

Institut für Nachrichtenvermittlung und Datenverarbeitung

Universität Stuttgart

Prof. Dr.-Ing., Dr.-Ing. E. h. A. Lotze

30. Bericht über verkehrstheoretische Arbeiten

Analyse zufallsabhängiger Prozesse in Systemen zur
Nachrichtenvermittlung und Nachrichtenverarbeitung

von

Paul Kühn

Institute of Switching and Data Technics

University of Stuttgart

Prof. Dr.-Ing., Dr.-Ing. E. h. A. Lotze

30th Report on Studies in Congestion Theory

Analysis of Stochastic Processes in
Communication Switching and Computer Systems

by

Paul Kühn

© 1982 Institut für Nachrichtenvermittlung und Datenverarbeitung Universität Stuttgart

Druck: Kurz & Co.

ISBN 3-922403-40-9

SUMMARY

The thesis is concerned with the analysis of stochastic processes in communication switching and data processing systems. The performance of such systems depends heavily on the properties of the traffic originated by a large number of independently acting customers as well as the number of system resources and the system organization. The studies aim at the modelling, performance evaluation, and traffic engineering of communications switching and computer systems.

1. Introduction

In the first chapter, some trends of actual development are traced showing the strong influence of computer and switching system technology on each other. The main subject of the traffic analysis and the methodology of performance evaluation are outlined: modelling, analysis, and system design.

2. Systems for Communications Switching and Data Processing

After referring to two fundamental aspects of system dynamics, functional and traffic-related consideration, systems for communications switching and data processing are described in more detail. Both subjects are treated in a systematic way by addressing the related aspects of system architecture, control principles, operating systems, functional procedures, and traffic engineering. For communications switching systems, an architecture with centralized stored programme control (SPC) serves as example. As a typical representative of modern computer architecture, a paged virtual memory system with automatic I/O-handling has been chosen. Both examples introduce the most important terms of reference and form the basis for later questions of modelling and performance evaluation.

3. Methods of Modelling

This chapter aims at the modelling technique. First, those items are identified which characterize the system dynamics as service requests, resources, system states, events, and transitions. The corresponding components of a traffic model are then developed systematically referring to system structure, system operation and traffic processes. Two extensive examples are given to demonstrate this methodology: 1. Control information transfer between periphery and central control in a computer-controlled communications switching system, 2. Data transfer between CPU and I/O-channel in a computer system with a two-level storage hierarchy. The chapter concludes with a systematic review of standardized traffic

models (single stage and network models).

4. Methods of Performance Analysis

Chapter 4 gives a systematic review on methods being applied to the analysis of stochastic processes: 1. Experimental methods (measurements, simulation), 2. Analytic methods (deterministic-combinatorial and probabilistic), and 3. Hybrid methods. In the remaining sections, probabilistic methods are described more specifically: Markovian service systems, non-Markovian service systems (phase methods, supplementary variable method, imbedded Markov chain, integral and moment methods), and various approximate techniques for system analysis.

5. Analysis of Queuing Networks

Queuing networks are of considerable interest for performance evaluation of computer systems, data communications networks, and switching system control. The recent advances of queuing network analysis and their results are briefly summarized both in case of exact state analysis (product-form solutions) as well as in case of approximate analysis for general queuing networks.

After a critical review of the known queuing network analysis methods the need of an approximate method for open queuing networks with general arrival and service processes is conjectured. Subsequently, a new method is developed which is based on the decomposition of a complex network into subsystems; the interaction between the various subsystems is described by means of input and output processes consistently with respect to the first and second moments. This analysis algorithm consists of a number of elementary steps as: traffic flows (first moments), average delay and coefficient of variation of the interdeparture time (output process), splitting of traffic flows, superposition of traffic flows, feedback stage replacement, traffic flows (second moments), and average job flow times. The chapter concludes with validation results on elementary steps and network examples and a discussion on possible extensions.

6. Selected Examples on System Analysis and Optimization

The final chapter addresses the application of various methods to performance evaluation. After a general review on principal models and their results, four specific problems are discussed in greater detail: 1. Throughput optimization in paged virtual memory computer systems with batch processing, 2. Throughput and response times in time-sharing computer systems with a two-level storage hierarchy, 3. Response times in interactive real-time systems, and 4. Analysis

of the control data flows in computer-controlled SPC switching systems. All these models have a comparatively complex system model in common; the analysis methods being applied range from exact multi-dimensional state variable methods to approximate decomposition methods. Most results are validated by simulation results.

In the first problem, the principal computer network model is complimented by programme behaviour and memory partition models. The analysis relates the various resource, programme, and traffic parameters to each other. From these studies, an optimum degree of multiprogramming can be identified numerically.

The second problem differs from the first one in the presence of user stations (terminals), where terminal-active programmes are swapped out from main memory intermediately. Modelling leads to a queuing network with a control switch for an adaptive admission of returning jobs from the terminals. This model is solved by bounding throughput and delay by two limit models of the product form network class. The results show that throughput maximization yields at the same time a delay minimization.

The third problem addresses an interactive computer system where typically many real-time requests have to be processed and stored. This model is analyzed by the new decomposition method of Chapter 5.

The fourth problem leads to a 3-stage hierarchical queuing network of mixed type with different service disciplines as cyclic service (scanning events), clocked I/O, and nonpreemptive scheduling. Models of such complexity can only be analyzed through decomposition. The decomposition algorithm of Chapter 5 is extended to allow the analysis of the corresponding scanning, clocked, and priority service disciplines. This analysis results to the control signal delays and an optimum time interval for the clocked I/O operations.

	Seite		
Inhaltsverzeichnis			
Schrifttumsverzeichnis			
1. EINFÜHRUNG	1	2.3.3 Betriebsarten	23
1.1 Entwicklungstrends	1	a) Simultane Ein/Ausgabe	23
1.2 Verkehrstheoretische Problemstellungen	2	b) Mehrprogrammbetrieb	23
1.3 Lösungsmethodik	3	c) Mehrprozessorbetrieb	25
1.4 Übersicht	4	d) Stapelverarbeitung	25
2. ABLAUFGESCHEHEN IN SYSTEMEN ZUR NACHRICHTENVERMITTLUNG UND NACHRICHTENVERARBEITUNG	6	e) Dialogbetrieb	26
2.1 Ablaufvorgänge und Betrachtungsweisen	6	2.3.4 Virtuelles Speicherprinzip	27
2.1.1 Funktionsbezogene Betrachtung	6	a) Aufgabenstellung und Voraussetzungen	27
2.1.2 Verkehrsbezogene Betrachtung	7	b) Abbildung des Virtuellen Speichers auf physikalische Speicher	28
2.2 Beispiel 1: Rechnergesteuerte Nachrichtenvermittlungssysteme	9	c) Das Paging-Verfahren	30
2.2.1 Architektur	10	d) Speicherverwaltungs-Strategien	31
a) Periphere Einrichtungen	11	2.3.5 Funktionsabläufe	34
b) Teilzentrale Steuer- und Informationsübertragungseinrichtungen	11	2.3.6 Verkehrsprobleme	36
c) Zentrale Steuereinrichtungen	11	a) Befehls- und Rechenwerke	38
2.2.2 Steuerungsprinzipien	11	b) Zentralprozessoren mit Schnellspeichern	38
a) Dezentrale Steuerung	12	c) Modulare Arbeitsspeicher	38
b) Zentrale Steuerung	12	d) Zentrale Leitungssysteme	38
c) Mischformen aus dezentraler und zentraler Steuerung	13	e) Speicherhierarchie	38
2.2.3 Betriebsarten	13	f) Teilnehmer-Rechensysteme, Rechnernetze	39
a) Teilhaberbetrieb	13	3. METHODEN DER MODELLEBILDUNG	40
b) Realzeitbetrieb	14	3.1 Zielsetzungen der Modellbildung	40
c) Mehrprozeßbetrieb	14	3.1.1 Modellbegriff	40
2.2.4 Funktionsabläufe	16	3.1.2 Kriterien zur Modellbildung	41
2.2.5 Verkehrsprobleme	16	3.2 Methodik der Modellbildung	41
a) Leitungsnetz	18	3.2.1 Charakterisierung des realen Ablaufgeschehens	42
b) Koppelnetz	18	a) Anforderungen und Betriebsmittel	42
c) Teilzentrale Steuereinrichtungen	18	b) Zustände, Ereignisse und Übergänge	42
d) Zentrale Steuereinrichtungen	18	c) Logische und zeitliche Zusammenhänge	43
2.3 Beispiel 2: Rechnersysteme mit hierarchischem Speicherprinzip	19	3.2.2 Modellkomponenten	43
2.3.1 Architektur	19	a) Strukturbezogene Komponenten	43
a) Zentraleinheit	19	b) Betriebsbezogene Komponenten	45
b) Peripherie	21	c) Verkehrsbezogene Komponenten	45
c) Konzeption der Speicherhierarchie	21	d) Beispiel eines einstufigen Bedienungssystem-Modelles	48
2.3.2 Steuerungsprinzipien	21	3.2.3 Abbildung	49
a) Verdrahtete (Hardware-) Steuerwerke	22	3.2.4 Modellbestätigung	50
b) Mikroprogrammierte Steuerwerke	22	3.3 Beispiele zur Modellbildung	50
c) Systemprogramme (Betriebssystem)	22	3.3.1 Beispiel 1: Informationsaustausch zwischen Peripherie und Zentralsteuerung in rechnergesteuerten Nachrichtenvermittlungen	51
		a) Charakterisierung des realen Ablaufgeschehens	51
		b) Modellkomponenten	53
		c) Modell	53
		3.3.2 Beispiel 2: Informationsaustausch zwischen Zentralprozessor und Kanalwerk in Rechnersystemen mit zweistufiger Speicherhierarchie	55
		a) Charakterisierung des realen Ablaufgeschehens	55
		b) Modellkomponenten	56
		c) Modell	57
		d) Vereinfachung des Modelles	58

3.4 Systematik verkehrstheoretischer Modelle	58	4.5.2 Methode der Gedächtnisvariablen	84
3.4.1 Klassifizierungsmerkmale und Kurznotation	58	4.5.3 Methode der Eingebetteten Markoff-Kette	85
3.4.2 Modelle mit einstufiger Bedienung	59	4.5.4 Lindleysche Integralmethode	87
a) Modelle mit einer Eingangs-Warteschlange ohne/mit Rückkopplung	60	4.5.5 Momentenmethoden	87
b) Modelle mit parallelen Eingangs-Warteschlangen und vollkommener Erreichbarkeit	60	4.6 Approximative Methoden	88
c) Modelle mit parallelen Eingangs-Warteschlangen und unvollkommener Erreichbarkeit (Mischungen)	61	4.6.1 Makrozustände	88
d) Modelle mit parallelen Eingangs-Warteschlangen und mehrstufigen Koppelanordnungen (Linkssysteme)	61	4.6.2 Kontinuierliche Zustände	89
e) Modelle mit Überlaufmöglichkeit	61	4.6.3 Unabhängigkeitsannahmen	89
3.4.3 Modelle mit mehrstufiger Bedienung (Netzmodelle)	61	4.6.4 Zerlegungsmethoden	89
a) Offene Warteschlangennetze	62	4.6.5 Ersetzungsmethoden	90
b) Geschlossene Warteschlangennetze	62	4.6.6 Momentenapproximationen	90
3.4.4 Allgemeine Bemerkungen	63	4.6.7 Grenzwerttheorien	91
4. METHODEN DER MODELLANALYSE	64	4.7 Zusammenfassung	91
4.1 Experimentelle Methoden	64	5. ANALYSE VON WARTESCHLANGENNETZEN	92
4.1.1 Messung	64	5.1 Exakte Ergebnisse für Warteschlangennetze	92
a) Hardware-Messungen	64	5.1.1 Warteschlangennetze mit Produktlösungsform und einer einzigen Klasse von Anforderungen	92
b) Software-Messungen	65	a) Voraussetzungen	
4.1.2 Simulation	66	b) Produktlösung	
a) Prinzip der zeittreuen Simulation	66	c) Eigenschaften	
b) Prinzip der Ruf-Löschzahlenmethode	67	5.1.2 Warteschlangennetze mit Produktlösungsform und mehreren Klassen von Anforderungen	96
c) Beispiel	68	a) Voraussetzungen	
4.2 Analytische Methoden	70	b) Produktlösung	
4.2.1 Deterministisch-kombinatorische Methoden	70	c) Eigenschaften	
a) Beispiel 1: Deterministische Prozessorzuteilung	70	5.1.3 Eine Verallgemeinerung der Warteschlangennetze mit Produktlösungsform	97
b) Beispiel 2: Blockierungsfreie mehrstufige Koppelanordnungen	71	a) Definition der allgemeinsten klassenunabhängigen Abfertigungsdisziplin	
4.2.2 Wahrscheinlichkeitstheoretische Methoden (Stochastische Prozesse)	72	b) Verallgemeinerung	
a) Stochastische Prozesse in Bedienungssystemen	72	5.1.4 Allgemeinere Voraussetzungen	100
b) Definition Stochastischer Prozesse	73	5.2 Übersicht über approximative Ergebnisse für Warteschlangennetze	100
c) Klassifizierung Stochastischer Prozesse	74	5.2.1 Diffusionsapproximationen	100
4.3 Hybride Methoden	76	5.2.2 Äquivalenzverfahren	101
4.4 Analyse Markoffscher Bedienungssysteme	76	5.2.3 Zerlegungsverfahren	101
4.4.1 Kolmogoroffsche Gleichungen	76	5.3 Schlußfolgerungen und Begründung für ein neues Zerlegungsverfahren	102
4.4.2 Systemzustandsprozeß	77	5.4 Ein neues Verfahren zur Analyse von Warteschlangennetzen nach der Zerlegungsmethode	103
4.4.3 Warteprozeß	80	5.4.1 Voraussetzungen	103
4.5 Analyse Nichtmarkoffscher Bedienungssysteme	81	a) Netzstruktur	
4.5.1 Phasenmethode	81	b) Ankunfts- und Bedienungsprozesse	
a) Darstellung allgemeiner VF durch Phasenmodelle	81	c) Betriebsweise	
b) Erlang-k-VF (E_k)	82	5.4.2 Lösungsübersicht	105
c) Hyperexponentielle VF k-ter Ordnung (H_k)	83	5.4.3 Lösungsschritte des Verfahrens	106
d) Beispiel: Wartesystem $M/E_k/1$	83	a) Bestimmung der Verkehrsflüsse (1. Momente)	
		b) Wartezeit und Ausgangsprozeß des Wartesystems GI/G/1	
		c) Verzweigung von Verkehrsflüssen	

d) Überlagerung von Verkehrsflüssen		6.3.3 Modellanalyse	154
e) Ersetzung von Rückkopplungen		a) Zwei analytische Grenzmodelle	
f) Bestimmung der Verkehrsflüsse (2. Momente)		b) Zustandswahrscheinlichkeiten und charakteristische Größen	
g) Bestimmung der Durchlaufzeiten		6.3.4 Ergebnisse	158
5.4.4 Algorithmus	118	6.4 Antwortzeiten in interaktiven Systemen	160
5.4.5 Genauigkeitsnachweise	118	6.4.1 Problemstellung und Voraussetzungen	160
a) Elementaroperationen		6.4.2 Systemmodell	161
b) Gesamtergebnisse		6.4.3 Modellanalyse	161
5.4.6 Kritische Diskussion und Verallgemeinerungen	124	6.4.4 Ergebnisse	162
a) Diskussion der Ergebnisse		6.5 Analyse des Steuerdatenflusses in rechnergesteuerten Nachrichtenvermittlungssystemen	164
b) Verallgemeinerungen		6.5.1 Problemstellung	164
6. AUSGEWÄHLTE BEISPIELE ZUR SYSTEMANALYSE UND OPTIMIERUNG	127	6.5.2 Voraussetzungen	165
6.1 Übersicht über Modelle und Ergebnisse	127	a) Struktur der Steuerung	
6.1.1 Prozessmodelle	127	b) Betriebsarten	
6.1.2 Speicherzugriffmodelle	128	c) Steuerinformations-Anreize	
6.1.3 Gesamtmodelle	128	6.5.3 Systemmodell	166
6.1.4 Standardisierte Modelle, Tabellen	129	a) Gesamtmodell	
6.2 Durchsatzoptimierung in Stapelverarbeitungssystemen mit Virtuellem Speicher	130	b) Teilmodell Signalsammelsystem	
6.2.1 Problemstellung	130	c) Teilmodell Signalpuffer- und Ein/Ausgabesystem	
6.2.2 Voraussetzungen	131	d) Teilmodell Zentralsteuerung	
a) Rechnerstruktur		6.5.4 Modellanalyse	169
b) Betriebsstrategien		a) Methodik und Übersicht	
c) Programmeigenschaften (Auftragsprofil)		b) Teilmodell Signalsammelsystem	
6.2.3 Systemmodell	135	c) Teilmodell Signalpuffer- und Ein/Ausgabesystem	
a) Warteschlangenmodell		d) Teilmodell Zentralsteuerung	
b) Speichermodell		6.5.5 Ergebnisse und Simulationsvergleiche	181
c) Benutzerprogramm-Modell		a) Steuerdatenflüsse und Durchlaufzeiten	
d) Festlegung der Parameter des Systemmodells		b) Optimierung und Überlaststeuerung	
6.2.4 Modellanalyse	141	c) Simulationsvergleiche	
a) Zustandsdefinition und Zustandsraum		7. ZUSAMMENFASSUNG	192
b) Zustandsgleichungen und Zustandswahrscheinlichkeiten		Anerkennung	
c) Charakteristische Größen			
6.2.5 Ergebnisse und Simulationsvergleiche	144		
a) Rechenergebnisse			
b) Simulationsvergleiche			
6.3 Durchsatz und Antwortzeiten in Teilnehmerrechensystemen mit zweistufiger Speicherhierarchie	150		
6.3.1 Problemstellung und Voraussetzungen	150		
a) Problemstellung			
b) Voraussetzungen			
6.3.2 Systemmodell	151		
a) Warteschlangenmodell			
b) Speichermodell			
c) Benutzerprogramm-Modell			
d) Festlegung der Parameter des Systemmodells			

SCHRIFTTUMSVERZEICHNIS

- [1] Lotze, A.: Vorlesungen "Nachrichtenvermittlung" an der Universität Stuttgart.
- [2] Gerke, P.R.: Rechnergesteuerte Vermittlungssysteme. Springer-Verlag, Berlin/Heidelberg/New York (1972).
- [3] Special Issues BSTJ on No. 1 ESS, BSTJ 43 (1964); No. 4 ESS, BSTJ 56 (1977); Common Channel Interoffice Signalling (CCIS), BSTJ 57 (1978).
- [4] Schmidt, G.: Vermittlungstechnik. R. Oldenbourg-Verlag, München/Wien (1966).
- [5] Botsch, D.: Characteristics of the Processes Employed in Telephone Switching Systems. Telefon Report 12 (1976), S. 160-165.
- [6] Bocker, P., Gaiser, R.: Der Aufbau von Datennetzen mit dem elektronischen Datenvermittlungssystem EDS. NTZ 26 (1973), S. 297-304.
- [7] Davies, D.W., Barber, D.L.A.: Communications Networks for Computers. John-Wiley and Sons, Inc., London/New York (1973).
- [8] Swoboda, J.: Vorlesungen "Architektur von Rechenanlagen" an der Universität Stuttgart.
- [9] Bell, G.C., Newell, A.: Computer Structures: Readings and Examples. McGraw-Hill Book Company, New York (1971).
- [10] Hellerman, H., Conroy, T.F.: Computer System Performance. McGraw-Hill Book Company, New York (1975).
- [11] (Autorenteam): Der Virtuelle Speicher. IBM Deutschland (1972).
- [12] Buzen, J.P.: I/O Subsystem Architecture. Proceedings of the IEEE, Vol. 63 (1975), S. 871-879.
- [13] Auslander, M.A., Jaffe, J.F.: Functional Structure of IBM Virtual Storage Operating Systems. Part I: Influences of Dynamic Address Translation on Operating System Technology. IBM Syst.J. 12 (1973), S. 368-381.
- [14] Swoboda, J.: Was ist Pipelining? El. Rechenanlagen 17 (1975), S. 80-83.
- [15] Graham, R.M.: Principles of Systems Programming. John Wiley and Sons, Inc., New York/London/Sydney/Toronto (1975).
- [16] Denning, P.J.: Virtual Memory. Computing Surveys 2 (1970), S. 153-189.
- [17] Aho, A.V., Denning, P.J., Ullman, J.D.: Principles of Optimal Page Replacement. J. ACM 18 (1971), S. 80-93.
- [18] Herzog, U., Krämer, W., Kühn, P., Witzgall, M.: Analyse von Betriebssystem-Modellen für Rechnersysteme mit Multiprogramming und Paging. Lecture Notes in Computer Science, No. 8. Springer-Verlag, Berlin/Heidelberg/New York (1974), S. 266-288.
- [19] Shedler, G.S., Yang, S.C.: Simulation of a Model of Paging Systems Performance. IBM Syst.J. 10 (1971), S. 113-128.
- [20] Herzog, U.: Verkehrsfluß in Datennetzen. Habilitationsschrift, Universität Stuttgart (1973).
- [21] Queuing Standardization Conference Report. May 11 (1971).
- [22] International Teletraffic Congresses (ITC). 1. ITC bis 8. ITC (1955-1976).
- [23] Int. Conferences on Computer System Modeling. IFIP Working Group 7.3 (1976,1977).
- [24] Conferences of the International Federation of Information Processing (IFIP).
- [25] International Conferences on Communications (ICC). IEEE (jährlich).
- [26] International Conferences on Computer Communications (ICCC, 1972-1978).
- [27] IRIA-IFIP Symposia on Modeling and Performance Evaluation of Computer Systems.
- [28] European Computing Symposia (ACM).
- [29] NTG-Fachtagungen über Struktur, Betrieb und Analyse von Rechensystemen (1967, 1970, 1972, 1974, 1978).
- [30] GI-Jahrestagungen und GI-Workshops über Modelle für Rechnersysteme (1970-1977).
- [31] Berichte über verkehrstheoretische Arbeiten des Instituts für Nachrichtenvermittlung und Datenverarbeitung der Universität Stuttgart.
- [32] Drummond, M.E.: Evaluation and Measurement Techniques for Digital Computer Systems. Prentice-Hall, Inc., Englewood Cliffs, N.J. (1973).
- [33] Stanley, W.I.: Measurement of System Operational Statistics. IBM Syst.J. 8 (1969), S. 299-308.
- [34] Wagner, H., Dietrich, G.: Bestimmung der Verkehrsleistung von Wartesystemen durch künstlichen Fernspreverkehr. NTZ 17 (1964), S. 273-279.
- [35] Huber, M., Wagner, W.: Simulation von Nachrichtenvermittlungssystemen. In: Nichtnumerische Informationsverarbeitung, Hsg. R. Gunzenhäuser, Springer-Verlag, Wien/New York (1968).
- [36] Kampe, G., Kühn, P., Langenbach-Belz, M.: Simulation in der Nachrichtenverkehrstheorie: Problemstellungen und Programmiersprachen. GI-Workshop über Simulationstechniken, Kernforschungszentrum Karlsruhe, 1973. KFK-Ber. 1845 (1973), S. 240-263.
- [37] Kampe, G.: Simscript. Vieweg-Verlag, Braunschweig (1971).
- [38] Kümmerle, K.: Simulation of the Performance of Computer Systems. Elektron. Rechenanl. 12 (1970), S. 324-328.
- [39] Conway, R.W., Maxwell, W.L., Miller, L.W.: Theory of Scheduling. Addison-Wesley, Reading, Mass. (1967).
- [40] Coffman, E.G., Denning, P.J.: Operating Systems Theory. Prentice-Hall, Inc., Englewood Cliffs, N.J. (1973).
- [41] Beneš, V.: Mathematical Theory of Connecting Networks and Telephone Traffic. Academic Press, New York (1965).
- [42] Clos, C.: A Study of Non-Blocking Switching Networks. Bell System Tech.J. 32 (1953), S. 406-423.
- [43] Kleinrock, L.: Queuing Systems. Vol. I: Theory, and Vol. II: Computer Applications. John Wiley & Sons, New York/London/Sydney/Toronto (1975, 1976).

