwerte in einem Intervall zwischen gedächtnislosen Punkten sind statistisch unabhängig vom nächsten Intervall. Diese Methode wird als 'Regeneration' bezeichnet.

#### 4.2.1.5 Implementierungsgesichtspunkte

Für die Durchführung der Simulation auf einer allgemeinen Datenverarbeitungsanlage stehen drei Möglichkeiten zur Verfügung:

- Verwendung eines allgemeinen Simulationstools,
- individuelle Implementierung des Simulationsprogrammes oder
- Implementierung mit Hilfe einer 'Modul-Bibliothek', die immer wiederkehrende Aufgaben abnimmt.

In Tabelle 4.2 sind Vor- und Nachteile der drei Möglichkeiten dargestellt.

Kriterium	Simulations-tool	Modul-Bibliothek	individuelles Simulationsprogramm
Entwicklungszeit	Kurz	Relativ kurz	Lang
Benutzeroberfläche	Grafisch	Dem Problem angepaßt	Individuell dem Problem angepaßt
Programmiersprache	-	Allgemeine Programmiersprache	Allgemeine Programmiersprache
Auf neue Modellkomponenten erweiterbar	Nein	Ja (Durch Spezialisten)	Ja (Meist nur durch Entwickler)
Mögliche Komplexität	Mittel	Hoch	Mittel-Hoch
Rechenzeitbedarf	Hoch	Niedrig	Niedrig
Speicherplatzbedarf	Hoch	Niedrig	Niedrig

Tabelle 4.2: Vor- und Nachteile der Alternativen bei der Durchführung der Simulation

Durch die Komplexität der Modelle und der speziellen Modellelemente, die in der vorliegenden Arbeit verwendet wurden, wurde die Verwendung eines Simulationstools sofort ausgeschlossen. Während der Arbeit wurden wesentliche Bestandteile der Modul-Bibliothek entwickelt. Die Grundzüge sind in den folgenden Abschnitten dargestellt. Für eine ausführlichere Diskussion sei auf Ottmar Gühr und Gert Willmann [37] verwiesen.

#### 4.2.1.6 Aufbau der Modul-Bibliothek

Die Grundzüge der Modul-Bibliothek orientieren sich an einem individuellen Simulationsprogramm, das für allgemeineren Gebrauch implementiert wurde. Die wesentlichsten Gesichtspunkte sind :

- Verwendung dynamischer Datenstrukturen,
- Verlagerung der Abläufe aus dem Programmcode in die Datenstruktur (ähnlich der objektorientierten Programmierung) und
- Lokalität der Ereignisse in Bezug auf das Gesamtmodell.

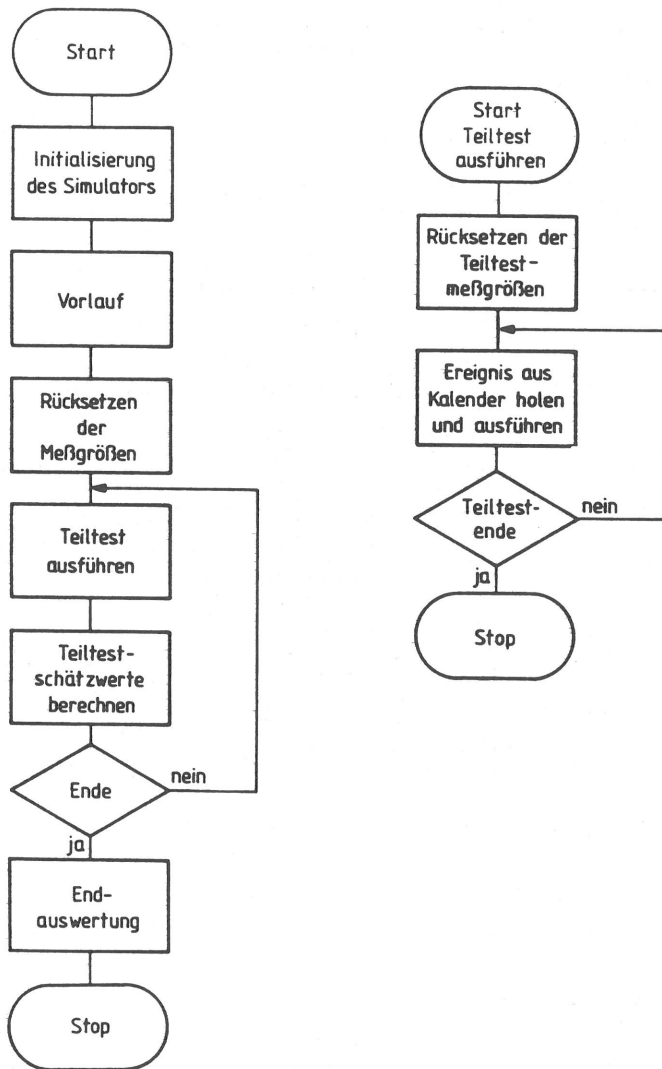


Bild 4.7: Ablaufdiagramm einer Simulation

Es wird davon ausgegangen, daß die Aufträge, bedingt durch ihre Komplexität, explizit durch das Netzwerk wandern. Für die Datenstruktur ergeben sich damit folgende 4 Bereiche.

Modell  
Station, Bus  
Generator  
Warteschlange

In dem ersten Teil wird die Struktur des Modells beschrieben. Das Modell enthält Verweise ('Pointer') auf die Teilmodelle, bis elementare Elemente wie Warteschlange oder Bedieneinheiten, auftreten. In den Elementen sind die entsprechenden Variablen für den Zustand des Prozesses enthalten. Die Datenstruktur als solche sollte nur zur Initialisierung und zur Ausgabe der Ergebnisse verwendet werden.

Ereignis-  
bearbeitung

Für die Ereignisbearbeitung sind in jedem Modellelement Verweise eingetragen, so daß ein Ereignis, das dieses Element betrifft, mit der Kenntnis dieses Elementes bearbeitet werden kann, ohne Kenntnis des Gesamtmodelles. (Lokalität der Ereignisse.)

Routing  
der Aufträge

Für den Weg der Aufträge durch das Modell sind in den betreffenden Elementen Verweise eingetragen, so daß auch der Weg der Aufträge durch das Modell ohne Kenntnis des Gesamtmodelles gefunden wird.

Statistik

Jedes Modellelement enthält die jeweils notwendigen statistischen Daten für die Meßgrößen. In der derzeitigen Version sind alle gleichartigen Modellelemente zu einer linearen Liste verknüpft, um auf die Statistikdaten die jeweiligen Prozeduren anwenden zu können, die unabhängig von dem Platz der Meßgrößen im Modell sind.

Die Modul-Bibliothek enthält Module, die vollständig unabhängig von dem simulierten Modell sind, Module, die bei Bedarf (z.B. neue Modellelemente) anzupassen sind und Module, die spezifisch für das Modell sind (z.B. Initialisierung, Ein-/Ausgabe). In Tabelle 4.3 sind die Module nach diesen Charakteristiken zusammengestellt.

Modul	Für die Simulation		
	unabhängig	anzupassen	spezifisch
Zufallszahlenerzeugung	x		
Statistik	x	(x)	
Datenstruktur		x	
Kalenderverwaltung	x		
Ereignisbearbeitung	(x)	x	(x)
Initialisierung, Ein-/Ausgabe		(x)	x
Hauptprogramm	(x)	x	

Tabelle 4.3: Module der Modul-Bibliothek

#### 4.2.1.7 Softwarestruktur der Modul-Bibliothek

Für die Erstellung der Modul-Bibliothek sind allgemeine Richtlinien für komplexe Software wie :

- Modularisierung mit
- klar definierten Schnittstellen und
- gute Dokumentation mit selbsterklärenden Namen, die für
- leichte Lesbarkeit sorgen,

beachtet worden. Für die Implementierung wurde die Programmiersprache Pascal verwendet und das Konzept der Trennung von Definition und Implementierung aus Modula II durch einen erweiterten Sprachumfang emuliert. Die Implementierung erfolgte auf Datenverarbeitungsanlagen des Typs VAX unter dem Betriebssystem VMS, ist im Kern aber im wesentlichen davon unabhängig. Die Module sind derart gebildet, daß sie hierarchisch aufeinander aufbauen. Die Module der untersten Stufe sind vom Modell vollständig unabhängig, während Module der höheren Stufen anzupassen oder spezifisch für das Modell sind (Bild 4.8).



Bild 4.8: Hierarchie der Simulationsmodule

### 4.3 Messung

Verwendet man für die Leistungsuntersuchung eines Systems die Messung, so kann unterschieden werden zwischen

- passiver Messung oder aktiver Messung und
- Messung am konzentrierten System oder Messung am verteilten System.

Eine Messung setzt immer ein real vorhandenes System voraus, das auch zugänglich ist. Der Aufwand für die Messung ist sowohl vom Aufwand an Material, wie auch Zeit, groß. In den nächsten Abschnitten werden die einzelnen Verfahren eingehender diskutiert.

#### 4.3.1 Passive Messung

Bei der passiven Messung wird das zu beobachtende System durch die Messung nicht oder nur in sehr geringem Maße beeinflusst (Bild 4.9). Das System wird in seiner normalen Umgebung für längere Zeiträume oder dauernd beobachtet. Die Messungen geben sowohl Aufschluß über die Leistungsfähigkeit des Systems, als auch das Verhalten der Umwelt

des Systems. Als Beispiel kann die Messung der Belastung einer Vermittlungsstelle im Telefonverkehr oder der Verkehr auf einem lokalen Netz, bei der Kommunikation zwischen Datenverarbeitungsanlagen, dienen.

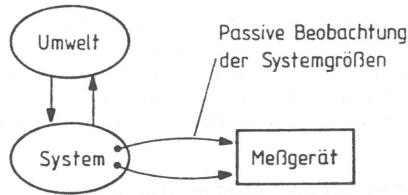


Bild 4.9: Passive Messung

Die passive Messung kann jedoch lediglich über die Leistungsfähigkeit des Systems während der Messung etwas aussagen. Oft können interessierende Bereiche, vor allem die kritischen Bereiche der Überlastung, nicht gezielt beobachtet werden.

#### 4.3.2 Aktive Messung

Bei der aktiven Messung wird die reale Umwelt des Systems ersetzt durch einen Lastgenerator oder einen Umweltsimulator, falls die Rückwirkungen des Systems mit berücksichtigt werden sollen. Diese Messung kann im Labor durchgeführt werden, was die praktische Durchführung außerordentlich erleichtert. Der Aufwand für den Lastgenerator oder den Umweltsimulator ist jedoch erheblich (Bild 4.10).

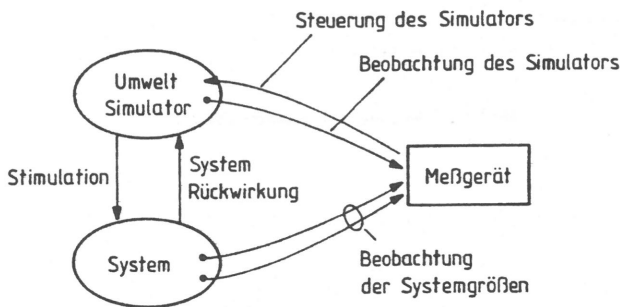


Bild 4.10: Aktive Messung

#### 4.3.3 Messung am konzentrierten System

Die Messung am konzentrierten System ist durch eine örtliche Vereinigung von System, Umwelt oder Umweltsimulator und Meßgerät gekennzeichnet. Das Meßgerät erfaßt und

wertet die Meßwerte zentral aus. Die zeitliche Zuordnung der Ereignisse und Meßwerte zueinander ist jederzeit gegeben.

#### 4.3.4 Messung am verteilten System

Die Messung am verteilten System erfordert eine gleichzeitige Beobachtung von Ereignissen und Meßwerten an verschiedenen Orten und deren zeitlich richtige Zuordnung zueinander. Die beobachteten Ereignisse und Meßwerte werden beim Erfassen mit einem Zeitstempel versehen und einer zentralen Meßwertauswertung übermittelt. Damit ergeben sich zwei weitere Gesichtspunkte für die Messungen an verteilten Systemen (Bild 4.11).

1. Synchronisation der Uhren bei der verteilten Meßwernerfassung.
2. Kommunikationsnetzwerk zur Übermittlung der verteilt erfaßten Meßwerte zu einer zentralen Auswertung.

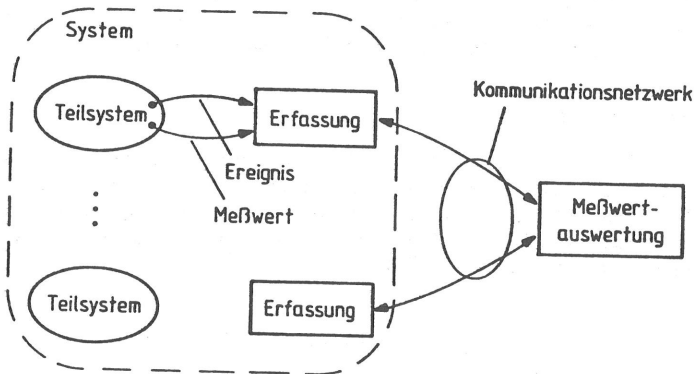


Bild 4.11: Messung am verteilten System



## Kapitel 5

# Analytische Leistungsuntersuchung von Flußregelungen

Nachdem im Kapitel 4 die Grundlagen der Leistungsuntersuchung dargestellt wurden, sollen in diesem Kapitel die analytischen Methoden auf den Protokollmechanismus Flußregelung angewendet werden. Es werden die Modelle aus Kapitel 3 zugrunde gelegt. Im ersten Abschnitt wird die prinzipielle Vorgehensweise anhand des 'Handshake'-Mechanismus dargestellt, bevor im zweiten Abschnitt der Fenstermechanismus analysiert wird. Der dritte Abschnitt befaßt sich mit der Ratenregelung bei verbindungsloser Kommunikation. Im vierten Abschnitt wird eine Optimierung der Fenstergröße, abhängig von der Zielfunktion, durchgeführt, und im fünften Abschnitt wird die Fehlerbehandlung bei Übertragungsfehlern in die Analyse mit einbezogen.

In Bild 5.1 sind die wesentlichsten Leistungsgrößen bei der Analyse der Flußregelung anhand des Grundmodells für den gleitenden Fenstermechanismus (Bild 3.7) dargestellt.

Anschaulich gesehen werden die Wartezeiten durch die Flußregulierung aus dem Innern des Netzwerkes in die Zugangswarteschlange verlagert. Außerdem ergibt sich, erwünscht oder nicht, ein erniedrigter maximaler Durchsatz, da das Netzwerk nicht voll ausgelastet werden kann.

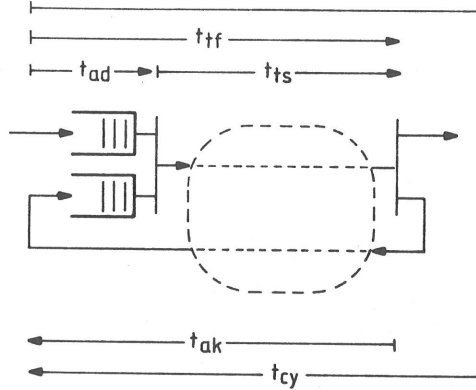
### 5.1 'Handshake'-Mechanismus

Anhand des 'Handshake'-Mechanismus bei einer Verbindung werden in diesem Abschnitt die grundsätzlichen Effekte der Flußregulierung dargestellt. Geht man von dem Grundmodell in Bild 3.5 aus, so kann das vereinfachte Modell nach Bild 3.6 gebildet werden, das im folgenden analysiert werden soll. Dazu werden alle Bedienphasen, die ein Auftrag erfährt, zusammengefaßt und als eine G-Bedienphase aufgefaßt, die einen Anteil für den Vorwärtszweig (Daten) und einen Anteil für den Rückwärtszweig (Quittierungen) enthält. Die Zugangsverzögerung kann somit exakt als Lösung des  $G/G/1-\infty$  Modells berechnet werden.

Für das Beispielnetzwerk in Bild 5.2 können die Leistungsgrößen angegeben werden, wenn der Ankunftsprozeß und die einzelnen Bedienzeitverteilungen festgelegt werden. In Bild 5.3 sind für die Festlegung

$$\begin{aligned} A : & \quad A(t) = 1 - e^{-\lambda t} \\ B_1 \dots B_3 : & \quad B(t) = 1 - e^{-\mu t} \quad ; \mu = 1/s \end{aligned}$$

dieser Verteilungen die Ergebnisse dargestellt.



- $T_{ad}$  - Zugangsverzögerung  
 $T_{ts}$  - Transit-Zeit  
 $T_{tf} = T_{ad} + T_{ts}$  - Transferzeit  
 $T_{ak}$  - Quittierungs-Transit-Zeit  
 $T_{cy} = T_{tf} + T_{ak}$  - Zykluszeit  
 $t_{ss}$  - Mittelwert von  $T_{ss}$

Bild 5.1: Leistungsgrößen an dem Grundmodell der flußregulierten Verbindung

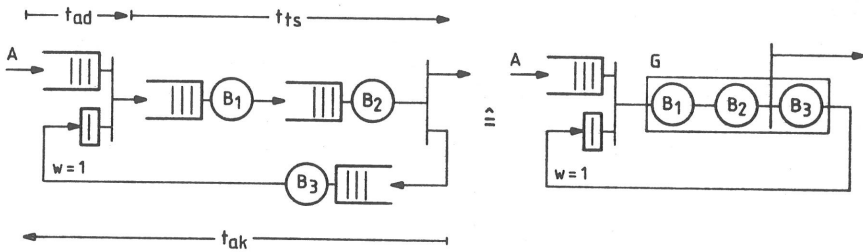


Bild 5.2: Beispielnetzwerk mit 'Handshake'-Protokoll

Solange der Durchsatz des Netzwerkes eine linear steigende Funktion des Angebotes ist, stellt eine Flußregelung eine Verlängerung der Transferzeit dar. Die Verlängerung der Transferzeit wird um so größer, je mehr Verarbeitungsinstanzen im Netzwerk seriell durchlaufen werden müssen. Der maximale Durchsatz ist begrenzt und garantiert eine nicht vollständige Ausnutzung der Netzwerkressourcen.

Die Vorteile der Flußregelung, bei einer Verbindung, können nur nachgewiesen werden, wenn der Durchsatz ab einer gewissen Auslastung des Netzwerkes fällt. Dies kann bei realen Systemen durch Ressourcen-Engpässe bei starker Auslastung auftreten, was Verluste von Dateneinheiten oder Ablauf von Zeitüberwachungen und anschließende Wiederholungen zur Folge hat. Werden mehrere Verbindungen betrachtet, so kommt der Gesichtspunkt der Fairneß der Verbindungen gegeneinander hinzu.

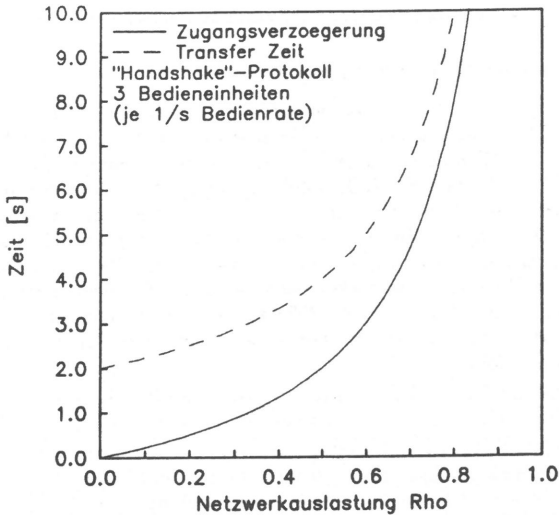


Bild 5.3: Leistungsgrößen für das 'Handshake'-Protokoll

## 5.2 Fenstermechanismus

Flußregelungen mit dem Fenstermechanismus sind sehr schwierig zu analysieren, wenn das Netzwerk nicht eine triviale Form hat. Die einzige exakte Methode, die bekannt ist, verwendet den globalen Zustandsraum des Markov-Prozesses. Angewendet werden kann diese Methode allerdings auch lediglich bei kleinen Netzwerken, die oft nicht realistisch sind, da die Größe des Zustandsraumes nicht mehr beherrscht werden kann.

Man ist daher auf Approximationen, die eine Einschränkung des Zustandsraumes erlauben, angewiesen. Die Wahl der geeigneten Approximation ist somit ein Hauptgegenstand bei der erfolgreichen Analyse von Flußregelungen. Dabei soll die Approximation möglichst allgemein anwendbar sein.

Aufbauend auf den Modellen aus Kapitel 3 werden in den folgenden Abschnitten die Analyseverfahren für eine und mehrere Verbindungen dargestellt und miteinander verglichen. Zur Einleitung wird jeweils eine Übersicht über bekannte Literatur dargestellt.

### 5.2.1 Verfahren für eine Verbindung

#### 5.2.1.1 Literaturübersicht

Es gibt eine Fülle von Aufsätzen über Flußregelungen, die jedoch jeweils spezielle Einschränkungen für das Modell und die Analyse aufweisen. Die folgenden Einschränkungen sind am gravierendsten:

- Die Lösung ist auf ein spezielles Problem zugeschnitten und läßt sich nicht verallgemeinern.

- Das äußere Gesamtsystem wird nicht als Warte- sondern als Verlustsystem modelliert und kann somit auf ein äquivalentes geschlossenes Netzwerk abgebildet werden, das exakt lösbar ist.
- Die Übertragung der Quittierungen wird vernachlässigt.

In der Diplomarbeit von Clemens Hauber [40] findet man eine detaillierte Darstellung der einzelnen Aufsätze über Flußregelung. Weitere Übersichten über den Stand der Literatur findet man bei Charles H. Sauer [98], Serge Fdida, Andrzej Milk, Harry Perros [27] und Edmundo de Souza e Silva, Richard R. Muntz [103]. Die Methoden, die bei der Analyse flußregulierter Verbindungen angewendet werden, wurden parallel ebenfalls zu der Leistungsbewertung von Rechensystemen mit 'Multiprogramming' entwickelt und eingesetzt. Die Anzahl der ablauffähigen Programme im Speicher einer Datenverarbeitungsanlage kann als Populationsbeschränkung aufgefaßt und dementsprechend auch modelliert werden. Kritische Sektionen in der Software sind ein weiterer möglicher Grund für Populationsbeschränkungen. Die wichtigsten Aufsätze über das Gebiet der Flußregelung sind im folgenden angeführt.

S.S. Lam [80] hat die Klasse der Produktform-Netzwerk erweitert auf zustandsabhängige Ankunftsdaten. Dadurch kann vor allem ein Verlustsystem exakt berechnet werden.

J.W. Wong und M.S. Unsoy [113] untersuchen an einem offenen Warteschlangenmodell die Einflüsse eines zweistufigen Flußregelmechanismus mit Ende-zu-Ende (Fenstermechanismus) und globaler ('Isarithmic Scheme') Flußregelung. Die Analyse baut auf der Produktlösungsform auf und verallgemeinert den Verlust von Aufträgen.

Martin Reiser [96] und Charles H. Sauer [98] beschreiben dasselbe Verfahren, das auf einer Aggregation des Netzwerkes in einen flußäquivalenten Ersatzknoten beruht. Während Martin Reiser das Verfahren auf Verbindungen anwendet, wendet Charles H. Sauer es auf Datenverarbeitungsanlagen an. Diese beiden Aufsätze können als klassisch angesehen werden, da die Methode der Aggregation von anderen Autoren häufig angewendet wird.

Nicolas D. Georganas [33] gibt ein iteratives Näherungsverfahren an, mit dem die Flußregelung auf den drei Stufen *Lokal*, *Global* und *Ende-zu-Ende* untersucht werden kann. Lokal auftretende Blockierungen, aufgrund überfüllter Pufferspeicher, werden mit berücksichtigt.

S.S. Lam und J.W. Wong [81] geben in ihrem Übersichtsaufsatz die gängigen Analyseverfahren für Kommunikationsnetze mit Flußregelung über den Fenstermechanismus wieder und beschäftigen sich darüber hinaus mit der Pufferspeicherzuweisung bei lokaler Flußregelung.

Eduard Arthurs, Gregory L. Chesson und Barton W. Stuck [3] befassen sich an einem speziellen Modell mit der Flußregelung zur Anpassung der Geschwindigkeit des Empfängers an die Geschwindigkeit des Senders bei einer virtuellen Verbindung. Auf die Dimensionierung der Fenstergröße wird verstärkt eingegangen.

Ottmar Gühr und Paul J. Kühn [36] beschreiben eine Flußregelung in einer mehrschichtigen Protokollarchitektur. Die Analyse erfolgt letztendlich durch das Verfahren von Charles H. Sauer [98] und Martin Reiser [96], nachdem die Prozessoren der LLC-Schicht dekomponiert wurden. Die Quittierungen werden explizit in die Analyse mit einbezogen.

Serge Fdida, Andrzej Milk und Harry Perros [27] verallgemeinert das Verfahren von Charles H. Sauer [98] und Martin Reiser [96] auf mehrere Hierarchiestufen durch wiederholten Einsatz von flußäquivalenten Ersatzknoten. Die Quittierungen werden explizit in die Analyse mit einbezogen.

Für die Analyse von Datenverarbeitungsanlagen mit Populationsbeschränkungen sind folgende Aufsätze anzuführen:

Y. Bard [4] beschreibt ein iteratives Näherungsverfahren auf der Basis der Dekomposition von Teilsystemen an dem Modell für die Datenverarbeitungsanlage VM/370 von IBM.

Charles H. Sauer [98] wurde bereits in Zusammenhang mit Martin Reiser [96] erwähnt.