- [44] Syski, R.: Markovian Queues. Proc. of the Symposium on Congestion Theory, University of North Carolina, 1964. The University of North Carolina Press (Ed. W.L. Smith, W.E. Wilkinson), Chapel Hill (1964), S. 170-227.
- [45] Kühn, P.: Über die Berechnung der Wartezeiten in Vermittlungs- und Rechnersystemen. Dissertationsschrift, Universität Stuttgart (1972).
- [46] Cox, D.R.: A Use of Complex Probabilities in the Theory of Stochastic Processes. Proc. Cambr. Phil. Soc. 51 (1955), S. 313-319.
- [47] Bux, W., Herzog, U.: The Phase Concept: Approximation of Measured Data and Performance Analysis. Proceedings of the Int. Symp. on Computer Performance Modeling, Measurement, and Evaluation (Ed. K.M. Chandy, M. Reiser). North-Holland Publ. Comp., Amsterdam/New York/Oxford (1977), S. 23-38.
- [48] Störmer, H., et.al.: Verkehrstheorie. R. Oldenbourg-Verlag, München/Wien (1966).
- [49] Cox, D.R.: The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables. Proc. Cambr. Phil. Soc. 51 (1955), S. 433-441.
- [50] Riordan, J.: Stochastic Service Systems. John Wiley and Sons, Inc., New York/London (1962).
- [51] Kendall, D.G.: Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of Imbedded Markov Chains. Ann. Math. Statistics 24 (1953), S. 338-354.
- [52] Cooper, R.B.: Introduction to Queuing Theory. The MacMillan Comp., New York/London (1972).
- [53] Cobham, A.: Priority Assignment in Waiting Line Problems. Opns. Res. 2 (1954), S. 70-76.
- [54] Herzog, U.: Priority Models for Communication Processors Including System Overhead. 8. ITC, Melbourne (1976). Congressbook, S. 623/1-7.
- [55] Syski, R.: Introduction to Congestion Theory in Telephone Systems. Oliver and Boyd, Edinburgh/London (1960).
- [56] Cohen, J.W.: The Single Server Queue. North-Holland, Amsterdam (1969).
- [57] Gross, D., Harris, C.M.: Fundamentals of Queueing Theory. John Wiley and Sons, New York/London/Sydney/Toronto (1974).
- [58] Kosten, L.: Stochastic Theory of Service Systems. International Series of Monogr. in Pure and Appl. Math., Vol. 103. Pergamon Press, Oxford/New York/Toronto/Sydney/Braunschweig (1973).
- [59] Schehrer, R.: Über die exakte Berechnung von Überlaufsystemen der Wählvermittlungstechnik. Dissertationsschrift, Universität Stuttgart (1969).
- [60] Lörcher, W.: Exakte Berechnung der Zustandswahrscheinlichkeiten und der charakteristischen Verkehrsgrößen mehrstufiger Koppelanordnungen mit konjugierter Durchschaltung. Dissertationsschrift, Universität Stuttgart (1973).
- [61] Herzog, U.: Die exakte Berechnung des Streuwertes von Überlaufverkehr hinter Koppelanordnungen beliebiger Stufenzahl mit vollkommener bzw. unvollkommener Erreichbarkeit. AEÜ 20 (1966), S. 180-184.
- [62] Kühn, P.: Waiting Time Distribution in Multi-Queue Delay Systems with Gradings. 7. ITC, Stockholm (1973). Congressbook S. 242/1-9 und AEÜ 29 (1975), S. 53-61.
- [63] Thierer, M.: Wartesysteme mit unvollkommener Erreichbarkeit. Dissertationsschrift, Universität Stuttgart 1968.
- [64] Kühn, P.: Combined Delay and Loss Systems with Several Input Queues, Full and Limited Accessibility. AEÜ 25 (1971), S. 449-454.
- [65] Kampe, G., Kühn, P.: Graded Delay Systems with Infinite or Finite Source Traffic and Exponential or Constant Holding Time. Part I: Theory. AEÜ 32 (1978), S. 8-18. Part II: Numerical Results and Validation. AEÜ 32 (1978), S. 59-61.
- [66] Krämer, W.: Untersuchung von Wartesystemen mit seriellem Warten. Dissertationsschrift, Universität Stuttgart (1975).
- [67] Kleinrock, L.: Communication Nets, Stochastic Message Flow and Delay. McGraw-Hill Book Comp., New York/San Francisco/Toronto/London (1964).
- [68] Lotze, A.: Optimum Link Systems. 5. ITC, New York (1967), Prebook S. 242-251.
- [69] Kümmerle, K.: An Analysis of Loss Approximations for Link Systems. AEÜ 25 (1971), S. 466-471.
- [70] Courtois, P.J.: Decomposability, Instabilities, and Saturation in Multiprogramming Systems. Comm. ACM 18 (1975), S. 371-377.
- [71] Cox, D.R., Miller, H.D.: The Theory of Stochastic Processes. Chapman and Hall Ltd., London (1965).
- [72] Burke, P.J.: The Output of a Queuing System. Opns. Res. 4 (1956), S. 699-704.
- [73] Burke, P.J.: Output Processes and Tandem Queues. Proceedings of the Symp. on Computer Comm. Networks and Teletraffic, New York (1972). Polytechnic Press of the PIB, Vol. XXII (1972), S. 419-428.
- [74] Wilkinson, R.I., Riordan, J.: Theories for Toll Traffic Engineering in the USA. Bell Syst. Tech. J. 35 (1956), S. 421-514.
- [75] Bretschneider, G.: Die Berechnung von Leitungsgruppen für überfließenden Verkehr in Fernsprechwählanlagen. NTZ 9 (1956), S. 533-540.
- [76] Lotze, A.: A Traffic Variance Method for Gradings of Arbitrary Type. 4. ITC, London (1964). Document 80.
- [77] Lotze, A., Schehrer, R.: The Design of Alternate Routing Systems with Regard to the Variance Coefficient. NTZ Comm. J. 7 (1968), S. 52-56.
- [78] Herzog, U.: Die Bemessung ein- und mehrstufiger Koppelanordnungen der Vermittlungstechnik für angebotenen Überlaufverkehr. Dissertationsschrift, Universität Stuttgart (1968).
- [79] Herzog, U., Lotze, A.: The RDA-Method, a Method Regarding the Variance Coefficient for Limited Access Trunk Groups. NTZ Comm. J. 7 (1968), S. 47-52.

- [80] Herzog, U., Kühn, P.: Comparison of some Multiqueue Models with Over-flow and Load Sharing Strategies for Data Transmission and Computer Systems. Proceedings of the Symp. on Computer Comm. Networks and Teletraffic, New York (1972). Polytechnic Press of the PIB, Vol. XXII (1972), S. 449-472.
- [81] Kümmerle, K.: A Model for the I/O-Channel Traffic in Computer Systems. NTZ 25 (1972), S. 137-141.
- [82] Kümmerle, K.: Berechnungsverfahren für mehrstufige Koppelanordnungen mit konjugierter Durchschaltung - Systematik und Analyse. Dissertationsschrift, Universität Stuttgart (1969).
- [83] Bazlen, D., Kampe, G., Lotze, A.: Design Parameters and Loss Calculation of Link Systems. IEEE-COM 22 (1974), S. 1908-1920.
- [84] Lotze, A., Thierer, G., Röder, A.: PPL - A Reliable Method for the Calculation of Point-to-Point Loss in Link Systems. 8. ITC, Melbourne (1976), Congressbook S. 547/1-14.
- [85] Hießer, L.: About Multi-Stage Link Systems with Queuing. 6. ITC, München (1970), Congressbook S. 233/1-7.
- [86] Kampe, G.: Analysis of Link Systems with Delay. Proceedings Int. Conf. on Communications (ICC), Chicago (1977). IEEE Cat. No. 77 CH 1209-6 CSCB, Bd. 1, S. 10.6-244 bis 10.6-248.
- [87] Wagner, W.: Über ein kombiniertes Warte-Verlustsystem mit Prioritäten. Dissertationsschrift, Universität Stuttgart (1968).
- [88] Basket, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J. ACM 22 (1975), S. 248-260.
- [89] Muntz, R.R.: Poisson Departure Processes and Queuing Networks. IBM Res. Rep. RC-4145 (1972).
- [90] Chandy, K.M., Howard, J.H., Towsley, D.F.: Product Form and Local Balance in Queuing Networks. J. ACM 24 (1977), S. 250-263.
- [91] Chandy, K.M., Herzog, U., Woo, L.: Parametric Analysis of Queuing Networks. IBM J. Res. Develop. 19 (1975), S. 36-42.
- [92] Buzen, J.P.: Computational Algorithms for Closed Queuing Networks with Exponential Servers. Comm. ACM 16 (1973), S. 527-531.
- [93] Reiser, M., Kobayashi, H.: Horner's Rule for the Evaluation of General Closed Queuing Networks. Comm. ACM 18 (1975), S. 592-593.
- [94] Jackson, J.R.: Networks of Waiting Lines. Opns. Res. 5 (1957), S. 518-521.
- [95] Gordon, W.J., Newell, G.F.: Closed Queuing Systems with Exponential Servers. Opns. Res. 15 (1967), S. 254-265.
- [96] Kobayashi, H., Reiser, M.: On Generalization of Job Routing Behavior in a Queuing Network Model. IBM Res. Rep. RC-5252 (1975).
- [97] Lam, S.S.: Queuing Networks with Population Size Constraints. IBM J. Res. Develop. 21 (1977), S. 370-378.

- [98] Kobayashi, H., Silverman, H.F.: Some Dispatching Priority Schemes and Their Effects on Response Time Distribution. IBM Res. Rep. RC-3584 (1971).
- [99] Lavenberg, S.S.: The Steady-State Queuing Time Distribution for the M/G/1 Finite Capacity Queue. IBM Res. Rep. RJ 1150 (1973).
- [100] Walke, B.: Queuing Networks with Degenerate Exponential Servers. Wiss. Ber. AEG-Telefunken 48 (1975), S. 153-157.
- [101] Kobayashi, H.: Application of the Diffusion Approximation to Queuing Networks. Parts I and II. J. ACM 21 (1974), S. 316-328 und S. 459-469.
- [102] Reiser, M., Kobayashi, H.: Accuracy of the Diffusion Approximation for Some Queuing Systems. IBM J. Res. Develop. 18 (1974), S. 110-124.
- [103] Gelenbe, E., Pujolle, G.: The Behavior of a Single Queue in a General Queuing Network. Acta Informatica 7 (1976), S. 123-136.
- [104] Chandy, K.M., Herzog, U., Woo, L.: Approximate Analysis of General Queuing Networks. IBM J. Res. Develop. 19 (1975), S. 43-49.
- [105] Sauer, C.H., Chandy, K.M.: Approximate Analysis of Central Server Models. IBM J. Res. Develop. 19 (1975), S. 301-313.
- [106] Disney, R.L., Cherry, W.P.: Some Topics in Queuing Network Theory. In: Lecture Notes in Economics and Math. Systems, Operations Research, No. 98, Springer-Verlag, Berlin/Heidelberg/New York (1974), S. 23-44.
- [107] Ando, A., Fisher, F.M.: Near-Decomposability, Partition and Aggregation, and the Relevance of Stability Discussions. Int. Econ. Rev. 4 (1963), S. 53-67.
- [108] Brandwajn, A.: A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods. Acta Informatica 4 (1974), S. 11-47.
- [109] Brizzi, M., Grillo, D.: Implementation of Algorithms for Performance Analysis of a Class of Multiprogrammed Computers. Proceedings of the IRIA-IFIP Symp. on Modeling and Performance Evaluation of Computer Systems. (Eds. H. Beilner, E. Gelenbe). North-Holland Publ. Comp. (1977), S. 131-150.
- [110] Bux, W., Kühn, P.: Untersuchungen zur verkehrsgerechten Dimensionierung von Speichervermittlungen. NTG-Fachtagung "Datennetze", Baden-Baden (1976). NTG-Fachberichte Bd. 55, VDE-Verlag, Berlin (1976), S. 83-93.
- [111] Bux, W.: Modelling and Analysis of Store- and Forward Data Switching Centres with Finite Buffer Memory and Acknowledgement Signalling. El. Rechenanlagen 19 (1977), S. 173-180.
- [112] Rothmaier, K., Kampe, G., Kühn, P., Röder, A.: Berechnung des Wähltonverzugs in registergesteuerten Vermittlungssystemen. Monographie Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1978).
- [113] Krämer, W., Langenbach-Belz, M.: Approximate Formulae for the Delay in the Queuing System GI/G/1. 8. ITC, Melbourne (1976), Congressbook S.235/1-8.

- [114] Pack, C.D.: The Output of an M/D/1 Queue. Opns. Res. 23 (1975), S. 750-760.
- [115] Heffes, H.: On the Output of a GI/M/N Queuing System with Interrupted Poisson Input. Opns. Res. 24 (1976), S. 530-542.
- [116] Daley, D.J.: Notes on Queuing Output Processes. In: Math. Methods on Queuing Theory. Springer-Verlag, Berlin/Heidelberg/New York (1974), S. 351-354.
- [117] Marshall, K.T.: Some Inequalities in Queuing. Opns. Res. 16 (1968), S. 651-665.
- [118] Kühn, P.: Analysis of Complex Queuing Networks by Decomposition. 8. ITC, Melbourne (1976), Congressbook S. 236/1-8.
- [119] Kühn, P.J.: Approximate Analysis of General Queuing Networks by Decomposition. Eingereicht bei IEEE Transactions on Communications.
- [120] Ertelt, R., Kühn, P.: Analysis of Complex Queuing Networks for Computer Systems. Monographie Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1975).
- [121] Takács, L.: A Single-Server Queue with Feedback. BSTJ 42: (1963), S. 505-519.
- [122] Herzog, U., Kühn, P., Zeh, A.: Klassifizierung und Analyse von Verkehrsmodellen für das Ablaufgeschehen in Rechnersystemen. Nachrichtentechn. Fachberichte 44 (1972), S. 181-198.
- [123] Kobayashi, H., Konheim, A.G.: Queuing Models for Computer Communications Systems Analysis. IEEE Transact. COM-25 (1977), S. 2-29.
- [124] Villar, J.E.: Traffic Calculations in SPC Systems. Electrical Comm. 52 (1977), S. 199-205.
- [125] Swoboda, J., Rosenbohm, W.: Modell für den Befehlsablauf in einer Rechenanlage: Eine Serverkette mit vorgebarbarer Varianz der Belegungsdauern. Lecture Notes in Computer Science, No. 1. Springer-Verlag, Berlin/New York (1973), S. 314-326.
- [126] Takács, L.: Delay Distributions for One Line with Poisson Input, General Holding Times and Various Orders of Service. BSTJ 42 (1963), S. 487-503.
- [127] Takács, L.: Delay Distributions for Simple Trunk Groups with Recurrent Input and Exponential Service Times. BSTJ 41 (1962), S. 311-320.
- [128] Kühn, P.: On a Combined Delay and Loss System with Different Queue Disciplines. Proceedings of the 6th Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes. Academia, Publ. House of the Czech. Acad. of Sciences, Prag (1973), S. 501-528.
- [129] Jaiswal, N.K.: Priority Queues. Academic Press, New York/London (1968).
- [130] Brandt, J.G.: Das preemptive Warteverlustsystem. Dissertationsschrift, Universität Stuttgart (1971).
- [131] Miller, R.G.: Priority Queues. Ann. Math. Stat. 31 (1960), S. 86-103.
- [132] Herzog, U.: Optimal Scheduling Strategies for Real-Time Computers. IBM J. Res. Develop. 19 (1975), S. 494-504.
- [133] Krämer, W.: Processor Priority Model with Different User Tasks and Operating System Phases. Proceedings of the Int. Symp. on Computer Performance Modeling, Measurement, and Evaluation. (Ed. K.M. Chandy, M. Reiser). North-Holland Publ. Comp., Amsterdam/New York/Oxford (1977), S.305-325.
- [134] Coffman, E.G.: On the Tradeoff between Response and Preemption Costs in a Foreground-background Computer Service Discipline. IEEE Transact. Comp. C-18 (1969), S. 942-947.
- [135] Kühn, P.: The Impact of Queuing Theory on the Optimization of Communications and Computer Systems. Proceedings TIMS XX, Tel Aviv (1973), S. 559-568.
- [136] Takács, L.: Two Queues Attended by a Single Server. Opns. Res. 16 (1968), S. 639-650.
- [137] Eisenberg, M.: Queues with Periodic Service and Changeover Time. Opns. Res. 20 (1972), S. 440-451.
- [138] Hashida, O., Ohara, K.: Line Accommodation Capacity of a Communication Control Unit. Res. of the Electr. Comm. Lab., Nippon Telegraph and Telephone Public Corp., 20 (1972), S. 231-239.
- [139] Kühn, P., Langenbach-Belz, M.: Über die Wirksamkeit zyklischer Abfertigungsstrategien in Realzeitsystemen. Lecture Notes in Computer Science, No. 26. Springer-Verlag, Berlin/Heidelberg/New York (1974), S. 463-472.
- [140] Kuehn, P.J.: Multiqueue Systems with Cyclic Service. Techn. Memorandum TM 78-3451-6, Bell Telephone Laboratories, 1978.
- [141] Langenbach-Belz, M.: Getaktete Wartesysteme bei Rechnern und zentralgesteuerten Nachrichtenvermittlungsanlagen. Dissertationsschrift, Universität Stuttgart (1973).
- [142] Langenbach-Belz, M.: Vergleich zweier Warteschlangenmodelle für Realzeit-Rechnersysteme mit Interrupt- bzw. Takt-gesteuerter Übernahme von Anforderungen aus der Peripherie. Lecture Notes in Computer Science, No. 1. Springer-Verlag, Berlin/Heidelberg/New York (1973), S. 304-313.
- [143] Schwaertzel, H.G.: Serving Strategies of Batch Arrivals in Common Control Switching Systems. NTZ 27 (1974), S. 341-345.
- [144] Weisschuh, H., Wizgall, M.: Investigations on the Traffic Behavior of the Common Control in SPC Switching Systems. 8. ITC, Melbourne (1976), Congressbook, S. 612/1-8.
- [145] Weisschuh, H.: Entwicklung der Steuersoftware für eine rechnergesteuerte Vermittlungsstelle. Dissertationsschrift, Universität Stuttgart (1977).
- [146] Eckberg, A.E., Jr.: Analysis of Inequity of Service for a Particular FEP Polling and Packet Removal Scheme. Proceedings of the Int. Symp. on Computer Performance Modeling, Measurement, and Evaluation. (Ed. K.M. Chandy, M. Reiser). North-Holland Publ. Comp., Amsterdam/New York/Oxford (1977), S. 469-484.
- [147] Walke, B.: Realzeit-Rechnermodelle. R. Oldenbourg-Verlag, München/Wien (1979).

- [148] Sahni, S.K.: Algorithms for Scheduling Independent Tasks. J. ACM 23 (1976), S. 116-127.
- [149] Swoboda, J., Bächle, A.: Effektivität eines Schnellspeichers zwischen Prozessor und Arbeitsspeicher - Über das Einschwingverhalten. NTF 44 (1972), S. 85-95.
- [150] Chow, C.K.: Determination of Cache's Capacity and its Matching Storage Hierarchy. IEEE C-25 (1976), S. 157-164.
- [151] Burnett, G.J., Coffman, E.G.: Analysis of Interleaved Memory Systems Using Blockage Buffers. Comm. ACM 18 (1975), S. 91-95.
- [152] Fuller, S.H.: Minimal-Total-Processing Time Drum and Disk Scheduling Disciplines. Comm. ACM 17 (1974), S. 376-381.
- [153] Oney, W.C.: Queuing Analysis of the Scan Policy for Moving-Head Disks. J. ACM 22 (1975), S. 397-412.
- [154] Teorey, T.J., Pinkerton, T.B.: A Comparative Analysis of Disk Scheduling Policies. Comm. ACM 15 (1972), S. 177-
- [155] Gaver, D.P.: Probability Models for Multiprogramming Computer Systems. J. ACM 14 (1967), S. 423-438.
- [156] Smith, J.I.: Multiprogramming under a Page-on-Demand Strategy. Comm. ACM 10 (1967), S. 636-646.
- [157] Wallace, V.L., Mason, D.L.: Degree of Multiprogramming in Page-On-Demand Systems. Comm. ACM 12 (1969), S. 305-318.
- [158] Spies, P.P.: A Queuing Model Analysis of a Multiplexed Use of a Central Processor Unit and an I/O-Channel. Proceedings Int. Comp. Symp. (ACM), Bonn (1970), S. 282-299.
- [159] Buzen, J.P.: Queuing Network Models of Multiprogramming. PhD-Thesis, Harvard-University, Cambridge/Mass. (1971). NTIS AD 731 575 (1971).
- [160] Shedler, G.S.: A queuing Model of a Multiprogrammed Computer with a Two-Level Storage System. Comm. ACM 16 (1973), S. 3-10.
- [161] Brown, J.C., Chandy, K.M., Hogarth, J., Lee, C.: The Effect on Throughput of Multiprocessing in a Multiprogramming Environment. IEEE C-22 (1973), S. 728-735.
- [162] Brandwajn, A., Buzen, J.P., Gelenbe, E., Potier, D.: A Model of Performance for Virtual Memory Systems. Proceedings ACM SIGMETRICS Symp. (1974).
- [163] Denning, P.J., Graham, G.S.: Multiprogrammed Memory Management. Proc. IEEE 63 (1975), S. 924-939.
- [164] Badel, M., Gelenbe, E., Leroudier, J., Potier, D.: Adaptive Optimization of a Time-Sharing System's Performance. Proc. IEEE 63 (1975), S. 958-965.
- [165] Muntz, R.R.: Analytic Modeling of Interactive Systems. Proc. IEEE 63 (1975), S. 946-953.
- [166] Chen, P.P.-S.: Queuing Network Model of Interactive Computing Systems. Proc. IEEE 63 (1975), S. 954-957.
- [167] Brown, J.C. et.al.: Hierarchical Techniques for the Development of Realistic Models of Complex Computer Systems. Proc. IEEE 63 (1975), S. 966-975.
- [168] Kühn, P.: Zur optimalen Steuerung des Multiprogrammgrades in Rechnersystemen mit virtuellem Speicher und Paging. Lecture Notes in Computer Science, No. 34. Springer-Verlag, Berlin/Heidelberg/New York (1975), S. 567-580.
- [169] Walke, B.: Durchsatzberechnung für Rechenanlagen bei wählbarer Aufteilung des Arbeitsspeichers unter mehrere Programme unterschiedlichen Platzbedarfs. Dissertationsschrift, Universität Stuttgart (1975).
- [170] Brandwajn, A.: A Model of a Time-Sharing System with Two Classes of Processes. Lecture Notes in Computer Science, No. 34. Springer-Verlag, Berlin/Heidelberg/New York (1975), S. 548-566.
- [171] Chow, W.-M., Chiu, W.W.: An Analysis of Swapping Policies in Virtual Storage Systems. IEEE SE-3 (1977), S. 150-156.
- [172] Dietrich, G.: Verkehrsmodell für zentralgesteuerte Vermittlungssysteme. Elektrisches Nachrichtenwesen 50 (1975), S. 30-36.
- [173] Schmid, H., Kühn, P., Witzgall, M.: Simulation der Steuerdatenflüsse in rechnergesteuerten Nebenstellenanlagen. Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1977).
- [174] Kühn, P.: Delay Problems in Communications Systems - Classification of Models and Tables for Application. Proceedings Int. Conf. on Communications (ICC), Chicago (1977). IEEE Cat. No. 77 CH 1209-6 CSCB, Bd.1., S. 10.5-237 bis 10.5-243.
- [175] Kühn, P.: Tables on Delay Systems. Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1976).
- [176] Oliver, N., Chu, W.W., Opperbeck, H.: Measurement Data on the Working Set Replacement Algorithm and their Applications. Proc. Symp. on Computer-Comm. Networks and Teletraffic, New York (1972). Polytechnic Press of the PIB, Vol. XXII (1972), S. 113-124.
- [177] Hatfield, D.J.: Experiments on Page Size, Program Access Patterns, and Virtual Memory Performance. IBM J. Res. Develop. 16 (1972), S. 58-66.
- [178] Wolf, P.: Eine Methode zur Untersuchung von Programmen bezüglich eines Betriebssystems mit Virtuellem Speicher - Anwendung zur Vorhersage des Programmverhaltens. Lecture Notes in Computer Science, No. 8. Springer-Verlag, Berlin/Heidelberg/New York (1974), S. 289-300.
- [179] Rodriguez-Rosell, J.: Empirical Working Set Behavior. Comm. ACM 16 (1973), S. 556-560.
- [180] Peter, F., Kühn, P., Weisschuh, H., Witzgall, M.: Modelle zur künstlichen Erzeugung von Seitenreferenzketten zur Simulation von Rechnersystemen mit Virtuellem Speicher. Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1975).

- [181] Schamal, W., Kühn, P., Weisschuh, H., Wizgall, M.: Simulation von Rechnersystemen mit Virtuellem Speicher und Ein/Ausgabeverkehr. Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1975).
- [182] Garreis, G., Kühn, P.: Durchsatz und Antwortzeiten in Teilnehmer-Rechnersystemen mit Virtuellem Speicher. Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1976).
- [183] Brandwajn, A.: A Model of a Time Sharing Virtual Memory System Solved Using Equivalence and Decomposition Methods. Acta Informatica 4(1974), S. 11-47.
- [184] Eisenberg, M.: Two Queues with Alternating Service. (forthcoming paper)
- [185] Dor, N.M.: Guide to the Length of Buffer Storage Required for Random (Poisson) Input and Constant Output Rates. IEEE Transact. Electr. Comp. EC-16 (1967), S. 683-684.
- [186] Chu, W.W.: Buffer Behavior for Poisson Arrivals and Multiple Synchronous Constant Outputs. IEEE Transact. Comp. C-19 (1970), S. 530-534.
- [187] Maritsas, D.G., Hartley, M.G.: Buffer Length for Erlang-Input and Constant Removal Rate. IEEE Transact. Comp. C-19 (1970), S. 839-843.
- [188] Drechsel, G., Kühn, P.: Untersuchung von Warteverlustsystemen mit getakteter Abfertigung. Monographie, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart (1976).
- [189] Burke, P.J.: Delays in Single Server Queues with Batch Input. Opns. Res. 23 (1975), S. 830-833.