G. Varghese, W. Chou und A. A. Nilson [112] wandeln das Verfahren von Charles H. Sauer [98] und Martin Reiser [96] leicht ab, indem sie für die zustandsabhängigen Bedienraten des Ersatzknotens für eine Population  $n$ , die größer als die Fenstergröße  $w$  ist ( $n > w$ ), eine separate Berechnung über die Zustände des Netzwerkes durchführen. Die Autoren betrachten die Zwischenankunftszeit bei der Zugangskontrolle in dem kurzgeschlossenen Netzwerk. Verläßt ein Auftrag den Bedienknoten  $N$ , der der letzte vor der kurzgeschlossenen Zugangskontrolle ist, so betrachten die Autoren die Wahrscheinlichkeiten, daß sich zwischen dem Knoten  $N$  und dem Knoten  $i$  keine Aufträge befinden. Der Auftrag im Knoten  $i$  sieht somit eine gesamte Bedienzeit, die der Summe der Bedienzeiten in allen Bedienknoten zwischen  $i$  und  $N$ , jeweils einschließlich, entspricht. Aus den Wahrscheinlichkeiten und den Bedienraten wird eine Cox-Repräsentation [23] gebildet, die nach der Methode von Raymond A. Marie [87] zu den gewünschten Bedienraten umgerechnet wird. Die Autoren validieren ihre Ergebnisse mit Simulationen und erhalten für ihre Modelle bessere Ergebnisse wie nach der Methode von Charles H. Sauer [98] und Martin Reiser [96].

Patricia A. Jacobson und Edward D. Lazowska [57, 58] befassen sich mit einem iterativen Ansatz zur Untersuchung von primären und sekundären Ressourcen, wobei die primäre Ressource ebenfalls eine Bedienung durchführen kann. Es erfolgt eine Iteration über zwei Teilmodelle, wobei jeweils eine Ressource durch einen 'Infinite Server'-Knoten ersetzt wird. Die Methode arbeitet daher mit Ersatzverzögerungszeiten ('Surrogate Delays').

Patricia A. Jacobson und Edward D. Lazowska [59] verwenden wiederum die Iteration zwischen zwei Teilmodellen, wobei die primäre Ressource durch ein  $M/M/w-\infty$  Warteschlangenmodell, bei der Fenstergröße  $w$ , nachgebildet wird. Bei dem zweiten Teilmodell wird die primäre Ressource durch einen Kurzschluß ersetzt und das entstehende geschlossene Netzwerk für eine mittlere Population, die der Auslastung des  $M/M/w-\infty$  Systems entspricht, berechnet. Der Durchsatz liefert wiederum die Bedienrate des  $M/M/w-\infty$  Systems.

Subhash C. Agrawal und Jeffery P. Buzen [1] führen ein iteratives Verfahren zur Serialisierung von Aufträgen bei dem Zugriff auf kritische Sektionen der Software ein. Die Verweildauer in der kritischen Sektion wird durch einen zusätzlichen Server ('Aggregate Server') nachgebildet.

Darrel J. Freund und James N. Bexfield [29] schildern ein iteratives Approximationsverfahren, das auf Aggregation und Dekomposition aufbaut. Simultanbelegungen werden durch eine neu definierte 'Multi-Entrance-Queue' nachgebildet, die der Produktform genügt.

A. Krzesinski und P. Teunissen [77] verfeinern die Methode von Patricia A. Jacobson und Edward D. Lazowska [59], so daß auch mehrere Token pro Auftrag verbraucht werden können.

Edmundo de Souza e Silva und Richard R. Muntz [103] erweitern diese Methode auf beliebige Anordnungen von geschachtelten Populationsbeschränkungen. Dazu wird

hauptsächlich eine neue Approximation von  $M/M/w-\infty$  Knoten, für den Einsatz in Produktformnetzen, eingeführt.

### 5.2.1.2 Flußäquivalentes Ersatznetzwerk

Dieses Verfahren geht auf die Autoren Charles H. Sauer [98] und Martin Reiser [96] zurück. Für das Netzwerk wird vorausgesetzt, daß es vom Produktformtyp ist. Durch die Dekomposition der Flußregelung und des Netzwerkes und der anschließenden Aggregation des Netzwerkes zu einem flußäquivalenten Ersatzknoten erfolgt die Analyse über den Zustandsraum des Ersatzknotens. Es wird davon ausgegangen, daß die Quittierungen die Aufträge nicht behindern (keine gemeinsamen Ressourcen). Der Algorithmus kann dann in drei Schritten dargestellt werden (siehe auch Bild 5.4).

1. Aggregation des Netzwerkes zu einem flußäquivalenten Ersatzknoten.
  - (a) Flußregelung entfernen und kurzschließen.
  - (b) Durchsatz  $\Phi(n)$  für alle möglichen Populationen  $n$  des Netzwerkes berechnen.
  - (c) Konstruktion des Ersatzknotens mit der Bedienratenfunktion  $\mu^*(n) = \Phi(n)$ .
2. Ersetzung des Netzwerkes durch den Ersatzknoten.
  - (a) Das Netzwerk wird entfernt und der flußäquivalente Ersatzknoten aus Schritt 1c) wird dafür eingesetzt.
  - (b) Elimination der Flußregelung durch eine Äquivalenzumformung. Die Bedienratenfunktion wird dabei angepaßt.

$$\mu(n) = \begin{cases} \mu^*(n) & ; n \leq w \\ \mu^*(w) & ; n > w \end{cases}$$

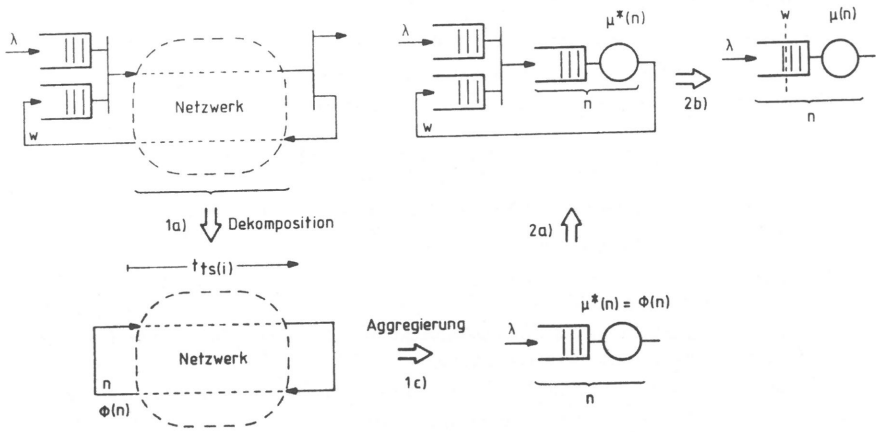
3. Berechnung der Leistungsgrößen.
  - (a) Berechnung der Zustandswahrscheinlichkeiten  $p$  des Knotens aus 2b).
  - (b) Berechnung der mittleren Zugangsverzögerungszeit  $t_{ad}$  bei geteilter Warteschlange aus den Zustandswahrscheinlichkeiten.
  - (c) Berechnung der mittleren Transitzeit  $t_{ts}$  von Aufträgen über die, mit den Zustandswahrscheinlichkeiten gewichtete, Summe der bedingten Transitzeiten  $t_{ts}(n)$ , die in Schritt 1b) bereits erfaßt werden können.

Die restlichen interessierenden Größen können daraus abgeleitet oder entsprechend bestimmt werden.

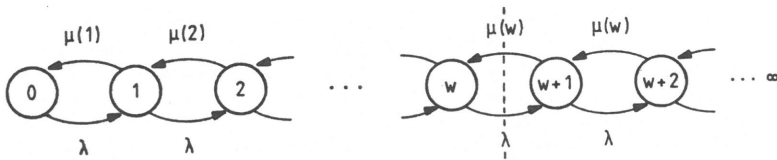
Es bleibt anzumerken, daß die einzige Näherung bei diesem Verfahren durch die Dekomposition gemacht wird, die nicht exakt ist, da das Gesamtmodell nicht der Produktform genügt, auch wenn das Netzwerk vom Produktformtyp ist.

Behindern sich Quittierungen und Aufträge im Netzwerk gegenseitig, so muß das Netzwerk durch eine 'Composite Queue' mit 2 Auftragsarten und einer Bedienratenfunktion, die von dem Vektor  $\underline{n} = (n_1, n_2)$  der beiden Populationen Aufträge und Quittierungen abhängig ist, nachgebildet werden (siehe auch Bild 5.5).

1. Aggregation des Ersatzknotens zu einem flußäquivalenten Ersatzknoten.
  - (a) Flußregelung entfernen und die Kette für Aufträge und die Kette für Quittierungen kurzschließen.



**Zustandsraum des Knotens aus 2b)**



$$p_n = p_0 \sum_{i=1}^n \prod_{i=1}^n \frac{\lambda}{\mu(i)} = p_0 \begin{cases} \prod_{i=1}^n \frac{\lambda}{\mu(i)} & ; n \leq w \\ \left(\frac{\lambda}{\mu(w)}\right)^{n-w} \prod_{i=1}^w \frac{\lambda}{\mu(i)} & ; n > w \end{cases}$$

$$p_0^{-1} = 1 + \sum_{n=1}^{\infty} \prod_{i=1}^n \frac{\lambda}{\mu(i)} = 1 + \sum_{n=1}^w \prod_{i=1}^n \frac{\lambda}{\mu(i)} + \frac{\lambda/\mu(w)}{1 - \lambda/\mu(w)} \prod_{i=1}^w \frac{\lambda}{\mu(i)}$$

$$t_{ad} = \sum_{i=w+1}^{\infty} (i-w) p_i / \lambda$$

$$t_{ts} = \sum_{i=1}^w t_{ts}(i) p_i + t_{ts}(w) \sum_{i=w+1}^{\infty} p_i = \sum_{i=1}^w t_{ts}(i) p_i + t_{ts}(w) \left(1 - \sum_{i=0}^w p_i\right)$$

Bild 5.4: Analyseverlauf beim fluquivalenten Ersatzknoten ohne Beeinflussung zwischen Auftrgen und Quittierungen

- (b) Durchsatzvektor  $\underline{\Phi}(\underline{n})$  für alle Populationen  $|\underline{n}| \leq w$  berechnen.  
 (c) Konstruktion des Ersatzknotens mit der Bedienratenfunktion  $\underline{\mu}^*(\underline{n}) = \underline{\Phi}(\underline{n})$ .
2. Ersetzung des Netzwerkes durch den Ersatzknoten.
- (a) Netzwerk entfernen und dafür Ersatzknoten einsetzen.  
 (b) Elimination der Flußregelung durch die Äquivalenzumformung

$$\underline{\mu}(\underline{n}) = \begin{cases} \underline{\mu}^*(n_1, n_2) & ; |\underline{n}| \leq w \\ \underline{\mu}^*(w - n_2, n_2) & ; |\underline{n}| > w \end{cases}$$

Innerhalb des Netzwerkes können sich höchstens  $w$  Aufträge befinden.

3. Berechnung der Leistungsgrößen
- (a) Berechnung der Zustandswahrscheinlichkeiten  $p(n_1, n_2, n_3)$  über die globalen Zustandsgleichungen (Bild 5.5).  
 (b) Berechnung der mittleren Zugangsverzögerung  $t_{ad}$  aus den Zustandswahrscheinlichkeiten bei geteilter Warteschlange.  
 (c) Berechnung der mittleren Transitzeit  $t_{ts}$  der Aufträge über die Zustandswahrscheinlichkeiten bei geteilter Warteschlange.

Sowohl Charles H. Sauer [98], wie auch Martin Reiser [96] haben das grundsätzliche Verfahren durch Simulationen validiert. Zdzislaw Papir [93] hat für ein Tandem-Netzwerk aus zwei Knoten eine exakte Analyse entwickelt und darüber das Verfahren ebenfalls validiert. Die Aussagen sind eindeutig und geben eine ausgezeichnete Genauigkeit für geringen Durchsatz.

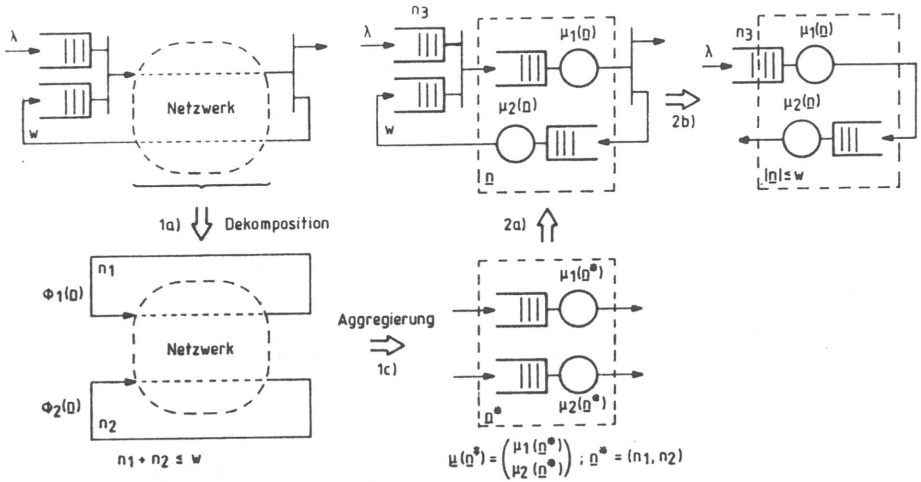
Steigt der Durchsatz bis in den Bereich des maximalen Durchsatzes, so ergeben sich gewisse Fehler. Die Transferzeit wird i.a. sehr gut abgeschätzt, während die Zugangsverzögerungszeit schneller ungenau wird. Zdzislaw Papir [93] beobachtete einen Fehler von unter 1% für die Transferzeit und 20% für die Zugangsverzögerungszeit. Bilden die Ketten, durch Verzweigungen nach Wahrscheinlichkeiten, Schleifen, so wirkt sich dies ebenfalls negativ auf die Genauigkeit aus. Für eine ausführlichere Validierung sei auf Clemens Hauber [40] verwiesen.

### 5.2.1.3 Lastäquivalentes Ersatznetzwerk

Bei diesem Verfahren, das von Patricia A. Jacobson und Edward D. Lazowska [59] eingeführt wurde, wird eine Iteration zwischen zwei Teilmodellen durchgeführt. Die gegebene mittlere Belastung des Netzwerkes wird dabei erhalten. Die Iteration erfolgt zwischen einem  $M/M/w-\infty$  Knoten, mit konstanter zustandsunabhängiger Bedienrate, und dem kurzgeschlossenen Netzwerk mit einer (nicht ganzzahligen) Population  $n$ . Bei dem  $M/M/w-\infty$  Knoten wird die mittlere Anzahl  $n$  von Aufträgen in den Bedieneinheiten bei gegebener Ankunftsrate und Bedienrate bestimmt. Diese mittlere Anzahl von Aufträgen in den Bedieneinheiten wird dann als mittlere Population im kurzgeschlossenen Netzwerk verwendet und der Durchsatz  $\Phi(n)$  bestimmt. Dieser Durchsatz dient wiederum als Bedienrate im  $M/M/w-\infty$  Knoten (Bild 5.6). Die Iteration ist in Bild 5.7 dargestellt und wird bis zur Konvergenz durchgeführt. Das Netzwerk wird als Produktform-Netzwerk vorausgesetzt.

Die eindeutige Konvergenz der Iteration ist gesichert, solange  $\Phi(n)$  eine monoton steigende Funktion darstellt.  $\Phi(n)$  kann als die Linearisierung des Durchsatzes im jeweils betrachteten Belastungsfall angesehen werden.





Zustandsraum des Knotens aus 2b)

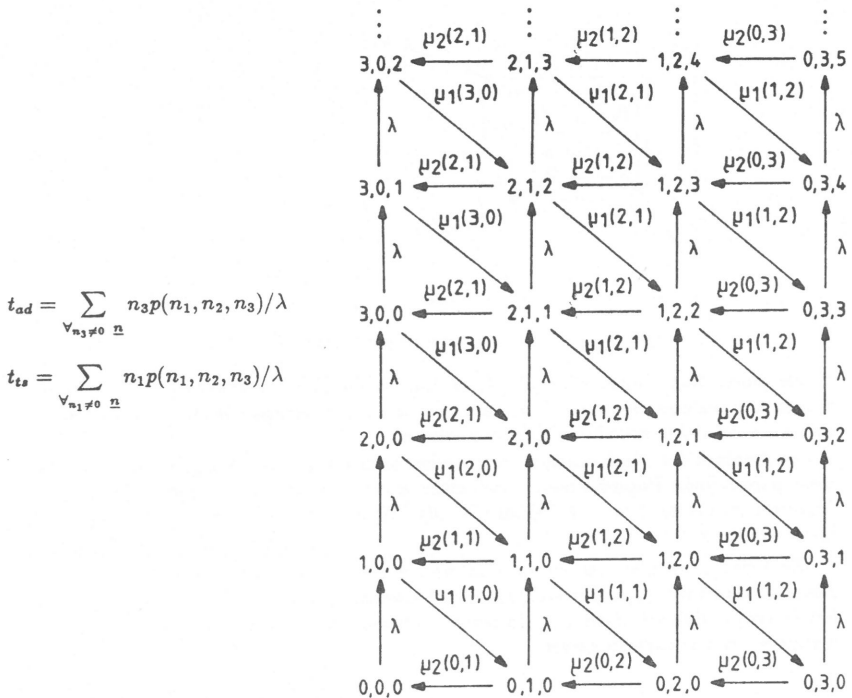


Bild 5.5: Analyseverlauf beim flußäquivalenten Ersatzknoten mit Beeinflussung zwischen Aufträgen und Quittierungen (z.B.  $w = 3$ )

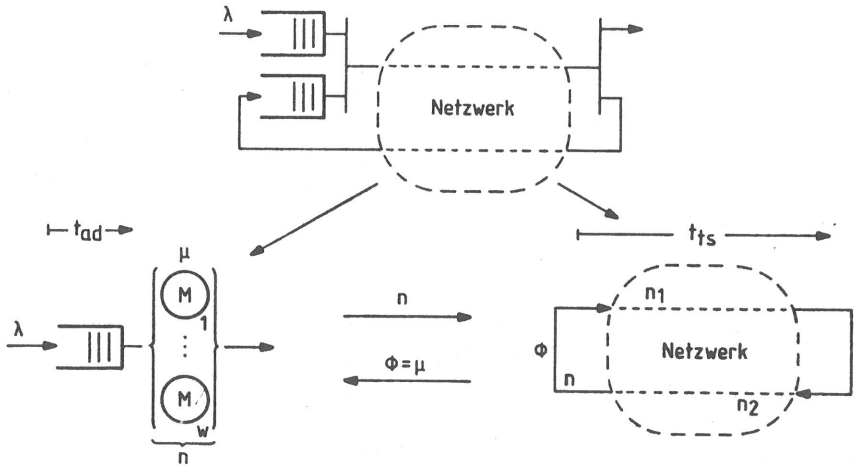


Bild 5.6: Analyseverlauf bei dem lastäquivalenten Ersatznetzwerk und einer flußregulierten Verbindung

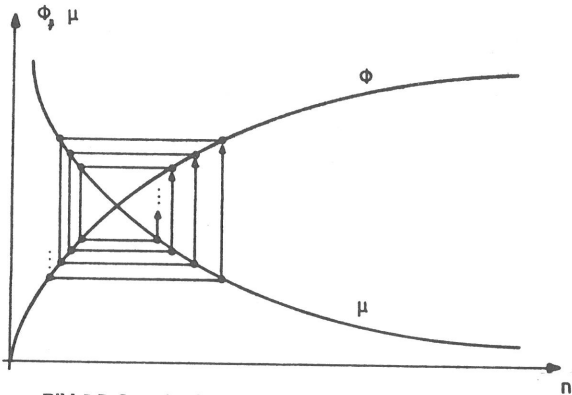


Bild 5.7: Iteration bei dem lastäquivalenten Ersatznetzwerk

Die eindeutige Konvergenz der Iteration ist gesichert, solange  $\Phi(n)$  eine monoton steigende Funktion darstellt.  $\Phi(n)$  kann als die Linearisierung des Durchsatzes im jeweils betrachteten Belastungsfall angesehen werden.

Die Analyse des geschlossenen Netzwerkes setzt einen Lösungsalgorithmus voraus, der nicht ganzzahlige Populationen verarbeiten kann. Es wird dafür meist der Linearizer-Algorithmus (Dong Neuse, K. Mani Chandy [92] und K. Mani Chandy, Dong M. Neuse [17]) verwendet.

Die Leistungsgrößen lassen sich anhand der zwei Teilmodelle bestimmen. Die Zugangsverzögerung  $t_{ad}$  ergibt sich als die Wartezeit im  $M/M/w-\infty$  Knoten. Die Transferzeit ergibt sich aus dem geschlossenen Netzwerk  $t_{ts} = n_1/\Phi$ , wenn  $n_1$  die Anzahl der Aufträge im Vorwärtszweig ist.

exemplarisch dargestellt werden wird.

#### 5.2.1.4 Maximum-Entropie-Verfahren

Demetres D. Kouvatso hat in seinen Arbeiten die Analyse von offenen und geschlossenen allgemeinen Netzwerken mit der Methode der maximalen Entropie dargestellt. Unterliegt ein offenes allgemeines Netzwerk mit einem Poisson-Ankunftsstrom für Aufträge der Flußregelung, so kann der folgende Algorithmus angewendet werden, der Methoden der Produktformnetzwerke und Methoden der maximalen Entropie verwendet. Es wird von dem Grundmodell (Bild 3.5) mit  $G/G/1-\infty$  Knoten ausgegangen. Es wird die einschränkende Annahme gemacht, daß die Aufträge und Quittierungen sich gegenseitig nicht beeinflussen. Das Ziel des Algorithmus ist es, das Netzwerk durch einen Ersatzknoten, entsprechend dem 'Norton-Theorem' bei Produktformnetzwerken, zu ersetzen.

Der Algorithmus läuft wiederum in drei Schritten ab (Bild 5.8).

1. Aggregation des Netzwerkes zu einem Ersatzknoten.
  - (a) Flußregulierung entfernen und kurzschließen.
  - (b) Fluß  $h_N^*(n)$ ,  $c_N^2(n)$  der Aufträge, an der Stelle der Flußregelung im kurzgeschlossenen Netzwerk für alle möglichen Populationen  $n$  bestimmen.
  - (c) Konstruktion des Ersatzknotens mit den zustandsabhängigen Bedienzeitverteilungen  $F_n^*(t)$ , die durch zwei Momente  $h_N^*(n)$  und  $c_N^2(n)$  approximiert werden.
2. Ersetzung des Netzwerkes durch den Ersatzknoten.
  - (a) Das Netzwerk wird entfernt und der Ersatzknoten aus Schritt 1c) eingesetzt.
  - (b) Elimination der Flußregelung durch eine Äquivalenzumformung. Die Bedienzeitverteilungsfunktion wird dabei angepaßt.

$$F_n(t) = \begin{cases} F_n^*(t) & ; n \leq w \\ F_w^*(t) & ; n > w \end{cases}$$

3. Berechnung der Leistungsgrößen
  - (a) Berechnung der Zustandswahrscheinlichkeiten  $p$  des Knotens aus 2b). (Siehe Anhang B.)
  - (b) Berechnung der mittleren Zugangsverzögerungszeit  $t_{ad}$  aus den Zustandswahrscheinlichkeiten.
  - (c) Berechnung der mittleren Transitzeit über die mit den Zustandswahrscheinlichkeiten gewichtete Summe der Aufträge im Vorwärtszweig  $n_1$ .

$$t_{ts} = \left[ \sum_{i=1}^w n_1(i)p_i + n_1(w) \left( 1 - \sum_{i=0}^w p_i \right) \right] / \lambda$$

$n_1$  wird am geschlossenen Netzwerk bei der Population  $i$  bestimmt. Falls der Rückwärtszweig entfällt, kann die Transitzeit direkt am Ersatzknoten bestimmt werden.

Für den Fall von Produktform-Netzwerken degeneriert das Verfahren ebenfalls zu dem Verfahren des flußäquivalenten Ersatzknotens. Die Genauigkeit des Verfahrens wird von zwei Approximationen beeinflusst. Allgemeine Netzwerke lassen sich nicht exakt, sondern lediglich approximativ analysieren. Die Dekomposition der Flußregelung und des Netzwerkes und die Aggregation des Netzwerkes sind nicht exakt.

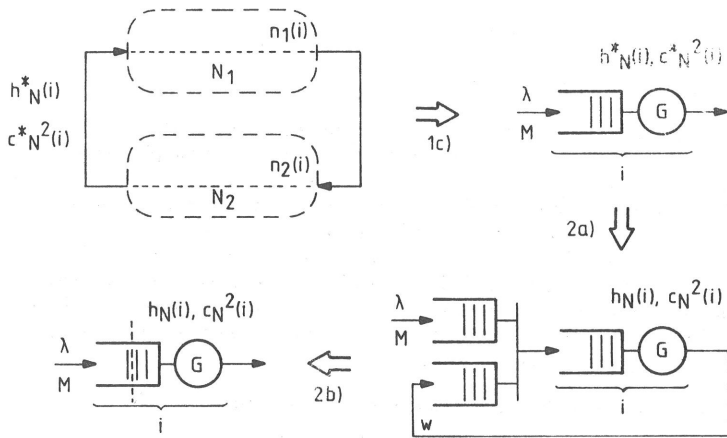


Bild 5.8: Analyseverlauf beim Maximum-Entropie-Verfahren

Im nächsten Abschnitt werden, für die vorgestellten Verfahren zu der Analyse flußregulierter Verbindungen mit dem Fenstermechanismus bei Folgenummern-Steuerung, beispielhafte Ergebnisse für ein flußreguliertes Tandem-Netzwerk dargestellt und jeweils mit Simulationen validiert.

### 5.2.1.5 Bewertung der Verfahren

Für die Bewertung der Analyseverfahren wurde das Tandem-Netzwerk aus Bild 5.9 herangezogen. In einem ersten Schritt werden die Verfahren mit flußäquivalentem und lastäquivalentem Ersatznetzwerk dargestellt und mit Simulationen validiert. Die Parameter des Netzwerkes müssen so gewählt werden, daß das Tandem-Netzwerk der Produktform-Klasse angehört. Das Maximum-Entropie-Verfahren liefert für die Klasse der Produktform-Netzwerke identische Ergebnisse, wie das Verfahren mit flußäquivalentem Ersatznetzwerk. Dieses Verfahren wird im zweiten Schritt separat für ein allgemeines Tandem-Netzwerk dargestellt und mit Simulationen validiert.

Alle Simulationsergebnisse wurden mit einem Vertrauensgrad von 95% ermittelt.

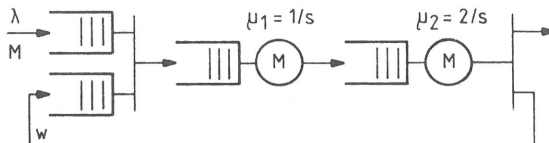


Bild 5.9: Tandem Netzwerk

Für die Fenstergröße  $w = 1$  liefern die Verfahren mit flußäquivalentem und lastäquivalentem Ersatznetzwerk dieselben Ergebnisse. Es sollte jedoch das exakte Verfahren

aus Abschnitt 5.1 verwendet werden. Zum Vergleich ist in Bild 5.10 die Analyse des 'Handshake'-Mechanismus und die Analyse über das Ersatznetzwerk für die Transfer-Zeit und die Zugangsverzögerung für die Parameter des Bildes 5.9 dargestellt.

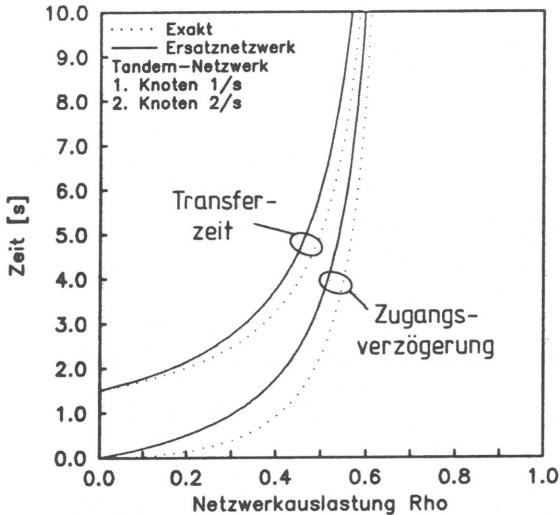


Bild 5.10: Transferzeit und Zugangsverzögerung bei flußreguliertem Tandem-Netzwerk und Fenstergröße  $w = 1$

Die Netzwerkauslastung bezieht sich auf die Auslastung des ersten Knotens  $\rho = \lambda/\mu_1 = \lambda[s]$ .

In Bild 5.11 ist die Transferzeit für die Fenstergröße  $w = 2$  und  $w = 4$  für die Analyse mit flußäquivalentem und lastäquivalentem Ersatznetzwerk dargestellt und mit Simulationen validiert. Die Analyse mit lastäquivalentem Ersatznetzwerk erfaßt zwar die Grenzwerte für  $\rho \rightarrow 0$  und  $\rho \rightarrow$  'Maximale Auslastung' richtig, hat aber im Zwischenbereich erhebliche Abweichungen, die bei größerem Fenster noch ausgeprägter werden. Die Analyse mit dem flußäquivalenten Ersatznetzwerk approximiert ebenfalls den Zwischenbereich genügend genau. Bei steigender Fenstergröße verbessert sich die Approximation immer mehr. Sie liefert für den Grenzfall  $w \rightarrow \infty$  die exakten Ergebnisse.

Die Validierung des Maximum-Entropie-Verfahrens wird in den Bildern 5.12 und 5.13 dargestellt. Es wird wiederum das Tandem-Netzwerk mit zwei Knoten verwendet. Die Parameter der Bedienzeitverteilungen wurden zu

1. Knoten :  $\mu_1 = 1/s$ ,  $c_1^2 = 2$  und
2. Knoten :  $\mu_2 = 2/s$ ,  $c_2^2 = 3$

gewählt. In Bild 5.12 ist ein Markov-Ankunftsprozeß verwendet worden, während in Bild 5.13 ein Ankunftsprozeß mit 'Generalized Exponential'-Verteilungsfunktion ( $c_A^2 = 2$ ) verwendet wurde.

Die Simulationsergebnisse zeigen, daß bei niedriger Netzwerkauslastung die Transferzeit gut abgeschätzt wird. Bei steigender Netzwerkauslastung treten Fehler auf, die auch

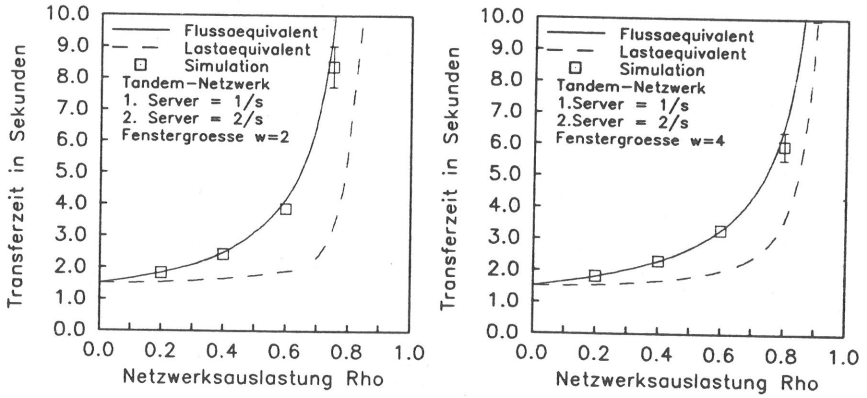


Bild 5.11: Transferzeit bei flureguliertem Tandem-Netzwerk und Fenstergre  $w = 2$  und  $w = 4$

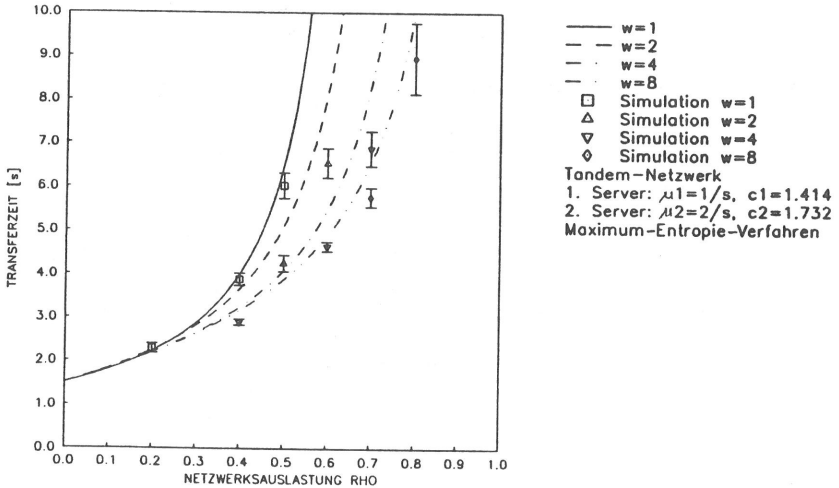


Bild 5.12: Validierung des Maximum-Entropie-Verfahrens bei Markov-Ankunftsproe

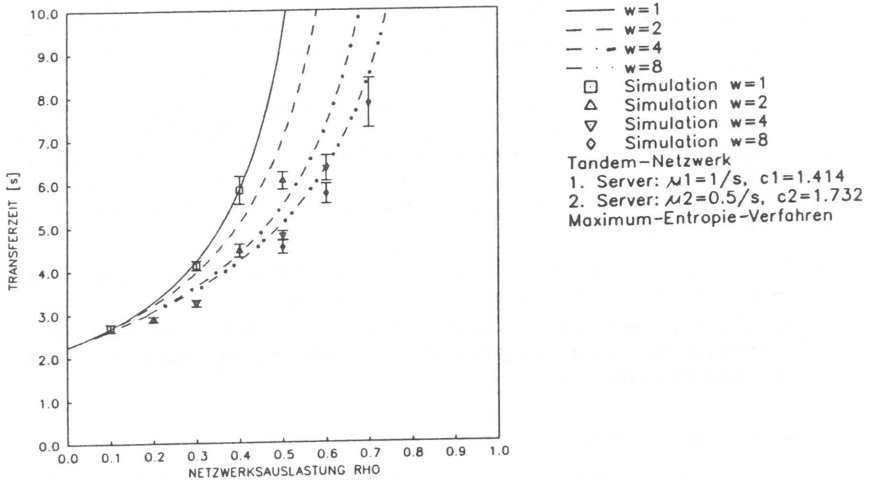


Bild 5.13: Validierung des Maximum-Entropie-Verfahrens bei 'Generalized Exponential'-Ankunftsstandsverteilung ( $c_A^2 = 2$ )

bei großen Fensterwerten nicht verschwinden. Die Ursache liegt in der nicht exakt möglichen Analyse allgemeiner geschlossener Netzwerke. Die Analyse der Flußregelung bringt noch weitere Fehler hinzu.

Zusätzliche Validierungen mit noch höheren Variationskoeffizienten ergaben noch ungünstigere Ergebnisse. Die Ergebnisse werden tendenziell für große Fensterwerte und komplexere Netzwerke (möglichst ohne Schleifen) genauer. Das Verfahren der maximalen Entropie kann demnach in einem begrenzten Bereich eingesetzt werden.

## 5.2.2 Verfahren für mehrere Verbindungen

Die Verfahren für mehrere Verbindungen bauen auf dem Verfahren für eine Verbindung auf. Durch geeignete Erweiterungen, die jeweils im allgemeinen approximativ sind, kann das Gesamtmodell jeweils in Teilprobleme mit einer flußregulierten Verbindung zerlegt werden.

### 5.2.2.1 Literaturübersicht

Bei Charles H. Sauer [98] wurde eine direkte Erweiterung der Methode des flußäquivalenten Ersatznetzwerkes für mehrere Verbindungen untersucht. Das Netzwerk wurde dazu in einen Ersatzknoten aggregiert, dessen Bedienrate eine Funktion des Zustandsvektors des Ersatzknotens ist. Dieser Ersatzknoten muß über den globalen Zustandsraum gelöst werden, da eine Äquivalenz-Umformung, wie im Fall einer flußregulierten Verbindung, die Bedingungen für Produktform-Netzwerke verletzt. Charles H. Sauer hat diese Äquivalenz-Umformung versuchsweise durchgeführt und erhielt ungenügende Ergebnisse.

Patricia A. Jacobson und Edward D. Lazowska [58] haben für die Methode des lastäquivalenten Ersatznetzwerkes ebenfalls eine Erweiterung auf mehrere Verbindungen durchgeführt. Die Erweiterung läßt sich ohne Schwierigkeiten durch einen  $M/M/w-\infty$  Ersatzknoten für jede Verbindung durchführen.

Edward D. Lazowska und John Zahorjan [82], Alexander Thomasian [107], Alexander Thomasian, P. Bay [108] und A. Krzesinski, P. Teunissen [77] haben das Verfahren des flußäquivalenten Ersatznetzwerkes jeweils wieder aufgegriffen. Die Analyse des Gesamtmodells wird durchgeführt, indem jeweils eine Flußregelung approximativ untersucht wird, während alle anderen Flußregelungen nach einem der zwei folgenden Verfahren ersetzt wurden:

1. Die Bedienraten der einzelnen Knoten werden entsprechend der 'Load Concealment'-Methode für die zu untersuchende Kette herabgesetzt.
2. Alle Flußregelungen, außer der zu untersuchenden, werden kurzgeschlossen und mit ihren mittleren Populationen gefüllt, die sich iterativ bei der Analyse der einzelnen Flußregelungen ergeben.

#### 5.2.2.2 Lastäquivalentes Ersatznetzwerk

Das Verfahren geht auf Patricia A. Jacobson und Edward D. Lazowska [58] zurück und ist eine konsequente Erweiterung des Verfahrens für eine flußregulierte Verbindung.

Das zu untersuchende Modell wird in zwei Arten von Modellen zerlegt (siehe Bild 5.14). Jede Flußregelung wird durch einen  $M/M/w-\infty$  Ersatzknoten repräsentiert, während das Netzwerk bei kurzgeschlossenen Flußregelungen und einem mittleren Populationsvektor, der sich aus der Auslastung der  $M/M/w-\infty$  Knoten ergibt, analysiert wird. Die Zykluszeit der Aufträge in den geschlossenen Ketten ergibt wiederum die Bedienzeit der  $M/M/w-\infty$  Knoten. Durch Iteration zwischen diesen Teilmodellen erhält man bei Konvergenz die Lösung. Die eindeutige Konvergenz ist gesichert, wenn das Netzwerk keine Knoten enthält, deren Bedienratenfunktion bei steigender Belastung fällt.

Die Berechnung der Leistungsgrößen erfolgt analog dem Fall einer flußregulierten Verbindung. Die Güte des Verfahrens leidet unter den gleichen Mängeln, wie sie bei einer flußregulierten Verbindung festgestellt wurden. Stark ausgelastete Verbindungen zeigen dabei besseres Verhalten als schwach ausgelastete Verbindungen, die meist über den gesamten Lastbereich relativ große Fehler aufweisen. Die mittlere Population  $\bar{n}_i$  wird am lastäquivalenten Ersatzknoten abgeschätzt durch:

$$\begin{aligned}\bar{n}_i &= \sum_{j=0}^{w_i} j p_i(j) + \sum_{j=w_i+1}^{\infty} w_i p_i(j) \\ &= p_i(0) \sum_{j=1}^{w_i} \frac{1}{(j-1)!} \left(\frac{\lambda_i}{\mu_i}\right)^j + p_i(0) \frac{1}{w_i!} \frac{w_i \lambda_i}{w_i \mu_i - \lambda_i} \left(\frac{\lambda_i}{\mu_i}\right)^{w_i} \\ p_i(0)^{-1} &= 1 + \sum_{n=1}^{w_i} \frac{1}{n!} \left(\frac{\lambda_i}{\mu_i}\right)^n + \frac{1}{w_i!} \left(\frac{\lambda_i}{\mu_i}\right)^{w_i} \frac{\lambda_i}{w_i \mu_i - \lambda_i}\end{aligned}$$

Die Berechnung der Leistungsgrößen erfolgt jeweils individuell für jede flußregulierte Verbindung entsprechend dem Fall einer flußregulierten Verbindung. Eine Validierung der Ergebnisse erfolgt im übernächsten Abschnitt.



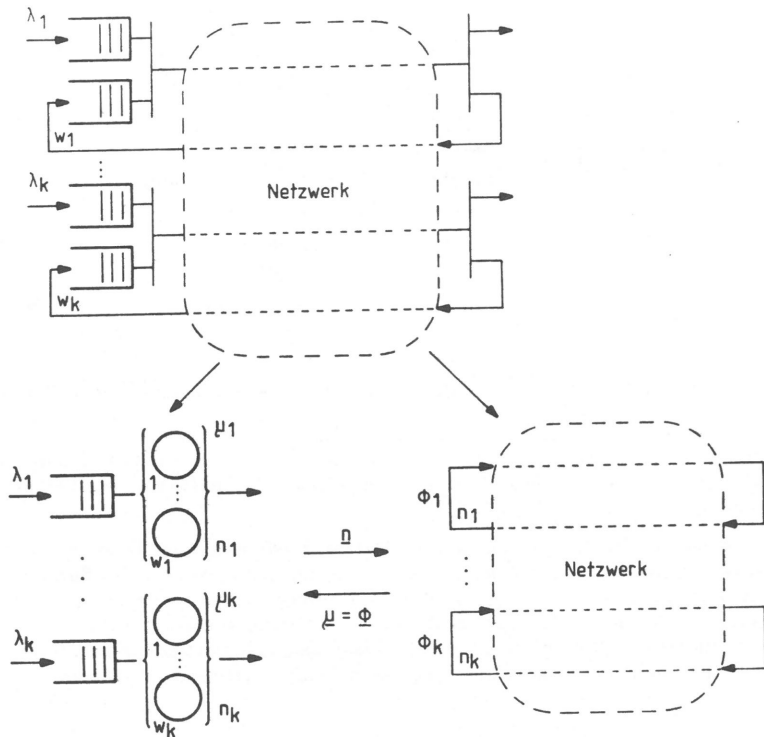


Bild 5.14: Analyseverlauf bei lastäquivalentem Ersatzknoten und mehreren flußregulierten Verbindungen

### 5.2.2.3 Flußäquivalentes Ersatznetzwerk

In diesem Abschnitt soll die Analyse mehrerer flußregulierter Verbindungen nach der Methode des flußäquivalenten Ersatznetzwerkes dargestellt werden. Die Analyse erfolgt iterativ, indem jeweils eine flußregulierte Verbindung analysiert wird. Die anderen Verbindungen werden ersetzt durch geschlossene Ketten mit der mittleren Population der entsprechenden flußregulierten Verbindung. Für jede flußregulierte Verbindung wird jeweils die mittlere Population im Netzwerk berechnet und bei der Analyse der anderen flußregulierten Verbindungen verwendet. Die Iteration wird wiederholt bis Konvergenz eintritt (siehe Bild 5.15).

### 5.2.2.4 Bewertung der Verfahren

Die Validierung der beiden vorgestellten Verfahren für mehrere Verbindungen wird anhand des Tandem-Netzwerkes (Bild 5.9) durchgeführt, das von zwei Ketten durchlaufen wird. Die Ankunftsrate der beiden Ketten wurden identisch gewählt, während die Fenstergrößen unsymmetrisch  $w_2 = 2w_1$  gewählt wurden. Werden die Fenstergrößen symme-

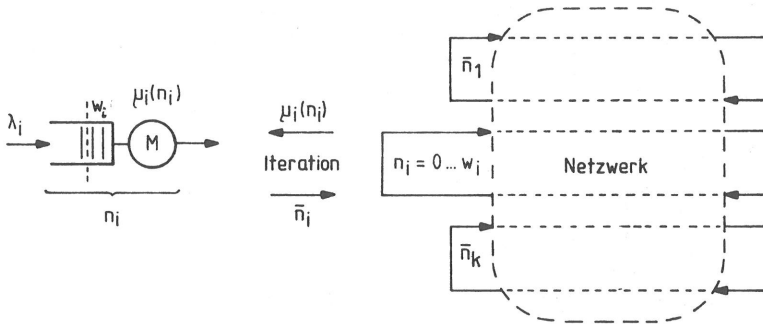


Bild 5.15: Analyseverlauf beim flußäquivalenten Ersatznetzwerk und mehreren Verbindungen

trisch, die Ankunftsrate aber unsymmetrisch gewählt, so ergaben sich ähnliche Ergebnisse. Symmetrische Fenstergrößen und symmetrische Ankunftsrate ergaben genauere Ergebnisse.

Während für das lastäquivalente Ersatznetzwerk mit steigender Fenstergröße die Ergebnisse ungenauer werden (Bild 5.16), werden die Ergebnisse für das flußäquivalente Ersatznetzwerk genauer (Bild 5.17). Für die Fenstergrößen  $w_1 = 1$  und  $w_2 = 2$  sind die Ergebnisse ähnlich genau. Das lastäquivalente Ersatznetzwerk unterschätzt die Transferzeit, außer bei Fenstergröße  $w_1 = 1$ , während das flußäquivalente Ersatznetzwerk die Transferzeit meist überschätzt und für grössere Fensterwerte sehr genau wird.

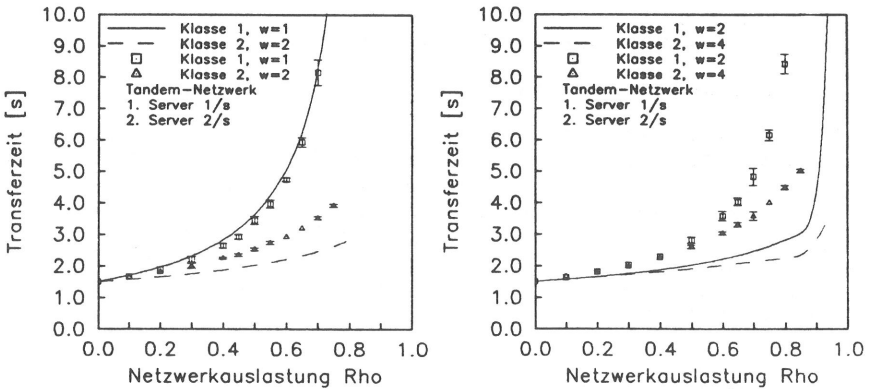


Bild 5.16: Lastäquivalentes Ersatznetzwerk

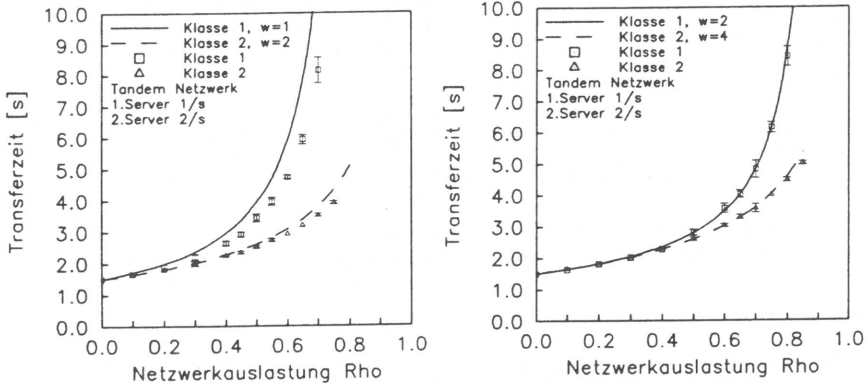


Bild 5.17: Flußäquivalentes Ersatznetzwerk

### 5.2.3 Schritthaltender Fenstermechanismus

Der schritthaltende Fenstermechanismus (Bild 3.10) ist extrem schwierig zu analysieren, da durch die Sammlung der 'Credits' beim Empfänger die Aggregation des Netzwerkes zusätzlich gestört wird. Es werden wiederum, wie bei der Zugangskontrolle, die Voraussetzungen für Produktform-Netzwerke verletzt. Für die einfacher zu analysierenden Verlustsysteme hat Mischa Schwartz [100] bei geeignet gewählter Fenstergröße, minimale Unterschiede zwischen gleitendem und schritthaltendem Fenstermechanismus nachgewiesen. Der schritthaltende Fenstermechanismus war, bedingt durch die Sammlung der 'Credits', die eine Verzögerung bewirken, leicht schlechter in seinem Leistungsverhalten. Von einer weiteren Behandlung des schritthaltenden Fenstermechanismus wird in dieser Arbeit daher abgesehen.

## 5.3 Ratenregelung

Die Ratenregelung bei verbindungsloser Kommunikation wird unterschieden in:

- Regelung des Flusses der Dateneinheiten und
- Regelung der Arbeit, die den Flußregelpunkt passiert.

Diese beiden Ausprägungen der Ratenregelung werden in den nächsten beiden Abschnitten analysiert. Der Fall a) kann als Spezialfall von Fall b) angesehen werden. Er soll jedoch wegen seiner einfachen Analyse separat betrachtet werden.

### 5.3.1 Ratenregelung der Dateneinheiten

Diese Art der Ratenregelung kann nach Kapitel 3.2.2.1.2 (Bild 3.19) durch ein G/D/m- $\infty$  Warteschlangensystem modelliert werden. Die Wartezeit in der Warteschlange repräsentiert dabei die Wartezeit vor dem Kontrollpunkt. Eine exakte Analyse des G/D/m- $\infty$

Warteschlangenmodells ist nicht möglich. Eine geeignete Approximation für die mittlere Wartezeit  $E[T_w]$  in einem  $G/G/m-\infty$  Warteschlangenmodell wurde von Toshikazu Kimura [66] entwickelt.

$$E[T_w(G/G/m)] = \frac{c_a^2 + c_b^2}{\frac{1 - c_a^2}{E[T_w(D/M/m)]} + \frac{1 - c_b^2}{E[T_w(M/D/m)]} + \frac{2(c_a^2 + c_b^2 - 1)}{E[T_w(M/M/m)]}} \quad (5.1)$$

$c_a^2$  - Variationskoeffizient der Verteilung der Ankunftsabstände  
 $c_b^2 = 0$  - Variationskoeffizient der Bedienzeitverteilung ( $G=D$ )

### 5.3.2 Ratenregelung der Arbeit

Die Modellierung der Ratenregelung über die Arbeit, die Dateneinheiten ins System bringen, geschieht über ein  $G/G/1-\infty$  System. Die Wartezeitverteilung  $W(t)$  zu den Zeitpunkten der Ankunft eines Auftrages stellt die unerledigte Arbeit dar, die ein Auftrag bei der Ankunft vor sich sieht. Abhängig von der Ausprägung der Zugangskontrolle ist die Analyse leicht unterschiedlich.

#### 5.3.2.1 Zulassung bei Unterschreitung von $U_{max}$

Kann der Auftrag den Kontrollpunkt passieren, wenn die unerledigte Arbeit vor ihm den Wert  $U_{max}$  unterschritten hat, so kann die Wartezeitverteilungsdichtefunktion  $\tilde{w}(\tilde{t})$  des Auftrages direkt angegeben werden (vgl. Abschnitt 3.2.2.1 und Bild 3.20).

$$\tilde{w}(\tilde{t}) = w(\tilde{t} + U_{max}) + \delta(\tilde{t}) \int_{0-}^{U_{max}} w(t) dt \quad ; \tilde{t} \geq 0 \quad (5.2)$$

In Bild 5.18 ist dieser Sachverhalt anschaulich dargestellt. Die Wartezeitverteilungsdichtefunktion  $w(t)$  wird neu skaliert mit  $\tilde{t} = t - U_{max}$  und die Anteile von  $w(t)$  für  $t < U_{max}$  werden zusammengefaßt zum Zeitpunkt  $\tilde{t} = 0$ .