1. EINFÜHRUNG

Die vorliegende Arbeit behandelt die Analyse zufallsabhängiger Vorgänge in Systemen zur Vermittlung und Verarbeitung von Nachrichten. Obwohl die Funktionen in derartigen Systemen nach vorher festgelegten Regeln eindeutig ablaufen, hängt das durch eine Vielzahl von "Teilnehmern" verursachte tatsächliche Ablaufgeschehen von einer Reihe zufallsabhängiger Größen ab. Aus der Erfahrung sowie aus zahlreichen verkehrstheoretischen Untersuchungen ist auch bekannt, daß diese Zufallsabhängigkeit die Leistungsfähigkeit in starkem Maße beeinflusst. Die Untersuchungen in dieser Arbeit sollen einen Beitrag liefern zur Modellierung und theoretischen Analyse von zufallsabhängigen Vorgängen in Systemen zur Nachrichtenvermittlung und Nachrichtenverarbeitung sowie zur verkehrsgerechten Dimensionierung solcher Systeme.

1.1 Entwicklungstrends

Der rasche Fortschritt auf dem Gebiet der Elektronik hat sich am deutlichsten in der Entwicklung der Technik von Rechenanlagen manifestiert. Zur Zeit zeichnen sich hier mehrere parallele Trends ab zu:

- Kleinrechnern für spezielle Aufgaben,
- Großrechnern für ein universelles Aufgabenspektrum und
- Rechnersystemen, welche beide Vorteile durch Kopplung von Einzelsystemen (Rechnernetze) anstreben.

Klein- und Kleinstrechner ("Mikroprozessoren" mit hochintegrierten Speichereinheiten) erschließen zur Zeit neue Bereiche wie etwa in der Automatisierungstechnik. Großrechnersysteme erleben seit mehr als zwei Jahrzehnten eine stetige Leistungssteigerung durch die Möglichkeiten von erhöhter Arbeitsgeschwindigkeit und Speicherkapazität. Insbesondere verspricht die Datenfernverarbeitungstechnik eine weitere Stufe von Möglichkeiten in der Nachrichtenverarbeitung.

Aus diesen Gründen ist die Rechnertechnik auch in zunehmendem Maße in die Nachrichtenvermittlung eingedrungen. Dies zeigt sich einmal im Einsatz von Steuerrechnern in Fernsprech- und Datenvermittlungssystemen sowie zum anderen in dem Aufbau von speziellen Datennetzen für den Rechnerverkehr selbst. Durch den Einsatz schneller Steuerrechner verlagert sich der Schwerpunkt von den klassischen, dezentral gesteuerten Vermittlungssystemen zu zentralgesteuerten Systemen hin. Neben der Realisierung einer Reihe neuer Leistungsmerkmale für den Teilnehmer und die Verwaltung erhofft man sich zudem eine größere Wirtschaftlichkeit. Die extrem hohe Verfügbarkeitsforderung in der Vermittlungstechnik bedingt allerdings einen recht komplexen Aufbau in der Systemstruktur und in der Systemsteuerung (Hardware und Software). Alle neuen Systeme verwenden deshalb aus Sicherheitsgründen

Mehrrechnerkonzepte. Diese arbeiten nach folgenden Sicherungsprinzipien:

- Mikrosynchronismus mit operationsweisem Vergleich
- Reserverechnerprinzip mit synchronem bzw. bedarfsweisem Auffrischen der Systemzustandsinformation (hot/cold standby)
- Lastteilungsprinzip (load sharing) mit Aufteilung der Aufgaben auf mehrere gleiche Baueinheiten.

Darüberhinaus zeigt sich bereits ein weiterer Trend von der Zentralsteuerung zu einer Mischform aus zentraler und dezentraler Steuerung, indem entweder die Zentralsteuerung durch Auslagerung von Teilfunktionen entlastet wird (dezentrale Vorverarbeitung) oder indem die Vermittlungsgrundfunktionen in funktionspezifische Mikrorechner verlagert werden, während die übergeordnete Zentralsteuerung die zusätzlichen "nichtlebensnotwendigen" Leistungsmerkmale realisiert.

1.2 Verkehrstheoretische Problemstellungen

Das Ablaufgeschehen in Vermittlungs- und Rechnersystemen ist durch einen regen Austausch (Verkehr) von Nutz- und Steuerungsinformationen zwischen dem Teilnehmer (Benutzer) und dem System sowie zwischen einzelnen Systemteilen selbst gekennzeichnet. Eine zufriedenstellende Planung wird außer zweckmäßigen Konzepten in Hard- und Software eine verkehrsgerechte Auslegung einbeziehen müssen. Es ist die Aufgabe der Nachrichtenverkehrstheorie, Verfahren zur Analyse und zur optimalen Auslegung bereitzustellen.

Die Grundfragen der verkehrstheoretischen Analyse beziehen sich auf Auslastungen von Systemteilen, auftretende Blockierungen, Wartezeiten sowie Antwort- bzw. Reaktionszeiten. Diese Größen hängen in starkem Maße ab von der zugrundegelegten Systemstruktur, der Systembetriebsweise sowie von den statistischen Eigenschaften der an das System gestellten "Anforderungen". In Nachrichtenvermittlungssystemen interessiert i. a. die Leistungsfähigkeit von Durchschaltenetzen und Steuerungen, wobei das Ablaufgeschehen von zufallsbedingten Größen abhängt wie z. B. Eintreffzeitpunkten von Anrufen, Wähldauern, Gesprächsdauern oder etwa Laufzeiten von Vermittlungsprogrammen innerhalb von Steuerrechnern. In Datenverarbeitungsanlagen interessieren Auslastungen einzelner Systemkomponenten wie Prozessoren, Speichermodule und Kanäle sowie Durchlaufzeiten für Programme oder Reaktionszeiten auf externe Realzeit-Anforderungen.

Von der Praxis des Systementwurfs her gesehen ist die Bereitstellung zuverlässigen Materials für die Auslegung und Dimensionierung von Interesse. Die sehr oft schwierigen Analyseverfahren sind dem Anwender i. a. nicht bekannt bzw. in der oft gebotenen Eile nicht auswertbar; hier stellt sich die Frage der Standardisierung von Teilproblemen und der entsprechenden Unterstützung durch prakti-

kables Material wie Tabellen, Diagramme oder Prozeduren.

An den Systementwurf knüpfen sich weitere Fragestellungen an wie der Vergleich alternativer Struktur- und Steuerungskonzepte, Fragen der Ausbaufähigkeit sowie des Überlastverhaltens. Die Frage einer Optimalsynthese von Systemen ist i.a. sehr schwierig zu beantworten. In Teilbereichen wie der optimalen Struktur von Durchschaltenetzen, der wirtschaftlichen Verkehrsaufteilung in einzelnen Netzknoten oder der optimalen Einteilung von Steuerungsaufrufen in Prioritätsklassen wurden bereits analytische Lösungen angegeben. Gesamtsysteme können bislang allerdings nur iterativ optimiert werden durch schrittweises Einsetzen von Teilergebnissen.

1.3 Lösungsmethodik

Allgemein gesprochen können Systeme zur Nachrichtenvermittlung und Nachrichtenverarbeitung als "Bedienungssysteme" aufgefaßt werden. Sie bestehen aus unterschiedlichen "Betriebsmitteln", welche anfordernden "Teilnehmern" zeitweise zugeteilt werden.

In einem Fernsprechvermittlungssystem fordert z.B. der rufende Teilnehmer in der Verbindungsaufbauphase zentralisierte Einrichtungen an wie Wahlaufnahmesätze, Speicher, Prozessoren, Kanäle, Einsteller etc.; nach erfolgtem Verbindungsaufbau sind ihm für die Dauer des Gesprächs die im Verbindungszug zum gerufenen Teilnehmer liegenden Leitungsabschnitte, Koppelinrichtungen und Verbindungssätze zugeteilt.

In einem Rechnersystem bestehen die Anforderungen letztlich aus Befehlen, welche der Teilnehmer (Benutzer) in Form eines Programmes oder direkt über eine Bedienungstation dem System mitteilt. Zur Ausführung dieser Befehle werden zeitweise zentrale Betriebsmittel zugeteilt wie z.B. Speicherbereiche, Kanäle, Prozessoren, E/A-Geräte usw.

Die Lösung verkehrstheoretischer Problemstellungen erfolgt i.a. nach folgendem Vorgehen:

1. Modellbildung

- a) Feststellung des funktionellen Zusammenhangs zwischen Anforderung, Betriebsmittelzuteilung und Reaktion.
- b) Ermittlung der stochastischen Eigenschaften von Anforderungen hinsichtlich Anforderungszeitpunkt und Belegungsdauer.
- c) Konstruktion eines Verkehrsmodelles, welches das Ablaufgeschehen mit Hilfe weniger standardisierter Modellkomponenten beschreibt. Modellkomponenten sind dabei z.B. Verkehrsquellen, Bedienungseinheiten, Warteschlangen usw.

2. Modellanalyse

- a) Definition von Zustandsvariablen
- b) Beschreibung des Ablaufgeschehens mittels stochastischer Prozesse
- c) Analyse des stochastischen Prozesses

Die Analyse kann dabei entweder exakt oder approximativ erfolgen. Zur Nachprüfung approximativer Lösungen bzw. zur Untersuchung sehr komplexer Zusammenhänge wird i.a. die Simulation eingesetzt.

3. Systementwurf

- a) Definition von allgemein anwendbaren Grundmodellen
- b) Bereitstellung von Hilfsmitteln zur Dimensionierung
- c) Optimierung

Im Verlaufe der Arbeit werden diese Schritte ausführlicher behandelt und durch Beispiele unterstützt.

1.4 Übersicht

Anschließend an die Einführung werden im 2. Kapitel der Arbeit Struktur, Betriebsarten und Ablaufvorgänge in Systemen zur Nachrichtenvermittlung und Nachrichtenverarbeitung beschrieben. Nach allgemeinen Bemerkungen zu funktions- und verkehrsbezogener Betrachtungsweise des Ablaufgeschehens werden zwei Beispiele ausführlicher behandelt: Rechnergesteuerte Fernsprechvermittlungssysteme und Rechnersysteme mit hierarchischem Speicherprinzip. Abschließend werden jeweils typische Verkehrsprobleme aufgezeigt.

Im 3. Kapitel wird versucht, die Methodik der Modellbildung zu umreißen. Vom Standpunkt der Anwendung aus gesehen, ist das Auffinden eines geeigneten Modells der wichtigste Schritt überhaupt, denn die Aussagen einer Analyse können nur im Rahmen der Modellgenauigkeit gültig sein. Zu diesem Punkt bestehen bislang nur wenig allgemein anwendbare Methoden. Im allgemeinen wird vielmehr aus der direkten Anschauung heraus ein spezielles Modell für den zu untersuchenden Anwendungsfall entwickelt. Nach der Vorstellung der wesentlichsten Bestandteile von Verkehrsmodellen wird das Vorgehen in systematische Einzelschritte unterteilt. Die entstehenden Modelle werden schließlich nach Struktur-, Betriebs- und Verkehrsgesichtspunkten klassifiziert.

Das 4. Kapitel gibt einen Überblick über die verschiedenen Methoden der Modellanalyse. Neben experimentellen Methoden (Messung, Simulation) wird auf exakte und approximative analytische Methoden eingegangen. Hierbei wird versucht, sowohl exakte als auch approximative Methoden nach grundsätzlichen Gesichtspunkten zu systematisieren, wobei die wesentlichsten Merkmale, Vorteile, Grenzen und die Anwendbarkeit diskutiert werden.

Im Anschluß an die einführenden Kapitel 2-4 zur Systembeschreibung, Verkehrsproblematik sowie der Beschreibung von Methoden zur Modellbildung und Modellanalyse werden in den Kapiteln 5 und 6 spezielle Methoden der Systemanalyse entwickelt und angewandt, insbesondere im Hinblick auf die Analyse von Systemen mit relativ komplexer Struktur bzw. komplexen Wechselwirkungen zwischen einzelnen Komponenten dieser Systeme.

Das 5. Kapitel behandelt die Analyse von Warteschlangennetzen. Nach einem zusammenfassenden Überblick über exakte und approximative Ergebnisse wird ein Verfahren zur Analyse von allgemeinen Warteschlangennetzen vorgestellt, welches auf einer neuen Zerlegungsmethode aufbaut. Die angewandte Zerlegungsmethode beruht auf einer systemtheoretischen Betrachtungsweise, indem das Netz in Teile zerlegt wird, welche im wesentlichen durch ihre Eingangs/Ausgangscharakteristik beschrieben werden. Das Analyseverfahren wird systematisch in eine Reihe von Einzelschritten aufgelöst und in einem allgemein anwendbaren Algorithmus realisiert. Zum Genauigkeitsnachweis einzelner Operationen bzw. des gesamten Verfahrens werden Simulationsvergleiche angegeben.

Das 6. und abschließende Kapitel ist der Anwendung gewidmet. Nach einer kurzen Übersicht über die wesentlichsten Modelle und Ergebnisse werden folgende Beispiele zur Systemanalyse und Optimierung ausführlicher behandelt:

- Durchsatzoptimierung in Stapelverarbeitungssystemen mit Virtuellem Speicher
- Durchsatz und Antwortzeiten in Teilnehmerrechensystemen
- Antwortzeiten in interaktiven Systemen
- Steuerdatenflußanalyse in rechnergesteuerten Nachrichtenvermittlungssystemen.

Die Gemeinsamkeiten dieser ausgewählten Anwendungsbeispiele bestehen in einer relativ komplexen System- bzw. Modellstruktur. Das Hauptziel dieses Kapitels liegt deshalb weniger in der Analyse von abgegrenzten bzw. isoliert betrachteten Teilproblemen als vielmehr in der Methodik der Behandlung der Gesamtmodelle, welche aus zum Teil sehr unterschiedlichen Teilmodellen zusammengesetzt und durch starke gegenseitige Wechselwirkungen gekennzeichnet sind. Methodisch werden hierbei exakte Verfahren, Zerlegungsmethoden sowie Grenzwertbetrachtungen angewandt. Die verschiedenen Anwendungsbeispiele werden exemplarisch durch numerische Auswertungen unterstützt, welche den prinzipiellen Einfluß der wesentlichsten Systemparameter aufzeigen bzw. Anhaltspunkte für den Systementwurf und die Optimierung liefern. Approximative Methoden werden schließlich einem Vergleich mit Simulationsergebnissen unterzogen, welcher Aufschluß über die Güte der Approximationen gibt.

2. ABLAUFGESCHEHEN IN SYSTEMEN ZUR NACHRICHTENVERMITTLUNG UND NACHRICHTENVERARBEITUNG

Ziel dieses Kapitels ist das Aufzeigen von wesentlichen Verkehrsproblemen in Nachrichtensystemen. Nach einer einleitenden Diskussion verschiedener Betrachtungsweisen für Ablaufvorgänge werden zwei Beispiele ausführlicher behandelt.

2.1 Ablaufvorgänge und Betrachtungsweisen

Systeme zur Vermittlung und Verarbeitung von Nachrichten sind modular aus einer Vielzahl funktionspezifischer Komponenten ("Betriebsmittel") aufgebaut. Zur Abwicklung eines "Auftrages" ist i.a. eine Kommunikation zwischen diesen Komponenten erforderlich. Der Informationsaustausch erfolgt dabei über vereinbarte "Schnittstellen" nach eindeutig definierten Prozeduren.

Die Beschreibung der ablaufenden Vorgänge kann dabei nach zwei unterschiedlichen Betrachtungsweisen erfolgen:

- funktionsbezogene Betrachtung
- verkehrsbezogene Betrachtung

2.1.1 Funktionsbezogene Betrachtung

Betrachtet werde ein System nach Bild 2.1.

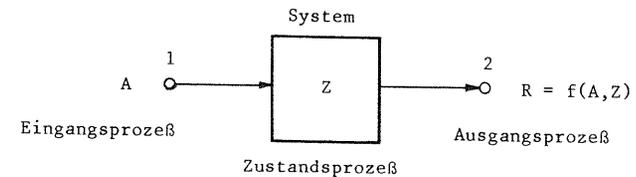


Bild 2.1. Funktionsbezogene Betrachtung

Eine "Anforderung" A (Eingangssignal, Meldung, Anreiz) werde dem System über den Eingang 1 zugeführt. Die Reaktion $R = f(A,Z)$ des Systems am Ausgang 2 (Ausgangssignal) ist i.a. eine Funktion der Anforderung A und des Systemzustands Z. Ist das System beispielsweise eine Vermittlungsstelle, die Anforderung A ein Anruf (bestehend aus einer Anzahl von Teilanforderungen wie Abheben und Wählzifferneingabe), so hängt die Reaktion R von der in A enthaltenen Zielinformation sowie vom Zustand Z des Systems ab: R kann ein abhängig vom Netzzustand ausgewählter und durchgeschalteter Verbindungsweg sein; die Reaktion R kann aber auch eine Anzeige eines erfolglosen Verbindungsversuches sein ("Verluststraf"). Stellt das System eine Datenverarbeitungsanlage und die Anforderung A ein Benutzerprogramm

dar, so ist i.a. die Reaktion R (die Ausgabedaten als Ergebnis eines Rechenlaufes) lediglich von A, nicht dagegen vom Systemzustand Z abhängig.

Die formale Beschreibung funktioneller Abläufe eines Systems kann allgemein nach folgenden Gesichtspunkten erfolgen:

- Festlegung von Eingangs/Ausgangs-Schnittstellen des Systems
- Definition eines eindeutigen Alphabets der Eingangs- bzw. Ausgangssignale
- Beschreibung der logischen Folge von Eingangssignalen (Eingangsprozeß)
- Beschreibung der Systemreaktionen, d.h. der logischen Folge von Systemzustandsänderungen (Zustandsprozeß) bzw. ausgelösten Ausgangssignale (Ausgangsprozeß).

Ihrer Natur nach ist diese Beschreibungsweise system- bzw. automatentheoretisch. Die Beschreibungsform hängt dabei sehr stark vom jeweiligen Anwendungsfall ab, wie die folgenden Beispiele zeigen mögen:

Kombinatorisches Netzwerk:	Schaltalgebraische Funktion
Sequentielles Netzwerk:	Folgeschaltfunktion
Steuerrechner:	Ablaufdiagramme für Hardware- und Software-Funktionen.

Die funktionelle Beschreibung von Systemen ist nicht Gegenstand dieser Arbeit. Im Zusammenhang mit der für eine verkehrstheoretische Analyse notwendigen Modellbildung ist die Kenntnis der funktionellen Abläufe jedoch von großer Wichtigkeit.

2.1.2 Verkehrsbezogene Betrachtung

Während bei der funktionsbezogenen Betrachtung das Hauptinteresse darin liegt, die funktionellen Zusammenhänge zwischen Eingangs-, Zustands- und Ausgangsgrößen eines Systems zu beschreiben, interessiert bei der verkehrsbezogenen Betrachtung die Systemreaktion im Hinblick auf die Abwicklung der von der Gesamtheit aller Anforderungen ausgelösten Vorgänge bzw. die Fähigkeit des Systems, seine Betriebsmittel möglichst optimal den zufallsabhängig eintreffenden Anforderungen zuzuteilen. Als Beispiel werde hierzu das System nach Bild 2.2 betrachtet:

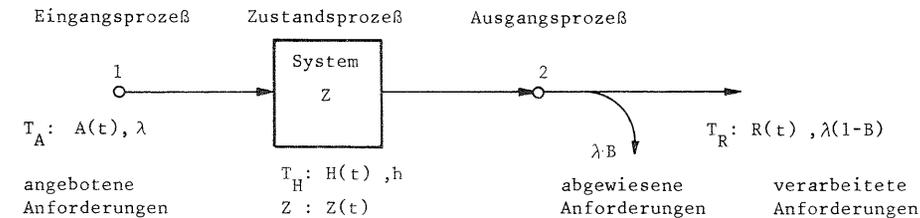


Bild 2.2. Verkehrsbezogene Betrachtung

Anforderungen mögen in zufallsabhängigen Abständen T_A dem System zur Verarbeitung ("Bedienung") angeboten werden. Die zeitliche Folge von Anforderungen bildet den Eingangs- oder Ankunftsprozeß, welcher z.B. mittels einer Ankunftsabstands-Verteilungsfunktion $A(t) = P\{T_A \leq t\}$ mit Ankunftsrate λ (λ = Anzahl eintreffender Anforderungen pro Zeiteinheit) beschrieben wird. Solange bei Ankunft einer Anforderung noch zuteilbare Betriebsmittel ("Bedienungseinheiten") innerhalb des Systems frei sind, kann die Anforderung sofort bedient werden. Je nach Systemverhalten im Falle der Ankunft einer Anforderung, welche auf einen Zustand trifft, in dem alle zuteilbaren Betriebsmittel belegt sind, unterscheidet man zwei grundsätzliche Betriebsarten:

- Verlustbetrieb: Die eintreffende Anforderung wird sofort abgewiesen.
- Wartebetrieb: Die eintreffende Anforderung kann zwischengespeichert und später bedient werden.

Innerhalb des Systems durchlaufen die Anforderungen gewisse Verarbeitungs-(Bedienungs-)Stationen. Im einfachsten Falle werden die Anforderungen nur einmal bedient, und es stehen nur gleichartige Bedienungseinheiten zur Verfügung. Der Bedienungsprozeß läßt sich dann durch die Verteilungsfunktion $H(t) = P\{T_H \leq t\}$ der zufälligen Bedienungszeiten T_H beschreiben mit der mittleren Bedienungsdauer h .

Das Zusammenwirken von Ankunftsprozeß und Bedienungsprozeß löst einen Zustandsprozeß $Z(t)$ des Systems aus, d.h. eine zeitliche Folge von Systemzuständen. Entsprechend bildet die zeitliche Folge von verarbeiteten Anforderungen (Systemreaktionen) einen Ausgangsprozeß. Dieser Ausgangsprozeß kann z.B. beschrieben werden durch die Rate abgewiesener bzw. verarbeiteter Anforderungen bzw. die Abstands-Verteilungsfunktion $R(t) = P\{T_R \leq t\}$ der Reaktionszeiten T_R für verarbeitete Anforderungen.

Die eintreffenden Anforderungen stellen einen "Verkehr" an das System dar, bei dessen Verarbeitung gegenseitige Behinderungen bzw. Verzögerungen auftreten können. Die formale Beschreibung der Verkehrsabläufe eines Systems kann nach folgenden Gesichtspunkten erfolgen:

- Beschreibung der zeitlichen Folge von Anforderungen (Eingangs- oder Ankunftsprozeß)
- Beschreibung der Bedienungsdauern (Bedienungsprozeß)
- Beschreibung der Systemstruktur (Anzahl und Art der Betriebsmittel)
- Beschreibung der Systemorganisation (Betriebsmittelverwaltung)
- Beschreibung des Zustandsprozesses mit Hilfe der Theorie Stochastischer Prozesse.

Aus der Verkehrsanalyse des Systemverhaltens werden charakteristische Kenngrößen abgeleitet. In einem Verlustsystem ist die wichtigste Kenngröße beispielsweise

die Verlustwahrscheinlichkeit B für eintreffende Anforderungen. Der Anteil $B \cdot \lambda$ der Ankunftsrate λ geht verloren, der Anteil $(1-B) \cdot \lambda$ wird verarbeitet. Bei WartebetrieB läßt sich das System beispielsweise durch die i.a. ebenfalls zufallsabhängigen Antwortzeiten T_R bzw. deren Antwortzeit-Verteilungsfunktion $R(t)$ beschreiben. Den Zusammenhang zwischen Eintreffzeitpunkt einer Anforderung und Reaktionszeit T_R des Systems verdeutlicht ein einfaches Beispiel eines reinen Wartesystems im nachfolgenden Bild 2.3:

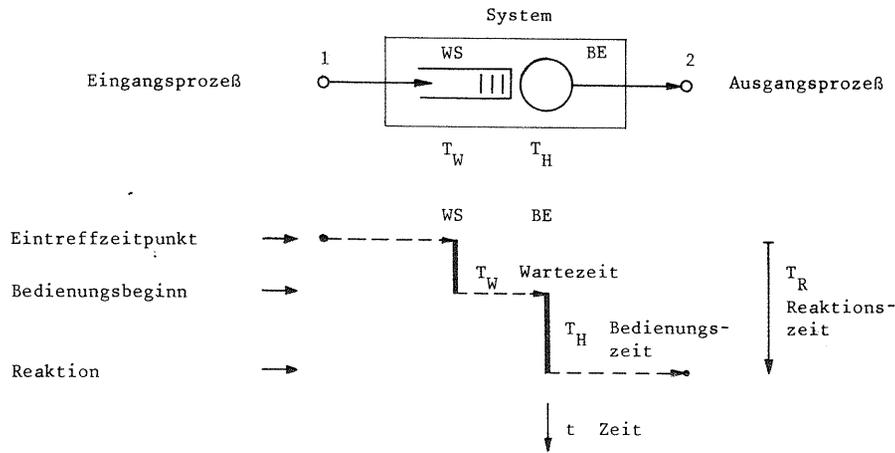


Bild 2.3. Zur verkehrsbezogenen Beschreibung von Wartesystemen
 WS: Warteschlange
 BE: Bedienungseinheit

Die Reaktionszeit T_R einer Anforderung setzt sich additiv aus der i.a. zufallsabhängigen Bedienungszeit T_H und der zufälligen Wartezeit T_W zusammen. In T_W drückt sich die Beeinflussung der betrachteten Anforderung durch die Gesamtheit der anderen Anforderungen aus.

Zur vollständigen Kennzeichnung des Ablaufgeschehens müssen sowohl die funktionellen als auch die verkehrsmäßigen Zusammenhänge beschrieben werden. Aus den funktionellen Zusammenhängen heraus erfolgt die Konstruktion von Baugruppen, während über die Verkehrsanalyse die Regeln zu ihrer Dimensionierung abgeleitet werden.

2.2 Beispiel 1: Rechnergesteuerte Fernsprechvermittlungssysteme

Neuere Fernsprechvermittlungen sind i.a. durch das Merkmal der Rechnersteuerung gekennzeichnet. Die interne Struktur der Rechnersteuerung ist zum Teil sehr unterschiedlich und hängt ab von Kriterien der Rechnerstechnologie, des Sicher-

heitskonzeptes, der Größe, der Ausbaufähigkeit u.a.m. Das folgende Beispiel beschränkt sich daher auf einige wesentliche Grundelemente, welche i.a. allen Konzeptionen gemeinsam sind. Dieses Beispiel dient gleichzeitig als Grundlage für die spätere Modellbildung und Modellanalyse.

2.2.1 Architektur

Die prinzipielle Struktur einer rechnergesteuerten Fernsprechvermittlung ist in Bild 2.4 dargestellt. Sie läßt sich in drei Ebenen unterteilen, welche sich in Arbeitsgeschwindigkeit und Zentralisierungsgrad unterscheiden, vergl. Ref. [1-3]:

- a) Periphere Einrichtungen
- b) Teilzentrale Steuer- und Informationsübertragungseinrichtungen
- c) Zentrale Steuereinrichtungen

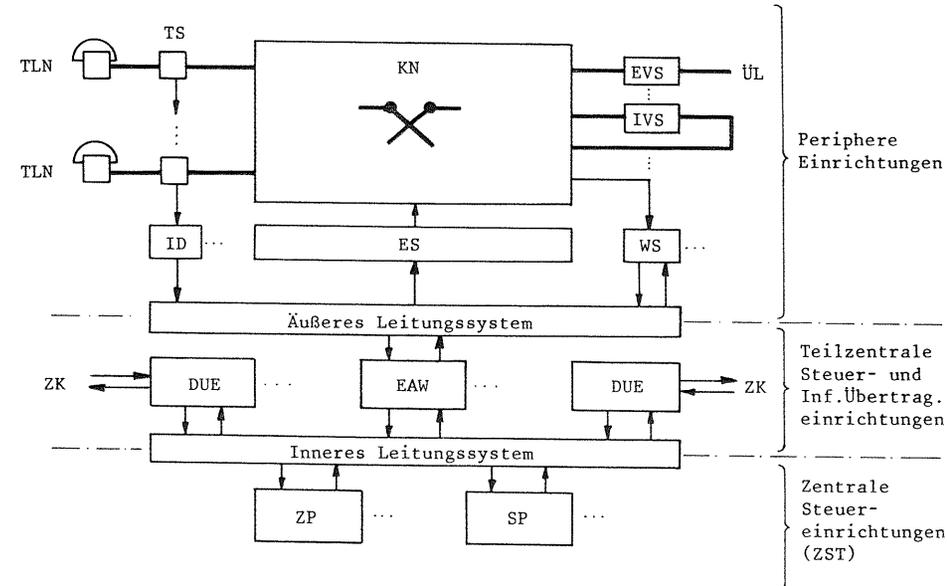


Bild 2.4. Grundstruktur einer rechnergesteuerten Fernsprechvermittlungsstelle

TLN	Teilnehmerapparat	IVS	Intern-Verbindungssatz	ZK	zentraler Datenkanal
TS	Teilnehmersatz	EVS	Extern-Verbindungssatz	ZP	Zentralprozessor
KN	Koppelnetz	EAW	Ein/Ausgabewerk	SP	Speichereinheit
ID	Identifizierer	DUE	Datenübertragungseinrichtung		
ES	Einstellsatz	ÜL	Übertragungsleitung		
WS	Wählsatz				

a) Periphere Einrichtungen

Die peripheren Einrichtungen umfassen die Teilnehmerapparate TLN, die teilnehmerindividuellen Steuersätze TS, das Koppelnetz KN, interne bzw. externe Verbindungsätze IVS bzw. EVS zur Teilnehmerspeisung und Überwachung des Verbindungszustandes, Identifizierer ID zur Feststellung eines rufenden Teilnehmers, Einsteller ES für das Koppelnetz sowie Wahlsätze WS zur Zwischenspeicherung von Wählinformationen. Die Nutzwege zwischen Teilnehmern derselben Vermittlungsstelle werden über das Koppelnetz unter Einschaltung eines Intern-Verbindungsnetzes bzw. bei Externverbindungen zwischen Teilnehmern und Übertragungsleitungen unter Einschaltung eines Extern-Verbindungssatzes hergestellt.

b) Teilzentrale Steuer- und Informationsübertragungseinrichtungen

Die peripheren Einrichtungen sind steuerungsmäßig über ein Äußeres Leitungssystem (Signalsammel- und verteilsystem) mit dem Ein/Ausgabewerk EAW verbunden. Das Ein/Ausgabewerk gehört zu den teilzentralisierten Einrichtungen; es puffert die Eingabemeldungen bzw. Ausgabebefehle, unternimmt in der Regel eine Vorverarbeitung (Adressierung, Codewandlung, Schnittstellenanpassung etc.) und stellt die Schnittstelle zu den zentralen Steuereinrichtungen dar. Zur Ebene der teilzentralen Einrichtungen gehören u.a. auch die Datenübertragungseinrichtungen DUE zum Austausch von Steuerinformationen zwischen Vermittlungsstellen.

c) Zentrale Steuereinrichtungen

Der Informationsaustausch zwischen den teilzentralen Einrichtungen und den zentralen Einrichtungen erfolgt über ein Inneres Leitungssystem. Je nach Art und Ausbau der Vermittlungsstelle, wird dieser Informationsaustausch entweder unterbrechungsgesteuert, taktgesteuert oder durch zyklische Abfragesteuerung realisiert. Die zu übertragenden Informationen werden über einen Pufferbereich der zentralen Speichereinheit ausgetauscht. Die Verarbeitung der von den peripheren bzw. teilzentralen Einrichtungen eingegebenen Anreize erfolgt programmgesteuert durch den Zentralprozessor ZP. Die Arbeits-(Vermittlungs-)Programme, die Organisationsprogramme ("Betriebssystem") sowie die Datenbereiche (Zustandsdaten der Vermittlungsstelle) sind dazu in der Speichereinheit SP abgelegt.

2.2.2 Steuerungsprinzipien

Die Steuerung eines Vermittlungssystems umfaßt alle organisatorischen Vorgänge, welche aufgrund der eingegebenen Zielinformation zum Auf- und Abbau von Verbindungen erforderlich sind. Die Gesamtaufgabe der Steuerung zerfällt in mehrere Teilsteueraufgaben, welche - je nach Realisierung - in sehr unterschiedlicher

Weise ausgeführt werden können. In diesem Zusammenhang sollen hier nur die drei grundsätzlichen Steuerungsprinzipien diskutiert werden:

- a) Dezentrale Steuerung
- b) Zentrale Steuerung
- c) Mischformen aus dezentraler und zentraler Steuerung.

a) Dezentrale Steuerung

Das dezentrale Steuerungsprinzip wurde vorwiegend in der klassischen direktgesteuerten Vermittlungstechnik angewandt, vergl. [1,4]. Die einzelnen Betriebsmittel wie Sätze, Koppelinrichtungen, Leitungen usw. besitzen eigene Steuereinrichtungen, in denen der Zustand des Betriebsmittels gespeichert ist und durch welche Steuerinformationen zum Auf- und Abbau von Verbindungen ausgetauscht bzw. verarbeitet werden. Die "Intelligenz" dieser dezentralisierten Steuereinrichtungen ist begrenzt; umgekehrt zeichnen sich diese Systeme durch Robustheit und geringe Wirkbreite bei Störungen aus. Durch die neuen Möglichkeiten der Mikroelektronik zeichnet sich eine neue Tendenz zur dezentralen Steuerung ab.

b) Zentrale Steuerung

In rechnergesteuerten Vermittlungssystemen wird i.a. das zentrale Steuerungsprinzip angewandt, welches mit Hilfe zentral angeordneter Rechnerbaugruppen realisiert wird. Zustandsspeicherung (Abbild der peripheren Betriebsmittelzustände) und Vermittlungsprozeduren werden in der Zentralsteuerung durchgeführt. Die Signalisiervorgänge zum Informationsaustausch spielen sich nun im wesentlichen zwischen den einzelnen peripheren Betriebsmitteln und der Zentralsteuerung ab, wodurch die Steuerleitungssysteme zur Sammlung bzw. Verteilung von Anreizen bzw. Befehlen sowie die Ein/Ausgabewerke zu den zentralen Steuereinrichtungen von großer Wichtigkeit sind. Die Vermittlungsprozeduren für den Auf- und Abbau der Verbindungen werden durch Programme in der Zentralsteuerung realisiert, welche durch äußere Anreize aufgerufen und deren Ergebnisse in Form von Einstell-, An- und Abschaltbefehlen der Peripherie übergeben werden.

Infolge der Zentralisierung kann die Zentralsteuerung mit einem hohen Intelligenzgrad ausgestattet werden, wodurch u.a. auch "neue Leistungsmerkmale" der Vermittlungstechnik realisierbar sind; umgekehrt entstehen in derartigen Systemen neue Probleme hinsichtlich der verkehrsmäßigen Leistungsfähigkeit des "Zugriffs"- und Steuerungssystems sowie der Wirkbreite von Störungen. Zentralsteuerungen bilden insbesondere im Hinblick auf das Überlastverhalten ein sehr kritisches Glied; dynamisch wirkende Laststeuerungsmechanismen müssen hier eingesetzt werden, um die Leistungsfähigkeit bei Überlast aufrechtzuerhalten.

c) Mischformen aus dezentraler und zentraler Steuerung

Dezentrale und zentrale Steuerung werden in der Praxis häufig in einer Mischform kombiniert angewandt. Die heute noch vorherrschenden klassischen Vermittlungssysteme z.B. besitzen bereits eine Reihe teilzentralisierter bzw. zentralisierter Einrichtungen (Register, Umwerter). Die Erfahrungen bei der Entwicklung zentralgesteuerter Systeme haben dazu geführt, daß in neueren Konzeptionen die Steuerungsaufgaben sinnvollerweise auf dezentrale, teilzentrale und zentrale Einrichtungen verteilt werden.

2.2.3 Betriebsarten

Die Betriebsweise zentralgesteuerter Vermittlungssysteme ist hinsichtlich der Steuerung durch drei wesentliche Merkmale gekennzeichnet:

- Teilhaberbetrieb
- Realzeitbetrieb
- Mehrprozeßbetrieb

a) Teilhaberbetrieb

Die Beschreibung der dabei ablaufenden Vorgänge kann mit Hilfe eines vermittlungstechnischen Prozeßbegriffes erfolgen, vergl. etwa [5]. Dieser Prozeßbegriff beruht auf der Beschreibung der in der Peripherie erzeugten Zustandsänderungen (Teilnehmer, Sätze etc.) und den dadurch ausgelösten Anforderungen (Meldungen) an die Zentralsteuerung; die Meldungen lösen ihrerseits entsprechende Zustandsänderungen innerhalb der Zentralsteuerung aus, welche gegebenenfalls in Form von Befehlen an die Peripherie ausgegeben werden.

Die Folge von Anforderungen, welche ein Teilnehmer bzw. die von ihm belegten peripheren Betriebsmittel im Zuge des Auf- und Abbaus einer Verbindung erzeugt, heiße "äußerer Prozeß". Mit Eingabe der ersten Anforderung ("Abheben des Teilnehmers") wird ein zugehöriger "innerer Prozeß" initiiert. Der innere Prozeß enthält alle Zustandsdaten über den Teilnehmer und die von ihm bis dahin aufgebaute Verbindung. Die von den einzelnen Anforderungen des äußeren Prozesses aufgerufenen Vermittlungsprogramme teilen dieser Verbindung periphere Betriebsmittel zu (z.B. Wahlsatz, Internsatz etc.) und bringen die Zustandsdaten des Teilnehmers bzw. der ihm zugewiesenen Betriebsmittel auf den neuesten Stand.

Die aufgerufenen Vermittlungsprogramme sind nicht teilnehmerindividuell vorhanden, sondern stehen entweder allen Teilnehmern oder allen Teilnehmern einer bestimmten Gruppe (z.B. der Gruppe von Teilnehmern mit Kurzwahlmöglichkeit) zur Verfügung. Die Vermittlungsprogramme werden auf Anforderung der einzelnen äußeren Prozesse diesen zeitweise zugeteilt, d.h. alle äußeren Prozesse "haben

daran teil" (diesem Prinzip steht die Betriebsweise "Teilnehmerbetrieb" in Rechnersystemen gegenüber, bei welcher jeder Teilnehmer i.a. ein individuelles Programm besitzt).

b) Realzeitbetrieb

Die Verarbeitung äußerer Anreize unterliegt zeitkritischen Grenzen, in denen die Reaktion der Zentralsteuerung ausgeführt sein muß. Betriebsabläufe mit Vorschriften bezüglich der Fertigstellung von Aufträgen fallen allgemein unter den Begriff der "Realzeitverarbeitung" (real time processing), wie sie vor allem in der Prozeßrechenstechnik auftreten. Um die zeitlichen Forderungen der unterschiedlich wichtigen Anreize einzuhalten, wird die Verarbeitung nach einer Prioritätsstruktur gesteuert. Dies erfolgt durch Einteilung der Anforderungen in

- Ebenen unterbrechender Prioritäten (Unterbrechungsebenen)
- Klassen nichtunterbrechender Prioritäten innerhalb jeder Ebene.

Als Beispiele für Anforderungen unterbrechender Priorität seien genannt

- Taktgesteuerte Ein/Ausgabe (z.B. 10 msec-Takt)
- Schnelle Alarme zur Fehlerbehandlung

Klassen nichtunterbrechender Prioritäten können sein:

- Zeitkritische Zustandsänderungen (Wählziffern)
- Weniger zeitkritische Zustandsänderungen (Abheben, Auflegen)
- Routineprüfprogramme als "Hintergrundlast"

Zur Einhaltung der zeitkritischen Bedingungen muß das Betriebssystem außerdem eine Zeitverwaltung durchführen. Die Zeitverwaltung erfolgt mit Hilfe einer "Realzeituhr", welche in periodischen Abständen ein Zeitverwaltungsprogramm aufruft. Aufgaben der Zeitverwaltung sind beispielsweise die Ein/Ausgabe-Unterbrechungssteuerung sowie das Abmessen von Zeiten z.B. zur Erkennung von Endlosschleifen oder zur Einhaltung von Schutzzeiten.

c) Mehrprozeßbetrieb

Ein wesentliches Merkmal des Vermittlungsbetriebs ist der hohe Parallelitätsgrad gleichzeitig existenter äußerer und innerer Prozesse. Die Ausführung der einzelnen zu verschiedenen Prozessen gehörenden Verarbeitungsschritte erfolgt teilweise echt parallel (z.B. in Steuerwerken der teilzentralen Ebene) bzw. quasi-parallel wie z.B. im Zentralsteuerwerk durch zeitlich verschachtelte Folge (Zeitteilung). Die Steuerung derartiger parallel ablaufender Rechenprozesse fällt allgemein unter den Begriff der "Prozeßverwaltung" (process oder task management).

Die einzelnen Verwaltungsaufgaben der Zentralsteuerung unterliegen dem Organisationsprogramm ("Betriebssystem"). Das Betriebssystem hat dabei drei prinzipielle Aufgaben:

- Prozeßverwaltung
- Zeitverwaltung
- Verwaltung zentraler Betriebsmittel

Die zentrale Betriebsmittelverwaltung erstreckt sich im wesentlichen auf den Zentralprozessor, den Speicher sowie die gemeinsam benutzten Arbeitsprogramme. Die Speicherorganisation wird wesentlich durch die Erfordernisse des Realzeit- und Mehrprozeßbetriebs bestimmt. Zeitkritische Organisations-, Vermittlungs- und Fehlerbehandlungsprogramme sowie die umfangreichen Zustandsdaten der peripheren Betriebsmittel und aktiven Prozesse werden im Arbeitsspeicher abgelegt (resident), während die restlichen Programme auf einem langsameren Hintergrundspeicher liegen (nichtresident).

Die prinzipielle Betriebsweise der Zentralsteuerung eines rechnergesteuerten Vermittlungssystems hinsichtlich der Verarbeitung einer Anforderung A eines "äußeren Prozesses", des daraufhin ausgelösten Abschnittes des "inneren Prozesses" sowie der Systemreaktion R an die peripheren Betriebsmittel ist im nachfolgenden Bild 2.5 und der zugehörigen Legende dargestellt.

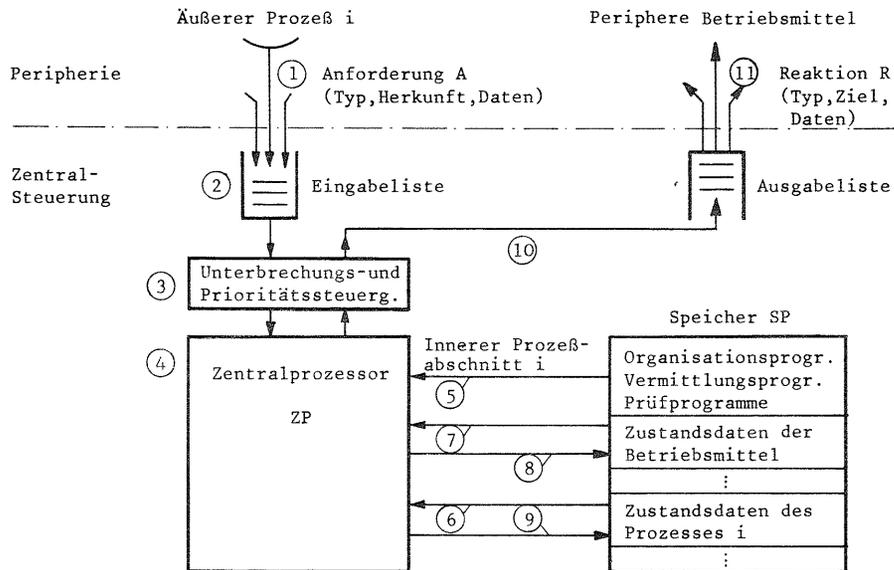


Bild 2.5. Prinzipielle Betriebsweise der Zentralsteuerung in rechnergesteuerten Vermittlungssystemen

Die prinzipielle Betriebsweise der Zentralsteuerung nach Bild 2.5 ist wie folgt:

- 1 Eintreffen einer Anforderung A des äußeren Prozesses i
- 2 Übernahme der Anforderung in eine Eingabeliste
- 3 Unterbrechung eines laufenden inneren Prozesses bzw. prioritätsgerechte Abfertigung bei Beendigung eines inneren Prozesses
- 4 Start des Ablaufs eines Abschnittes des inneren Prozesses i
- 5 Befehlsfolge des Organisationsprogrammes bzw. aufgerufenen Vermittlungs- oder Prüfprogrammes
- 6 } Bereitstellung der Zustandsdaten des inneren Prozesses i
- 7 } bzw. der peripheren Betriebsmittel
- 8 } Aktualisierung der Zustandsdaten nach Ablauf eines
- 9 } Vermittlungsprogrammes
- 10 Eintragen eines Befehls in die Ausgabeliste
- 11 Übergabe des Befehls (Reaktion R) an ein oder mehrere periphere Betriebsmittel.

2.2.4 Funktionsabläufe

Als Beispiel für einen Funktionsablauf werde der Aufbau einer Internverbindung zwischen einem Teilnehmer A und einem Teilnehmer B in einer rechnergesteuerten Vermittlungsstelle entsprechend Bild 2.4 betrachtet. Bild 2.6 zeigt die grundsätzlichen Signalisiervorgänge, welche zwischen den verschiedenen Betriebsmitteln ablaufen.

Die einzelnen Betriebsmittel werden dabei sehr unterschiedlich beansprucht (belegt), hinsichtlich Häufigkeit und Zeitbedarf, z.B.

- | | |
|-------------------|---|
| Wahlsätze: | für die Dauer des Wählvorganges |
| Verbindungssätze: | im wesentlichen für die Dauer des Gespräches |
| Zentralsteuerung: | mehrfach kurzzeitig zur Verarbeitung von Zustandsänderungen des Teilnehmers bzw. einzelnen Wählziffern. |

Entsprechend der schnellen Arbeitsweise zentralisierter Betriebsmittel wie ID, ES, EAW, ZST, werden diese im Zeitmultiplex mehreren oder allen aktiven Teilnehmern zugeteilt. Bei der Betriebsmittelvergabe kommt es daher vor, daß mehrere Anforderungen gleichzeitig anstehen. Das Zuteilungsproblem wird durch Pufferung gelöst: Die Anforderungen werden in Warteschlangen eingereiht und - entsprechend ihrer Wichtigkeit - nach einer hinsichtlich Ankunftszeitpunkt oder Anforderungstyp geordneten Reihenfolge seriell bzw. bei besonders zeitkritischen Verhältnissen nach unterbrechenden Prioritäten bedient. Die dabei auftretenden Wartezeiten verzögern die Abfertigung zusätzlich (sie wurden in Bild 2.6 aus Übersichtlichkeitsgründen nicht extra hervorgehoben).

2.2.5 Verkehrsprobleme

Die simultane Bedienung sehr vieler sich unabhängig voneinander verhaltender Teilnehmer wirft eine Reihe von Verkehrsproblemen auf. Die Verkehrsprobleme

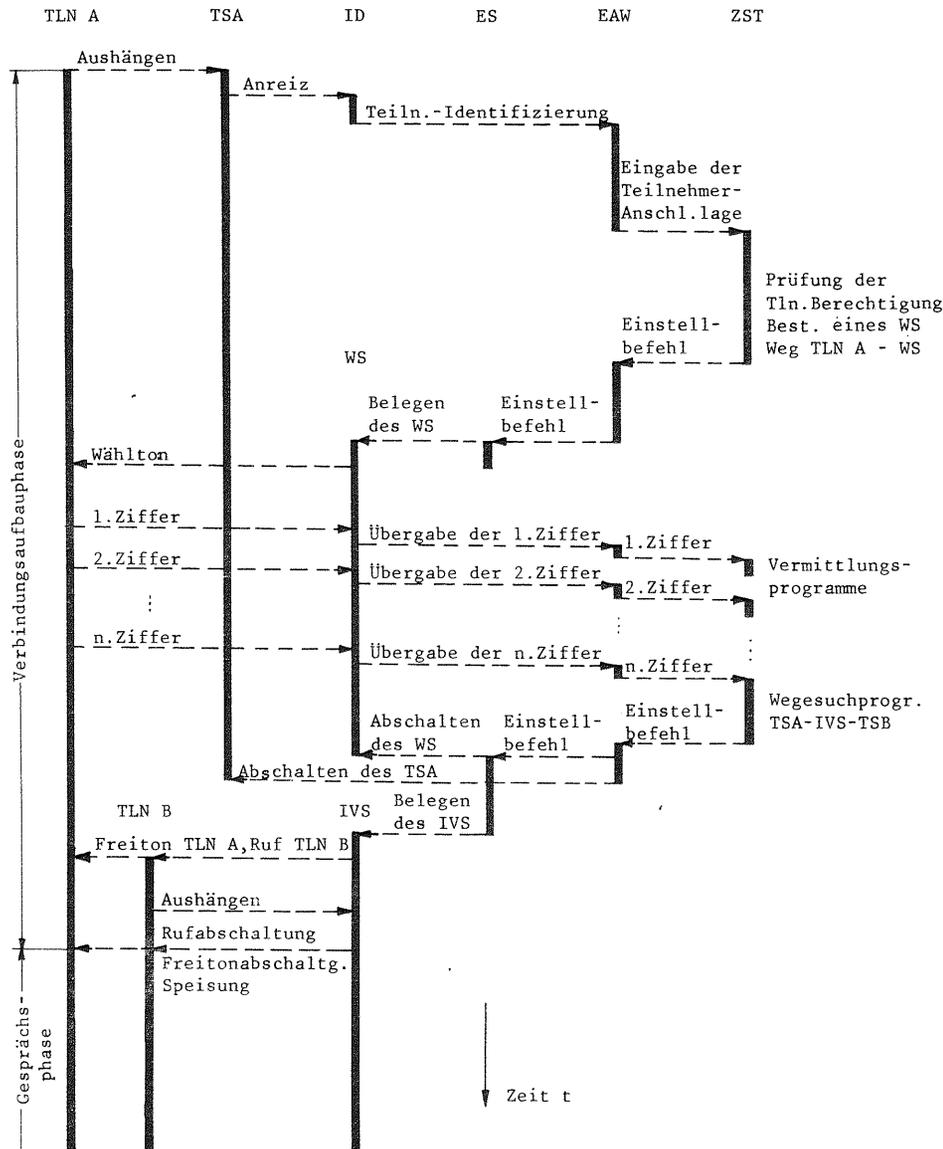


Bild 2.6. Prinzipieller Funktionsablauf beim Aufbau einer Internverbindung in einem rechnergesteuerten Vermittlungssystem nach Bild 2.4

werden in allen Ebenen des Vermittlungssystems sichtbar und sind bei der Auslegung zu berücksichtigen. Als Beispiele seien genannt:

a) Leitungsnetz

Die einzelnen Vermittlungsstellen sind über ein Netz aus Übertragungsleitungen miteinander verbunden. Die verkehrsmäßige Leistungsfähigkeit des Netzes hängt ab von dessen Struktur, den Bündelgrößen, den Absuchverfahren der Bündel, der Wegelenkung sowie den statistischen Eigenschaften der Ankunftsprozesse und der Belegungsdauern. Die verkehrsgerechte Auslegung des Leitungsnetzes ist ein klassisches Problem der Fernsprechverkehrstheorie.

b) Koppelnetz

Die Vermittlungsgüte des Koppelnetzes hängt ab von dessen Struktur, den Wegesuchverfahren sowie von den statistischen Eigenschaften der Belegungsversuche und Belegungsdauern. Die passende Dimensionierung des Koppelnetzes und seiner Abnehmerschaltglieder (Leitungen, Wahlsätze) gehört ebenfalls zu den klassischen Aufgaben der Fernsprechverkehrstheorie.

c) Teilzentrale Steuereinrichtungen

Die teilzentralen Steuereinrichtungen puffern die Anforderungen der Peripherie und übergeben diese der Zentralsteuerung. Umgekehrt werden die Befehle der Zentralsteuerung zwischengepuffert, bevor sie den peripheren Empfängern zugeführt werden. Der Informationsaustausch mit der Zentralsteuerung kann auf verschiedene Art und Weise erfolgen: a) taktgesteuert, b) unterbrechungsgesteuert, c) zyklische Abfrage. Die Leistungsfähigkeit hängt ab von der Art des Informationsaustausches sowie von den statistischen Eigenschaften der Meldungen bzw. Befehle.

d) Zentrale Steuereinrichtungen

Die Zentralsteuerung arbeitet die Anforderungen der Peripherie entsprechend ihrer Dringlichkeit ab. Ihre Leistungsfähigkeit hängt u.a. ab von der Ablauforganisation der Programme und Ein/Ausgabevorgänge, der Prioritätsstruktur, den Programmlaufzeiten sowie von der Statistik der Anforderungen. Die Leistungsfähigkeit wird ferner beeinflusst von den Sicherungseinrichtungen in Hardware (Mehrrechnerkonzepte mit Mikrosynchronismus, Lastteilung oder Reserveschaltung) und Software (Programme zur Prüfung, Fehlerdiagnose und Ersatzschaltung).