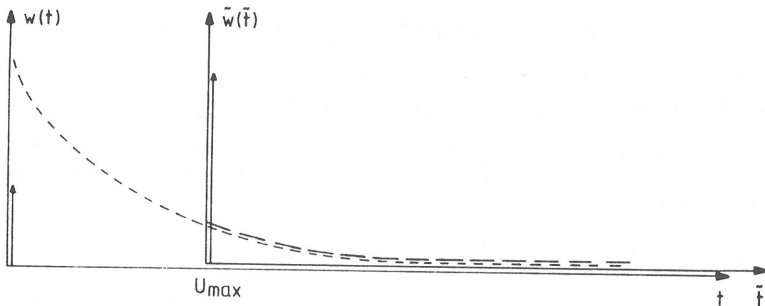


Bild 5.18: Skalierung der Wartezeitverteilungsdichtefunktion bei Ratenregelung

### 5.3.2.2 Zulassung bei Unterschreitung von $U_{max}$ mit Auftrag

Kann der Auftrag den Kontrollpunkt passieren, wenn die unerledigte Arbeit und seine eigene Arbeit, die er im System verursacht,  $U_{max}$  unterschreitet, so ist die Analyse in zwei Schritten vorzunehmen.

1. Die Zufallsvariable für die Wartezeit  $T_w$  und die Zufallsvariable für die Arbeit des Auftrages  $T_B$  werden addiert. Dies entspricht einer Faltung der Verteilungsdichtefunktionen oder einer Multiplikation der entsprechenden Laplace-Stieltjes-Transformierten. Man erhält die neue Verteilungsdichtefunktion  $w_a(t) = w(t) * b(t)$ .
2. Diese Verteilungsdichtefunktion  $w_a(t)$  wird wiederum neu skaliert.

$$\tilde{w}(\tilde{t}) = w_a(\tilde{t} + U_{max}) + \delta(\tilde{t}) \int_0^{U_{max}} w_a(t) dt \quad ; \tilde{t} \geq 0 \quad (5.3)$$

Die Arbeit des Auftrages sollte  $U_{max}$  nicht überschreiten. In der Realität wäre dann die unerledigte Arbeit vor ihm abgearbeitet, er könnte jedoch den Kontrollpunkt trotzdem nicht passieren, da seine Arbeit größer als  $U_{max}$  ist, was zu einem Verklemmungszustand führt. In der Analyse wird lediglich die unerledigte Arbeit betrachtet.

Die Ratenregelung über die Dateneinheiten ergibt sich als Spezialfall der Ratenregelung der Arbeit wenn eine konstante Arbeit  $d$  vorausgesetzt wird und  $U_{max} = md$  gewählt wird.

### 5.3.2.3 Beispiele

In diesem Abschnitt sollen zwei Beispiele für die Analyse bei Ratenregelung über die Arbeit dargestellt werden. Das erste Beispiel verwendet einen Markov-Ankunftsstrom von Aufträgen und ebenfalls negativ exponentielle Arbeitsanforderungen der Aufträge. Ein eintreffender Auftrag sieht dann folgende Verteilungsdichtefunktion der Arbeit (oder äquivalent der Wartezeit) vor sich:

$$w(t) = (1 - \rho)\delta(t) + \lambda(1 - \rho)\exp(-\mu(1 - \rho)t) \quad ; t \geq 0.$$

Für den Fall a), Zulassung bei Unterschreitung von  $U_{max}$ , ergibt sich:

$$\tilde{w}(\tilde{t}) = (1 - \rho \exp(-\mu(1 - \rho)U_{max}))\delta(\tilde{t}) + \lambda(1 - \rho)\exp(-\mu(1 - \rho)(\tilde{t} + U_{max})) \quad ; \tilde{t} \geq 0.$$

$$E[\tilde{T}_w] = \int_{-\infty}^{\infty} \tilde{t}\tilde{w}(\tilde{t})d\tilde{t} = \frac{\rho}{\mu(1 - \rho)}\exp(-\mu(1 - \rho)U_{max}).$$

Für den Fall b), Zulassung bei Unterschreitung von  $U_{max}$  mit Auftrag, ergibt sich:

$$\begin{aligned} w_a(t) &= w(t) * b(t) \quad ; b(t) = \mu \exp(-\mu t) \quad ; t \geq 0 \\ w_a(t) &= \mu(1 - \rho)\exp(-\mu(1 - \rho)t) \end{aligned}$$

$$\tilde{w}(\tilde{t}) = [(1 - \exp(-\mu(1 - \rho)U_{max}))]\delta(\tilde{t}) + \mu(1 - \rho)\exp(-\mu(1 - \rho)(\tilde{t} + U_{max})) \quad ; \tilde{t} \geq 0.$$

$$E[\tilde{T}_w] = \int_{-\infty}^{\infty} \tilde{t}\tilde{w}(\tilde{t})d\tilde{t} = \frac{1}{\mu(1 - \rho)}\exp(-\mu(1 - \rho)U_{max}).$$

In Bild 5.19 ist jeweils die mittlere Wartezeit für die beiden Fälle, für eine mittlere Arbeitsanforderung von 1s, dargestellt.

In einem zweiten Beispiel soll das GE/GE/1- $\infty$  System verwendet werden. Damit können jeweils 2 Momente der Verteilungsfunktion vorgegeben werden. Ein eintreffender Auftrag sieht die Verteilungsdichtefunktion (siehe auch Anhang A)

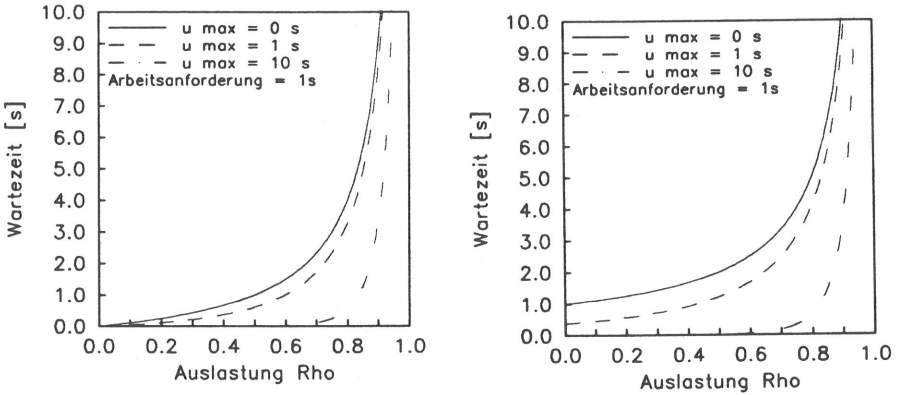


Bild 5.19: Mittlere Wartezeit bei Ratenregelung und negativ exponentiell verteilter Arbeitszeitanforderung. Links (Fall a): Zulassung bei Unterschreitung von  $U_{max}$  und rechts (Fall b): Zulassung bei Unterschreitung von  $U_{max}$  mit Auftrag.

$$w(t) = (1 - \rho) \frac{1 + c_b^2}{c_a^2 + c_b^2} \delta(t) + 2\mu(1 - \rho) \frac{c_a^2 - 1 + \rho(1 + c_b^2)}{(c_a^2 + c_b^2)^2} \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} t\right) \quad ; t \geq 0$$

für seine Wartezeit bei der Ankunft. Für den Fall a), Zulassung bei Unterschreitung von  $U_{max}$ , ergibt sich:

$$\begin{aligned} \tilde{w}(\tilde{t}) &= \delta(\tilde{t}) \int_0^{U_{max}} w(t) dt + w(\tilde{t} + U_{max}) \\ &= \left[ 1 - \frac{c_a^2 - 1 + \rho(1 + c_b^2)}{c_a^2 + c_b^2} \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} U_{max}\right) \right] \delta(\tilde{t}) \\ &\quad + 2\mu(1 - \rho) \frac{c_a^2 - 1 + \rho(1 + c_b^2)}{(c_a^2 + c_b^2)^2} \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} (\tilde{t} + U_{max})\right) \quad ; \tilde{t} \geq 0 \end{aligned}$$

Für die mittlere Wartezeit ergibt sich:

$$E[\tilde{T}_w] = \frac{c_a^2 - 1 + \rho(1 + c_b^2)}{2\mu(1 - \rho)} \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} U_{max}\right).$$

Für den Fall b), Zulassung bei Unterschreitung von  $U_{max}$  mit Auftrag, ergibt sich für die mittlere Wartezeit

$$E[\tilde{T}_w] = \frac{c_a^2 + c_b^2}{2\mu(1 - \rho)(1 + c_b^2)} \left[ 2 + \frac{(c_b^2 - 1)(c_a^2 - 1 + \rho(1 + c_b^2))}{c_a^2 + c_b^2} \right] \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} U_{max}\right).$$

In Bild 5.20 ist für beide Fälle für die Variationskoeffizienten  $c_a^2 = 2$  und  $c_b^2 = 2$  ( $\mu=1/s$ ) die mittlere Wartezeit für unterschiedliche  $U_{max}$  dargestellt.

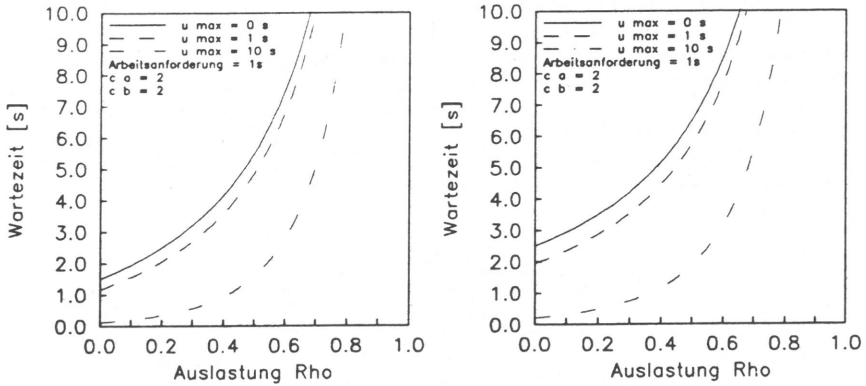


Bild 5.20: Mittlere Wartezeit bei Ratenregelung und verallgemeinerter negativ exponentieller Arbeitsanforderung. Links (Fall a): Zulassung bei Unterschreitung von  $U_{max}$  und rechts (Fall b): Zulassung bei Unterschreitung von  $U_{max}$  mit Auftrag.

## 5.4 Optimierung

Ziel der Optimierung ist es, die frei wählbaren Parameter der Kommunikation so einzustellen, daß eine optimale Kommunikation zustande kommt. Als frei wählbare Parameter werden in dieser Arbeit, bei verbindungsorientierter Kommunikation, die Fenstergrößen  $w$  und bei verbindungsloser Kommunikation die Werte  $U_{max}$  angesehen. Die Ankunftsprozesse werden als gegeben vorausgesetzt. Eine optimale Kommunikation kann je nach Standpunkt unterschiedlich sein. Ein Gesichtspunkt kann der maximale Durchsatz sein. Geht man von einem gegebenen Durchsatz aus, so ist eine möglichst kleine Transferzeit gewünscht, bei ebenfalls kurzer Transitzeit durch das Netzwerk. Der Netzbetreiber ist an einem möglichst guten Gesamtdurchsatz interessiert, der sich bei öffentlichen Netzen auf die Einnahmen niederschlägt. Da das Netzwerk meist mehrere Benutzer befriedigen muß, kommt als wichtiger Gesichtspunkt die Fairneß des Netzwerkes gegenüber diesen Benutzern hinzu. In den nächsten Abschnitten werden diese Gesichtspunkte der Optimierung nacheinander aufgegriffen und die jeweilige Zielfunktion aufgestellt.

### 5.4.1 Verbindungsorientierte Kommunikation

#### 5.4.1.1 Eine Verbindung

Bei der Optimierung der Fenstergröße für eine Verbindung gibt es die zwei Gesichtspunkte

1. maximaler Durchsatz und
2. minimale Transferzeit bei minimaler Transitzeit.

Die Optimierung auf maximalen Durchsatz sorgt ebenfalls für eine gute, oft maximale, Auslastung des Netzwerkes. Durch diese Optimierung wird sowohl die Geschwindigkeit des Senders an die Geschwindigkeit des Empfängers angepaßt, als auch einer Überlastung der Netzwerkressourcen entgegengetreten.

**Maximaler Durchsatz** Die Zielfunktion bei dieser Optimierung ist der maximale Durchsatz. Der freie Parameter ist die Fenstergröße  $w$ . Meist ist durch die Implementierung eine Einschränkung des Parameterbereiches durch die Puffer-Kapazität gegeben (d.h.  $1 \leq w \leq w_{max}$ ). Geht man davon aus, daß eine eindeutige optimale Fenstergröße  $w_{opt}$  existiert und, ausgehend von  $w_{opt}$ , der Durchsatz  $\Phi(w)$  bei fallender wie auch bei steigender Fenstergröße fällt, so kann diese Fenstergröße  $w_{opt}$  als Maximum von  $\Phi(w)$  ermittelt werden. Algorithmisch kann  $w_{opt}$  durch Intervallschachtelung gefunden werden. Das Vorgehen wird an dem Beispielnetzwerk in Bild 5.21 demonstriert. Die Durchsatzwerte werden bei kurzgeschlossener Flußregelung ermittelt. Die Lösung dieses Netzwerkes ist exakt möglich, solange die Knoten aus der Klasse der Produktformknoten sind.

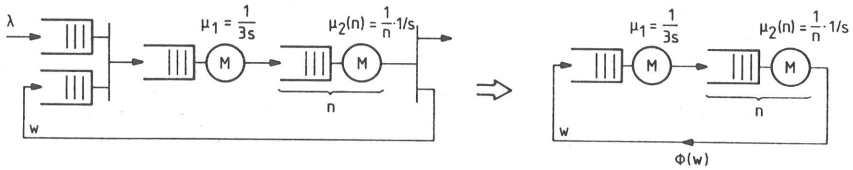


Bild 5.21: Beispielnetzwerk für Optimierung auf maximalen Durchsatz

In Bild 5.22 ist der maximale Durchsatz über der Fenstergröße aufgetragen. Man erkennt, daß die Fenstergröße  $w = 3$  das Optimum darstellt.

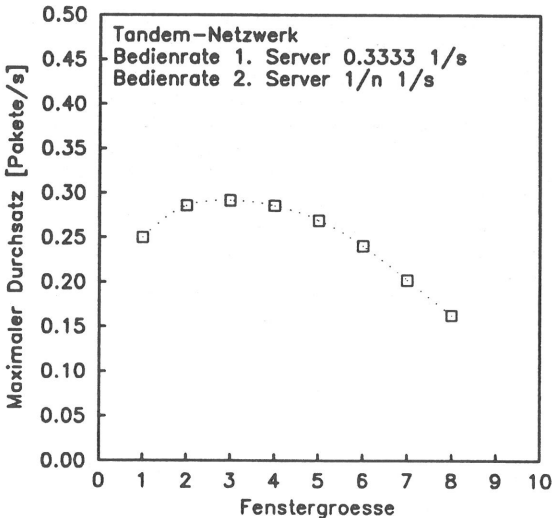


Bild 5.22: Maximaler Durchsatz als Funktion der Fenstergröße

Haben alle Knoten eine monoton steigende Bedienratenfunktion, so ist die optimale Fenstergröße mit  $w_{opt} = w_{max}$  sofort gegeben.



**Minimale Transferzeit** In diesem Abschnitt soll auf minimale Transferzeit bei minimaler Transitzeit optimiert werden. Die minimale Transferzeit ist hauptsächlich im Interesse des Benutzers, während die minimale Transitzeit im Interesse des Netzwerkes, bzw. des Netzbetreibers, ist. Die minimale Transferzeit sorgt für eine möglichst geringe mittlere Anzahl von Aufträgen im Netzwerk. Die Aufträge werden in die Zugangskontrolle verschoben. Als Zielfunktion wird daher  $P = t_s t_f$  verwendet. In Bild 5.23 ist  $P$  über der Fenstergröße für verschiedene Ankunftsraten aufgetragen. Es wurde dazu das Tandem-Netzwerk aus Bild 5.9 mit  $\mu_1 = 1/s$  und  $\mu_2 = 2/s$  verwendet. Aus  $P$  läßt sich das optimale Fenster für unterschiedliche Ankunftsraten ableiten (siehe Bild 5.24).

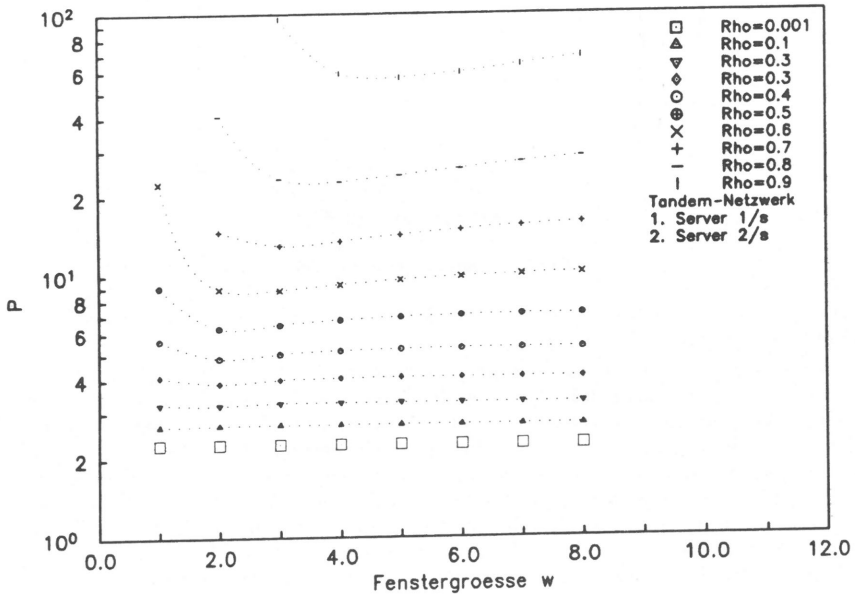


Bild 5.23: Optimierung auf minimale Transferzeit und minimale Transitzeit

Ein ähnlicher Ansatz ist von Mario Gerla und H.W. Chan [35] entwickelt worden.

#### 5.4.1.2 Mehrere Verbindungen

Für mehrere Verbindungen kann sowohl auf maximalen wie auch ausgewogenen Durchsatz für alle Verbindungen optimiert werden. Die Zielfunktion ist als Summe über alle Verbindungen zu definieren. Die Suche nach dem Optimum kann nicht mehr durch Inspektion aller Möglichkeiten geschehen. Stattdessen wird jeweils der Parameter einer Verbindung verändert, bis sich keine Verbesserung der Zielfunktion mehr ergibt. Dieser Vorgang wird mit allen Verbindungen sukzessive durchgeführt. Läßt sich keine Verbesserung der Zielfunktion durch ändern der Fenstergröße irgendeiner Verbindung erreichen, so geht man davon aus, daß man das Optimum erreicht hat. Dies setzt voraus, daß das Optimum ein-

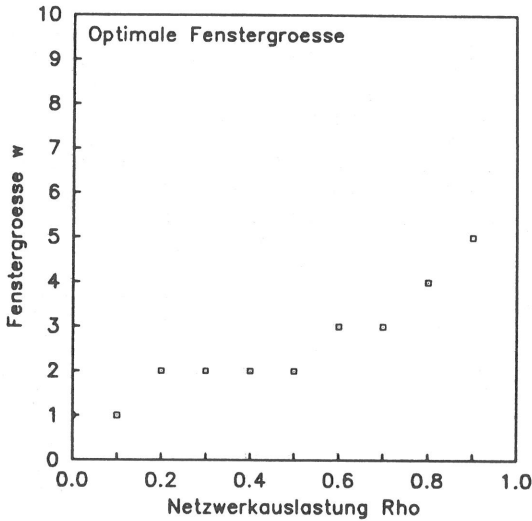


Bild 5.24: Optimale Fenstergröße

deutig ist und die mehrdimensionale Zielfunktion, vom Punkt des Optimums aus, nach allen Seiten steigt.

Als wesentlicher neuer Gesichtspunkt kommt bei mehreren Verbindungen die Fairneß hinzu. Es soll damit ausgedrückt werden, daß keine Verbindung ohne Grund gegenüber anderen Verbindungen bevorzugt oder vernachlässigt wird. Je nach Umgebungsbedingung kann die Zielfunktion unterschiedlich definiert werden. Jeffery M. Jaffe [61, 62] definiert seine Zielfunktion so, daß die Kapazität an den Engpässen gleichmäßig auf alle Verbindungen aufgeteilt wird. R.G. Gallager und S.J. Golestani [30] schlagen eine Kombination von Routing und Flußregelung mit ausgewogenem Durchsatz vor. Mario Gerla und M. Staskauskas [34] verallgemeinern diesen Ansatz, während J.W. Wong, Jacques P. Sauve und James A. Field [115] die Tarifierung bei minimaler Transferzeit als Kriterium heranziehen.

In dieser Arbeit wird die Zielfunktion

$$F = \sum_k \alpha_k (t_{if,k} - t_{if,k}^*)^2 (t_{ts,k} - t_{ts,k}^*)^2 \quad (5.4)$$

verwendet, die an die Definition von J.W. Wong, Jacques P. Sauve und James A. Field [115] angelehnt ist. Der Faktor  $\alpha_k$  ist ein frei wählbarer Faktor für die Bedeutung der Verbindung. Die Differenz  $t_k - t_k^*$  gibt den Abstand der aktuellen Transferzeit  $t_k$  von der Transferzeit  $t_k^*$  an, wenn lediglich Verbindung  $k$  im Netzwerk wäre. Die Zielfunktion versucht die Transferzeit für alle Verbindungen möglichst nahe bei der minimal möglichen Transferzeit zu halten.

An dem Netzwerk in Bild 5.25 wird das Vorgehen demonstriert.

In Tabelle 5.1 sind sowohl  $P$  als auch  $F$  für  $\alpha_k = 1$  dargestellt. Die optimale Wahl des Fensters ist jeweils  $w_1 = 1$  und  $w_2 = 3$ .

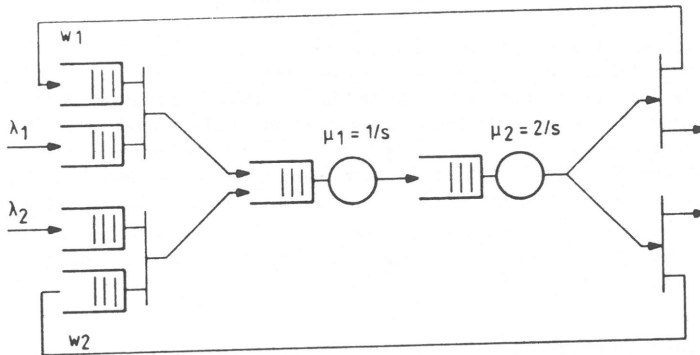


Bild 5.25: Beispielnetzwerk für faire Fensterzuteilung

$w_1$	$w_2$	$t_{ts}$		$t_{tf}$		$P$	$F$
		Klasse 1	Klasse 2	Klasse 1	Klasse 2		
1	2	4.103	2.484	6.957	12.620	59.893	185.429
1	3	4.103	3.166	6.957	7.293	51.634	183.937
1	4	4.103	3.720	6.957	6.322	52.062	184.175
2	2	5.204	2.484	5.920	12.620	62.156	238.539
2	3	5.204	3.166	5.920	7.293	53.897	237.047
2	4	5.204	3.720	5.920	6.322	54.326	237.285
3	3	5.616	3.166	5.841	7.293	55.893	281.041
3	4	5.616	3.720	5.841	6.322	56.321	281.279
4	4	5.760	3.720	5.834	6.322	57.122	300.892

Tabelle 5.1:  $P$  und  $F$  als Funktion der Fenstergrößen

#### 5.4.2 Verbindungslose Kommunikation

Die Optimierung bei verbindungsloser Kommunikation hat die gleichen Zielfunktionen, wie bei verbindungsorientierter Kommunikation, wenn man anstatt der Verbindungen die verwendeten Kommunikationsbeziehungen betrachtet. Der diskrete Parameterbereich der Fenstergröße bleibt bei Ratenregelung über die Dateneinheiten diskret oder wird zu einem kontinuierlichen Parameter bei der Ratenregelung über die Arbeit. Bei kontinuierlichem Parameter kann nicht jeder Parameterwert explizit überprüft werden. Bei eindeutigem Optimum wird er durch sukzessive Näherung approximiert. Existiert kein eindeutiges Optimum, so besteht die Gefahr, daß der Algorithmus bei einem 'Sub-Optimum' hängen bleibt. In der Praxis ist dieses 'Sub-Optimum' jedoch meist in der Nähe des Optimums, so daß weitere Anstrengungen, das Optimum zu finden, meist nicht unternommen werden.

## 5.5 Einbeziehung der Fehlerbehebung

In diesem Abschnitt soll die Fehlererkennung und Fehlerbehebung in ihrem Einfluß auf die Leistungsfähigkeit des Netzwerkes analytisch behandelt werden. In zwei Teilabschnitten wird für die Einzelquittierung und die Reihenfolge-Überwachung eine analytische Behandlung für verbindungsorientierte Protokolle dargestellt. Auf die analytische Behandlung der Untätigkeitsüberwachung ('Inactivity Control') wird verzichtet. Dieser Mechanismus könnte als eine Kette mit schwachem Verkehr behandelt werden, wobei die Intensität des Verkehrs aus dem Ankunftsprozeß, bei der entsprechenden Verbindung, abgeleitet wird.

Die Fehlerbehebung bei verbindungslosen Protokollen beschränkt sich auf Einzelquittierungen. Sie wird nicht gesondert behandelt, da sie bereits bei verbindungsorientierter Kommunikation behandelt wird.

### 5.5.1 Einzelquittierung

Für den Fall einer Verbindung kann das Modell aus Bild 3.12 zu einem  $G/G/1-\infty$  Warteschlangensystem vereinfacht werden. Die  $G$ -Bedieneinheit wird bei Fehlern mehrfach belegt (siehe Bild 5.26).

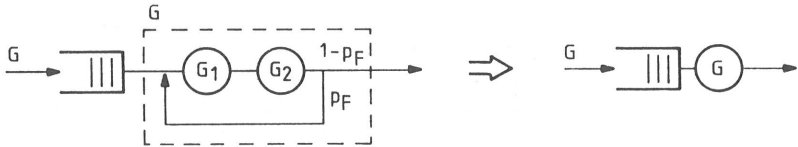


Bild 5.26: Analytisches Modell der Einzelquittierung

Die Zugangsverzögerung ergibt sich aus der Wartezeit des  $G/G/1-\infty$  Systems. Die Transitzeit der Dateneinheiten kann aus den Bedienphasen und der Fehlerwahrscheinlichkeit  $p_F$  berechnet werden. Für mehrere Verbindungen kann diese Vereinfachung nicht durchgeführt werden. Durch Einführung zweier neuer Klassen von Aufträgen für jede Verbindung können die Fehler jedoch repräsentiert werden. Die Fehlerwahrscheinlichkeit entspricht der Übergangswahrscheinlichkeit von Dateneinheiten zu der neuen Klasse, der negativen Quittierungen, die zu den wiederholten Dateneinheiten beim Sender werden. Diese Modelle können somit ohne zusätzliche Probleme mit den Verfahren von Abschnitt 5.2 gelöst werden.

### 5.5.2 Reihenfolgeüberwachung

Die analytische Behandlung der Reihenfolgeüberwachung wurde in der Literatur für die Datensicherungsschicht sowohl für 'Go-Back-N' als auch 'Selective Repeat' dargestellt.

Mischa Schwartz [100] analysiert die Verfahren unter idealisierten Bedingungen. Für vereinfachte Versionen des 'Go-Back-N' Protokolles haben A. Konheim [69] und D.M. Chitre [18] Analysen durchgeführt. Werner Bux, Karl Kümmerle, Hong Linh Truong [10], J. Labetoulle, Guy Pujolle [79] und Malcolm C. Easton [25] haben die Verfahren für die realen Protokolle HDLC, ADCCP und SDLC untersucht. M. Bechtold, Guy Pujolle, Otto Spaniol [6] und D.W. Andrews [2] untersuchten das 'Selective Repeat'-Verfahren für

Verbindungen über Satelliten. Durch das Aufkommen von breitbandigen Übertragungskanälen bei lokalen Netzen wurden weitere Untersuchungen, eines modifizierten 'Selective Repeat'-Verfahrens, durch Werner Bux, Parviz Kermani, Wolfgang Kleinöder [11] und Phouc Tran-Gia, Hamid Ahmadi, Parviz Kermani [109] durchgeführt. Z. Rosberg und N. Shacham [97] haben sich speziell mit der Wartezeitverteilung beim Empfänger durch das Wiederherstellen der Reihenfolge beschäftigt.

Hauptgesichtspunkt dieser Untersuchungen war es jeweils, die Ausnutzung des Übertragungskanales in Abhängigkeit der Übertragungsgeschwindigkeit, der Fehlerrate und der Rahmenlänge zu berechnen. Bei dem 'Selective Repeat'-Verfahren wurde die Dimensionierung der Empfangsspeicher als weiterer einschränkender Parameter betrachtet.

Der Sender wurde bei diesen Untersuchungen als 'gesättigt' angesehen, d.h. er hatte immer Dateneinheiten zum Übertragen.

In diesem Abschnitt soll das 'Selective Repeat'-Verfahren bei gegebenem Ankunftsprozeß für eine Verbindung in den höheren Schichten untersucht werden. Der Übertragungskanal soll jeweils durch ein Netzwerk dargestellt werden, im Gegensatz zu den Untersuchungen auf der Datensicherungsschicht. Dort kann der Übertragungskanal als die Bedienung der Dateneinheiten entsprechend ihrer Rahmenlänge und einer Laufzeit modelliert werden.

Ausgehend von dem Grundmodell in Bild 5.27 wird unter der 'Produktform-Voraussetzung' für das Netzwerk der flußäquivalente Ersatzknoten konstruiert. Die Wartezeit bei der Wiederherstellung der Reihenfolge muß dabei abgeschätzt werden. Diese Abschätzung wird für geringe Fehlerwahrscheinlichkeiten von Dateneinheiten durchgeführt.

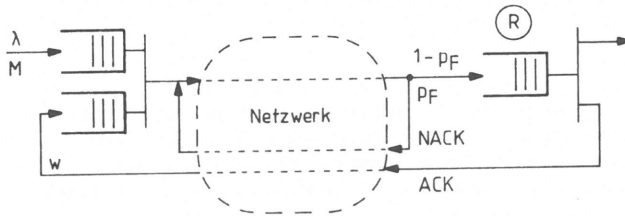
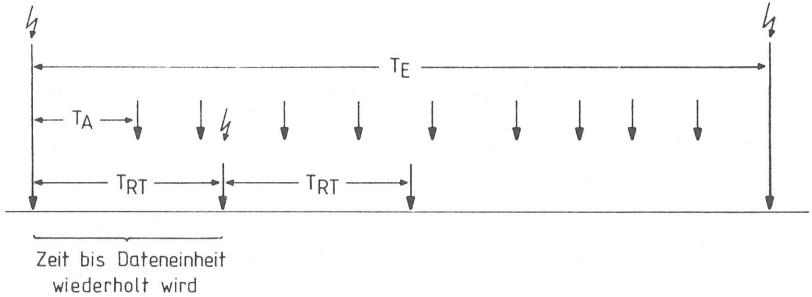


Bild 5.27: Grundmodell für 'Selective Repeat'-Verfahren

In Bild 5.28 ist das Intervall bei der Warteschlange R zwischen zwei fehlerhaft ankommenden Dateneinheiten dargestellt, die das erstmalig übertragen werden.

Aus der Wiederholungs-Zeit  $T_{RT,x}$ , bei  $x$  Aufträgen im Netzwerk, kann mit der Fehlerwahrscheinlichkeit  $p_F$  eine virtuelle Wiederholungszeit  $T_{VRT,x}$  berechnet werden.

$$\begin{aligned}
 T_{VRT,x} &= (1-p_F)T_{RT,x} + p_F(1-p_F)(T_{RT,x} + T_{RT,x}) \\
 &\quad + p_F^2(1-p_F)(T_{RT,x} + T_{RT,x} + T_{RT,x}) + \dots \quad (5.5) \\
 E[T_{VRT,x}] &= h_{VRT,x} = \frac{1}{1-p_F} h_{RT,x}
 \end{aligned}$$



$T_A$	-	Ankunftsabstand	$F_{T_A} = 1 - e^{-\lambda t}$	$h_A = E[T_A] = 1/\lambda$
$T_E$	-	Abstand der Fehler	$F_{T_E} = 1 - e^{-t/h_E}$	$h_E = E[T_E] = h_A/p_E$
$T_{RT,x}$	-	Wiederholungszeit	$F_{T_{RT,x}} = 1 - e^{-ct}$	$h_{RT} = E[T_{RT,x}] = 1/c$
$x$	-	Anzahl der Aufträge im Netzwerk		

Bild 5.28: Zeitintervall zwischen zwei Fehlern bei der Erstübertragung

Die mittlere Wartezeit  $E[T_w]$  einer beliebigen Dateneinheit ist durch

$$E[T_{w,x}] = h_{w,x} = 1/\mu_{w,x} = q_{w,x} \frac{E[T_{VRT,x}^2]}{2E[T_{VRT,x}]} \quad (5.6)$$

gegeben. Es wird von negativ exponentiell verteilten Ankunftsabständen ausgegangen, so daß  $E[T_{VRT,x}^2] = E^2[T_{VRT,x}]$  ist.  $q_{w,x}$  stellt dabei die Wahrscheinlichkeit dar, daß die Dateneinheit warten muß. Diese Wahrscheinlichkeit wird approximiert durch

$$q_{w,x} = \frac{n_x}{N}. \quad (5.7)$$

$N$  drückt die Anzahl der Dateneinheiten aus, die in dem Intervall zwischen zwei Fehlern bei Erstübertragungen im Mittel ankommen:  $N = \lambda h_E = 1/p_F$ .  $n_x$  drückt die Anzahl der Dateneinheiten aus, die in diesen Intervall ankommen, aber warten müssen.  $n_x$  ist beschränkt durch die Population des Netzwerkes  $x$  und wird als Summe über das Ankommen von  $i$  Dateneinheiten, gewichtet mit deren Wahrscheinlichkeit  $n_{x,i}$ , berechnet.

$$n_x = \sum_{i=1}^x i n_{x,i} \quad (5.8)$$

Die virtuelle Wiederholzeit wird als negativ exponentiell vorausgesetzt, womit  $n_{x,i}$  durch

$$\begin{aligned} n_{x,i} &= \int_0^\infty \frac{(\lambda t)^i}{i!} e^{-\lambda t} dF_{T_{VRT}} = \int_0^\infty \frac{(\lambda t)^i}{i!} e^{-\lambda t} \frac{1-p_F}{p_F h_{RT}} \epsilon \exp\left(-\frac{t(1-p_F)}{p_F h_{RT}}\right) dt \\ &= \frac{h_A}{h_A + h_{RT,i}} \left( \frac{h_{RT,i}}{h_A + h_{RT,i}} \right)^i \end{aligned} \quad (5.9)$$

approximiert wird. Mit dieser mittleren Wartezeit in der Warteschlange R läßt sich, zusammen mit dem flußäquivalenten Ersatznetzwerk, die weitere Analyse durchführen, indem ein neuer flußäquivalenter Ersatzknoten gebildet wird (siehe Bild 5.29).

$$\mu(x)^* = \sum_{i=1}^x \mu(i) p_i = \sum_{i=1}^x \frac{\mu(i)}{1 + \sum_{k=1}^x \prod_{j=1}^k \frac{(x-j+1)\mu_2}{\mu_1(j)}} \prod_{j=1}^i \frac{(x-j+1)\mu_{w,x}}{\mu(j)} \quad (5.10)$$

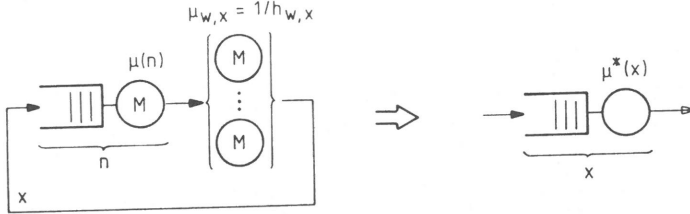


Bild 5.29: Flußäquivalentes Ersatznetzwerk bei 'Selective Repeat'-Verfahren

Dieser flußäquivalente Ersatzknoten wird, als begrenzt zustandsabhängig, bis zur Fenstergröße  $w$  und mit geteilter Warteschlange, betrachtet (vgl. Abschnitt 5.2.1.2). In Bild 5.30 ist für das Tandem-Netzwerk (Bild 5.9) die Transferzeit über der Ankunftsrate für die Fenstergröße  $w = 2$  und die drei Fehlerwahrscheinlichkeiten  $p_F = 0\%, 1\%$  und  $10\%$  dargestellt. Für kleine Ankunftsraten stimmen simulatives Ergebnis und approximatives Ergebnis sehr gut überein. Für steigende Ankunftsraten werden die Diskrepanzen leicht größer.

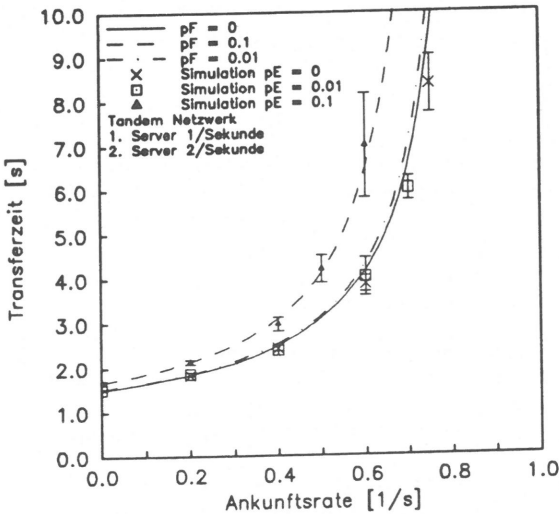


Bild 5.30: Ergebnisse für Tandem-Netzwerk mit Flußregelung

## Kapitel 6

# Analytische Leistungsuntersuchung von Flußregelungen in Protokollarchitekturen

In diesem Kapitel wird die Analyse von mehrschichtigen Flußregelungen in Protokollarchitekturen und Sequenzen von Verbindungen dargestellt. Jeder Verbindungsabschnitt einer Sequenz kann seinerseits wieder aus einer mehrschichtigen Protokollarchitektur aufgebaut sein. Die Analyse beschränkt sich auf verbindungsorientierte Protokolle, da höhere Protokolle meist verbindungsorientiert arbeiten und dies somit in realen Systemen häufig anzutreffen ist. Die Analyse geht von der Annahme der Produktform für die Teilnetzwerke aus und aggregiert sie jeweils zu einem flußäquivalenten Ersatzknoten. Für allgemeine Teilnetzwerke kann diese Aggregation nur mit starken Approximationen durchgeführt werden. Damit ist der Fehler i.a. nicht kontrollierbar.

### 6.1 Literaturübersicht

Die Analyse von Protokollmechanismen wurde bereits von zahlreichen Autoren behandelt. Die Analyse mehrschichtiger Protokollarchitekturen beschränkt sich auf einige wenige Artikel, von denen hier die wichtigsten zitiert werden sollen.

Arthur Goldberg und Gerald Popek [39] haben die Analyse von einer Datenverarbeitungsanlage auf mehrere Datenverarbeitungsanlagen ausgedehnt, die über ein Netzwerk miteinander verbunden sind und ein verteiltes Betriebssystem verwendeten. Das Netzwerk wurde zu einem flußäquivalenten Ersatzknoten aggregiert, so daß insgesamt ein Netzwerk entstand, das die Bedingungen der Produktform-Netzwerke erfüllte. Die Autoren verwendeten die approximative Mittelwertanalyse zur Lösung des Netzwerkes.

Mehrere Autoren haben Anfang der 80er Jahre den Anschluß von Stationen an ein lokales Netz untersucht. Dabei wurde jeweils von zwei bis drei Schichten ausgegangen:

- obere Schicht: Kommunikationsprozessor oder Anwendung,
- mittlere Schicht: Netzwerkanschluß ('Network Interface Unit') und
- untere Schicht: Zugriffsverfahren.

Die Analysen der Schichten wurden teilweise individuell, ohne Berücksichtigung der Rückwirkungen der anderen Schichten, durchgeführt, oder es wurde jeweils ein flußäquivalenter



Ersatzknoten der niederen Schicht gebildet. Zusätzliche Protokollmechanismen wurden meist nicht berücksichtigt.

Einige dieser Autoren sind G. Bernard [7], Tsvi Lissack, Basil Maglaris, Hubert Chin [85] und Wikhard Kiesel [65]. M. Murata, H. Takagi [91] berücksichtigen zusätzlich Prioritäten für die Prozessoren und setzen die approximative Mittelwertanalyse mit Prioritäten ein.

In den Artikeln von J.W. Wong, J.A.B. Moura, J.A. Field [114] und L.C. Mitchell, D.A. Lide [90] wird dasselbe 3-schichtige Modell mit dem zusätzlichen Protokollmechanismus Flußregelung betrachtet. Die Artikel unterscheiden sich lediglich im Detaillierungsgrad. J.W. Wong, J.A.B. Moura und J.A. Field [114] lösen die untere Schicht (MAC) simultan, während L.C. Mitchell und D.A. Lide [90] eine approximative analytische Lösung verwenden. Nachdem die untere Schicht zu einem flußäquivalenten Ersatzknoten aggregiert wurde, werden die untere und mittlere Schicht jeweils mit der Mittelwertanalyse gelöst, wobei die Flußregelungen nach der Methode des flußäquivalenten Ersatznetzwerkes (siehe Kapitel 5) repräsentiert werden. Die Übertragung der Quittierungen vom Empfänger zum Sender wird jeweils vernachlässigt.

Bei Ottmar Gühr und Paul J. Kühn [36] wird eine 2-stufige Anordnung mit den Schichten

- Zugriffsprotokoll (MAC) und
- Datensicherungsschicht (LLC) mit Flußregelung betrachtet.

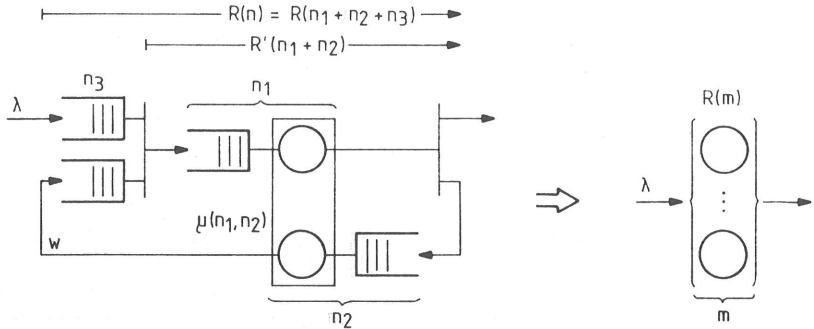
Das Zugriffsprotokoll wird approximativ analytisch gelöst und zu einer Verzögerungszeit aggregiert. Der Prioritätsprozessor auf der LLC-Schicht wird dekomponiert für die individuellen Verbindungen, die Sende- und die Empfangsrichtung. Das Modell wird durch Aggregation (Norton-Theorem) und Lösung des Zustandsraums berechnet.

Bei Serge Fdida, Andrzej Milk und Harry Perros [27] wird eine hierarchisch flußregulierte Verbindung betrachtet. Die Zugangswartezeiten werden mit der Methode des flußäquivalenten Ersatznetzwerkes berechnet. Die Argumentation greift jedoch nur indirekt darauf zurück. Die Rückübertragung der Quittierungen vom Empfänger zum Sender wird berücksichtigt, aber die Konstruktion des flußäquivalenten Ersatzknotens für die Übertragung der Daten vernachlässigt die Population durch Quittierungen, wenn die Quittierungen die gleichen Ressourcen des Netzwerkes wie die Dateneinheiten verwenden. In Bild 6.1 sind das flußregulierte Netzwerk und der Ersatzknoten dargestellt.

Die Transferzeit  $R(n)$  wird in Gleichung 6.1 richtig dargestellt, aber im Ersatzknoten wird eine andere Population zugrunde gelegt.

## 6.2 Hierarchisch flußregulierte Verbindungen

Für die Analyse hierarchisch flußregulierter Verbindungen muß, als wesentlicher Bestandteil, die Flußregelung auf der unteren Schicht ersetzt werden (siehe Bild 3.25). Die Ersatzdarstellung wird so gewählt, daß sie in die Analyse der oberen Schicht eingebunden werden kann. Im nächsten Abschnitt wird die Konstruktion des flußäquivalenten Ersatzknotens und dessen Einbettung in die Analyse der höheren Schicht dargestellt.



$$R(n) = \begin{cases} R'(n) & ; n \leq w \\ R'(w) + (n - w)/\gamma(w) & ; n > w \end{cases} \quad (6.1)$$

$\gamma(w)$  - Durchsatz des geschlossenen Netzwerkes bei Population  $w$

Bild 6.1: Flußäquivalenter Ersatzknoten bei Serge Fdida, Andrzej Milk und Harry Perros [27]

### 6.2.1 Konstruktion des flußäquivalenten Ersatzknotens

Es wird davon ausgegangen, daß die ( $N$ )-Schicht und die unteren Schichten keine gemeinsamen Ressourcen haben, daß aber die Dateneinheiten und Quittierungen auf den unteren Schichten ( $1 \dots N-1$ ) die Ressourcen gemeinsam verwenden. Existieren getrennte Ressourcen für Daten und Quittierungen, so kann der flußäquivalente Ersatzknoten aus Abschnitt 5.2.1.2 (siehe Bild 5.4) nach dem Schritt 2b verwendet werden. Existieren gemeinsame Ressourcen auf der ( $N$ )-Schicht und den unteren Schichten, so muß ein Weg gesucht werden, diese Ressourcen zu dekomponieren. Dies kann zum Beispiel nach der Methode des 'Load Concealment' geschehen. Der Fall gemeinsamer Ressourcen von Dateneinheiten und Quittierungen auf derselben Schicht wird in den nächsten Abschnitten behandelt.

In Abschnitt 5.2.1.2 (siehe Bild 5.5) wurde eine Verbindung für gemeinsame Ressourcen von Daten und Quittierungen analysiert. Dazu wurde die zweidimensionale Bedienratenfunktion  $\mu(\underline{n})$  berechnet und der Zustandsraum aufgestellt. Nach Berechnung der Zustandswahrscheinlichkeiten ließen sich daraus die Zugangsverzögerung und die Transitzeit berechnen.

Diese Zustandswahrscheinlichkeiten werden nun zur Konstruktion des flußäquivalenten Ersatzknotens verwendet. Die Bedienraten für Dateneinheiten werden, gewichtet mit der Zustandswahrscheinlichkeit, zeilenweise aufsummiert und bezogen auf die Randwahrscheinlichkeit der Zeile.

$$\mu^*(n) = \begin{cases} \sum_{\forall n_1+n_3=n} \frac{\mu(n_1, n_2) P(n_1, n_2, n_3)}{P^*(n_1, n_3)} & ; n \leq w \\ \sum_{\forall n_1+n_3 \geq w} \frac{\mu(n_1, n_2) P(n_1, n_2, n_3)}{\left(1 - \sum_{\forall n_1+n_3 < w} P^*(n_1, n_3)\right)} & ; n > w \end{cases}$$

$$P^*(n_1, n_3) = \sum_{n_2=0}^w P(n_1, n_2, n_3) \quad ; \text{ Randwahrscheinlichkeit}$$

Damit ist die Bedienrate des fluquivalenten Ersatzknotens

$$\mu(n) = \begin{cases} \mu^*(n) & ; n < w \\ \mu^*(w) & ; n \geq w \end{cases}$$

gegeben. Da eine Halb-Duplex-Verbindung auf der (N)-Schicht eine Voll-Duplex Verbindung auf der (N-1)-Schicht zur Folge hat, mu die (N-1)-Schicht entsprechend Abschnitt 5.2.2.3 (Bild 5.15) iterativ gelst werden, indem immer eine Verbindung durchgerechnet wird und die mittlere Anzahl der Auftrge der anderen Kette entsprechend bercksichtigt wird. Jede Halb-Duplex-Verbindung wird dann durch einen fluquivalenten Ersatzknoten reprsentiert. Sind auf der (N)-Schicht ebenfalls mehrere Verbindungen zu untersuchen, so mu dort auch eine Iteration durchgefhrt werden. Meist wird dann schnellere Konvergenz erreicht, wenn die Iterationen nicht getrennt durchgefhrt werden, sondern jeweils ein Iterationsschritt auf der (N)-Schicht und dann ein Iterationsschritt auf der (N-1)-Schicht durchgefhrt wird.

### 6.3 Analyse der MAP-Architektur

In diesem Abschnitt soll die MAP-Architektur, wie sie als vereinfachtes Modell in Bild 3.24 dargestellt ist, analysiert werden. Dazu werden in Unterabschnitten die genaueren Parameter definiert, die dem Modell zugrunde gelegt werden, und die Ersetzung des Zugriffsverfahrens (CSMA/CD) durch einen fluquivalenten Ersatzknoten dargestellt. Abschlieend werden die Analyse der Protokollarchitektur durchgefhrt, die Ergebnisse dargestellt und mit Simulationen validiert.

#### 6.3.1 Modell und Modellparameter

Die Aufteilung der Bedienphasen auf die in der Hardware vorhandenen Prozessoren wird entsprechend Abschnitt 3.5.1 durchgefhrt. In Bild 6.2 ist das Modell mit dieser Aufteilung und den gewhlten Parametern dargestellt. Alle Bedienphasen werden als negativ-exponentiell angenommen und die Abarbeitungsreihenfolge der gemeinsamen Warteschlange sei FIFO. (Zur besseren Verstndlichkeit ist vor jeder Bedienphase eine Warteschlange dargestellt, die jedoch gemeinsam fr alle Auftrge ist, die den Prozessor durchlaufen). Die Wahl der Parameter ist in Tabelle 6.1 dargestellt. Fr die Schichten 5, 6 und 7a wurde die Bedienzeit einheitlich zu 3ms gesetzt. Sie spiegelt in guter Nherung die Bedienzeiten des real zugrunde gelegten Systems wieder. Zur Motivation dieser Werte sei auf Martin Bosch, Ottmar Gahr, Wikhard Kiesel [8, 9] verwiesen. Die Werte der Bedienzeiten auf den Schichten 2b, 3 und 4 wurden einheitlich auf 1ms gesetzt. Als Grundlage diente der Artikel von Bernd Werner Meister und Philippe A. Janson [89].