Die angegebenen Beispiele charakterisieren typische Verkehrsprobleme innerhalb verschiedener Ebenen von Vermittlungssystemen. Unter genauerer Berücksichtigung der speziellen Gegebenheiten läßt sich eine Vielzahl weiter konkretisierter Teilprobleme angeben.

Als Beispiel für das Ablaufgeschehen in Nachrichtenvermittlungssystemen wurde im vorangehenden ein rechnergesteuertes Fernsprechvermittlungssystem betrachtet. Entsprechende Überlegungen können auch für Datenvermittlungssysteme angestellt werden, welche entweder nach dem Durchschalte- oder nach dem Speicher- vermittlungsprinzip arbeiten. Aus Umfangsgründen wird hier von einer ausführ- licherer Diskussion abgesehen und auf das Schrifttum verwiesen, vergl. z.B. [6-7].

2.3 Beispiel 2: Rechnersysteme mit hierarchischem Speicherprinzip

Heutige Rechnersysteme sind durch eine Vielzahl unterschiedlicher Funktionsteile (Baugruppen, Betriebsmittel) charakterisiert. Während des Betriebs werden diese Funktionsteile den einzelnen Programmen zeitweise oder ganz zugeteilt, wodurch ein umfangreicher Austausch von Nutz- und Steuerinformationen entsteht. Als Bei- spiel hierfür wird im folgenden ein Rechnersystem mit hierarchischem Speicher- prinzip betrachtet. Dieses Beispiel dient gleichzeitig als Grundlage für die spätere Modellbildung und Modellanalyse (Kapitel 3 und 6).

2.3.1 Architektur

Bild 2.7 zeigt die prinzipielle Konfiguration eines Großrechnersystems. Sie läßt sich grob unterteilen in zwei Ebenen:

- a) Zentraleinheit
- b) Peripherie.

Zur detaillierteren Diskussion werde auf das einschlägige Schrifttum verwiesen, vergl. z.B. [8-12].

a) Zentraleinheit

Die Zentraleinheit besteht aus dem Zentralprozessor ZP (central processing unit, CPU), dem Arbeitsspeicher ASP, dem Ein/Ausgabe-Prozessor EAP mit den angeschlos- senen Schnellkanälen SK bzw. Standard-(Multiplex-)Kanälen MK sowie einem zentral- len Leitungssystem (BUS), über das die verschiedenen Baugruppen der Zentralein- heit Informationen austauschen können.

Der Zentralprozessor umfaßt im wesentlichen das Befehlswerk BW und das Rechen- werk RW. Das Befehlswerk organisiert die Auswahl und den Transport der Befehle und zugehörigen Daten eines Programmes, während das Rechenwerk für die Ausführung der Befehle selbst zuständig ist. Die Befehle und Daten stehen im Arbeitsspeicher, auf den über das zentrale Leitungssystem zugegriffen wird. Zur beschleunigten Befehlsausführung im Zentralprozessor wird häufig ein lokaler Schnellspeicher SSP eingesetzt; in diesem Falle muß dafür gesorgt werden, daß sich nach Möglichkeit

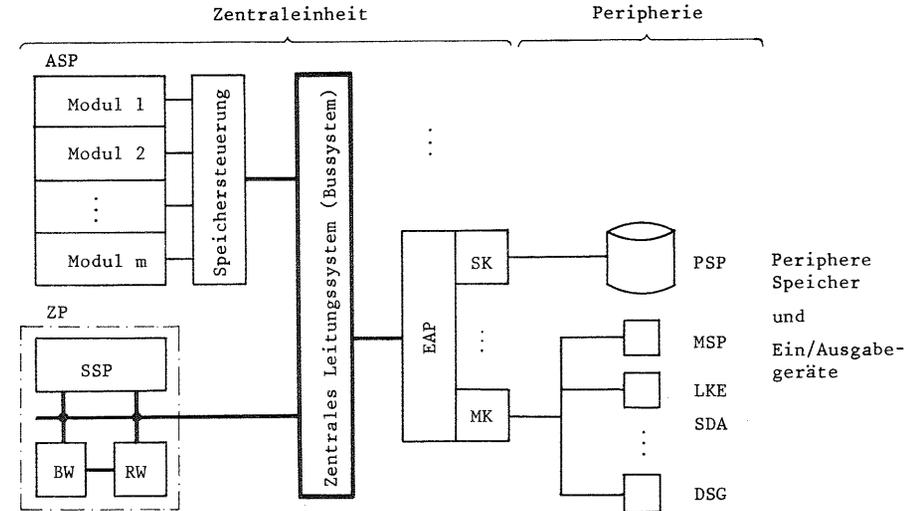


Bild 2.7. Konfiguration eines Rechnersystems

ASP	Arbeitsspeicher	EAP	Ein/Ausgabe-Prozessor
SSP	Schnellspeicher	SK	Schnellkanal
PSP	Plattenspeicher	MK	Multiplexkanal
ZP	Zentralprozessor	MSP	Magnetbandspeicher
RW	Rechenwerk	LKE	Lochkarteneingabe
BW	Befehlswerk	SDA	Schnelldruckerausgabe
KW	Kanalwerk	DSG	Datensichtgerät

stets die aktuellen Teile eines Programmes (Befehle und Daten) in diesem Schnell- speicher befinden, damit die Häufigkeit für "Fehlzugriffe" (welche einen lang- sameren ASP-Zugriff erfordern) niedrig bleibt. Bei virtueller Adressierung (siehe 2.3.4) umfaßt der Zentralprozessor ferner die Einrichtungen zur dynamischen Adreßumsetzung.

Der Arbeitsspeicher ASP ist häufig unterteilt in mehrere Modulen, auf welche parallel zugegriffen werden kann. Konfliktsituationen, welche beim Zugriff auf den Arbeitsspeicher durch andere Einheiten (ZP, EAP) entstehen können, werden durch ein Vorrangwerk in der Speichersteuerung gelöst.

Zur Initiierung, Ein- und Ausgabe von Programmen, Programmteilen und Daten ist ein Informationsaustausch zwischen Zentraleinheit und Peripherie erforderlich. In heutigen Systemen erfolgt dies über Kanalwerke, welche parallel zum Zentral- prozessor arbeiten können. Ein Ein/Ausgabevorgang wird durch das Betriebssystem

(siehe 2.3.3) "angestoßen", indem der Ein/Ausgabe-Prozessor EAP ein - ebenfalls im Arbeitsspeicher abgelegtes - Kanalprogramm verarbeitet; die eigentliche Ein/Ausgabe erfolgt danach über die angeschlossenen Kanäle unter der Regie des Ein/Ausgabeprozessors.

b) Peripherie

Die Peripherie des Rechnersystems umfaßt verschiedene Ein/Ausgabegeräte (LKE, SDA, DSG usw.), die Bedienungskonsole sowie externe Massenspeicher ("Hintergrundspeicher") wie Plattenspeicher PSP und Magnetbandspeicher MSP.

c) Konzeption der Speicherhierarchie

Die Architektur heutiger Rechnersysteme ist durch technologische wie auch durch verkehrsmäßige Gegebenheiten bedingt. Die Entwicklung auf dem Gebiet der Mikroelektronik ermöglicht sehr schnelle Verarbeitungseinheiten (Rechen- und Befehlswerke). Zur Anpassung an die Geschwindigkeit dieser schnellen Verarbeitungseinrichtungen wird eine nach Zugriffszeiten und Speicherkapazität gestaffelte Speicherhierarchie angewendet:

- Schnellspeicher (SSP)
Elektronische Speicher ("cache") mit kleinen Zugriffszeiten (z.B. 100-500 nsec) und relativ kleiner Kapazität (z.B. bis zu einigen Tausend Worten)
- Arbeitsspeicher (ASP)
Elektronische oder magnetische Speicher mit Direktzugriff (Zugriffszeiten z.B. 500-1000 nsec) und mittelgroßer Kapazität
- Hintergrundspeicher (HSP)
Magnetische Speicher mit mechanischem Zugriff (Platte, Trommel, Band) mit relativ großer Zugriffszeit (z.B. 10 msec und mehr) und sehr großer Kapazität

Eine derartige Speicherhierarchie ermöglicht eine nahezu optimale Ausnutzung, wenn die jeweils aktuellen Teile von Programmen bzw. Daten in den unteren Speicherebenen (SSP, ASP) verfügbar gemacht werden können.

2.3.2 Steuerungsprinzipien

Die Organisation heutiger Großrechnersysteme erfolgt nach einem hierarchischen Prinzip, bei welchem Steuerungsebenen schalenförmig aufeinander aufbauen. Im Prinzip erfolgt die Organisation in 3 Ebenen:

- a) Verdrahtete (Hardware-)Steuerwerke
- b) Mikroprogrammierte Steuerwerke
- c) Systemprogramme (Betriebssystem)

a) Verdrahtete (Hardware-)Steuerwerke

Hardware-Steuerwerke sind "festverdrahtete" Funktionsgeneratoren, welche auf Anstoß eine bestimmte und nicht veränderbare Folge von Operationen ausführen. Sie bilden die unterste Steuerungsebene und sind Bestandteil aller Baugruppen. Als Beispiel sei die Steuerung zur dynamischen Adreßumsetzung bei virtueller Adressierung angeführt, vergl. 2.3.4 und entsprechende Literatur, z.B. [13].

b) Mikroprogrammierte Steuerwerke

Die Ausführung der Befehle eines Anwender- oder Systemprogrammes (Makro- oder Maschinenbefehl) erfolgt durch eine Folge von "Mikrobefehlen". Beispielsweise wird die Funktion des Zentralprozessors in eine Reihe von Einzeloperationen zerlegt, welche zur beschleunigten Ausführung oft nach dem Fließbandprinzip (pipelining) organisiert sind. Es können dann z.B. die Einzeloperationen "Befehl Holen", "Adreßrechnung", "Operand Holen", "Befehl Ausführen" für nacheinander anstehende verschiedene Befehle zeitlich überlappt ablaufen, wenn für die Ausführung dieser Einzeloperationen entsprechende Hardware-Funktionsblöcke bereitstehen [14]. Diese Einzeloperationen werden von einem Mikroprogramm gesteuert. Die Mikroprogramme befinden sich in einem speziellen, der Baugruppe zugeordneten Mikroprogramm Speicher und können gewechselt werden, wodurch eine wesentlich größere Flexibilität erzielt wird.

c) Systemprogramme (Betriebssystem)

Unter dem "Betriebssystem" wird die Gesamtheit von (System)Programmen verstanden, welche den Ablauf von Anwenderprogrammen unterstützen. Entsprechend ihrer Funktion bilden die Systemprogramme selbst ein hierarchisch aufgebautes Programmsystem. Die wesentlichsten Teile bzw. Aufgaben eines Betriebssystems sind:

1. Dienstprogramme zur Unterstützung des Programmierens
 - Sprachübersetzung
 - Ein/Ausgabe
 - Dateiaufbereitung
 - Einbinden von Bibliotheksprogrammen u.a.m.
2. Organisationsprogramme zur Steuerung des Programmablaufs
 - Aktivierung und Terminierung von Programmen
 - Speicherverwaltung
 - Verklemmungsfreie und zeitgerechte Betriebsmittelzuteilung
 - Optimierung des Betriebsablaufs nach Gesichtspunkten der Betriebsmittelausnutzung bzw. Benutzererfordernisse

Die wichtigeren Organisationsprogramme sind stets im Arbeitsspeicher geladen (residente Systemprogramme), während die weniger wichtigen Dienst- und Organisationsprogramme bei Bedarf vom Hintergrundspeicher HSP aus zugeladen werden (nichtresidente Systemprogramme).

2.3.3 Betriebsarten

Im folgenden werden die wichtigsten Betriebsarten angeführt, welche in den Organisationsprogrammen realisiert sind und welche einen wesentlichen Einfluß auf das Ablaufgeschehen und die Leistungsfähigkeit eines Rechnersystems haben. Dabei sollen unterschieden werden:

- Simultane Ein/Ausgabe
- Mehrprogrammbetrieb
- Mehrprozessorbetrieb
- Stapelverarbeitung
- Dialogbetrieb
- Virtuelle Speicherung

Zur ausführlicheren Diskussion wird auf das Schrifttum verwiesen, z.B. [10,15,16].

a) Simultane Ein/Ausgabe

Voraussetzung für eine wirkungsvolle Ausnutzung der Systemeinheiten ist die zeitlich überlappte Arbeit verschiedener Systemkomponenten, insbesondere zwischen Zentralprozessor ZP (Verarbeitung) und Kanälen (Ein/Ausgabe). Ermöglicht wird dies durch den Ein/Ausgabe-Prozessor EAP, welcher parallel zum Zentralprozessor ZP ein Kanalprogramm abwickeln kann. Das Kanalprogramm wird von einem Systemprogramm aufgrund des E/A-Befehles konstruiert, im ASP abgelegt und gestartet. Bei virtuellem Speicher (vergl. 2.3.4) muß das Kanalprogramm außerdem von den virtuellen in die physikalischen ASP-Adressen übersetzt werden. Ferner müssen zur Ein/Ausgabe im Arbeitsspeicher Pufferbereiche bereitgestellt werden, welche für die Dauer der Ein/Ausgabe fest zugeweiht bleiben (Fixierung).

b) Mehrprogrammbetrieb

Die befehls-sequentielle Abarbeitung von Anwenderprogrammen bedingt, daß ein Anwenderprogramm i.a. nur ein Betriebsmittel gleichzeitig belegen kann. Simultane Arbeit ist dann nur möglich, wenn an mehreren Anwenderprogrammen parallel gearbeitet werden kann. Während beispielsweise ein Anwenderprogramm den Zentralprozessor belegt, kann ein zweites eine Eingabe und ein drittes eine Ausgabe zwischen Arbeits- und Hintergrundspeicher abwickeln. Voraussetzung hierfür ist die Aufteilung des Arbeitsspeichers ASP unter mehrere Anwenderprogramme. Das simultane Abarbeiten mehrerer Anwenderprogramme wird als Mehrprogrammbetrieb (Multiprogramming) bezeichnet; die Anzahl gleichzeitig aktivierter und in den Arbeitsspeicher geladener Anwenderprogramme heißt Multiprogramminggrad. Ist nur ein Anwenderprogramm geladen, spricht man von Monoprogramming.

In diesem Zusammenhang treten zwei grundsätzliche Probleme auf:

- Speicherplatzzuteilung
- Adressierung

Die Probleme sollen kurz anhand eines Beispiels nach Bild 2.8 diskutiert werden.

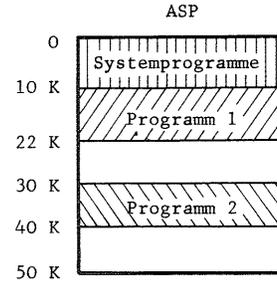


Bild 2.8. Arbeitsspeicher-Belegung bei Multiprogramming

Systemprogramme: 10 K Worte
 Programm 1 : 12 K Worte
 Programm 2 : 10 K Worte
 ASP-Kapazität : 50 K Worte
 (1 K $\hat{=}$ 1024)

In Bild 2.8 ist eine angenommene Arbeitsspeicher-Belegung bei Multiprogramming skizziert. Dabei umfasse der residente Teil des Betriebssystems 10 K Worte, ferner seien zwei Anwenderprogramme mit 12 K bzw. 10 K Worten geladen. Bei einer angenommenen ASP-Kapazität von 50 K Worten sind noch 18 K Worte frei. Ein drittes Anwenderprogramm mit beispielsweise 15 K Worten könnte demnach nicht geladen werden infolge Speicherfragmentierung.

Anwenderprogramme werden i.a. unabhängig von einer speziellen Rechnerkonfiguration oder einer speziellen ASP-Belegungssituation geschrieben und sind relativ adressiert. Vor Ausführung eines Programmes muß dieses zunächst in einen noch freien ASP-Bereich geladen werden. Zur Abarbeitung der Befehlsfolge dieses Programmes müssen seine absoluten (realen) Adressen bekannt sein. Dies kann entweder im voraus beim Ladevorgang (static load-time relocation) oder dynamisch während der Programmausführung mit Hilfe eines Basisadressregisters (dynamic relocation) erfolgen.

Die beschriebene Arbeitsweise weist einen entscheidenden Nachteil auf, der in der "Speicherzerstückelung" liegt. Um den Speicher dennoch besser auszunutzen, wären Umspeichervorgänge erforderlich ("garbage collection"). Ein weiterer prinzipieller Nachteil besteht darin, daß Anwenderprogramme entweder ganz geladen werden müssen bzw. daß - bei Segmentierung - der Anwender in seinem Programm die Ein- und Ausgabe von Programmsegmenten selbst organisieren muß. Beiden Nachteilen wird mit dem unter Abschnitt 2.3.4 beschriebenen "Virtuellen Speicherprinzip" begegnet.

c) Mehrprozessorbetrieb

Die bisher beschriebene Einprozessor-Betriebsweise kann erweitert werden auf eine Mehrprozessorbetriebsweise (multiprocessing), wenn mehrere Zentralprozessoren eingeführt werden, welche parallel an verschiedenen Programmen arbeiten. Zeitliche Engpässe können dabei u.a. bei Speicherzugriffen entstehen; diesen wird i.a. durch Speichermodularisierung begegnet, wodurch parallele Speicherzugriffe ermöglicht werden. Die Organisation von Mehrprozessorsystemen erfordert einen wesentlich höheren Aufwand im Vergleich zu Einprozessorsystemen.

d) Stapelverarbeitung

Unter Stapelverarbeitung (batch processing) wird die Abarbeitung von Anwenderprogrammen verstanden, bei welcher keine strengen Zeitanforderungen hinsichtlich der Fertigstellung gestellt sind. Es wird dabei versucht, die Betriebsmittel möglichst gut auszunutzen, um einen hohen Durchsatz zu erzielen. Die übliche Form der Stapelbetriebsweise ist im folgenden Bild 2.9 schematisch dargestellt:

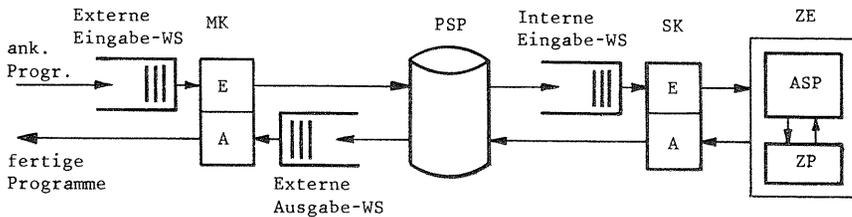


Bild 2.9. Stapelbetriebsweise

MK	Multiplexkanal	ZE	Zentraleinheit	WS	Warteschlange
SK	Schnellkanal	ASP	Arbeitsspeicher		
PSP	Plattenspeicher	ZP	Zentralprozessor		

Die angelieferten Programme bilden zunächst eine externe Warteschlange ("Stapel"). Vor der eigentlichen Verarbeitung werden sie über ein Eingabegerät und einen Multiplexkanal MK auf dem Plattenspeicher PSP zwischengespeichert. Sie sind nun dem System bekannt und bilden eine interne Warteschlange. Die interne Warteschlange wird durch ein spezielles Systemprogramm (job initiator) abgearbeitet, indem die Programme vom Plattenspeicher PSP in den Arbeitsspeicher ASP über einen Schnellkanal SK geladen werden. Die Ergebnisse der fertig bearbeiteten Programme werden ebenfalls auf dem Plattenspeicher PSP zwischengespeichert, bevor sie über das Ausgabegerät ausgegeben werden. Die Ein/Ausgabe über den Plattenspeicher läuft dabei parallel zur Arbeit der Zentraleinheit ZE ab, wobei der Zentralprozessor ZP nur kurzzeitig durch Unterbrechungssignale hoher Priorität angefordert wird (Systemprogramme für Lesen (reader) bzw. Drucken (writer)). Das beschriebene

Verfahren wird als "spooling" bezeichnet (simultaneous peripheral output on-line). Der Stapelbetrieb wird i.a. kombiniert mit Multiprogramming ausgeführt. Eine weitere Variante ermöglicht die Ein/Ausgabe von entfernten Datenendstationen (remote batch processing).

e) Dialogbetrieb

Dialogbetrieb liegt vor, wenn der Benutzer über angeschlossene Ein/Ausgabestationen ("Terminals") interaktiv mit dem Rechnersystem verkehren kann. Diese Verarbeitungsform ist durch den Dialog Benutzer-Maschine gekennzeichnet; dieser kann entweder programm- oder benutzerkontrolliert ablaufen. Verkehren die Benutzer dabei vorwiegend nur mit individuellen Programmen, so spricht man auch von "Teilnehmerbetrieb". Im Gegensatz zum Stapelbetrieb kommt es beim Dialogbetrieb wesentlich auf die Einhaltung bestimmter Reaktionszeiten (Antwortzeiten) an.

Die primäre Aufgabe des Betriebssystems ist es, eine größere Anzahl gleichzeitig aktiver Teilnehmer zeitgerecht zu bedienen. Dies wird durch das Zeitscheibenprinzip (time sharing) erreicht, bei welchem der Zentralprozessor in periodischer Wiederkehr zeitweise den einzelnen Benutzerprogrammen zugeteilt wird (z.B. 100 msec-Zeitscheiben/akt. Teilnehmer). Mit diesem Prinzip wird gleichzeitig eine sich an der Rechenzeit eines Programmes ausrichtende Priorität realisiert: Kürzere Programme werden entsprechend schneller abgearbeitet als längere. Im Grenzfall verschwindend kleiner Zeitscheiben erhält man das sog. "processor sharing": Sind n Teilnehmer gleichzeitig aktiv, so wird jedem Teilnehmer ein eigener Prozessor mit einer um den Faktor n erniedrigten Arbeitsgeschwindigkeit vorgetäuscht. Infolge der notwendigen Systemverwaltungszeiten ist die tatsächliche Leistung jedoch weit geringer.

Neben der Zeitteilung tritt bei größerer Anzahl aktiver Teilnehmer das Problem der Raumteilung im Arbeitsspeicher auf. Können nicht alle Programme gleichzeitig im Arbeitsspeicher Platz finden, wird die zeitweise Auslagerung auf den Hintergrundspeicher angewandt ("swapping"). Dies kann erfolgen entweder bei Erreichen eines Zeitscheibenendes und/oder bei Ende einer "Transaktion", wenn ein vom Teilnehmer aufgerufenes Programm beendet ist und die Kontrolle an das Terminal zurückgegeben wird bzw. der Teilnehmer in eine längere "Denkphase" eintritt. Das Speicherproblem wird heute vielfach in Kombination mit dem Virtuellen Speicher gelöst (vergl. 2.3.4).

2.3.4 Virtuelles Speicherprinzip

a) Aufgabenstellung und Voraussetzungen

Eine wesentliche Aufgabe des Betriebssystems besteht in der Verwaltung des Betriebsmittels "Speicher". Wie schon unter 2.3.3 b) ausgeführt, kann durch die Mehrprogramm-Betriebsweise das Problem der Speicherfragmentierung auftreten. Vor der Einführung des unten beschriebenen virtuellen Speicherprinzips wurde diesem Problem durch eine vom Benutzer ausgeführte "Segmentierung" begegnet: Der Programmierer unterteilt sein Programm in Segmente, welche für sich bearbeitbar sind und legt die Anweisungen zum Ein- und Ausspeichern von Programmsegmenten im Programm fest ("overlay structure"). Mit diesem Prinzip kann der Arbeitsspeicher besser ausgenutzt werden. Ferner ist es möglich, Programme zu bearbeiten, welche als ganzes die Arbeitsspeicher-Kapazität übersteigen. Hiermit ist auch ein Mehrprogrammbetrieb zur parallelen Nutzung von Betriebsmitteln durchführbar.