Da die MAP-Architektur hauptschlich in der Fabrikautomation eingesetzt wird, wird auf der Schicht 2a bzw. 1 die mittlere Rahmenlnge entsprechend den dortigen Verhltnissen auf  $d = 100$  Byte gesetzt. Bei Ottmar Gahr und Michael Weixler [38] ist die Messung der Rahmenlnge auf dem lokalen Netz in einer Fabrik dargestellt, die zu dieser Annahme berechtigt.

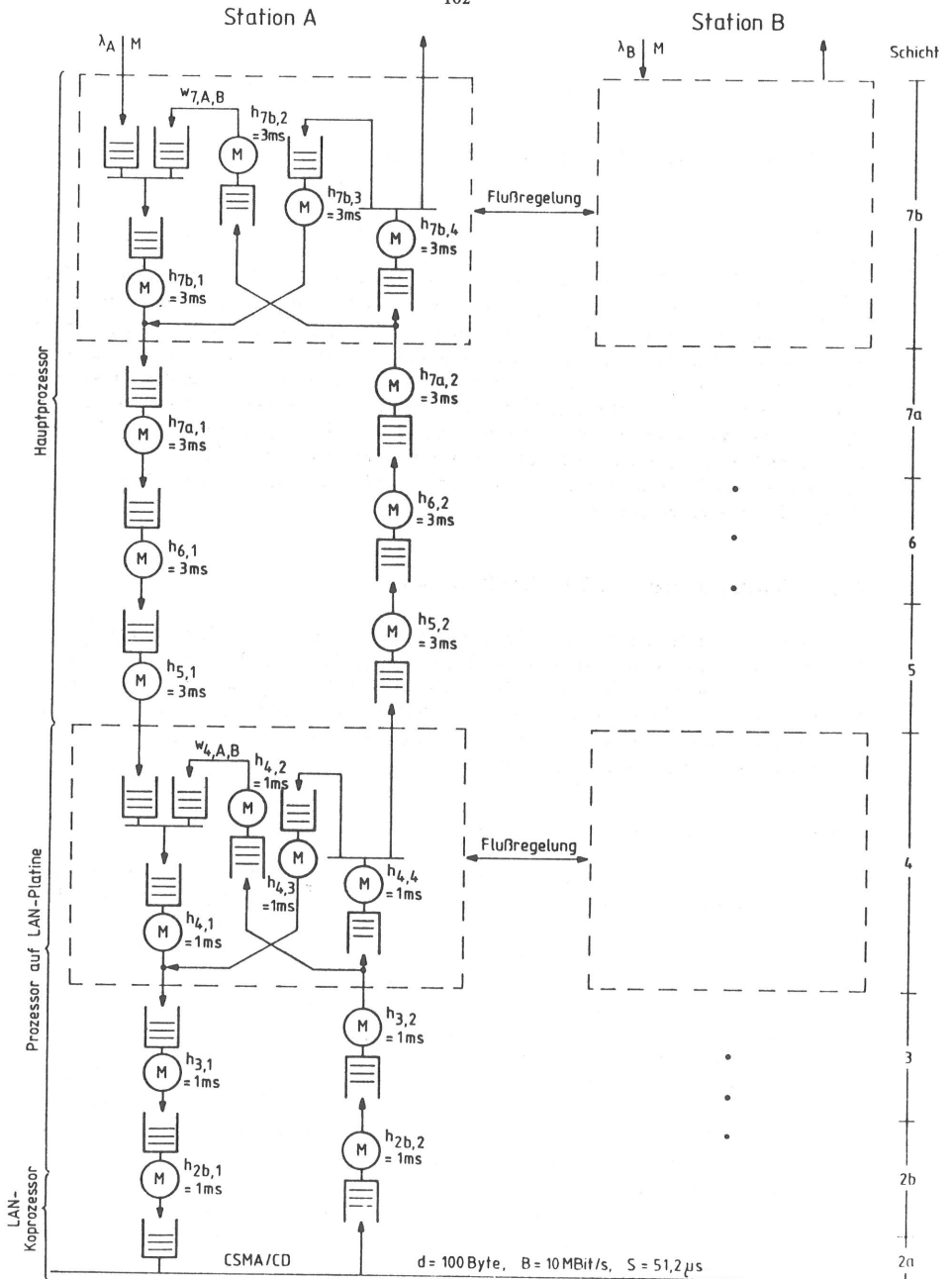


Bild 6.2: MAP-Modell

Schicht	Wert
7b	$h_{7b,x} = 3 \text{ ms}$
7a	$h_{7a,x} = 3 \text{ ms}$
6	$h_{6,x} = 3 \text{ ms}$
5	$h_{5,x} = 3 \text{ ms}$
4	$h_{4,x} = 1 \text{ ms}$
3	$h_{3,x} = 1 \text{ ms}$
2b	$h_{2b,x} = 1 \text{ ms}$
2a	$d = 100 \text{ Byte}$
	$B = 10 \text{ MBit/s}$
	$S = 51.2 \mu\text{s}$

- $h$  - Mittlere Bedienzeit  
 $d$  - Mittlere Rahmenlänge  
 $B$  - Übertragungsrate  
 $S$  - 'Slot' für Kollisionsauflösung

Tabelle 6.1: Modellparameter

### 6.3.2 Flußäquivalenter Ersatzknoten für Kanalzugriff

In diesem Abschnitt wird ein approximativer flußäquivalenter Ersatzknoten des Kanalzugriffsverfahrens CSMA/CD konstruiert. Die Ableitung folgt dem Vorschlag von Edward D. Lazowska, John Zahorjan, G. Scott Graham und Kenneth C. Sevcik [83]. Die Zeit wird in Zeitschlitze ('Slots') eingeteilt. Wenn  $n$  Stationen sendewillig sind und jede Station mit der Wahrscheinlichkeit  $1/n$  in dem betrachteten 'Slot' anfängt zu senden, dann ist die Wahrscheinlichkeit  $A$ , daß eine Station den Kanal erhält, gleich der Wahrscheinlichkeit, daß genau eine Station anfängt zu senden.

$$A = n \frac{1}{n} \left(1 - \frac{1}{n}\right)^{n-1} \quad (6.2)$$

Die mittlere Anzahl der 'Slots', welche bis zu der erfolgreichen Vergabe des Kanals vergehen, ist dann :

$$C = \sum_{i=1}^{\infty} i A (1 - A)^i = \frac{1 - A}{A}. \quad (6.3)$$

Die Effektivität des Kanales läßt sich damit ausdrücken durch

$$\begin{aligned}
 E(n) &= \frac{\text{Länge des Übertragungsintervalles}}{\text{Länge des Übertragungsintervalles} + \text{Zeit bis Kanal erfolgreich vergeben ist}} \\
 &= \frac{d/B}{d/B + CS}.
 \end{aligned}$$

Die lastabhängige Bedienrate ergibt sich damit zu

$$\mu_{LAN}(n) = \frac{B}{d} E(n) \quad (6.4)$$

Validierungen mit Simulationen haben gezeigt, daß dieser Ersatzknoten das Verhalten von CSMA/CD genügend genau nachbildet.

### 6.3.3 Auslastung der Prozessoren

Durch die Betrachtung der Auslastungen der zwei Prozessoren und des Übertragungsmediums bei gegebenem Angebot an die Schicht 7b, kann ein Überblick über das Systemverhalten ohne Flußregelung gewonnen werden. In Bild 6.2 ist die Auslastung der Prozessoren über der Paketrate der Anwendung (Benutzer der Schicht 7) dargestellt.

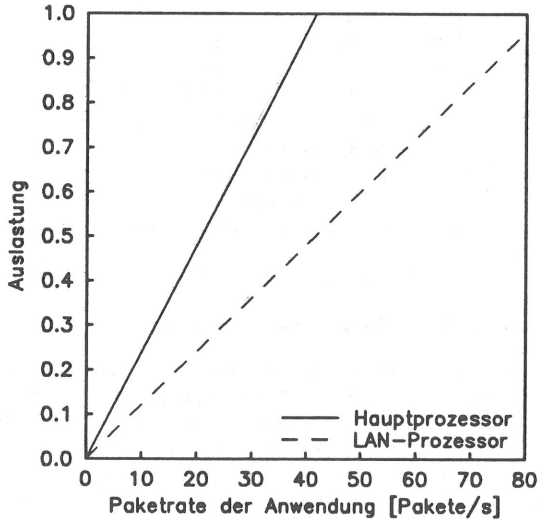


Bild 6.3: Auslastung der Prozessoren

Der maximale Durchsatz ergibt sich zu

Hauptprozessor = 37,04 Anwendungs-Pakete /s und

LAN-Prozessor = 71,43 Anwendungs-Pakete /s.

Der LAN-Prozessor überträgt für jedes Anwendungs-Paket zwei 'Schicht 4'-Pakete, da in der Schicht 7b eine Quittierung erzeugt wird. Die Schicht 4 verwendet ebenfalls Quittierungen, so daß auf dem Medium vier Rahmen übertragen werden für ein Anwendungs-Paket. Das Übertragungsmedium hat für den ausschließlichen Gebrauch durch zwei Stationen einen maximalen Durchsatz, der bei über 5000 Anwendungs-Paketen/s liegt. Die Auslastung des Mediums ist daher so gering, daß sie im Diagramm nicht dargestellt werden kann.

### 6.3.4 Analyse der Schichten 2b-4

Nachdem aus der Schicht 2a (MAC) ein flußäquivalenter Ersatzknoten gebildet wurde, werden die Schichten 2b-4 ebenfalls in die Analyse mit einbezogen. Der flußäquivalente Ersatzknoten wird, entsprechend Abschnitt 5.2.1.2, mit Hilfe des 'Norton Theorems' für jede Paketrate der Anwendung neu berechnet. In Tabelle 6.2 ist die Bedienrate für einige Paketraten und einige Zustände dargestellt.

	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$
$\lambda = 10/s$	149.500	205.223	233.914	251.394	263.151	271.287	269.645	279.924
$\lambda = 20/s$	133.828	188.129	217.225	235.356	247.620	254.231	234.615	261.276
$\lambda = 30/s$	117.303	169.569	198.934	217.754	230.271	232.517	191.712	239.581
$\lambda = 40/s$	99.723	149.035	178.415	197.956	210.078	204.520	145.161	214.311
$\lambda = 50/s$	80.661	125.719	154.710	174.999	185.405	168.659	98.517	184.877
$\lambda = 55/s$	70.546	112.776	141.372	162.022	170.660	147.270	76.181	168.281
$\lambda = 60/s$	60.104	98.964	127.082	148.042	153.860	123.429	55.200	150.224
$\lambda = 65/s$	49.614	84.732	112.560	133.679	134.948	97.555	36.451	130.664

Tabelle 6.2: Bedienrate  $\mu(n)$  (Einheit [1/s]) des fluquivalenten Ersatzservers der Schichten 2b-4

Die fluquivalenten Ersatzknoten sind identisch fr beide Richtungen, da die Datenpakete und Quittierungspakete der Schicht 7 dieselben Charakteristiken haben. Die separate Untersuchung der Schicht 4 ergibt eine 'Schicht 4'-Transferzeit, wie sie in Bild 6.4 dargestellt ist.

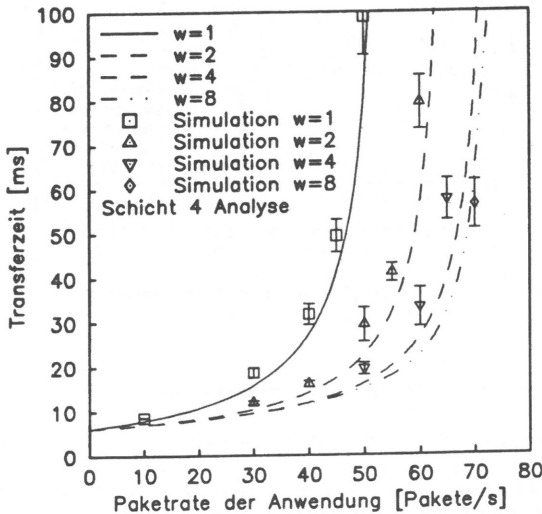


Bild 6.4: 'Schicht 4'-Transferzeit

Die dargestellten Ergebnisse beziehen sich auf eine Halb-Duplex-Verbindung, nachdem die Aggregation zu dem fluquivalenten Ersatzknoten durchgefhrt wurde. Es wurden dann anhand des eindimensionalen Zustandsraumes die Leistungsgren berechnet. Die Halb-Duplex-Verbindungen auf der Schicht 4 sind symmetrisch jeweils mit demselben Angebot belastet. Daher kann Bild 6.4 auch als Validierung der Methode der Bildung des fluquivalenten Ersatzknotens angesehen werden. Die Simulationsergebnisse, mit 95% Vertrauensintervallen, zeigen eine gute bereinstimmung.

### 6.3.5 Separate Analyse der Schichten 5-7

Bevor die Analyse der gesamten Protokollarchitektur im nächsten Abschnitt durchgeführt wird, wird in diesem Abschnitt das Verhalten der höheren Schichten 5-7 separat dargestellt, indem die darunterliegenden Schichten kurzgeschlossen werden. Die Analyse wird mit dem Verfahren nach Bild 4.5 durchgeführt. Die Ergebnisse sind in Bild 6.5 dargestellt und mit Simulationen validiert.

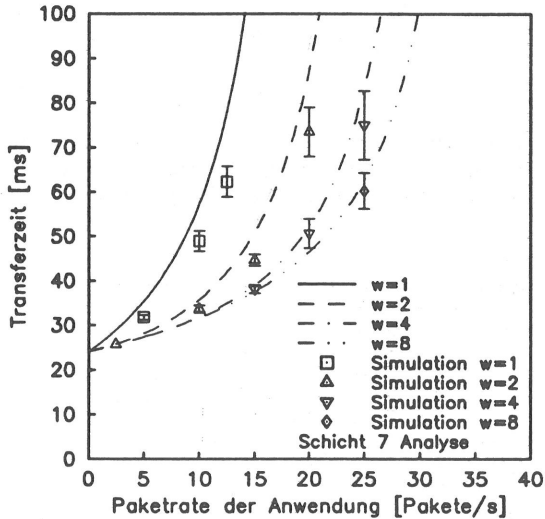


Bild 6.5: 'Schicht 7'-Transferzeit bei kurzgeschlossener Schicht 4

### 6.3.6 Analyse der gesamten Protokollarchitektur

Die Analyse der gesamten Protokollarchitektur wird in drei Schritten durchgeführt:

1. Ersetzung des Kanalzugriffsverfahrens (CSMA/CD) durch einen flußäquivalenten Ersatzknoten. Dazu müssen die Anwendungs-Paketraten an die Schicht 7 in die Kanal-Rahmenraten umgerechnet werden. Jedes Anwendungs-Paket hat in diesem Beispiel vier Rahmen auf dem Kanal zur Folge. Auf dem Kanal muß die gesamte Konfiguration mit allen angeschlossenen Stationen betrachtet werden.
2. Es werden die zwei interessierenden Stationen betrachtet und die Voll-Duplex-Verbindung auf der Schicht 4 zu zwei flußäquivalenten Ersatzknoten aggregiert. Das Kanal-Angebot anderer Stationen wird als offene Kette mit einem Summenangebot im flußäquivalenten Ersatzknoten des Kanals mit berücksichtigt.
3. Die oberen Schichten werden mit den flußäquivalenten Ersatzknoten aus Schritt 2 analysiert.



In Bild 6.6 ist die Transferzeit sowohl für die Schicht 4, wie auch die Schicht 7, für unterschiedliche Fenstergrößen der Schicht 7 dargestellt. Die Fenstergröße der Schicht 4 wurde einheitlich zu  $w_4 = 2$  gewählt.

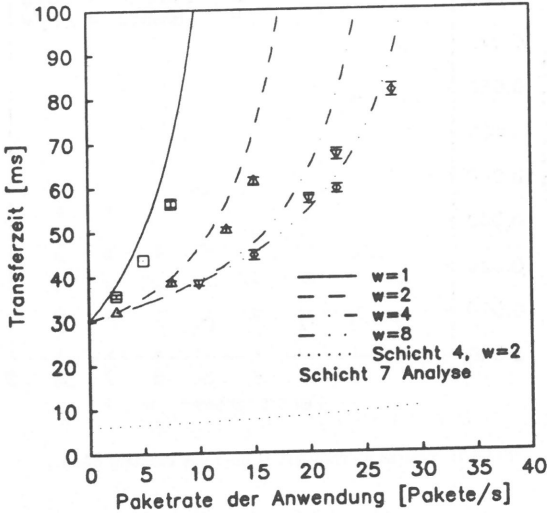


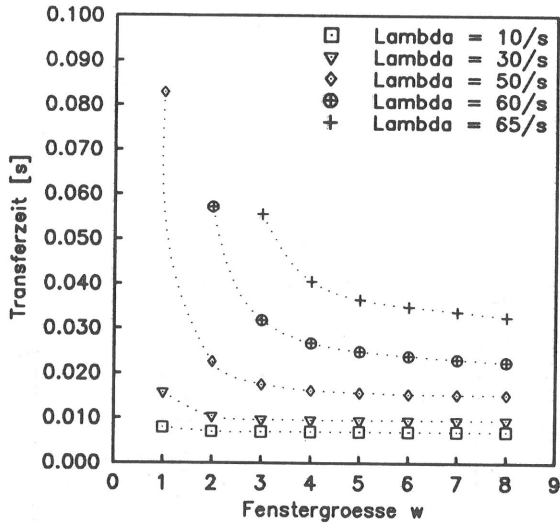
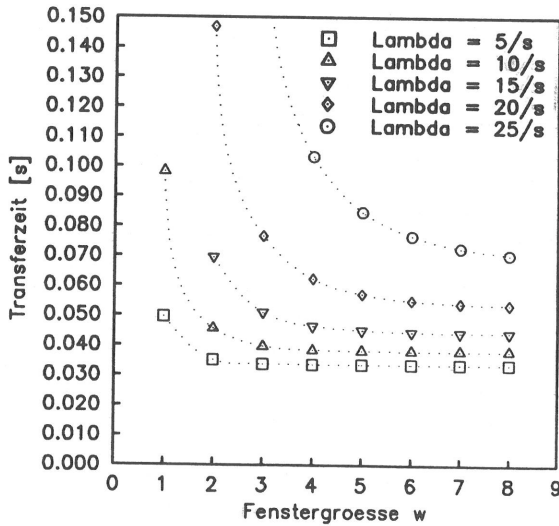
Bild 6.6: Transferzeit über die gesamte Protokollarchitektur

### 6.3.7 Optimierung der Fenstergröße

In den Bildern 6.7 und 6.8 sind die Transferzeiten über die Schicht 4 und die Schicht 7 für unterschiedliche Paketraten der Anwendung in Abhängigkeit von der Fenstergröße dargestellt. Bedingt durch den nur etwa halb so großen maximalen Durchsatz von Anwendungspaketen des Hauptprozessors gegenüber dem LAN-Prozessor genügt auf der Schicht 4 bereits ein Fenster von  $w_4 = 1$  oder höchstens  $w_4 = 2$ , um nicht eine zu große Verzögerung zu verursachen.

Die Zielfunktion  $P = t_{i_s} t_{i_f}$  tendiert für dieses Beispiel dazu, das Fenster so groß wie möglich zu wählen. Es wird daher die modifizierte Zielfunktion  $Q = t_{i_f} \sqrt{w}$  verwendet, die minimiert wird. Der Term  $t_{i_f}$  drückt die Interessen der Anwendung nach einer kurzen Transferzeit aus und der Term  $\sqrt{w}$  drückt das Interesse des Netzwerkes aus, eine möglichst kleine Anzahl von Paketen für die Übertragung im Netzwerk zuzulassen. Wird der Term  $w$  anstatt  $\sqrt{w}$  verwendet, so ergibt sich ein Übergewicht der Interessen des Netzwerkes.

In den Tabellen 6.3 und 6.4 ist die Zielfunktion  $Q$  und die optimale Fenstergröße  $w_7$ , nach dieser Zielfunktion, für unterschiedliche Paketraten der Anwendung dargestellt. In Tabelle 6.3 wurde die Fenstergröße der Schicht 4 zu  $w_4 = 1$  vorausgesetzt. In Tabelle 6.4 wurde  $w_4 = 2$  vorausgesetzt. Man erkennt, daß die Zielfunktion für  $w_4 = 2$  lediglich leicht verbessert werden kann, vor allem bei hohem Angebot. Die optimale Fenstergröße auf der Schicht 7 wird fast identisch gewählt. Lediglich für den Fall  $\lambda = 10$  Anwendungs-Pakete/s

Bild 6.7: 'Schicht 4'-Transferzeit über der Fenstergröße  $w_4$ Bild 6.8: 'Schicht 7'-Transferzeit über der Fenstergröße  $w_7$

$\lambda[1/s]$	$w_7 = 1$	$w_7 = 2$	$w_7 = 3$	$w_7 = 4$	$w_7 = 5$	$w_7 = 6$	$w_7 = 7$	$w_{opt}$
0.001000	0.030083	0.042541	0.052102	0.060162	0.067263	0.073683	0.079587	1
2.500000	0.038067	0.045311	0.054951	0.063414	0.070897	0.077664	0.083886	1
5.000000	0.049491	0.049506	0.058400	0.067130	0.075011	0.082166	0.088746	1
10.000000	0.098213	0.064457	0.068662	0.076862	0.085297	0.093267	0.100695	2
12.500000	0.166896	0.077621	0.076723	0.083700	0.092048	0.100329	0.108198	3
15.000000	0.447250	0.098265	0.088208	0.092820	0.100578	0.108956	0.117204	3
17.500000	8.326621	0.133919	0.105342	0.105558	0.111875	0.119929	0.128367	3
20.000000	-	0.207706	0.132684	0.124336	0.127664	0.134622	0.142831	4
22.500000	-	0.441922	0.181415	0.154076	0.151203	0.155555	0.162698	5
25.000000	-	5.566768	0.288362	0.206596	0.189435	0.187829	0.192113	6
27.500000	-	7.862464	0.688908	0.319464	0.260200	0.243335	0.240279	7

Tabelle 6.3: Optimierungsfunktion  $Q$  und optimaler Fensterwert  $w_7$  für  $w_4 = 1$ 

$\lambda[1/s]$	$w_7 = 1$	$w_7 = 2$	$w_7 = 3$	$w_7 = 4$	$w_7 = 5$	$w_7 = 6$	$w_7 = 7$	$w_{opt}$
0.001000	0.030083	0.042540	0.052100	0.060160	0.067261	0.073681	0.079584	1
2.500000	0.038067	0.045235	0.054868	0.063318	0.070789	0.077546	0.083759	1
5.000000	0.049491	0.049324	0.058219	0.066932	0.074792	0.081926	0.088487	2
10.000000	0.098213	0.063880	0.068212	0.076426	0.084836	0.092767	0.100155	2
12.500000	0.166896	0.076639	0.076047	0.083106	0.091444	0.099687	0.107510	3
15.000000	0.447249	0.096530	0.087186	0.092008	0.099802	0.108152	0.116355	3
17.500000	8.326620	0.130535	0.103727	0.104424	0.110873	0.118937	0.127340	3
20.000000	-	0.199543	0.129940	0.122670	0.126329	0.133380	0.141587	4
22.500000	-	0.407090	0.176103	0.151416	0.149322	0.153941	0.161163	5
25.000000	-	4.324879	0.274980	0.201728	0.186508	0.185583	0.190129	6
27.500000	-	7.830957	0.618062	0.307932	0.254802	0.239781	0.237459	7

Tabelle 6.4: Optimierungsfunktion  $Q$  und optimaler Fensterwert  $w_7$  für  $w_4 = 2$ 

ergibt sich ein Unterschied von  $w_7 = 2$  zu  $w_7 = 3$ . Ein niedrigerer Wert der Zielfunktion ist auf jeden Fall zu erwarten, da für größere Fensterwerte auf der Schicht 4 die Transferzeit der Schicht 4 immer niedriger ist bei konstanter Bedienrate der Prozessoren.

## 6.4 Unabhängige Sequenz von Verbindungen

### 6.4.1 Analyse

Das Modell einer unabhängigen Sequenz von Verbindungen ist in Bild 3.26 dargestellt. Für die Analyse werden die beiden Teile dekomponiert und bei gegebenem Angebot separat untersucht. Die gesamte Transferzeit ergibt sich als die Summe der einzelnen Transferzeiten (siehe Bild 6.9).

Die Netzwerke werden mit den Analysemethoden, die in Kapitel 5 dargestellt wurden, zu flußäquivalenten Ersatzknoten aggregiert. Anhand des Beispielmodells aus Bild 6.10 wird die Transferzeit in Bild 6.11 für symmetrische und unsymmetrische Fensterwerte ( $w_1, w_2$ ) dargestellt.