Das virtuelle Speicherprinzip geht von folgenden Zielen aus:

- Bereitstellung eines von der Größe des Arbeitsspeichers unabhängigen, zusammenhängenden "virtuellen" Adreßraums für jeden Benutzer
- Ausnutzung von großer Speicherkapazität und schneller Zugriffszeit, welche ein hierarchisches Speicherprinzip bietet (Schnellspeicher - Arbeitsspeicher - Hintergrundspeicher)
- Automatisierung der Segmentierungsaufgabe und des dazugehörigen Datentransportes zwischen den verschiedenen Ebenen der Speicherhierarchie.

Voraussetzungen für die Realisierung des virtuellen Speicherprinzips sind:

- "Lokalitätseigenschaften" von Anwenderprogrammen, welche eine vom System vorgenommene Segmentierung erfolgreich erscheinen lassen. Unter "Lokalitätsverhalten" soll dabei die Eigenschaft verstanden werden, wenn sich die Zugriffe von Programmen abschnittsweise jeweils auf eine beschränkte Teilmenge des Adreßraumes (die sog. momentane "Lokalität") konzentrieren.
- Schnelles Transportsystem zwischen den Ebenen der Speicherhierarchie, durch welches die aktuellen Programmteile schnell verfügbar gemacht werden können
- Automatische Speicherplatzverwaltung durch das Betriebssystem
- Einrichtungen zur effektiven Adreßumsetzung. Unter Adreßumsetzung soll dabei die Ausführung der Abbildungsfunktion (mapping) zwischen virtuellem Adreßraum und physikalischem Adreßraum verstanden werden. Der physikalische Adreßraum bezieht sich dabei auf die realen Adressen in Schnell-, Arbeits- und Hintergrundspeicher.

b) Abbildung des virtuellen Speichers auf physikalische Speicher

Durch das virtuelle Speicherprinzip erhält jedes Benutzerprogramm einen zusammenhängenden Adreßraum mit den Adressen $A = 1, 2, \dots, N$, wobei N den maximalen Adreßumfang bezeichnen möge. Die Abarbeitung eines Programmes erfolgt in diesem "virtuellen" Adreßraum, wobei zur Ausführung jeweils die physikalische Speicheradresse ermittelt werden muß. Die prinzipiellen Zusammenhänge sind im folgenden Bild 2.10 skizziert:

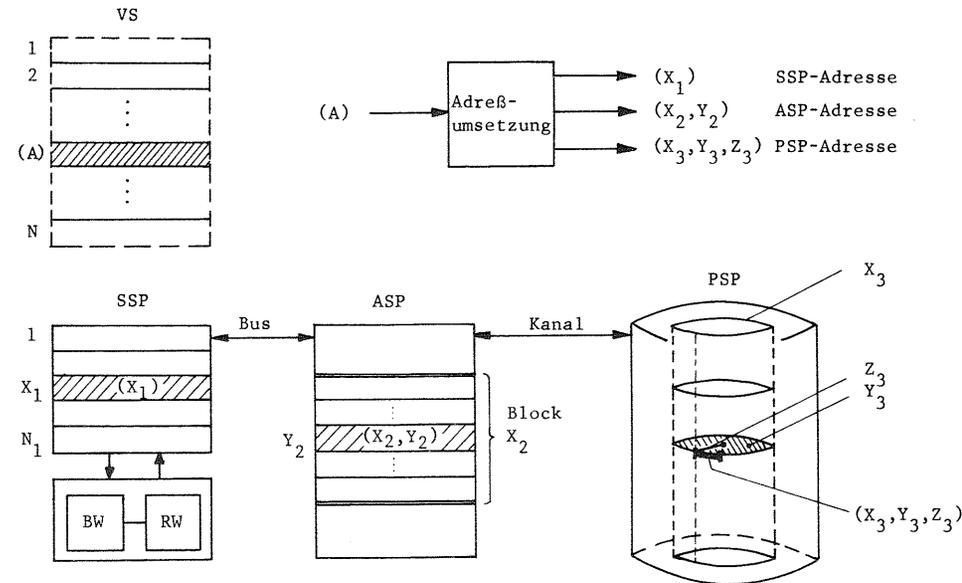


Bild 2.10. Abbildung von virtuellem Adreßraum auf physikalischen Adreßraum

VS	Virtueller Speicher	BW	Befehlswerk
SSP	Schnellspeicher	RW	Rechenwerk
ASP	Arbeitsspeicher		
PSP	Plattenspeicher		

In Bild 2.10 wird von einer 3-stufigen Speicherhierarchie ausgegangen bestehend aus verschiedenen Speichermedien mit sog. Direktzugriff, welche hinsichtlich Kapazität und Zugriffszeit gestaffelt sind:

- Schnellspeicher SSP
- Arbeitsspeicher ASP
- Plattenspeicher PSP

Das Bild 2.10 zeigt den Zusammenhang zwischen der virtuellen Adresse A eines Programmes und den physikalischen Adressen, wobei angenommen ist, daß das Wort, auf das sich die virtuelle Adresse A bezieht, sich gleichzeitig in allen drei physikalischen Speichern befinden möge:

- Schnellspeicher: Adresse (X_1) X_1 : Position innerhalb des SSP
- Arbeitsspeicher: Adresse (X_2, Y_2) X_2 : Block-Nummer
 Y_2 : Position innerhalb eines Blockes
- Plattenspeicher: Adresse (X_3, Y_3, Z_3) X_3 : Zylindernummer (Spur)
 Y_3 : Oberflächennummer
 Z_3 : Position innerhalb einer Spur (Sektor)

Der Virtuelle Speicher eines Programmes wird vollständig auf dem Hintergrundspeicher (Plattenspeicher PSP oder Trommelspeicher) abgebildet. Im Arbeitsspeicher ASP bzw. im Schnellspeicher SSP befinden sich in der Regel nur Teile des Programmes. Die Reihenfolge der Umrechnung von virtueller Adresse A erfolgt stufenweise entsprechend folgendem Bild 2.11:

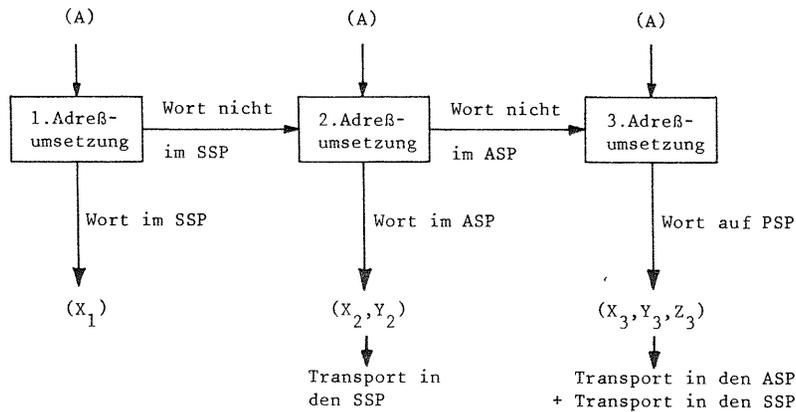


Bild 2.11. Stufen der Adreßumsetzung

Wird in den unteren Speicherebenen das zu einer Adresse gehörende Wort nicht gefunden, so muß ein Ergänzungstransport stattfinden. Entsprechend der Zugriffszeit zu den höheren Speicherebenen und deren Organisation werden die Ergänzungstransporte blockweise vorgenommen. Dies soll im folgenden am Beispiel des heute üblichen Paging-Verfahrens näher erläutert werden.

c) Das Paging-Verfahren

Die bekannteste Realisierungsform im Zusammenhang mit dem Virtuellen Speicher ist das Seitenteilungs- oder Paging-Verfahren. Um das Prinzipielle hervorzuheben, soll im folgenden dieses Verfahren für eine zweistufige Speicherhierarchie, bestehend aus Arbeitsspeicher ASP und Plattenspeicher PSP, beschrieben werden. Abgesehen von einigen Besonderheiten (kleinere Blockgrößen, paralleles "Durchschreiben" auf SSP und ASP u.a.m.), verlaufen die Vorgänge zwischen einem Schnellspeicher SSP und dem Arbeitsspeicher ASP prinzipiell in ähnlicher Weise.

Beim Paging-Verfahren wird der virtuelle Adreßraum des Benutzers vom System in Blöcke gleicher Größe (z.B. 2K Bytes), die sog. Seiten (pages), unterteilt. Entsprechend wird auch der physikalische Arbeitsspeicher in sog. Seitenrahmen (page frames, "Kacheln") unterteilt und die Plattenspeicher-Organisation auf der Blockgröße von 1 Seite aufgebaut. Die virtuelle Adresse A setzt sich aus zwei Teilen zusammen, der Seitenadresse i und der Position ("Distanz") d innerhalb einer Seite. Die zwei Stufen der Adreßumsetzung sind in Bild 2.12 skizziert:

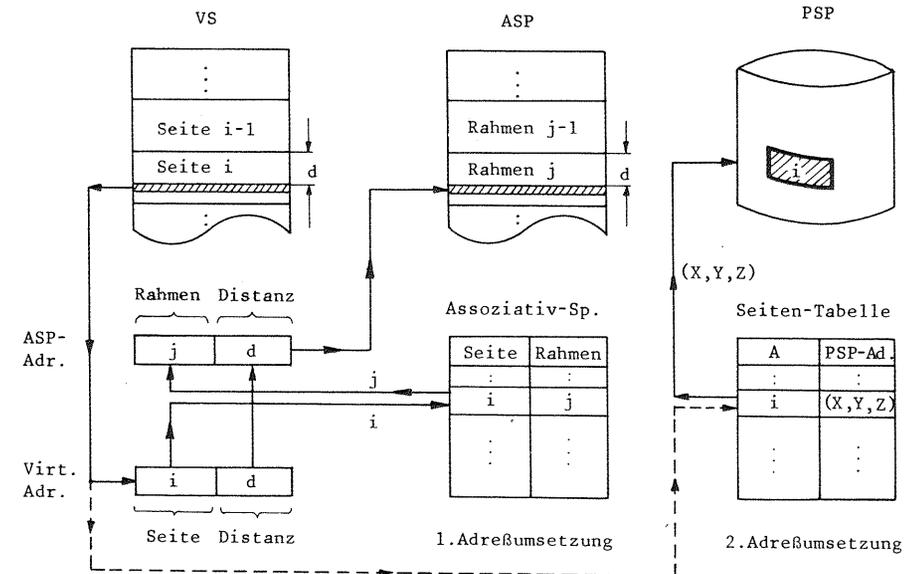


Bild 2.12. Adreßumsetzung beim Paging-Verfahren

- VS Virtueller Speicher
- ASP Arbeitsspeicher
- PSP Plattenspeicher

Die 1. Adreßumsetzung erfolgt dynamisch, d.h. zum Zeitpunkt jeder einzelnen Adreßreferenz. Da dieser Umsetzungsvorgang die Ausführung nicht verzögern sollte, wird die 1. Adreßumsetzung rein hardwaremäßig mit Hilfe eines Assoziativ-Speichers vorgenommen. Der Assoziativ-Speicher enthält eine Zuordnungstabelle zwischen den Seiten, welche sich momentan im Arbeitsspeicher befinden, und den zugehörigen Seitenrahmen. Aufgrund einer angebotenen Seitennummer *i* findet der Assoziativ-Speicher die Rahmennummer *j*, welche in Zusammensetzung mit der Distanz *d* die physikalische ASP-Adresse ergibt, zu der die virtuelle Adresse *A* gehört. In der Zuordnungstabelle des Assoziativ-Speichers sind i.a. noch weitere (in Bild 2.12 nicht dargestellte) Kenn-Bits angegeben, welche die Anwesenheit der Seite im ASP (presence bit), deren jüngste Benutzungshäufigkeit (activity bits) sowie deren Zustand hinsichtlich Veränderung (change bit) kennzeichnen.

Die 2. Adreßumsetzung wird aufgerufen, falls die Seite nicht im Arbeitsspeicher gefunden wurde (Fehlseitenzugriff, "page fault"). Das Programm wird unterbrochen und ein entsprechendes Systemprogramm gestartet. Dieses Systemprogramm verwaltet eine softwaremäßig realisierte Seiten-Tabelle, aus der die Position der Fehlseite auf dem Plattenspeicher PSP entnommen werden kann. Es wird ein entsprechendes Kanalprogramm konstruiert, welches von einem Ein/Ausgabe-Prozessor abgewickelt wird; der Zentralprozessor kann währenddessen einem anderen Anwenderprogramm zugeteilt werden.

In der Praxis wird - in Erweiterung zu den beschriebenen Verfahren - die virtuelle Adresse *A* i.a. dreidimensional angegeben. Hierzu wird der virtuelle Adreßraum zunächst einmal in Segmente unterteilt. Eine virtuelle Adresse setzt sich dann zusammen aus Segment-Nummer, Seiten-Nummer innerhalb eines Segmentes sowie der Distanz. Die Umsetzungsvorgänge verlaufen prinzipiell jedoch in gleicher Weise.

d) Speicherverwaltungs-Strategien

Das Paging-Verfahren wirft eine Reihe von Problemen auf hinsichtlich des Zeitpunktes eines Seitentransfers sowie hinsichtlich der ASP-Verwaltung. Die verschiedenen Verfahren sollen im folgenden unter dem Begriff der Speicherverwaltungs-Strategien (memory management) kurz behandelt werden.

Die Strategien zur Speicherverwaltung lassen sich grob nach folgendem Schema einteilen, vergl. Bild 2.13.

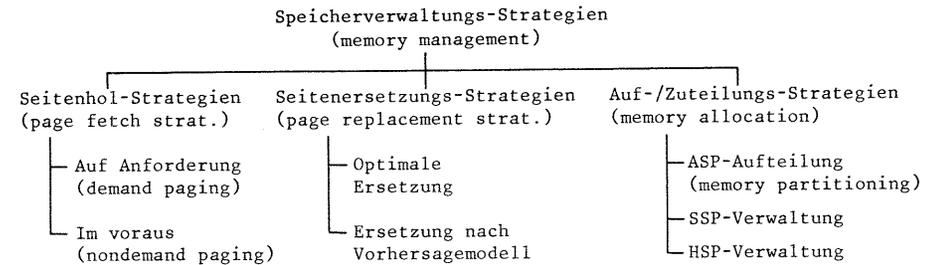


Bild 2.13. Übersicht über Strategien zur Speicherverwaltung

Seitenhol-Strategien

Die Seitenhol-Strategie beschreibt Zeitpunkt und Anzahl von Seiten, welche in den ASP geladen werden. Es werden unterschieden:

- demand paging
Das Holen einer Seite (der Fehlseite) wird stets nur bei Auftreten eines page fault veranlaßt.
- nondemand paging
Das Holen von Seiten erfolgt im voraus, obwohl das Programm noch arbeiten kann. Vorausschauendes Seitenholen ist nur dann sinnvoll, wenn eine hinreichend gute Vorhersage über das zukünftige Zugriffsverhalten möglich ist. Eine Mischform ist das vorausschauende Seitenholen zu den speziellen Zeitpunkten des Auftretens von page faults (demand prepaging).

Seitenersetzungs-Strategien

Das Ersetzen von Seiten ist erforderlich, wenn bei einem Seitenwechsel der für das betreffende Programm verfügbare ASP-Platz belegt ist. Die ersetzte Seite kann überschrieben werden, wenn sie nicht verändert wurde (change bit = 0); bei Veränderung (change bit = 1) muß die ersetzte Seite vorher auf den Hintergrundspeicher transportiert werden. Es können unterschieden werden:

- Optimale Ersetzungsstrategie
Ist das zukünftige Seitenzugriffsverhalten exakt bekannt, so ist es optimal, diejenige Seite zu ersetzen, welche ihren nächsten Zugriff am weitesten in der Zukunft erfahren wird. Da das Seitenzugriffsverhalten nur in den seltensten Fällen exakt bekannt ist, müssen reale Ersetzungsstrategien auf einem Vorhersagemodell für das zukünftige Zugriffsverhalten aufbauen.
- Ersetzung nach Vorhersagemodell
Die meisten Vorhersagemodelle orientieren sich am Zugriffsverhalten der unmittelbaren Vergangenheit. Kriterien zur Entscheidung, welche Seite ersetzt werden soll, können sein [17,18]:

Kriterium	Strategie	Ersetzte Seite
Aufenthaltsdauer im ASP	FIFO	längste Aufenthaltsdauer (first-in, first-out)
	LIFO	kürzeste Aufenthaltsdauer
Reihenfolge zurückliegender Zugriffe	LRU	weitest zurückliegender Zugriff (least-recently-used)
	MRU	kürzest zurückliegender Zugriff (most-recently-used)
Zugriffswahrscheinlichkeit bzw. -häufigkeit	A _o	geringste Zugriffswahrscheinlichkeit
	LFU	geringste Häufigkeit des Zugriffs (least-frequently-used)
	MFU	größte Häufigkeit des Zugriffs (most-frequently-used)
Page fault-Häufigkeit	PPF	Überschreitung einer page fault-Häufigkeitsgrenze (page-fault-frequency)
Zufall	RANDOM	zufällig ausgewählt
Zurückliegendes Zugriffsintervall	WORKING SET	Nichtzugehörigkeit zur Menge der Seiten, welche bei den letzten τ Seitenzugriffen mindestens einmal vertreten waren

Tabelle 1. Übersicht über Vorhersagemodelle für Seitenersetzung

Die Organisation des Zugriffsgedächtnisses sei anhand der beiden Ersetzungsstrategien FIFO und LRU verdeutlicht, vergl. Bild 2.14:

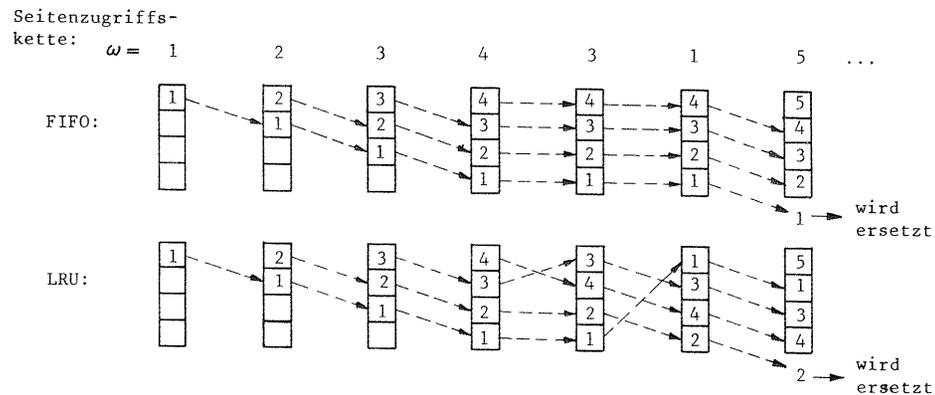


Bild 2.14. Organisation des Zugriffsgedächtnisses bei den Seitenersetzungsstrategien FIFO und LRU

Die "Seitenreferenzkette" ω in Bild 2.14 stellt die Folge der Seitennummern dar, auf welche sich die Adreßreferenzen beziehen. Das Programm habe 4 Seitenrahmen im ASP zur Benutzung. Die Reihung der "Wichtigkeit" erfolgt dabei von oben nach unten bei FIFO entsprechend dem Prinzip der Aufenthaltsdauer und bei LRU nach dem sog. "Kellerprinzip" (stack).

Auf-/Zuteilungs-Strategien

Die Aufteilung des Arbeitsspeichers unter die verschiedenen Programme bei Multiprogramming kann nach verschiedenen Gesichtspunkten erfolgen:

- feste Aufteilung (fixed partitioning)

Bei fester Aufteilung erhält ein Programm eine gleichbleibend konstante Anzahl von Seitenrahmen im ASP. Die am Multiprogramming beteiligten Programme können dabei entweder alle gleichviel oder unterschiedlich viel Platz erhalten (balanced/unbalanced partitioning).

- variable Aufteilung (variable partitioning)

Bei variabler Aufteilung kann der den einzelnen Programmen zugeteilte ASP-Platz an die momentanen Speicherplatzbedürfnisse angepaßt werden.

- Schnellspeicher-Verwaltung

Die Schnellspeicher-Verwaltung umfaßt das Holen und Ersetzen von Befehls- und Datenworten. Das Holen aus dem ASP erfolgt i.a. blockweise (z.B. Blöcke à 32 bytes); veränderte Worte werden häufig parallel in SSP und ASP geschrieben, wodurch der ASP stets aktualisiert ist (store-through). Das Ersetzen von Blöcken erfolgt i.a. nach einem LRU-Alterungsmechanismus, wodurch "Trefferraten" bis zu 95% erzielt werden.

- Hintergrundspeicher-Verwaltung

Auch die Zugriffszeiten zu Hintergrundspeichern (Platte, Trommel) können durch strategische Maßnahmen verbessert werden. Dies kann einerseits dadurch erfolgen, daß die häufiger benutzten Programnteile (z.B. nicht-residente System- und Bibliotheksprogramme) so abgelegt werden, daß sich eine im Mittel kürzere Zugriffszeit der mechanisch bewegten Schreib/Leseköpfe ergibt. Andererseits kann die Reihenfolge der jeweils anstehenden Platten- bzw. Trommelzugriffe an der momentanen Stellung der Schreib/Leseköpfe orientiert werden, wodurch ebenfalls im Mittel kürzere Zugriffszeiten erzielt werden, z.B.

SATF: shortest-access-time-first

SSTF: shortest-seek-time-first

SCAN: SSTF unter Beibehaltung der Arm-Bewegungsrichtung bis zur Umkehr usw.

2.3.5 Funktionsabläufe

Funktionelle Abläufe innerhalb von Rechnersystemen können in sehr unterschiedlichen Ebenen betrachtet werden, wie folgende Beispiele zeigen:

- Hardware-Steuerbefehls-Ebene
- Mikrobefehls-Ebene
- Makro-(Maschinen-)Befehls-Ebene
- Blocktransfer-Ebene
- Seitentransfer-Ebene
- Programm-Ebene

Im folgenden sollen die Funktionsabläufe am Beispiel des Zusammenspiels unterschiedlicher Betriebsmittel auf Seitentransfer-Ebene näher diskutiert werden, vergl. hierzu [19].

Vorausgesetzt werde ein Rechnersystem mit einer zweistufigen Speicherhierarchie bestehend aus Arbeitsspeicher ASP und Hintergrundspeicher HSP. Um das funktionelle Ablaufgeschehen auf Seitentransfer-Ebene zu verfolgen, werden zwei "Betriebsmittel" betrachtet: der Zentralprozessor ZP und das Kanalwerk KW. Das zentrale Leitungssystem (Bus) sowie das spezielle Zusammenspiel zwischen EAP, Kanal und HSP-Zugriffssystem innerhalb des Kanalwerkes soll nicht im einzelnen diskutiert werden. Die Interaktion zwischen ZP und KW wird mittels dreier Unterbrechungssignale gesteuert:

- Systemprogramm-Aufruf durch ein Benutzerprogramm zwecks Ein/Ausgabe (supervisor call interrupt, SVC)
- Systemprogramm-Aufruf durch ein Hardware-Unterbrechungssignal infolge Zugriff auf eine nicht vorhandene Hauptspeicheradresse (page fault, PF)
- Systemprogramm-Aufruf durch den Ein/Ausgabe-Prozessor (I/O interrupt)

Zur Unterbrechungssignal-Behandlung werden entsprechende Systemprogramme aufgerufen, welche der Zentralprozessor ZP bearbeitet. Bezüglich des Aktiv-Zustandes des Zentralprozessors ZP können somit zwei prinzipielle Zustände unterschieden werden:

- α : Überwacherstatus (supervisor state)
ZP bearbeitet ein Systemprogramm
- β : Benutzerstatus (user state)
ZP bearbeitet ein Benutzerprogramm

Dem Überwacherstatus α wird generell höhere Priorität gegenüber dem Benutzerstatus β eingeräumt. Innerhalb des Überwacherstatus α können dabei folgende Teilzustände unterschieden werden:

- α_1 : Behandlung von I/O-Interrupts:
Unterbrechung eines ablaufenden Benutzerprogrammes
Abspeichern des Maschinenzustandes (Registersätze)
Zustandseintrag des Betriebsmittels "Kanal"
Warteschlangeneintrag des Programmes mit abgeschlossenem I/O für α_4 -Phase

- α_2 : Behandlung von SVC-Interrupts bzw. PF-Interrupts:
Unterbrechung des ablaufenden Benutzerprogrammes
Abspeichern des Maschinenzustandes
Ausführung des Seitenersetzungs-Algorithmus bei PF-Interrupts
Konstruktion des Kanalprogrammes für den Seitentransfer bzw. die Ein/Ausgabe
Warteschlangeneintrag für α_3 -Phase
- α_3 : Auswahl eines Programmes für I/O
Starten des Kanalprogrammes (Seitentransfer bzw. Ein/Ausgabe)
- α_4 : Auswahl eines Benutzerprogrammes mit abgeschlossenem I/O
Bereitstellen des Maschinenzustandes
Starten des Benutzerprogrammes

Der Zentralprozessor ZP wird durch eine Unterbrechungs-Vorrangsteuerung verwaltet nach folgender Strategie (process dispatching, scheduling):

1. Ein I/O-Interrupt unterbricht ein ablaufendes Benutzerprogramm (β -Phase) sofort und startet anschließend eine α_1 -Phase.
Trifft ein I/O-Interrupt während eines ablaufenden Systemprogrammes (α -Phase) ein, so muß er warten.
2. Ein SVC-Interrupt bzw. ein PF-Interrupt unterbricht das ablaufende Benutzerprogramm (β -Phase) sofort und startet anschließend eine α_2 -Phase.
3. Nach Abschluß einer der α -Phasen werden anstehende ZP-Anforderungen entsprechend nachfolgender Rangordnung bedient:
 - 3.1 I/O-Interrupt (α_1 -Phase)
 - 3.2 Auswahl eines Benutzerprogrammes für I/O und Start des Kanalprogrammes (α_3 -Phase), falls Kanal frei
 - 3.3 Fortsetzung eines durch I/O-Interrupt unterbrochenen Benutzerprogrammes (β -Phase)
 - 3.4 Start des ausgewählten Benutzerprogrammes (β -Phase), falls - abgesehen von α_1 und α_3 - eine α_4 -Phase vorausgegangen ist.
 - 3.5 Auswahl eines Benutzerprogrammes (α_4 -Phase), falls - abgesehen von α_1 und α_3 - eine α_2 -Phase vorausgegangen ist.
4. Trifft keiner der o.a.Fälle zu, so bleibt der Zentralprozessor frei bis zum Eintreffen des nächsten I/O-Interrupts bzw. bis zur nächsten Initiierung eines Benutzerprogrammes.