### 6.4.2 Optimierung der Fenstergröße

Nachdem die Leistungsanalyse durchgeführt werden kann, sollen in diesem Abschnitt die Fenstergrößen  $w_1$  und  $w_2$  optimiert werden. Die Optimierung wird für das Modell aus

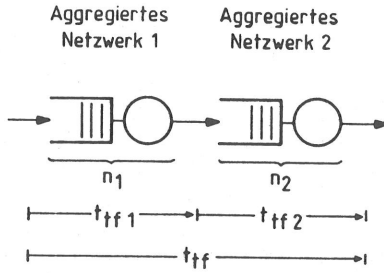


Bild 6.9: Aggregiertes Ersatzmodell für unabhängige Sequenz von Verbindungen

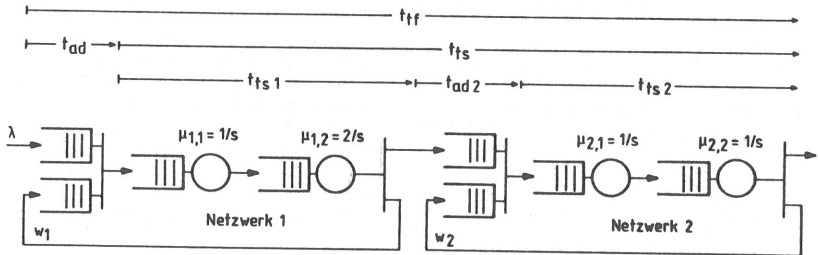


Bild 6.10: Beispielnetzwerk für unabhängige Sequenz von Verbindungen

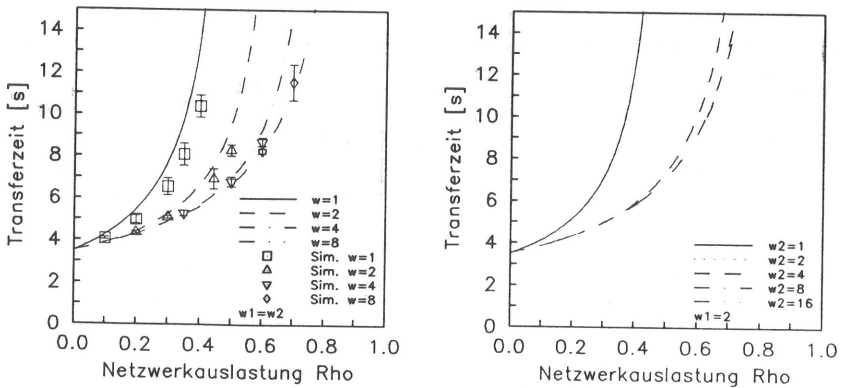


Bild 6.11: Transferzeit für unabhängige Sequenz von Verbindungen

	$w_2 = 1$	$w_2 = 2$	$w_2 = 3$	$w_2 = 4$	$w_2 = 5$	$w_2 = 6$	$w_2 = 7$	$w_2 = 8$
$w_1 = 1$	-	-	-	-	-	-	-	-
$w_1 = 2$	-	-	45.227	31.341	28.896	28.504	28.814	29.428
$w_1 = 3$	-	-	44.063	30.177	27.732	27.340	27.650	28.264
$w_1 = 4$	-	-	44.482	30.596	28.151	27.759	28.069	28.684
$w_1 = 5$	-	-	45.195	31.309	28.865	28.472	28.782	29.397
$w_1 = 6$	-	-	45.967	32.081	29.636	29.244	29.554	30.169
$w_1 = 7$	-	-	46.730	32.843	30.399	30.006	30.316	30.931
$w_1 = 8$	-	-	47.461	33.575	31.130	30.738	31.048	31.663

Tabelle 6.5: Optimierungsfunktion  $Q$  für  $\lambda = 0.7/s$ 

$\lambda [1/s]$	$w_1$	$w_2$
0.000001	1	1
0.050000	1	1
0.100000	1	1
0.150000	1	1
0.200000	1	1
0.250000	1	1
0.300000	1	2
0.350000	1	2
0.400000	2	2
0.450000	2	3
0.500000	2	3
0.550000	2	3
0.600000	2	4
0.650000	3	5
0.700000	3	6
0.750000	3	7
0.800000	4	10
0.850000	4	13
0.900000	5	16

Tabelle 6.6: Optimale Fensterwerte für unabhängige Sequenz von Verbindungen

Bild 6.10 durchgeführt.

Als Zielfunktion für die Optimierung wird  $Q = t_{t_1} \sqrt{w_1} + t_{t_2} \sqrt{w_2}$  verwendet. In Tabelle 6.5 ist  $Q$  für unterschiedliche Fenstergrößen und eine Ankunftsrate von  $\lambda = 0.7/s$  dargestellt. Es ergibt sich eine optimale Wahl der Fensterwerte mit  $w_1 = 3$  und  $w_2 = 6$ .

In Tabelle 6.6 sind die optimalen Fenstergrößen für unterschiedliche Ankunftsrate dargestellt.

## 6.5 Abhängige Sequenz von Verbindungen

### 6.5.1 Analyse

Die abhängige Sequenz von Verbindungen ist in Bild 3.27 als Modell dargestellt. Die Analyse muß die Interaktion der Flußregelungen berücksichtigen. Es wird dabei das

Ziel verfolgt, das zweite Netzwerk durch einen Ersatzknoten zu ersetzen. Es kann kein fluaquivalenter Ersatzknoten verwendet werden, da der Zustand von Netzwerk 2 fur die geschlossene Kette, die Netzwerk 1 durchlauft, nicht sichtbar ist. Fur die Berechnung des Ersatzknotens wird die mittlere Zugangsverzogerung zu Netzwerk 2 mit der mittleren Durchlaufzeit des Ersatzknotens gleichgesetzt bei einer hypothetischen Ankunftsrate von  $\lambda'$ . Die Zugangsverzogerung ergibt dann die  $M/M/1-\infty$  Bedienrate

$$\mu = \frac{1 + \lambda' t_{ad}}{t_{ad}}$$

Die hypothetische Ankunftsrate  $\lambda'$  ergibt sich aus der Uberlegung, da der Zustandsraum des fluaquivalenten Ersatzknotens von Netzwerk 2 auf  $0 \dots w_1 + w_2$  beschrankt ist. In dem Zustand  $w_1 + w_2$  werden weitere Anforderungen in der Zugangswarteschlange zu Netzwerk 1 blockiert. Um den mittleren Durchsatz  $\lambda$  zu erhalten, mu die Ankunftsrate in den restlichen Zustanden hoher sein als  $\lambda$ . Diese zustandsunabhangig angenommene Ankunftsrate  $\lambda'$  wird iterativ durch Losen des Ersatzknotens mit Ankunftsrate  $\lambda'$  und der Gleichung  $\lambda = \lambda'(1 - p_{w_1 + w_2})$  bestimmt.

Anschlieend kann das Netzwerk 1 mit dem zusatzlichen Ersatzknoten fur Netzwerk 2 gelost werden. Die Gesamt-Transferzeit  $t_{if}$  ergibt sich zu

$$t_{if} = t_{ad_1} + t_{ts_1} + t_{ts_2}$$

$t_{ts_1}$ , schliet die Zugangsverzogerung  $t_{ad_2}$  zu Netzwerk 2 mit ein.

Zusammenfassend lauft die Berechnung in drei Schritten ab:

1. Losung von Netzwerk 2 bei gegebener Ankunftsrate  $\lambda$ . Bestimmung der mittleren Zugangsverzogerung  $t_{ad_2}$  und der mittleren Transitzeit  $t_{ts_2}$ .
2. Netzwerk 2 durch einen Ersatzknoten ersetzen. Es kann aquivalent ein 'Infinite Server' oder eine  $M/M/1-\infty$  Warteschlange verwendet werden.
3. Losung von Netzwerk 1 mit Ersatzknoten und Bestimmung der Leistungsgroen.

In den Bildern 6.12 und 6.13 sind die Transferzeiten fur symmetrische und unsymmetrische Fensterwerte dargestellt. Es wurde das Beispielnetzwerk aus Bild 6.10 verwendet, wobei sich die Fluregelungen in der Mitte uberlappen.

Betrachtet man die vereinfachten Modelle mit keiner Verzogerung der Quittierungen, so konnen die Netzwerke zu jeweils einem fluaquivalenten Ersatzknoten aggregiert werden. Das Ersatzmodell kann dann uber den globalen Zustandsraum gelost werden, der lediglich zwei Dimensionen enthalt, von der eine zusatzlich auf  $w_1 + w_2$  beschrankt ist.

### 6.5.2 Optimierung der Fenstergroe

In Tabelle 6.7 sind wiederum die Zielfunktion  $Q^* = t_{if} \sqrt{w_1 + w_2}$  fur unterschiedliche Fenstergroen und eine Ankunftsrate von  $\lambda = 0.7/s$  dargestellt.

Vergleicht man die optimale Wahl der Fenstergroe in diesem Abschnitt mit der Wahl der Fenstergroe in Abschnitt 6.4.2, so ergeben sich fur  $w_2$  fast dieselben Werte, wahrend fur  $w_1$  bei hohem Angebot leicht groere Fensterwerte zu verwenden sind. Dies lat sich durch eine eventuelle zusatzliche Verzogerung der Quittierung erklaren, bevor Netzwerk 2 durchlaufen werden kann.

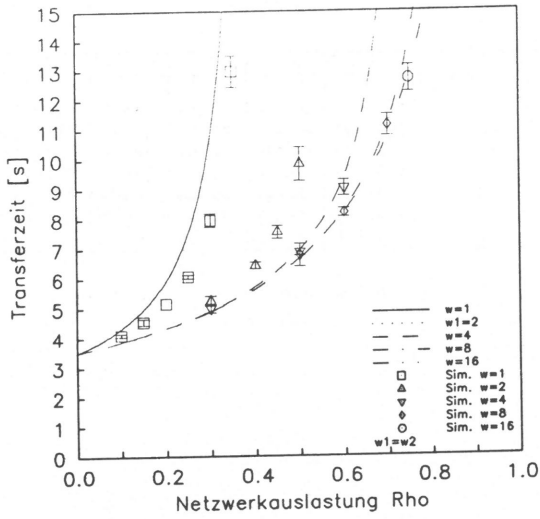


Bild 6.12: Transferzeit für abhängige Sequenz von Verbindungen und symmetrische Fensterwerte

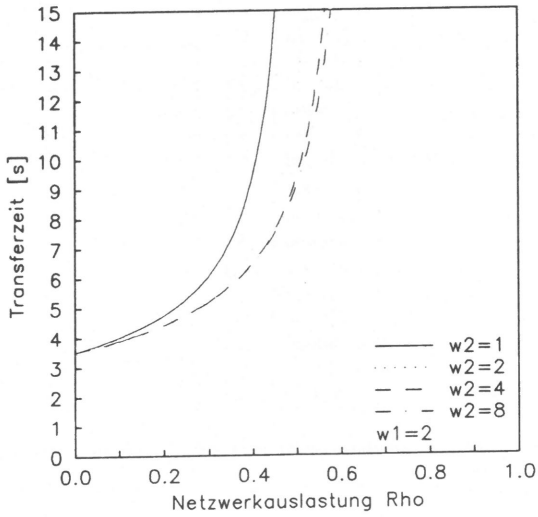


Bild 6.13: Transferzeit für abhängige Sequenz von Verbindungen und unsymmetrische Fensterwerte

	$w_2 = 1$	$w_2 = 2$	$w_2 = 3$	$w_2 = 4$	$w_2 = 5$	$w_2 = 6$	$w_2 = 7$	$w_2 = 8$
$w_1 = 1$	-	-	-	-	-	-	-	-
$w_1 = 2$	-	-	-	92.179	37.610	30.639	28.906	28.756
$w_1 = 3$	-	-	54.494	31.258	26.963	25.970	26.133	26.798
$w_1 = 4$	-	-	39.035	29.544	27.063	26.546	26.849	27.542
$w_1 = 5$	-	-	38.205	30.706	28.463	27.920	28.143	28.748
$w_1 = 6$	-	-	39.930	32.629	30.223	29.504	29.575	30.054
$w_1 = 7$	-	-	42.562	34.828	32.091	31.128	31.014	31.349
$w_1 = 8$	-	-	45.619	37.132	33.970	32.726	32.410	32.594

Tabelle 6.7: Optimierungsfunktion  $Q^*$  für  $\lambda = 0.7/s$ 

$\lambda[1/s]$	$w_1$	$w_2$
0.000001	1	1
0.050000	1	1
0.100000	1	1
0.150000	1	1
0.200000	1	1
0.250000	1	2
0.300000	2	2
0.350000	2	2
0.400000	2	2
0.450000	2	3
0.500000	2	3
0.550000	3	3
0.600000	3	4
0.650000	3	5
0.700000	4	5
0.750000	4	7
0.800000	5	9
0.850000	6	12
0.900000	8	16

Tabelle 6.8: Optimale Fensterwerte für abhängige Sequenz von Verbindungen



## Kapitel 7

# Zusammenfassung und Ausblick

In dieser Arbeit wurde die Leistungsuntersuchung des Protokollmechanismus 'Flußregelung' in Protokollen und mehrschichtigen Protokollarchitekturen durchgeführt.

Dazu wurde die Modellierung der Protokollmechanismen bei verbindungsloser und verbindungsorientierter Kommunikation entwickelt und dargestellt.

Für den Protokollmechanismus Flußregelung wurden Lösungsalgorithmen für Protokolle und mehrschichtige Protokollarchitekturen mit unterschiedlichen Randbedingungen entwickelt und durch Simulationen validiert. Die Optimierung der Fenstergröße wurde für verbindungsorientierte Kommunikation durchgeführt.

Die mehrschichtige Protokollarchitektur MAP, die in der Fertigungsautomatisierung eingesetzt wird, diente als Beispiel bei der Untersuchung.

Für die Flußregelung in einem verbindungsorientierten Protokoll wurden unter der Voraussetzung von Produktform-Netzwerken die zwei Lösungsalgorithmen

- lastäquivalentes Ersatznetzwerk und
- flußäquivalentes Ersatznetzwerk

dargestellt. Der Lösungsalgorithmus des flußäquivalenten Ersatznetzwerkes erfordert einen erhöhten Berechnungsaufwand, vor allem bei großen Fensterwerten. Die Validierungen mit Simulationen haben jedoch eine wesentlich bessere Genauigkeit ergeben als der Lösungsalgorithmus über das lastäquivalente Ersatznetzwerk.

Für allgemeine Netzwerke wurde ein Lösungsalgorithmus entwickelt, der bei Bedienzeitverteilungen, deren Variationskoeffizienten nicht zu weit von 1 entfernt sind, verwendet werden kann.

Die Lösungsalgorithmen für mehrschichtige Protokollarchitekturen aggregieren die unteren Schichten zu flußäquivalenten Ersatzknoten. Nach dieser Aggregation kann die Analyse der oberen Schichten durchgeführt werden. Die Genauigkeit der Analyse hängt stark von dem Einfluß der unteren Schichten und der Genauigkeit der Aggregation ab.

Für die Flußregelung bei verbindungsloser Kommunikation wurde die Ratenregelung über die Anzahl der Dateneinheiten und über die Arbeit der Dateneinheiten, die sie bei der Übertragung im Netzwerk verursachen, dargestellt und modelliert. Ferner wurde ein Lösungsalgorithmus entwickelt und an Beispielen mit Simulationen validiert.

Über die Optimierung können die optimalen Werte der freien Parameter für eine Konfiguration gefunden werden. Sowohl die Ankunftsprozesse der Dateneinheiten, wie auch das Netzwerk werden als gegeben vorausgesetzt.

Für den Betrieb von Kommunikationsnetzen müssen weiterhin Algorithmen entwickelt werden, wie diese freien Parameter statisch und dynamisch über das Netzwerk Management eingestellt werden. Die Validierung des Algorithmus kann anhand der Lösungsalgorithmen durchgeführt werden, die in dieser Arbeit vorgestellt wurden.

Durch den rasanten Fortschritt in der Halbleitertechnik, und damit auch der Kommunikationstechnik, werden immer schnellere und leistungsfähigere Kommunikationsmedien entwickelt und eingesetzt. Neue Anwendungen und neue Dienste sind die Folge. Im Augenblick zeichnet sich auf lokalen Netzen ein Geschwindigkeitsbereich von 100MBit/s ab. Für öffentliche Netze wird das neue Vermittlungsprinzip ATM ('Asynchronous Transfer Mode') diskutiert. Für die neuen Bedingungen sind die Fragen der Flußregelung, der Überlastabwehr und der Einbettung dieser Mechanismen in das Netzwerk Management neu zu diskutieren und zu lösen. Für die Datenkommunikation sind mögliche Ansätze sowohl in dem Einsatz dedizierter Hardware, die eine erhebliche Geschwindigkeitssteigerung in den oberen Schichten bringen könnte, als auch in abgemagerten Protokollarchitekturen zu sehen. Die Dekomponierbarkeit der Schichten des ISO-Referenzmodelles und einzelner Protokollmechanismen auf jeder Schicht kann bei der Implementierung in dedizierter Hardware ausgenützt werden, um Parallelität bei der Abarbeitung zu ermöglichen. Solche Spezialprozessoren können von der Bitübertragungsschicht (1) bis in die Darstellungsschicht (6) eingesetzt werden. Die Anwendungsschicht (7) ist aufgeteilt in die Teilschichten 7a und 7b. Während die Schicht 7a allgemeine Funktionen enthält, die ebenfalls in Hardware implementiert werden könnten, sind auf der Schicht 7b eine Vielzahl von Protokollen definiert. Die Komplexität dieser Protokolle ist vergleichbar mit jener auf der Transportschicht, meist jedoch noch größer. Der Einsatz von dedizierter Hardware wäre für jedes Protokoll einzeln nach den Gesichtspunkten technische Machbarkeit, Stabilität in der Standardisierung und Wirtschaftlichkeit zu prüfen.

# Literatur

- [1] Subhash C. Agrawal, Jeffery P. Buzen; The Aggregate Server Method for Analyzing Serialization Delays in Computer Systems; ACM Transactions on Computer Systems, Volume 1, Number 2, May 1983, Seiten 116-143
- [2] D.W. Andrews; Throughput Efficiency of Logical Links over Satellite Channels; Satellites and Computer Communication, (Editor J.L. Grange), Elsevier Science Publishers B.V. (North Holland), Amsterdam, 1983, Seiten 231-241
- [3] Eduard Arthurs, Gregory L. Chesson, Barton W. Stuck; Theoretical Performance Analysis of Sliding Window Flow Control; IEEE Journal on Selected Areas in Communications, Volume SAC-1, Number 5, November 1983, Seiten 947-959
- [4] Y. Bard; An Analytic Model of the VM/370 System; IBM Journal of Research and Development, Volume 22, Number 5, September 1978, Seiten 498-508
- [5] F. Baskett, K. Mani Chandy, R.R. Muntz, F. Palacios; Open, Closed and Mixed Networks of Queues with Different Classes of Customers; Journal of the ACM, Volume 22, Number 2, April 1975, Seiten 248-260
- [6] M. Bechtold, Guy Pujolle, Otto Spaniol; Throughput of a Satellite Channel Communication; Acta Informatica 22, Springer Verlag, Berlin-Heidelberg, 1985, Seiten 1-14
- [7] G. Bernard; Interconnection of Local Computer Networks: Modeling and Optimization Problems; IEEE Transactions on Software Engineering, Volume SE-9, July 1983, Seiten 463-470
- [8] Martin Bosch, Ottmar Gühr, Wikhard Kiesel; Modulare Simulationstechnik für komplexe Anwendungsprotokolle in der Fertigungsautomatisierung; Proceedings Prozes-srechnesysteme '88, (Herausgeber R. Lauber), Stuttgart, März 1988, Seiten 193-204
- [9] Martin Bosch, Ottmar Gühr, Wikhard Kiesel; Performance Comparison of two Application Layer Protocols for Manufacturing Automation; Proceedings of the 12th International Teletraffic Congress (ITC), Torino, June 1-8, 1988, Paper 3.4iB, Seiten 1-10
- [10] Werner Bux, Karl Kümmerle, Hong Linh Truong; Balanced HDLC Procedures: A Performance Analysis; IEEE Transactions on Communications, Volume COM-28, Number 11, November 1980, Seiten 1889-1898
- [11] Werner Bux, Parviz Kermani, Wolfgang Kleinöder; Performance of an Improved Data Link Control Protocol; Proceedings of the 9th International Conference on Computer Communication (ICCC), Tel Aviv, November 1988, Seiten 251-258
- [12] J.P. Buzen; Computational Algorithms for Closed Queueing Networks with Exponential Servers; Communications of the ACM, Volume 19, Number 9, September 1973, Seiten 527-531
- [13] K. Mani Chandy, Ulrich Herzog, L. Woo; Parametric Analysis of Queueing Networks;

- IBM Journal of Research and Development, Volume 19, Number 1, January 1975, Seiten 36-42
- [14] K. Mani Chandy, Ulrich Herzog, L. Woo; Approximate Analysis of General Queuing Networks; IBM Journal of Research and Development, Volume 19, Number 1, January 1975, Seiten 43-49
  - [15] K. Mani Chandy, J.H. Howard, D.F. Towsley; Product Form and Local Balance in Queueing Networks; Journal of the ACM, Volume 24, Number 2, April 1977, Seiten 250-263
  - [16] K. Mani Chandy, Charles H. Sauer; Computational Algorithms for Product Form Queueing Networks; Communications of the ACM, Volume 23, Number 10, October 1980, Seiten 573-583
  - [17] K. Mani Chandy, Doug Neuse; LINEARIZER: A Heuristic Algorithm for Queuing Network Models of Computer Systems; Communications of the ACM, Volume 25, Number 2, February 1982, Seiten 126-134
  - [18] D.M. Chitre; Analysis of Throughput Efficiency and Delay in ARQ Systems; COM-SAT Technical Review, Volume 11, Number 2, Fall 1981, Seiten 345-368
  - [19] A. Cobham; Priority Assignment in Waiting Line Problems; Journal of the Operations Research Society of America, Volume 2, Number 1, February 1954, Seiten 70-76
  - [20] J.W. Cohen; The Single Server Queue; North-Holland Publishing Company, 1969
  - [21] A.E. Conway, N.D. Georganas; RECAL - A New Efficient Algorithm for the Exact Analysis of Multiple-Chain Closed Queuing Networks; Journal of the ACM, Volume 33, Number 4, October 1986, Seiten 768-791
  - [22] P.J. Courtois; Decomposability; Academic Press, ISBN 0-12-193750-X, 1977
  - [23] D.R. Cox; A Use of Complex Probabilities in the Theory of Stochastic Processes; Proceedings of the Cambridge Philosophical Society, Volume 51, Part 2, April 1955, Seiten 313-319
  - [24] D.R. Cox, H.D. Miller; The Theory of Stochastic Processes; Chapman and Hall LTD, 1965
  - [25] Malcolm C. Easton; Batch Throughput Efficiency of ADCCP/HDLC/SDLC Selective Reject Protocol; IEEE Transactions on Communications, Volume COM-28, Number 2, 1980, Seiten 187-195
  - [26] M.A. El-Affendi, Demetres D. Kouvatso; A Maximum Entropy Analysis of the M/G/1 and G/M/1 Queueing Systems at Equilibrium; Acta Informatica, Volume 19, 1983, Seiten 339-355
  - [27] Serge Fdida, Andrzej Milk, Harry Perros; Semaphore Queues : Modelling Window Flow Control Mechanisms; Technical Report Number 183 of Laboratoire Methodologie & Architecture des Systemes Informatiques, Universite P. et M. Curie, 4 Place Jussieu 75252 Paris Cedex 05, FRANCE
  - [28] George S. Fishman; Concepts and Methods in Discrete Event Digital Simulation; John Wiley & Sons, ISBN 0-471-26155-6, 1973
  - [29] Darrel J. Freund, James N. Bexfield; A New Aggregation Approximation Procedure for Solving Closed Queueing Networks with Simultaneous Resource Possession; Proceedings of the ACM/SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Minneapolis, August 1983, Special Issue: Performance Evaluation Review, Seiten 214-223

- [30] R.G. Gallager, S.J. Golestani; Flow Control and Routing Algorithms for Data Networks; Proceedings International Conference on Computer Communication (ICCC), Atlanta, Georgia, October 1980, Seiten 779-784
- [31] E. Gelenbe, R.R. Muntz; Probabilistic Models of Computer Systems-Part II; Acta Informatica, Volume 12, 1976, Seiten 285-303
- [32] E. Gelenbe, I. Mitrani; Analysis and Synthesis of Computer Systems; Academic Press, ISBN 0-12-279350-1, 1980
- [33] Nicolas D. Georganas; Modeling and Analysis of Message Switched Computer-Communication Networks with Multilevel Flow Control; Computer Networks 4, 1980, Seiten 285-294
- [34] Mario Gerla, M. Staskauskas; Fairness in Flow Controlled Networks; Journal of Telecommunication Networks, Spring 82, Seiten 29-39
- [35] Mario Gerla, H.W. Chan; Window Selection in Flow Controlled Networks; Ninth Data Communications Symposium, Whistler Mountain, September 1985; ACM SIGCOM Computer Communication Review, Volume 15, Number 4, September 1985, Seiten 84-92
- [36] Ottmar Gühr, Paul J. Kuehn; Comparison of Communication Services with Connection-Oriented and Connectionless Data Transmission; International Seminar on Computer Networking and Performance Evaluation, Tokyo, September 1985, Seiten 173-186
- [37] Ottmar Gühr, Gert Willmann; Dynamische Datenstrukturen zur effizienten Simulation komplexer Verkehrsmodelle; Workshop Werkzeuge zur Modellierung und/oder Leistungsbewertung von Rechensystemen, (Herausgeber J. Mäter), Dortmund, Juni 1987
- [38] Ottmar Gühr, Michael Weixler; Messung der Datenverkehrsprofile in lokalen Netzen; Proceedings Kommunikation in verteilten Systemen, Stuttgart 22-24 Februar 1989, Seiten 861-877
- [39] Arthur Goldberg, Gerald Popek; A Validated Distributed System Performance Model; Performance '83, 1983, Seiten 251-268
- [40] Clemens M. Hauber; Analyseverfahren für flußkontrollierte Verbindungen; Diplomarbeit Nr. 833 am Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1987
- [41] ISO 3309, HDLC Frame Structure,  
ISO 4335, HDLC Elements of Procedure,  
ISO 6256; HDLC Balanced Classes of Procedure
- [42] Ulrich Herzog; Priority Models for Communication Processors Including System Overhead; Proceedings of the 8th International Teletraffic Congress (ITC), Melbourne, 1976, Volume 2, Session 62, Paper 3, Seiten 1-7
- [43] IEEE 802.2 : Local Area Networks - Logical Link Control  
ISO DIS 8802/2; Local Area Networks - Logical Link Control; 1985
- [44] IEEE 802.3 : Local Area Networks - CSMA/CD Access Method  
ISO DIS 8802/3; Local Area Networks - CSMA/CD Access Method; 1985
- [45] IEEE 802.4 : Local Area Networks - Token Passing Bus Access Method  
ISO DIS 8802/4; Local Area Networks - Token Passing Bus Access Method; 1985
- [46] ISO 7498 : Information Processing Systems - Open Systems Interconnection - Basic Reference Model; November 1983