Bild 2.15 zeigt den prinzipiellen Funktionsablauf zwischen Zentralprozessor ZP und Kanalwerk KW beim Seitentransfer. Es wurde dabei angenommen, daß zwei gleichberechtigte Benutzerprogramme A und B im Multiprogramming bearbeitet werden.

2.3.6 Verkehrsprobleme

Obwohl die einzelnen Vorgänge innerhalb eines Rechnersystems determiniert ablaufen, können die Anforderungen, welche hinsichtlich der verschiedenen Betriebsmittel gestellt werden, als "Verkehr" betrachtet werden. Die i.a. komplexen

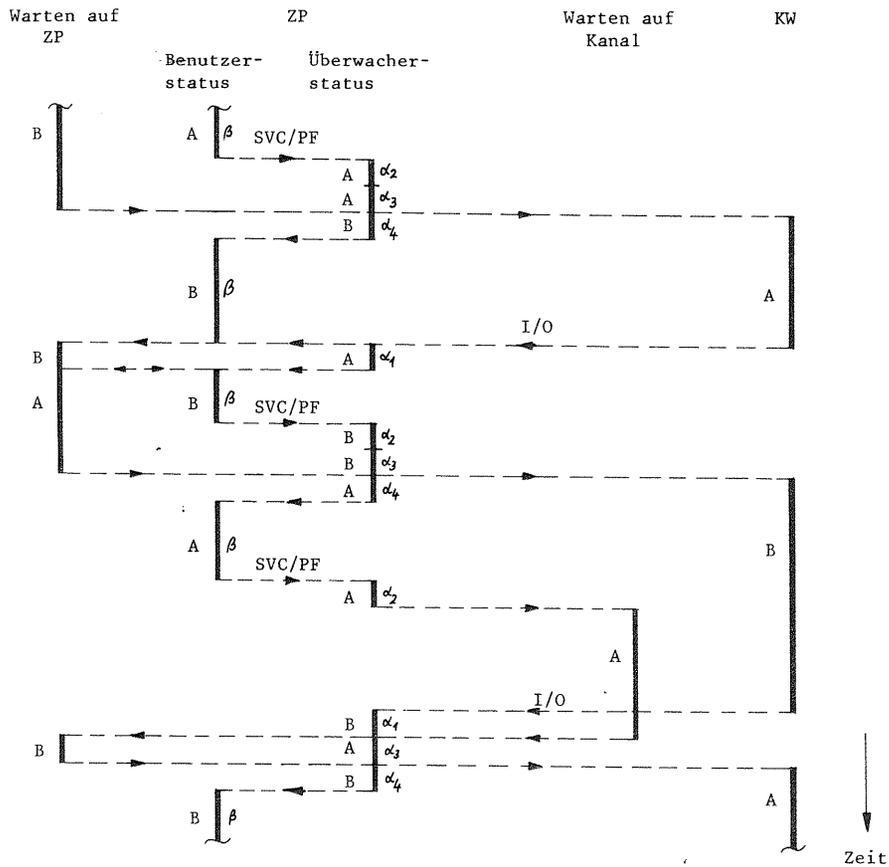


Bild 2.15. Prinzipieller Funktionsablauf zwischen Zentralprozessor ZP und Kanalwerk KW für den Seitentransfer bei Multiprogramming

Zusammenhänge zwischen Betriebsmittelanforderungen eines einzelnen Programmes wie auch die durch das Zusammenspiel mehrerer Programme bedingten Einflüsse auf Anzahl und Zeitpunkt von Betriebsmittelanforderungen insgesamt lassen es ratsam erscheinen, das Ablaufgeschehen vom statistischen Gesichtspunkt aus zu betrachten. Die Verkehrsprobleme wirken sich in Form von Blockierungen bzw. Verzögerungen aus, welche die optimale Nutzung der Betriebsmittel unter Umständen drastisch einschränken können. Im folgenden seien einige typische Verkehrsprobleme aufgezeigt:

a) Befehls- und Rechenwerke

Zur Erhöhung der Arbeitsgeschwindigkeit werden die Teilfunktionen "Befehl Holen", "Adreßrechnung", "Operand Holen" und "Befehl Ausführen" zeitlich überlappt für aufeinanderfolgende Befehle nach dem Fließbandprinzip (pipelining) durch autonome Unterwerke ausgeführt. Der Durchsatz durch diese pipeline hängt u.a. ab von den statistisch schwankenden Ausführungszeiten von Teilfunktionen sowie von der Sprungbefehls-Charakteristik des Befehlsstromes (bei Sprungbefehlen wird der o.a. Ablauf unterbrochen und muß mit der Sprungadresse neu begonnen werden).

b) Zentralprozessoren mit Schnellspeichern

Die Leistungsfähigkeit von Zentralprozessoren mit Schnellspeichern hängt von der "Trefferrate" ab, mit welcher Programm- und Datenworte im Schnellspeicher gefunden werden. Fehlreferenzen verursachen dagegen Ergänzungstransporte und vermindern damit die Verarbeitungsgeschwindigkeit des Zentralprozessors. Der Durchsatz wird wesentlich von den Lokalitätseigenschaften der Benutzer- und Systemprogramme sowie der Schnellspeicher-Verwaltung mitbestimmt.

c) Modulare Arbeitsspeicher

Zur zeitlich überlappten Ausführung von Schreib/Lesevorgängen werden große Arbeitsspeicher in Moduln unterteilt mit autonomer Speichermodul-Steuerung. Die Leistungsfähigkeit hängt in starkem Maße von der statistisch beschreibbaren Zugriffstreuung zu einzelnen Moduln sowie von der Speicherverwaltung ab (z. B. bei sog. "Adreßverschränkung", wobei Folgeadressen in zyklischer Folge in verschiedenen Moduln abgelegt werden).

d) Zentrale Leitungssysteme

Der Informationsaustausch zwischen zentralen Baugruppen erfolgt über ein im Zeitmultiplex betriebenes Bussystem und wird von einem Vorrangwerk gesteuert. Die Leistungsfähigkeit dieses "Vermittlungssystems" hängt u.a. ab von der Charakteristik der Übertragungs-Anforderungen, der Organisation der Übertragung (seriell/parallel, einzeln/blockweise) sowie der Zuteilungsstrategie.

e) Speicherhierarchie

Der Transport von Befehlen und Daten zwischen den verschiedenen Ebenen der Speicherhierarchie (Arbeitsspeicher-Hintergrundspeicher) verursacht einen Kanalverkehr, welcher unter Umständen zum Systemengpaß werden kann. Dies kann insbesondere bei hohem Multiprogrammingsgrad oder bei zu kleinem Arbeitsspeicher auftreten, wenn die Benutzerprogramme zu viele page faults erzeugen (thrashing). Die

Leistungsfähigkeit eines Rechnersystems mit Speicherhierarchie hängt in entscheidender Weise ab von den Lokalitätseigenschaften der Benutzerprogramme, den Arbeitsgeschwindigkeiten von Zentralprozessor und Kanälen, der Arbeitsspeichergröße, dem Multiprogrammgrad und der bei Unterbrechungen notwendigen Systemverwaltungszeit ("system overhead").

f) Teilnehmer-Rechensysteme, Rechnernetze

Teilnehmer-Rechensysteme und Rechnernetze sind durch einen intensiven Datenaustausch zwischen Teilnehmern und Rechnersystem bzw. verschiedenen Rechnersystemen über ein Daten(vermittlungs)netz gekennzeichnet. Struktur, Betriebsweise und der von einer Vielzahl von Teilnehmern verursachte Verkehr bedingen verhältnismäßig komplexe Ablaufvorgänge mit möglichen Engpässen hinsichtlich der Speicherung, Verarbeitung und Übertragung von Informationen. Dabei entstehen u.a. folgende Verkehrsprobleme:

- Datenverkehr innerhalb der einzelnen Rechnersysteme zwischen den verschiedenen Ebenen der Speicherhierarchie
- Datenverkehr zwischen der Zentraleinheit und den Teilnehmerstationen über direkt angeschlossene Ein/Ausgabekanäle oder über Vorverarbeitungsprozessoren zur Sammlung und Verteilung des Teilnehmerverkehrs
- Datenverkehr innerhalb des Datennetzes, welches verschiedene Rechnersysteme miteinander verbindet.

Voraussetzung für einen reibungslosen Ablauf des Betriebs ist die verkehrsgerechte Dimensionierung der Betriebsmittel innerhalb der einzelnen Ebenen und eine wirkungsvolle Organisation des Datenverkehrs innerhalb und zwischen diesen Ebenen.

Zur ausführlichen Darstellung der Problematik und Analyse des Verkehrsflusses in Datennetzen sei auf die Arbeit von Herzog [20] verwiesen.

3. METHODEN DER MODELLEBILDUNG

Ziel dieses Kapitels ist es, eine prinzipielle und systematische Methodik zur Modellbildung anzugeben. Nach einer einleitenden Diskussion der Zielsetzungen werden die einzelnen Schritte der Modellbildung beschrieben und anschließend an zwei Beispielen durchgeführt. Eine Systematik verkehrstheoretischer Modelle beschließt das Kapitel.

Die in diesem Kapitel angestellten Überlegungen sind überwiegend allgemein gehalten. In den Beispielen wird die Modellbildung jedoch schwerpunktmäßig vom Standpunkt der Steuerung her betrachtet.

3.1 Zielsetzungen der Modellbildung

Die Arbeitsweise in Naturwissenschaften und Technik ist außerordentlich stark durch das Modelldenken gekennzeichnet. Modelle führen dabei komplexe Vorgänge auf einfacher durchschaubare, unter Umständen aber auch auf fehlerbehaftete Vergleichsvorgänge zurück. Beispiele hierfür sind das mechanistische Atommodell (Bohr) oder etwa das aus konzentrierten Elementen bestehende elektrische Transistor-Ersatzschaltbild (Giacoletto). Obwohl diese Modelle im Vergleich zur Realität einfacher durchschaubar und beschreibbar sind, liegen ihre Nachteile in der Begrenztheit des Gültigkeitsbereiches. Zu jeder Modell-Vorstellung gehört deshalb eine Gültigkeitsbetrachtung und mit Überschreiten des Gültigkeitsbereiches ist i.a. auch eine Änderung der Modell-Vorstellung verbunden. Die "Modellbildung" gewinnt daher eine zentrale Rolle bei der Systemanalyse.

3.1.1 Modellbegriff

Unter einem "Modell" soll allgemein die Beschreibung des Ablaufgeschehens eines realen Systems in allen seinen logischen und zeitlichen Zusammenhängen mit Hilfe weniger abstrakter, d.h. von der jeweiligen Realisierung unabhängiger Grundkomponenten verstanden werden. Die "Modellbildung" beschreibt die Abbildung der tatsächlich ablaufenden Vorgänge auf äquivalente Vorgänge innerhalb des Modells.

Je nach Anwendungsfall soll zwischen "Gesamtmodellen" und "Teilmodellen" unterschieden werden, wenn sich die Abbildung auf das ganze System bzw. nur auf Teilsysteme oder Teilaspekte des ganzen Systems bezieht. Ferner kann zwischen funktions- und verkehrsbezogenen Modellen unterschieden werden, wenn in den entsprechenden Modellen der Aussageschwerpunkt mehr auf Funktions- bzw. Verkehrszusammenhänge gelegt wird.

3.1.2 Kriterien zur Modellbildung

Die Zielsetzung bei der Modellbildung unterliegt unterschiedlichen Kriterien. Die wichtigsten dieser Kriterien sind:

- Modellierungstiefe

Die Abbildung der tatsächlichen Vorgänge kann unterschiedlich detailliert sein, je nachdem ob die Zusammenhänge makroskopisch (z.B. Eingangs/Ausgangsverhalten eines Teilsystems) oder mikroskopisch (z.B. einschließlich aller ablaufenden Zustandsänderungen innerhalb eines Teilsystems) aufgelöst werden.

- Modellkomponenten

Die Beschreibung des Ablaufgeschehens im Modell sollte mit möglichst wenigen und standardisierten Grundkomponenten erfolgen, um die Zusammenhänge auf einer von der Realisierung unabhängigen abstrakten Ebene einheitlich und durchsichtig zu gestalten.

- Abbildungstreue

Das Modell sollte die logischen und zeitlichen Zusammenhänge in einer Form widerspiegeln, welche den tatsächlichen Gegebenheiten möglichst äquivalent ist.

- Modularität

Ein Modell sollte nach Möglichkeit aus mehr oder weniger abgeschlossenen Teilmodellen aufgebaut sein, deren Schnittstellen leicht beschreibbar sind. Dies ermöglicht i.a. leichtere Verallgemeinerungen oder Vereinfachungen. Die Modulbildung kann oft auch entscheidende Hinweise für eine entsprechend modularisierte Modellanalyse geben.

- Modellanalyse

Die Auswahl und das Zusammenspiel der Modellkomponenten sollte so erfolgen, daß sowohl formale Beschreibungen des funktionellen Zusammenspiels zur Prüfung der Vollständigkeit und Widerspruchsfreiheit als auch formale Beschreibungen des dynamischen Ablaufgeschehens zur Verkehrsanalyse ermöglicht werden.

In der Praxis werden diese Kriterien nicht immer in idealer Weise erfüllbar sein. Die Orientierung an derartigen Kriterien an sich kann jedoch häufig schon wertvolle Gesichtspunkte liefern.

3.2 Methodik der Modellbildung

Die Vorgehensweise bei der Modellbildung ist häufig von der Intuition her geprägt. Im folgenden soll versucht werden, die Vorgehensweise systematisch in eine Reihe von Einzelschritten aufzulösen.

3.2.1 Charakterisierung des realen Ablaufgeschehens

Das Ablaufgeschehen innerhalb von Systemen zur Nachrichtenverarbeitung oder Nachrichtenvermittlung kann beschrieben werden durch Anforderung, Zuteilung und Belegung (Inanspruchnahme) von Betriebsmitteln. Logische und zeitliche Zusammenhänge sind dabei durch Zustände, Ereignisse und Übergänge darstellbar.

a) Anforderungen und Betriebsmittel

Als "Anforderung" sollen alle jene Steuersignale verstanden werden, welche eine Inanspruchnahme von Betriebsmitteln einleiten. Anforderungen können sein: Teilnehmer-Verbindungsanreize, Wählziffern, Programme, Ein/Ausgabebefehle usw. Anforderungen sind zu klassifizieren nach

- Klassen (Zusammenfassung von Anforderungen gleichen Typs)
- Herkunft (Quelle)
- Ziel (Betriebsmittel).

Als "Betriebsmittel" ist jede (Hardware-)Einrichtung aufzufassen, welche bestimmte Teilfunktionen aufgrund einer "Anforderung" ausführt. Betriebsmittel können beispielsweise sein: Prozessoren, Unterwerke, Kanäle, Leitungen, Speicher, usw. Die Betriebsmittel sind hinsichtlich der gestellten Anforderungen zu klassifizieren nach:

- Bedienungsfunktion
- Puffer- oder Wartefunktion
- Klasse der bedienten Anforderungen.

b) Zustände, Ereignisse und Übergänge

Anforderungen und Betriebsmittel sind in jedem Zeitpunkt durch einen "Zustand" gekennzeichnet entsprechend ihrer momentanen Ausübung einer Funktion. Das Betriebsmittel "Zentralprozessor ZP" kann beispielsweise die Zustände "frei", " α_1 ", ..., " α_4 ", " β " annehmen (vergl. Abschnitt 2.3.5). Die Anforderung "Programm" kann Zustände "Warten auf ZP", "ZP-Bearbeitung", "Unterbrochen durch I/O-Interrupt" usw. annehmen. Eine ausführliche Zustandskennung umfaßt

- Zustands-Mengen aller Anforderungsklassen
- Zustands-Mengen aller Betriebsmittel

Unter "Ereignis" soll jede Ursache verstanden sein, welche eine Zustandsänderung bewirkt, d.h. einen Übergang von einem Zustand in einen anderen. Ereignisse sind:

- Eintreffen von Anforderungen
- Belegen und Freigeben von Betriebsmitteln
- Systembedingte Zustandsänderungen (Ein/Ausgabe-Unterbrechungstakte etc.)

c) Logische und zeitliche Zusammenhänge

Das funktionelle Ablaufgeschehen unterliegt i.a. gewissen Nebenbedingungen bedingt durch zugrundeliegende logische Zusammenhänge. Dies drückt sich in bestimmten Regeln aus, denen die Zustandsübergänge von Anforderungen bzw. Betriebsmitteln unterliegen. Beispielsweise können zwei zu verschiedenen Benutzerprogrammen gehörende β -Zustände eines Zentralprozessors nur über dazwischenliegende α -Zustände ineinander übergehen. Logische Zusammenhänge werden erfaßt durch

- Regeln für die Zustandsübergänge von Anforderungen
- Regeln für die Zustandsübergänge von Betriebsmitteln
- Strategien zur Betriebsmittelverwaltung

Zur Kennzeichnung des zeitlichen Ablaufgeschehens sind festzustellen:

- Systembedingte Belegungszeiten von Betriebsmitteln
- Benutzerbedingte Belegungszeiten von Betriebsmitteln
- Ankunftszeit-Charakter von systembedingten Anforderungen
- Ankunftszeit-Charakter von benutzerbedingten Anforderungen.

Auf systembedingte Belegungszeiten hat die Benutzer-Anforderung keinen Einfluß, wie z.B. bei Systemverwaltungszeiten von Zentralprozessoren. Unter benutzerbedingten Belegungszeiten sollen dagegen die Nutz-Bedienungszeiten verstanden werden, z.B. die Ausführungszeiten des Zentralprozessors für Benutzerprogramme. Entsprechend sollen hinsichtlich der zeitlichen Ankünfte von Anforderungen systembedingte bzw. benutzerbedingte Anforderungen unterschieden werden.

3.2.2 Modellkomponenten

Zur abstrakten, d.h. von der speziellen Realisierungsform losgelösten Beschreibung des Ablaufgeschehens werden allgemein verwendbare Modellkomponenten eingeführt, mit deren Hilfe Struktur, Betriebsweise und Verkehr innerhalb des Modells näher bezeichnet werden.

a) Strukturbezogene Komponenten

Strukturmäßig können Modelle aus wenigen Grundkomponenten aufgebaut werden, welche in der Tabelle 2 zusammengestellt sind.

Diese Komponenten sind verschiedener Zustände fähig: eine Bedienungseinheit kann beispielsweise "frei", "bedienbelegt" oder "blockierbelegt" sein. Zusätzliche Zustandsinformationen können sich beziehen auf Belegungsbeginn, Belegungsdauer, Belegungsende, Typ der belegenden Anforderung etc.

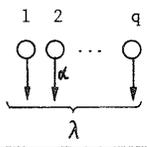
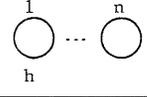
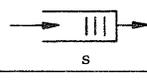
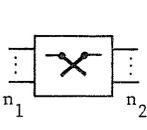
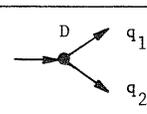
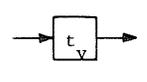
Komponente (Strukturelement)	Funktion	Symbol	Parameter
Verkehrsource(n)	Erzeugung von Anforderungen		q Anzahl h _I Mittlere Freizeit/Quelle α Anforderungsrate/freie Quelle λ Anforderungsrate aller freier Quellen
Bedienungseinheit(en)	Ausführung der von Anforderungen gewünschten Betriebsmittelfunktion ("Bedienung")		n Anzahl h Mittlere Belegungsdauer
Warteschlange	Zwischenspeichern (Pufferung) wartender Anforderungen ("Warten")		s Anzahl der Wartepplätze
Koppelvielfach (KVf)	Einstufiges Durchschalten wahlweiser Verbindungen zwischen Eingängen und Ausgängen mit vollkommener Erreichbarkeit		i Anzahl der Eingänge k Anzahl der Ausgänge
Koppelnetzwerk	Durchschalten wahlweiser Verbindungen zwischen Eingängen und Ausgängen		n ₁ Anzahl v. Eingängen n ₂ Anzahl v. Ausgängen k Erreichbarkeit S Stufenzahl g _j Anzahl der KVfe in Stufe j
Verzweigung	Verzweigen von Anforderungen nach Richtungen		Verzweigungsstrategie z.B. q _i Wahrscheinlichkeit für Richtung i
Schalter	Anschalten bzw. Abschalten von Verbindungswegen		Anschaltebedingung z.B. zeitabhängig zustandsabhängig
Laufzeitglied	Verzögern von Anforderungen beim Übergang von einem Element zum nächsten		t _v Verzögerungszeit

Tabelle 2. Strukturbezogene Komponenten verkehrstheoretischer Modelle

b) Betriebsbezogene Komponenten

Zustandsänderungen werden von Ereignissen ausgelöst und finden unter bestimmten Regeln statt, welche den "Betrieb" kennzeichnen. Dieses können sein:

- Zuteilungsstrategien: Auswahl von Bedienungseinheiten (Abschmodus erreichbarer Bedienungseinheiten)
Beispiele: sequentielles Absuchen
zyklisches Absuchen
- Abfertigungsstrategien: Auswahl einer von mehreren wartenden Anforderungen innerhalb einer Warteschlange (queue discipline) oder zwischen mehreren Warteschlangen (interqueue discipline)
Beispiele: FIFO (first-in, first-out)
RANDOM (zufällig)
CYCLIC (zyklisch)
- Prioritätsstrategien: Auswahl bzw. Behandlung von Anforderungen unterschiedlicher Dringlichkeit (priority disciplines)
Beispiele: unterbrechende Prioritäten
nichtunterbrechende Prioritäten
dynamische Prioritäten
- Verzweigungsstrategien: Wegelenkung von Anforderungen (routing strategies)
Beispiele: feste Wegelenkung
wahrscheinlichkeitsmäßige Wegelenkung
alternative Wegelenkung
adaptive Wegelenkung
- Blockierungsverhalten: Behandlung eintreffender Anforderungen in Blockierungsfällen
Verlustsystem: Abweisung im Falle vollbelegter erreichbarer Bedienungseinheiten
Wartesystem: Warten im Falle vollbelegter erreichbarer Bedienungseinheiten
Warteverlustsystem: Warten im Falle vollbelegter erreichbarer Bedienungseinheiten und freiem Speicherplatz, ansonsten Abweisung

c) Verkehrsbezogene Komponenten

Die zeitlichen Abläufe innerhalb eines Modelles sind im wesentlichen bestimmt durch die Eintreffzeitpunkte von Anforderungen und deren Bedienungszeitbedürfnisse. Allgemein können diese Eigenschaften beschrieben werden mit den Hilfsmitteln zufallsabhängiger Prozesse.

- Ankunftsprozesse: Beschreibung der Ankunftsabstände T_A (interarrival time) von Anforderungen bzw. der Freizeitdauern T_I (idle time) einzelner Verkehrsquellen.

Ankunftsprozesse können nach weiteren Kriterien unterteilt werden, wie z.B.

Einzel- oder Gruppenankünfte
Endlich oder unendlich große Quellenzahl
Kontinuierliche oder diskrete (getaktete) Ankunftszeitpunkte etc.

Im einfachsten Falle von Einzelankünften kann der Ankunftsprozeß beschrieben werden durch die VF der Ankunftsabstände T_A

$$A(t) = P \{ T_A \leq t \}$$

mit

$$\lambda = 1/E[T_A] \text{ Ankunftsrate (Anzahl eintreffender Anforderungen je Zeiteinheit)}$$

- Bedienungsprozesse: Beschreibung der Bedienungsdauern T_H (holding time) von Bedienungseinheiten für Anforderungen.

Bedienungsprozesse können nach weiteren Kriterien unterschieden werden, wie z.B.

Kontinuierliche oder diskrete (getaktete) Beginnzeitpunkte
Einzel- oder Gruppenbedienung etc.

Bedienungsdauern T_H werden i.a. beschrieben durch eine VF

$$H(t) = P \{ T_H \leq t \}$$

mit

$$h = E [T_H] \text{ mittlere Bedienungsdauer}$$

Die bei Ankunfts- bzw. Bedienungsprozessen auftretenden zufallsabhängigen Ankunftsabstände bzw. Belegungsdauern werden mit Hilfe von Wahrscheinlichkeitsverteilungsfunktionen (VF) beschrieben. Typ und Parameter der VF legen die Charakteristik des Prozesses fest. Zur Anpassung an real beobachtete Charakteristiken werden verschiedene VF-Typen angewandt, welche sich vor allem in der Streuung der zufälligen Zeiten T unterscheiden. Die wichtigsten VF sind in der folgenden Tabelle 3 angegeben.