- [47] ISO 8348 : Network Service Definition; 1984  
ISO DIS 8473 : Protocol for Providing the Connectionless-Mode Network Service; 1984
- [48] ISO 8072; Transport Service Definition; May 1984  
ISO 8073; Transport Protocol Specification; May 1984
- [49] ISO 8326; Basic Connection Oriented Session Service Definition, September 1984  
ISO 8327; Basic Connection Oriented Session Protocol Specification; September 1984
- [50] ISO DP 8822; Connection Oriented Presentation Service Definition May 1986  
ISO DP 8823; Connection Oriented Presentation Protocol Specification; May 1986
- [51] ISO DIS 8824; Specification of Abstract Syntax Notation One (ASN.1); 2nd DIS, September 1986  
ISO DIS 8825; Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1); 2nd DIS, September 1986
- [52] ISO DIS 9506; Manufacturing Message Specification, Part 1: Service Definition; December 1987  
ISO DIS 9506; Manufacturing Message Specification, Part 2: Protocol Specification; December 1987
- [53] ISO DP 8571; File Transfer, Access and Management; April 1985
- [54] ISO DIS 8649; Service Definition for Common Application Service Elements – Part 2 : Association Control; April 1986  
ISO DIS 8650; Protocol Specification for Common Application Service Elements – Part 2 : Association Control; April 1986
- [55] J.R. Jackson; Networks of Waiting Lines; Operations Research, Volume 5, Number 4, August 1957, Seiten 518-521
- [56] J.R. Jackson; Jobshop-Like Queueing Systems; Management Science, Volume 10, Number 1, October 1963, Seiten 131-142
- [57] Patricia A. Jacobson, Edward D. Lazowska; The Method of Surrogate Delays; Performance Evaluation Review, Volume 10, Number 3, Fall 1981, Seiten 165-174
- [58] P.A. Jacobson, E.D. Lazowska; Analyzing Queueing Networks with Simultaneous Resource Possession; Communications of the ACM, Volume 25, Number 2, February 1982, Seiten 142-151
- [59] P.A. Jacobson, E.D. Lazowska; A Reduction Technique for Evaluating Queueing Networks with Serialization Delays; 9th International Symposium on Computer Performance Modelling, Measurement and Evaluation, College Park, 1983; In 'Performance 83' (Editors A.K. Agrawala and S.K. Tripathi), North-Holland Publishing Company, Amsterdam, 1983, Seiten 45-59
- [60] Patricia A. Jacobson; Approximate Solution Techniques for Queueing Networks with Simultaneous Resource Possession; Technical Report 84-03-02, University of Washington, 1984
- [61] Jeffery M. Jaffe; A Decentralized, Optimal, Multiple User, Flow Control Algorithm; Proceedings International Conference on Computer Communication (ICCC), Atlanta, Georgia, October 1980, Seiten 839-844
- [62] Jeffery M. Jaffe; Bottleneck Flow Control; IEEE Transactions on Communications, Volume COM-29, Number 7, July 1981, Seiten 954-962
- [63] N.K. Jaiswal; Priority Queues; Academic Press, 1968

- [64] Edwin T. Jaynes; Prior Probabilities; IEEE Transactions on Systems Science and Cybernetics, Volume SCC-4, Number 3, September 1968, Seiten 227-241
- [65] Wikhard M. Kiesel; End-to-End Delay in Local Area Networks; Proceedings of 10th International Teletraffic Conference (ITC), Montreal, Canada, 1983, Paper 3.1-5, Seiten 1-7
- [66] Toshikazu Kimura; A Two-Moment Approximation for the Mean Waiting Time in the GI/G/s Queue; Management Science, Volume 32, Number 6, June 1986, Seiten 751-763
- [67] Hisashi Kobayashi; Application of the Diffusion Approximation to Queueing Networks, I. Equilibrium Queue Distributions; Journal of the Association for Computing Machinery, Volume 21, Number 2, 1974, Seiten 316-328
- [68] Hisashi Kobayashi; Modeling and Analysis; Addison-Wesley Publishing Company, 1978
- [69] A. Konheim; A Queueing Analysis of Two ARQ Protocols; IEEE Transactions on Communications, Volume COM-28, Number 7, 1981, Seiten 1004-1014
- [70] Dieter König, Dietrich Stoyan; Methoden der Bedienungstheorie; Wissenschaftliche Taschenbücher, Akademie Verlag, Berlin, 1976
- [71] L. Kleinrock; Queueing Systems, Volume 1: Theory; John Wiley & Sons Inc., New York, 1975, ISBN 0-471-49110-1; Volume 2: Computer Applications; John Wiley & Sons Inc., New York, 1976, ISBN 0-471-49111-X;
- [72] Demetres D. Kouvatsos; Maximum Entropy Methods for General Queueing Networks; Proceedings of Modelling Techniques and Tools for Performance Analysis, May 1984, (Editor D. Potier), North-Holland, 1985, Seiten 589-608
- [73] Demetres D. Kouvatsos, Panagiotis H.E. Geotgatsos, Nasreddine M. Tabet-Aouel; A Universal Maximum Entropy Algorithm for General Multiple Class Open Queueing Networks with Mixed Service Disciplines; Fourth International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Palma (Balearic Islands, Spain), September 14-16, 1988, Seiten 473-492
- [74] Wolfgang Kraemer, M. Langenbach-Belz; Approximate Formulae for the Delay in the Queueing System GI/G/1; Proceedings of the 8th International Teletraffic Congress (ITC), Melbourne, 1976
- [75] Wolfgang Kraemer; Processor Priority Model with Different User Tasks and Operation System Phases; In 'Computer Performance' (Editors K. Mani Chandy and Martin Reiser), North-Holland Publishing Company, Amsterdam, 1977, ISBN 0-444-85038-4, Seiten 305-325
- [76] P.S. Kritzinger, S. van Wyk, A.E. Krzesinski; A Generalisation of Norton's Theorem for Multiclass Queueing Networks; Performance Evaluation, Volume 2, Number 2, May 1982, Seiten 98-107
- [77] A. Krzesinski, P. Teunissen; Multiclass Queueing Networks with Population Constrained Subnetworks; Proceedings of the ACM/SIGMETRICS Conference on Measurement and Modeling of Computer Systems, 1985, und Performance Evaluation Review, Volume 13, Number 2, 1985, Seiten 128-139
- [78] Paul J. Kuehn; Analysis of Complex Queueing Networks by Decomposition; Proceedings of the 8th International Teletraffic Congress, Melbourne (ITC), 1976, Volume 1, Session 23, Paper 6, Seiten 1-8

- [79] J. Labetoulle, Guy Pujolle; HDLC Throughput and Response Time for Bidirectional Data Flow with Nonuniform Frame Sizes; IEEE Transactions on Computers, Volume C-30, Number 6, 1981, Seiten 405-413
- [80] S.S. Lam; Queuing Networks with Population Size Constraints; IBM Journal of Research and Development, Volume 21, Number 4, July 1977, Seiten 370-378
- [81] S.S. Lam, J.W. Wong; Queuing network models of packet switching networks, part 2: Networks with population size constraints; Performance Evaluation, Volume 2, Number 3, August 1982, Seiten 161-180
- [82] Edward D. Lazowska, John Zahorjan; Multiple Class Memory Constrained Queuing Networks; Performance Evaluation Review, ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Seattle Washington 1982, Volume 11, Number 4, Winter 1982-1983, Seiten 130-140
- [83] Edward D. Lazowska, John Zahorjan, G. Scott Graham, Kenneth C. Sevcik; Quantitative System Performance; Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1984
- [84] D.V. Lindley; The Theory of Queues with a Single Server; Proc. Cambridge Philosophical Society, 48, 1952, Seiten 277-289
- [85] Tsvi Lissack, Basil Maglaris, Hubert Chin; Impact of Microprocessor Architectures on Performance of Local Network Interface Adaptors; Proceedings of Local Networks and Distributed Office Systems, London, England, May 1981, Seiten 543-560
- [86] M. Maglaris, Tsvi Lissack; Performance Evaluation of Interface Units for Broadcast Local Area Networks; CompCon Proc., Fall 1982
- [87] Raymond A. Marie; An Approximate Analytical Method for General Queuing Networks; IEEE Transactions on Software Engineering, Volume SE-5, Number 5, September 1979, Seiten 530-538
- [88] K.T. Marshall; Some Inequalities in Queuing; Operations Research 16, 1968, Seiten 651-665
- [89] Bernd W. Meister, Philippe A. Janson, Liba Svobodova; Connection-Oriented Versus Connectionless Protocols: A Performance Study; IEEE Transactions on Computers, Volume C-34, Number 12, December 1985, Seiten 1164-1173
- [90] L.C. Mitchell, D.A. Lide; End-to-End Performance Modeling of Local Area Networks; IEEE Journal on Selected Areas in Communications, Volume SAC-4, Number 6, September 1986, Seiten 975-985
- [91] M. Murata, H. Takagi; Two-Layer Modeling for Local Area Networks; IEEE Transactions on Communications, Volume 36, Number 9, September 1988, Seiten 1022-1034
- [92] Doug Neuse, K. Mani Chandy; SCAT: A Heuristic Algorithm for Queuing Network Models of Computing Systems; 1981 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Las Vegas, 1981; ACM SIGMETRICS Performance Evaluation Review, Volume 10, Number 3, Fall 1981, Seiten 59-79
- [93] Zdzislaw Papir; Validation of Admission Delay Model for Two-Link Virtual Route with Window Flow Control; Performance Evaluation, Volume 7, 1987, Seiten 83-86
- [94] Martin Reiser, H. Kobayashi; Queuing Networks with Multiple Closed Chains: Theory and Computational Algorithms; IBM Journal of Research and Development, Volume 19, May 1975, Seiten 283-294



- [95] Martin Reiser, S.S. Lavenberg; Mean Value Analysis of Closed Multichain Queueing Networks; *Journal of the ACM*, Volume 27, Number 2, April 1980, Seiten 313-322
- [96] Martin Reiser; Admission delays on virtual routes with window flow control; International Conference on Performance of Data Communication Systems and their Applications, Paris, 1981; In 'Performance of Data Communication Systems and their Applications' (Editor Guy Pujolle), North-Holland Publishing Company, Amsterdam, 1981, ISBN 0-444-86283-8, Seiten 67-76
- [97] Z. Rosberg, N. Shacham; Re-Sequencing and Buffer Occupancy Under the Selective-Repeat ARQ; *Data Communication Systems and their Performance*, (Editors L.F.M. Moraes, E. de Souza e Silva and L.F.G. Soares), Elsevier Science Publishers B.V. (North Holland), IFIP 1988, Seiten 66-82
- [98] Charles H. Sauer; Approximate Solution of Queueing Networks with Simultaneous Resource Possessions; *IBM Journal of Research and Development*, Volume 25, Number 6, November 1981, Seiten 894-903
- [99] Charles H. Sauer, K. Mani Chandy; *Computer Systems Performance Modeling*; Prentice-Hall Inc., Englewood Cliffs, 1981
- [100] Mischa Schwartz; *Telecommunication Networks: Protocols, Modeling and Analysis*; Addison-Wesley Publishing Company, 1987
- [101] John E. Shore; Derivation of Equilibrium and Time-Dependent Solutions to  $M/M/\infty/N$  and  $M/M/\infty$  Queueing Systems Using Entropy Maximization; Proceedings National Computer Conference, AFIPS, 1978, Seiten 483-487
- [102] J.E. Shore, Rodney W. Johnson; Axiomatic Derivation of the Principle of Maximum Entropy and the Principle of Minimum Cross-Entropy; *IEEE Transactions on Information Theory*, Volume IT-26, January 1980, Seiten 26-37
- [103] Edmundo de Souza e Silva, Richard R. Muntz; Approximate Solutions for a Class of Non-Product Form Queueing Network Models; *Performance Evaluation*, Volume 7, Number 3, August 1987, Seiten 221-242
- [104] Johann Christoph Strelen; Wahrscheinlichkeitsverteilungen maximaler Entropie für die Wartezeiten in  $G/G/1$ -Systemen; 4. GI/ITG-Fachtagung über Messung, Modellierung und Bewertung von Rechensystemen, Erlangen, 1987; In 'Informatik-Fachberichte, Band 154: Messung, Modellierung und Bewertung von Rechensystemen' (Herausgeber U. Herzog und M. Paterok), Springer-Verlag, Berlin, 1987, ISBN 3-540-18406-6, Seite 343-359
- [105] 'Student': Gosset; The Probable Error of a Mean; *Biometrika*, Volume 6, Number 1, March 1908, Seiten 1-25
- [106] Hideaki Takagi; A Survey of Queueing Analysis of Polling Models; Proceedings of International Conference on Data Communication Systems and their Performance, Rio de Janeiro, 1987, Seiten 277-295
- [107] Alexander Thomasian; Queueing Network Models to Estimate Serialization Delays in Computer Systems; *Performance '83*, 1983, Seiten 61-81
- [108] A. Thomasian, P. Bay; Analysis of Queueing Network Models with Population Size Constraints and Delayed Blocked Customers; ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Cambridge, 1984; *ACM Performance Evaluation Review*, Volume 12, Number 3, August 1984, Seiten 202-216
- [109] Phouc Tran-Gia, Hamid Ahmadi, Parviz Kermani; Throughput Analysis of a Class of Selective Repeat Protocols in High-Speed Environments; *IBM Research Report*, 1988

- [110] Alan Mathison Turing; Computing Machinery and Intelligence; In: Anderson, Alan Ross; Minds and Machines; Prentice Hall, Englewood Cliffs/N.J., 1964
- [111] Hendrik Vantilborgh; Exact Aggregation in Exponential Queueing Networks; Journal of the Association for Computing Machinery, Volume 25, Number 4, October 1978, Seiten 620-629
- [112] G. Varghese, W. Chou, A.A. Nilson; Queueing Delays on Virtual Circuits Using a Sliding Window Flow Control Scheme; Proceedings of the 1983 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, ACM Performance Evaluation Review, August 1983, Seiten 275-281
- [113] J.W. Wong, M.S. Unsoy; Analysis of Flow Control in Switched Data Networks; Proceedings of IFIP Congress Information Processing 77, (Editor B. Gilchrist), 1977, Seiten 315-320
- [114] J.W. Wong, J.A.B. Moura, J.A. Field; Hierarchical Modelling of Local Area Computer Networks; Proceedings of National Telecommunication Conference, December 1980, Paper 37.1, Seiten 1-7
- [115] J.W. Wong, Jacques P. Sauve, James A. Field; A Study of Fairness in Packet-Switching Networks; IEEE Transactions on Communications, Volume COM-30, Number 2, February 1982, Seiten 346-353

## Appendix A

# Berechnung der Wartezeitverteilungsfunktion für das GE/GE/1-∞ Warteschlangensystem

Gegeben: Ankunftsabstandsverteilung und Bedienzeitverteilung des GE/GE/1-∞ Warteschlangensystems (FIFO-Abfertigungsstrategie).

$$a(t) = (1-r_a)\delta(t) + \lambda r_a^2 \exp(-\lambda r_a t)$$

$$A(t) = 1 - r_a \exp(-\lambda r_a t)$$

$$b(t) = (1-r_b)\delta(t) + \mu r_b^2 \exp(-\mu r_b t)$$

$$B(t) = 1 - r_b \exp(-\mu r_b t)$$

$$r_a = \frac{2}{1+c_a^2}$$

$$r_b = \frac{2}{1+c_b^2}$$

$$a(t) \circ \bullet \mathcal{A}^*(s) = (1-r_a) + \frac{\lambda r_a^2}{\lambda r_a + s} = \frac{s(1-r_a) + \lambda r_a}{\lambda r_a + s} = s\mathcal{A}(s)$$

$$A(t) \circ \bullet \mathcal{A}(s) = \frac{1}{s} - r_a \frac{1}{\lambda r_a + s} = \frac{s(1-r_a) + \lambda r_a}{s(\lambda r_a + s)} = \frac{\mathcal{A}^*(s)}{s}$$

$$b(t) \circ \bullet \mathcal{B}^*(s) = (1-r_b) + \frac{\mu r_b^2}{\mu r_b + s} = \frac{s(1-r_b) + \mu r_b}{\mu r_b + s} = s\mathcal{B}(s)$$

$$B(t) \circ \bullet \mathcal{B}(s) = \frac{1}{s} - r_b \frac{1}{\mu r_b + s} = \frac{s(1-r_b) + \mu r_b}{s(\mu r_b + s)} = \frac{\mathcal{B}^*(s)}{s}$$

Berechnung der Wartezeitverteilungsfunktion über die Lindley-Integralgleichung durch Faktorisierung im Spektralbereich (siehe [71]).

$$W(t) = \begin{cases} \int_{u=-0}^{\infty} C(t-u)dW(u) & ; t \geq 0 \\ 0 & ; t < 0 \end{cases}$$

$$C(t) = \int_{t=0}^{\infty} B(u+t)dA(t)$$

$$W_-(t) = \begin{cases} 0 & ; t \geq 0 \\ \int_{u=-\infty}^t W(t-u)dC(u) & ; t < 0 \end{cases}$$

Laplace Transformation:

$$W(t) \circ \bullet \Phi_+(s) = \int_{-\infty}^{\infty} W(t) \exp(-st) dt = \int_{0-}^{\infty} W(t) \exp(-st) dt$$

$$W_-(t) \circ \bullet \Phi_-(s) = \int_{-\infty}^{\infty} W_-(t) \exp(-st) dt = \int_{0-}^0 W_-(t) \exp(-st) dt$$

$$\Phi_-(s) = \Phi_+(s) [\mathcal{A}^*(-s)\mathcal{B}^*(s) - 1]$$

Faktorisierung des Spektrums:

$$\frac{\Psi_+(s)}{\Psi_-(s)} = \mathcal{A}^*(-s)\mathcal{B}^*(s) - 1$$

$\Psi_+(s)$  ist analytisch für  $\operatorname{Re}(s) > 0$

$\Psi_-(s)$  ist analytisch für  $\operatorname{Re}(s) < D$  ( $D$  beliebig)

$$\Phi_+(s) = \frac{K}{\Psi_+(s)} ; \Phi_-(s)\Psi_-(s) = \Phi_+(s)\Psi_+(s) = K$$

$$\begin{aligned} s\Phi_+(s) &= \mathcal{W}^*(s) & \bullet \circ w(t) \\ \Phi_+(s) &= \mathcal{W}(s) & \bullet \circ W(t) \end{aligned}$$

$$C^*(s) = \mathcal{A}^*(-s)\mathcal{B}^*(s) = \frac{-s(1-r_a) + \lambda r_a}{\lambda r_a - s} \frac{s(1-r_b) + \mu r_b}{\mu r_b + s}$$

Berechnung des Ausdrucks:  $\mathcal{A}^*(-s)\mathcal{B}^*(s) - 1$

$$\mathcal{A}^*(-s)\mathcal{B}^*(s) - 1 = C^*(s) - 1 = \frac{(\lambda r_a - s + s r_a)(\mu r_b + s - s r_b) - (\lambda r_a - s)(\mu r_b + s)}{(\lambda r_a - s)(\mu r_b + s)}$$

Bestimmung der Pole:  $(\lambda r_a - s)(\mu r_b + s) = 0$

$$\begin{aligned} s_{P1} &= -\mu r_b \\ s_{P2} &= \lambda r_a \end{aligned}$$

Bestimmung der Nullstellen :  $(\lambda r_a - s + s r_a)(\mu r_b + s - s r_b) - (\lambda r_a - s)(\mu r_b + s) = 0$

$$s [s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda)] = 0$$

$$s_{N1} = 0$$

$$s_{N2} = \frac{r_a r_b(\lambda - \mu)}{r_a + r_b - r_a r_b} < 0 \quad ; \quad 0 < r_a < 2 \text{ und } 0 < r_b < 2$$

Anmerkung:  $(c_a, c_b) \in [(0, 0), (\infty, \infty)]$  ist damit ausgeschlossen.

Faktorisierung:

$$\Psi_+(s) = \frac{s(s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda))}{s + \mu r_b}$$

$$\Psi_-(s) = \lambda r_a - s$$

$$K = \lim_{s \rightarrow 0} \frac{\Psi_+(s)}{s} = \lim_{s \rightarrow 0} \frac{s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda)}{s + \mu r_b} = r_a(1 - \rho)$$

$$K = r_a(1 - \rho) \quad ; \quad \rho = \lambda/\mu$$

$$\Phi_+(s) = \frac{K}{\Psi_+(s)} = \frac{r_a(1 - \rho)(s + \mu r_b)}{s [s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda)]}$$

$$\Phi_+(s) = \frac{r_a(1 - \rho)}{s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda)} + \frac{\mu r_a r_b(1 - \rho)}{s [s(r_a + r_b - r_a r_b) + r_a r_b(\mu - \lambda)]}$$

$$W(t) = \frac{r_a(1 - \rho)}{r_a + r_b - r_a r_b} \exp\left(-\frac{r_a r_b(\mu - \lambda)}{r_a + r_b - r_a r_b} t\right) + 1 - \exp\left(-\frac{r_a r_b(\mu - \lambda)}{r_a + r_b - r_a r_b} t\right)$$

$$W(t) = 1 - \frac{r_b + \rho r_a - r_a r_b}{r_a + r_b - r_a r_b} \exp\left(-\frac{r_a r_b(\mu - \lambda)}{r_a + r_b - r_a r_b} t\right) \quad ; \quad t \geq 0$$

$$w(t) = \left(1 - \frac{r_b + \rho r_a - r_a r_b}{r_a + r_b - r_a r_b}\right) \delta(t) + \frac{r_a r_b(\mu - \lambda)(r_b + \rho r_a - r_a r_b)}{(r_a + r_b - r_a r_b)^2} \exp\left(-\frac{r_a r_b(\mu - \lambda)}{r_a + r_b - r_a r_b} t\right)$$

$$w(t) = (1 - \rho) \frac{1 + c_b^2}{c_a^2 + c_b^2} \delta(t) + 2\mu(1 - \rho) \frac{c_a^2 - 1 + \rho(1 + c_b^2)}{(c_a^2 + c_b^2)^2} \exp\left(-\frac{2\mu(1 - \rho)}{c_a^2 + c_b^2} t\right)$$

Mittlere Wartezeit:

$$E[T_w] = \frac{1}{2\mu} \left[ \frac{c_a^2 + \rho c_b^2}{1 - \rho} - 1 \right]$$

Mittlere Anzahl der wartenden Aufträge:

$$\bar{n}_w = E[T_w]\lambda = \frac{\rho}{2} \left[ \frac{c_a^2 + \rho c_b^2}{1 - \rho} - 1 \right]$$

Mittlere Anzahl von Aufträgen im System:

$$\bar{n} = \bar{n}_w + \rho = \frac{\rho}{2} \left[ 1 + \frac{c_a^2 + \rho c_b^2}{1 - \rho} \right]$$

## Appendix B

# Analyse des begrenzt zustandsabhängigen GE/GE/1-∞ Warteschlangensystems

Das Warteschlangensystem GE/GE/1-∞ hat eine begrenzt zustandsabhängige Bedienzeitverteilungsfunktion  $B_n(t)$  bis zum Zustand  $x$ . Ab dem Zustand  $x$  ist die Bedienzeitverteilungsfunktion  $B_x(t)$ .

$$\begin{aligned} A(t) &= 1 - r_a \exp(-\lambda r_a t) \\ B_n^*(t) &= 1 - r_b(n) \exp(-\lambda r_b(n)t) \\ r_a &= \frac{2}{1 + c_a^2} \\ r_b(n) &= \frac{2}{1 + c_b^2(n)} \end{aligned}$$

$$B_n(t) = \begin{cases} B_n^*(t) & ; n \leq x \\ B_x^*(t) & ; n > x \end{cases}$$

Es wird davon ausgegangen, daß die Zustandswahrscheinlichkeiten ab  $p(x)$  geometrisch abnehmen, entsprechend der Zustandsverteilung beim M/GE/1-∞ Warteschlangensystem (siehe Demetres D. Kouvatso [72]), mit

$$p(n) = p(n-1) \frac{\bar{n}_x - \rho_x}{\bar{n}_x} ; n > x.$$

$$\rho_i = \lambda / \mu(i)$$

$$\bar{n}_i = \frac{\rho_i}{2} \left( 1 + \frac{c_a^2 + \rho_i c_{s,i}^2}{1 - \rho_i} \right)$$

Die Zustandswahrscheinlichkeiten  $p(1) \dots p(x)$  ergeben sich aus den Gleichungen:

$$p(1) = \frac{\rho_1^2}{\bar{n}_1(1 - \rho_1)} p(0)$$

und

$$p(i) = \frac{\rho_i^2}{\bar{n}_i(1 - \rho_i)} p(i - 1) \quad ; \quad 1 < i \leq x.$$

$p(0)$  ergibt sich aus der Normierungsbedingung  $\sum_{i=0}^{\infty} p(i) = 1$ .

Aus den Zustandswahrscheinlichkeiten  $p(i)$  lassen sich die interessierenden Leistungsgrößen berechnen.