Typ der VF	Kurz- bez.	Mathematische Beschrei- bung $F(t) = P\{T \leq t\}$	Mittelwert $E[T]$	Variations- koeffizient c
Exponential-VF (Markovian)	M	$F(t) = 1 - e^{-\frac{t}{h}}$	h	1
Konstante VF (Deterministic)	D	$F(t) = \begin{cases} 0, & t < h \\ 1, & t \geq h \end{cases}$	h	0
Erlang-k-VF (Erlangian with order \underline{k})	E_k	$F(t) = 1 - e^{-k\mu t} \sum_{i=0}^{k-1} \frac{(k\mu t)^i}{i!}$ $k = 1, 2, \dots$	$\frac{1}{\mu}$	$\frac{1}{\sqrt{k}}$
Hyperexponential- VF (Hyperexponential with order \underline{k})	H_k	$F(t) = 1 - \sum_{i=1}^k p_i e^{-\epsilon_i t}$ $k = 1, 2, \dots; \sum_{i=1}^k p_i = 1$	$\sum_{i=1}^k \frac{p_i}{\epsilon_i}$	$\sqrt{\frac{2 \sum_{i=1}^k p_i / \epsilon_i^2}{(\sum_{i=1}^k p_i / \epsilon_i)^2}} - 1$

Tabelle 3. Typen der wichtigsten Verteilungsfunktionen

In Tabelle 3 bedeuten:

T	Zufallsvariable (Zeit)
$F(t) = P\{T \leq t\}$	Wahrscheinlichkeits-Verteilungsfunktion (3.1)
$E[T^k] = \int_{t=0}^{\infty} t^k F'(t) dt$	Gewöhnliches Moment k-ter Ordnung speziell $k=1: E[T] = \text{Mittelwert}$ (3.2)
$VAR[T] = E[T^2] - E[T]^2$	Varianz (=Streuungsquadrat σ^2) (3.3)
$c = \frac{\sigma}{E[T]}$	Variationskoeffizient (3.4)

Das Streuungsverhalten der Zufallsvariablen T wird am kürzesten mit dem (mittelwertbereinigten) Variationskoeffizienten c ausgedrückt. VF mit $c < 1$ heißen allgemein "hypoexponentiell", diejenigen mit $c > 1$ "hyperexponentiell". Beliebige Verteilungsfunktionen werden mit dem Buchstaben G (general) abgekürzt.

Bemerkung:

Die verkehrsmäßige Belastung eines Modelles ist durch die Angabe von Ankunfts- und Bedienungsprozessen im Prinzip festgelegt. Je nach Modell können diese jedoch in mehr oder weniger komplexer Weise mit den übrigen Modellkomponenten verknüpft sein, wie folgende Beispiele zeigen:

1. Beispiel: Die Anreize, welche ein Fernsprechteilnehmer im Zuge des Verbindungsaufbaus als Anforderungsfolge an das System stellt, hängen ab von dem Teilnehmerverhalten (Schnelligkeit, Falschbedienung etc.) sowie von der Rückwirkung des Vermittlungssystems (Signalisierungen über Anschalte- und Besetztzustände etc.)
2. Beispiel: Die Belegungsphasen eines Benutzerprogrammes auf dem Zentralprozessor bzw. dessen Anforderungen für Seitentransfer hängen u.a. ab von Größe und Organisation des Arbeitsspeichers.

In derartigen Fällen muß die Beschreibung der "Arbeitslast" (workload) unter Umständen durch entsprechend detailliertere Teilmodelle erfolgen.

d) Beispiel eines einstufigen Bedienungssystem-Modelles

In Bild 3.1 ist ein einfaches Bedienungssystem-Modell skizziert. Das System besitzt eine Bedienungsstufe bestehend aus n gleichartigen, vollkommen erreichbaren Bedienungseinheiten und einem begrenzten Wartespeicher der Kapazität s.

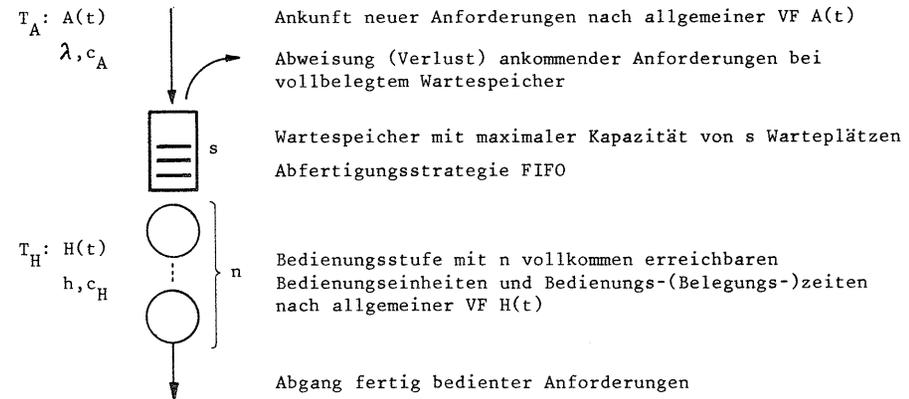


Bild 3.1. Beispiel eines einstufigen Bedienungssystems

Neue Anforderungen werden von einem Ankunftsprozeß erzeugt entsprechend der VF $A(t) = P\{T_A \leq t\}$. Mittelwert und Streuung der Ankunftsabstände sind dabei durch die Ankunftsrate $\lambda = 1/E[T_A]$ bzw. durch den Variationskoeffizienten c_A gekennzeichnet. Die Bedienungszeiten T_H sind ebenfalls durch eine VF $H(t) = P\{T_H \leq t\}$ beschrieben mit mittlerer Bedienungszeit h und Variationskoeffizient c_H .

Sind bei Eintreffen einer Anforderung noch Bedienungsorgane frei, so wird eines davon sofort belegt. Eine Anforderung verläßt das System sofort nach Bedienungsende bzw. im Falle des Antreffens des vollbelegten Systems (alle n Bedienungseinheiten und alle s Warteplätze belegt). Sind bei Eintreffen einer Anforderung alle Bedienungseinheiten belegt, aber noch freie Warteplätze vorhanden, kann diese Anforderung auf das Freiwerden einer Bedienungseinheit warten. Bei jedem Bedienungsende wird eine der wartenden Anforderungen entsprechend der Abfertigungsdisziplin zur Bedienung ausgesucht, im Falle von FIFO die am längsten wartende Anforderung. Der Warteplatz wird dabei sofort freigegeben.

3.2.3 Abbildung

Ein Verkehrsmodell ist aus den in 3.2.2 beschriebenen Modellkomponenten aufgebaut. Prinzipiell wird ein Verkehrsmodell dadurch gewonnen, daß Steuersignale, Betriebsmittel, Betriebsweisen und Eintreff- bzw. Belegungszeiten des realen Systems abgebildet werden auf "Anforderungen", "Strukturelemente", "Strategien" und "Verkehre" unter Einhaltung der logischen Zusammenhänge. Die Vorgehensweise kann in folgende prinzipiellen Einzelschritte aufgelöst werden:

1. Zuordnung von jenen realen Systemteilen, welche Steuersignale zur Anforderung von Betriebsmitteln erzeugen, zu "Verkehrsquellen" für "Anforderungen".
2. Zuordnung von realen Betriebsmitteln mit Bedienungsfunktion bzw. Pufferfunktion für Anforderungen zu "Bedienungseinheiten" bzw. "Warteschlangen".
3. Zuordnung von Bedienungszuständen bzw. Wartezuständen von Anforderungen zu Bedienungseinheiten bzw. Warteschlangen.
4. Verknüpfung von Verkehrsquellen, Warteschlangen und Bedienungseinheiten durch Verbindungswege entsprechend den in 3.2.1 festgelegten Angaben über Herkunft und Ziel von Anforderungen unter Berücksichtigung der Regeln für logische Zustandsübergänge von Anforderungen (individuell für jede Klasse von Anforderungen).
5. Festlegung von Verzweigungsstrategien für Richtungsaufteilungen entsprechend der Regeln für logische Zustandsübergänge, Wegelenkungsverfahren etc.
6. Festlegung der Zuteilungs-, Abfertigungs- und Prioritätsstrategien für die einzelnen Bedienungseinheiten unter Berücksichtigung der Regeln zur Betriebsmittelverwaltung bzw. der logischen Zustandsübergänge von Betriebsmitteln.
7. Zuordnung von Ankunfts- bzw. Bedienungsprozessen zur Beschreibung der zeitlichen Zusammenhänge zwischen dem Eintreffen von Anforderungen bzw. zwischen Anforderungen und Betriebsmittelbelegungen.
8. Festlegung einzelner Struktur-, Betriebs- und Verkehrsparameter.
9. Nachprüfung der logischen Zusammenhänge auf Richtigkeit.
10. Untersuchung des Modelles hinsichtlich möglicher Vereinfachungen oder Zerlegungen in Teilmodelle.

In der Praxis der Modellbildung wird i.a. mehr von der direkten Anschauung her intuitiv vorgegangen. Bei sehr komplexen Problemen empfiehlt sich allerdings eine stärkere Formalisierung, mit deren Hilfe zur Nachprüfung der Richtigkeit auch formal beschreibbare, und damit auch automatisierbare Prozeduren eingesetzt werden könnten.

3.2.4 Modellbestätigung

Bei der Modellbildung werden zwangsläufig gewisse Idealisierungen und damit verbundene Fehler in Kauf genommen. Aus diesem Grunde sollten die an einem Modell gewonnenen Ergebnisse stets im Zusammenhang mit der Abbildungstreue gesehen werden. Ferner sollte die Nachprüfung der Zulässigkeit getroffener Vereinfachungen im Auge behalten werden. Die Modellbestätigung (validation) kann prinzipiell entsprechend nachfolgendem Schema erfolgen, vergl. Bild 3.2:

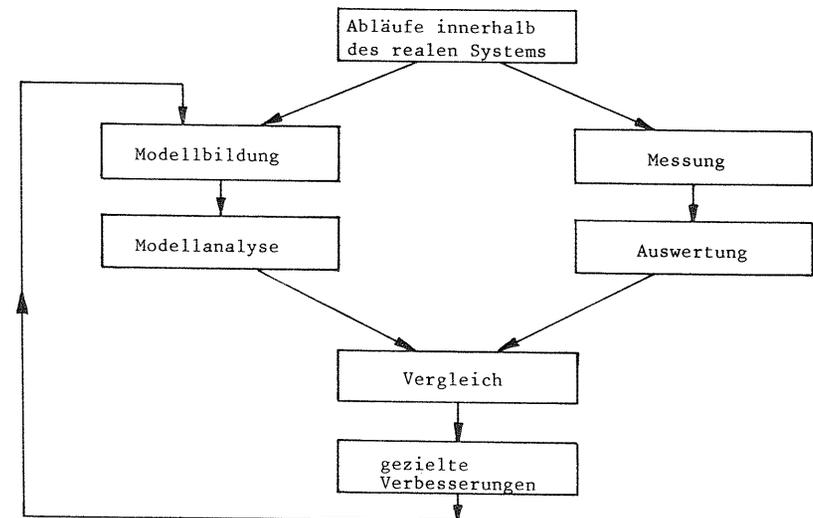


Bild 3.2. Prinzipielle Vorgehensweise zur Modellbestätigung

3.3 Beispiele zur Modellbildung

In diesem Abschnitt soll das Verfahren der Modellbildung anhand zweier vereinfachter Beispiele exemplarisch gezeigt werden. Die Beispiele beziehen sich auf Teilaspekte rechnergesteuerter Nachrichtenvermittlungen bzw. Rechnersysteme mit zweistufiger Speicherhierarchie.

3.3.1 Beispiel 1: Informationsaustausch zwischen Peripherie und Zentralsteuerung in rechnergesteuerten Nachrichtenvermittlungen

Betrachtet werde der Informationsaustausch zwischen peripheren Einrichtungen PE und Zentralsteuerung ZST entsprechend Abschnitt 2.2. Dieser Austausch besteht darin, daß Anreize der Peripherie an die Zentralsteuerung übertragen werden, welche darauf - nach einer entsprechenden Verarbeitungsphase - mit Befehlen an die Peripherie reagiert. Die einzelnen Schritte zur Modellfindung werden im folgenden kurzgefaßt dargestellt.

a) Charakterisierung des realen Ablaufgeschehens

Das reale Ablaufgeschehen ist durch eine Vielzahl unterschiedlicher Anreize ("Meldungen") und ihrer Behandlung gekennzeichnet. Aus Übersichtlichkeitsgründen sollen daher in diesem Abschnitt nur die grundsätzlichen Zusammenhänge berücksichtigt werden.

Betriebsmittel:	- Äußeres Leitungssystem	EBUS	(Übertragung von Meldungen PE → EAW)
	- Äußeres Leitungssystem	ABUS	(Übertragung von Befehlen EAW → PE)
	- Ein/Ausgabewerk	EAW	(Pufferung v. Meldungen u. Befehlen)
	- Inneres Leitungssystem (Ein/Ausgabekanal)	EAK	(Übertragung von Meldungen bzw. Befehlen EAW ↔ ZST)
	- Speicher	SP	(Speicherung v. Programmen u. Daten)
	- Zentralprozessor	ZP	(Bedienung von Meldungen)
	- Organisationsprogramm (Betriebssystem)	BS	(SP-residente Systemprogramme, Listen von Anforderungen)

Anforderungen:

- Meldungen von PE an die ZST sowie die daraus entstehenden Befehle der ZST an die PE
- I/O-Interrupts durch Realzeituhr

Klasseneinteilung:
 Klasse 0: I/O-Interrupt
 Klasse 1: Meldungen an ZST ohne Ausgabebefehl
 Klasse 2: Meldungen an ZST mit Ausgabebefehl an PE

Ereignisse:

- Ankunft von Anforderungen aus PE
- Übertragungsende auf EBUS bzw. ABUS
- Bedienungsende des ZP
- I/O-Interrupt für EAK zu äquidistanten starren Zeitpunkten (systembedingte Zustandsänderung)

I/O-Interrupts werden von einer Realzeituhr erzeugt und unterbrechen die laufende Bearbeitung des ZP für die Dauer der Ein/Ausgabe über den EAK

Zustände des I/O-Interrupts:
 0 I/O-Interrupt inaktiv
 Z0 I/O-Interrupt aktiv

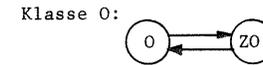
Zustände der Anforderungen Klassen 1 und 2:	Z1	Warten auf EBUS	} Klasse 1	} Klasse 2
	Z2	Übertragung durch EBUS		
	Z3	Warten auf EAK		
	Z4	Übertragung durch EAK		
	Z5	Warten auf ZP		
	Z6	Bearbeitung durch ZP		
	Z7	Warten auf EAK		
	Z8	Übertragung durch EAK		
	Z9	Warten auf ABUS		
	Z10	Übertragung durch ABUS		

Zustände des ZP:
 0 ZP frei
 α Behandlung von I/O-Interrupts (Eingabe bzw. Ausgabe)
 β Bearbeitung von Meldungen einschliesslich des jeweiligen Organisationsprogrammaufrufes vor Beginn der eigentlichen Bearbeitung

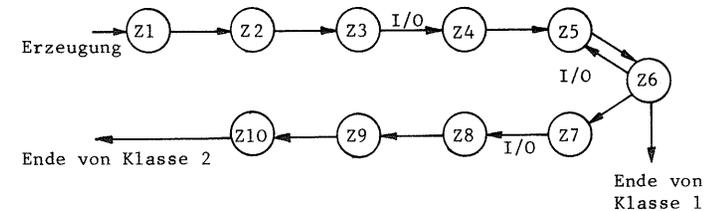
Zustände des EAK:
 0 EAK deaktiviert
 δ_1 EAK durch I/O-Interrupt aktiviert (Eingabe)
 δ_2 EAK durch I/O-Interrupt aktiviert (Ausgabe)

Zustände von EBUS, ABUS:
 0 EBUS (ABUS) frei
 $\delta (\epsilon)$ EBUS (ABUS) übertragungsbelegt

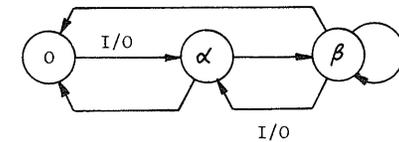
Zustandsübergänge von Anforderungen:



Klassen 1,2:

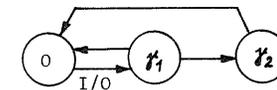


Zustandsübergänge des ZP:

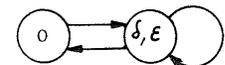


Zustandsübergänge von EAK, EBUS, ABUS:

EAK:



EBUS bzw. ABUS:



- Zeitliche Zusammenhänge:
- Beschreibung des Ankunftsprozesses eintreffender Anforderungen durch Wahrscheinlichkeits-VF je Klasse i
 $A_i(t) = P\{T_{Ai} \leq t\}$, $i = 1, 2$.
 - Beschreibung der Übertragungszeiten auf EBUS (T_{HE}) und ABUS (T_{HA}) durch Wahrscheinlichkeits-VF
 $H_E(t) = P\{T_{HE} \leq t\}$ bzw. $H_A(t) = P\{T_{HA} \leq t\}$
 - Konstante Ein- bzw. Ausgabe-Übertragungszeit h_{EAK} je Meldung bzw. Befehl.
 (Die Anzahl der pro I/O-Interrupt ein- bzw. auszugebenden Meldungen bzw. Befehle sei vorgebbar).
 - Beschreibung der ZP-Bedienungszeiten T_{HZi} durch Wahrscheinlichkeits-VF je Anforderungsklasse i
 $H_{Zi}(t) = P\{T_{HZi} \leq t\}$, $i = 1, 2$.
 - Konstante Taktperiode τ zwischen zwei I/O-Interrupts.

b) Modellkomponenten

Bedienungseinheiten: ZP, EAK, EBUS, ABUS

Warteschlangen: Q1 (Warten der Verkehrsquellen auf EBUS)
 Q2 (äußere Eingabeliste im EAW)
 Q3 (äußere Ausgabeliste im EAW vor ABUS)
 Q4 (innere Eingabeliste vor ZP)
 Q5 (innere Ausgabeliste vor EAK)

Vorrangsteuerwerk: Unterbrechung des ZP zur Ein/Ausgabe (SCHEDULER)

c) Modell (vergl. Bild 3.3)

Zuordnung von Bedienungseinheiten zu Bedienungszuständen der Anforderungen:

ZP:	α	\longleftrightarrow	Z0	(Klasse 0)
	β	\longleftrightarrow	Z6	(Klassen 1,2)
EAK:	γ_1	\longleftrightarrow	Z4	(Klassen 1,2)
	γ_2	\longleftrightarrow	Z8	(Klasse 2)
EBUS:	δ	\longleftrightarrow	Z2	(Klassen 1,2)
ABUS:	ϵ	\longleftrightarrow	Z10	(Klasse 2)

Zuordnung von Warteschlangen zu Wartezuständen der Anforderungen:

Q1	\longleftrightarrow	Z1	(Klassen 1,2)
Q2	\longleftrightarrow	Z3	(Klassen 1,2)
Q3	\longleftrightarrow	Z9	(Klasse 2)
Q4	\longleftrightarrow	Z5	(Klassen 1,2)
Q5	\longleftrightarrow	Z7	(Klasse 2)

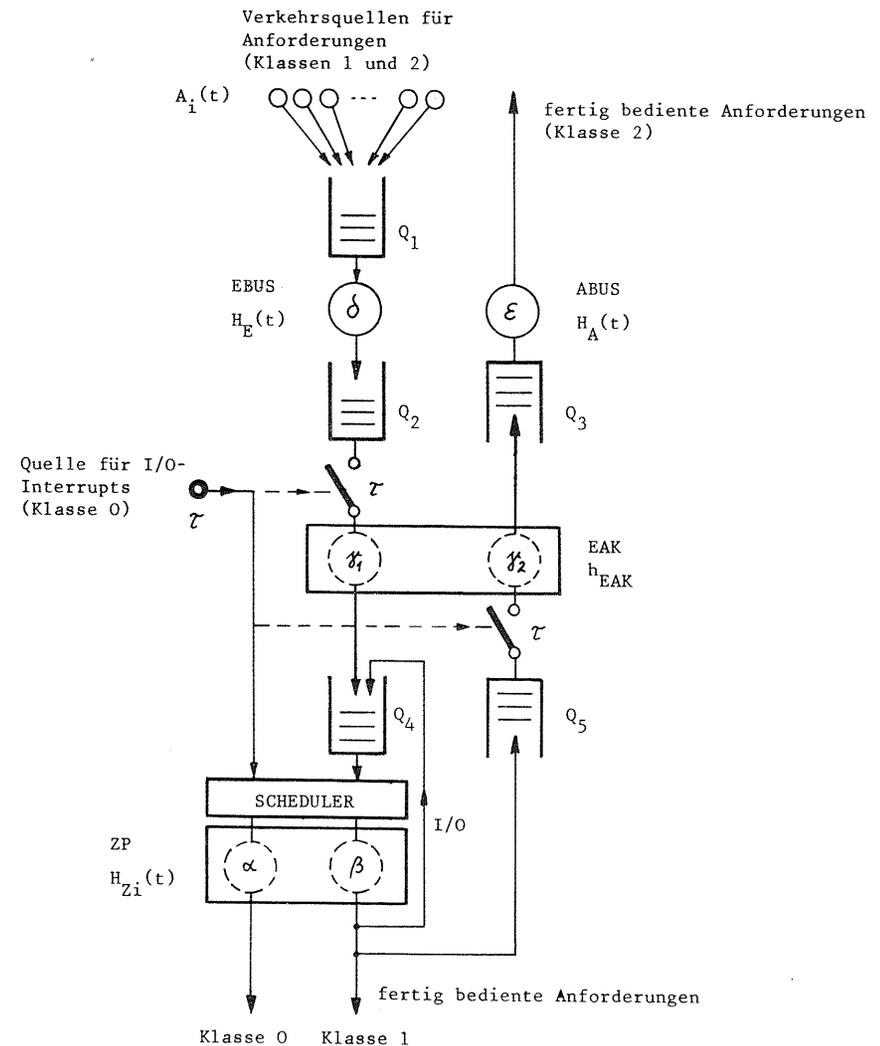


Bild 3.3. Modell zur Beschreibung des Informationsaustausches zwischen Zentralprozessor ZP und Peripherie

Zu dem Modell nach Bild 3.3 ist noch anzumerken, daß die Dauer der durch den I/O-Interrupt ausgelösten Ein/Ausgabe (Belegungen γ_1, γ_2 von EAK) von der Anzahl der übertragenen Anforderungen abhängt. Parallel dazu befindet sich der ZP im Zustand α .

3.3.2 Beispiel 2: Informationsaustausch zwischen Zentralprozessor und Kanalwerk in Rechnersystemen mit zweistufiger Speicherhierarchie

Betrachtet werde das Beispiel des Funktionsablaufes zwischen Zentralprozessor ZP und Kanalwerk KW aus Abschnitt 2.3.5. Die Schritte zur Umsetzung in ein Modell seien im folgenden kurzgefaßt dargestellt.

a) Charakterisierung des realen Ablaufgeschehens

- Betriebsmittel:
- Zentralprozessor ZP (Bedienung von Benutzer- u. Systemprogrammen)
 - Kanalwerk KW (Bedienung von Benutzerprogrammen)
 - Arbeitsspeicher ASP (Speicherung von Benutzer-u. Systemprogrammen)
 - Hintergrundsp. HSP (Speicherung von Benutzerprogrammen)
 - Betriebssystem BS (ASP-residente Systemprogramme Listen von Programm-Anforderungen)

- Anforderungen:
- Benutzerprogramme bzw. von Benutzerprogrammen ausgelöste Systemprogramme
- Herkunft und Ziel der Benutzerprogramme ist die Rechnerperipherie (nicht erfaßt im Modell)
- Alle Benutzerprogramme sind gleichrangig (1 Klasse)

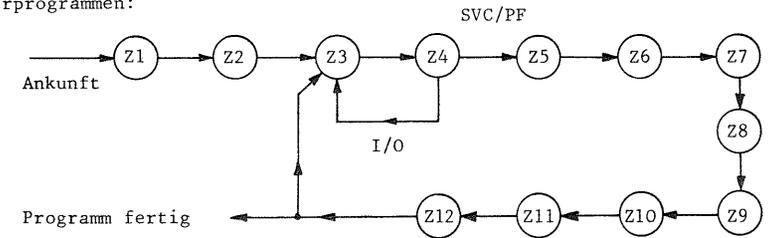
- Ereignisse:
- Ankunft eines neuen Benutzerprogrammes
 - Bedienungsende des ZP durch Benutzerprogramm (SVC,PF-Interrupt)
 - Bedienungsende des ZP durch Systemprogramm
 - Bedienungsende des KW durch Benutzerprogramm (I/O-Interrupt)

- Zustände von Benutzer-progr.:
- Z1 Warten auf Initiierung
 - Z2 Initiierung durch Systemprogramm (ZP)
 - Z3 Warten auf ZP-Zuteilung
 - Z4 Bearbeitung (ZP)
 - Z5 Unterbrechungsbehandlung (SVC,PF) durch Systemprogramm (ZP)
 - Z6 Warten auf Seitentransfer
 - Z7 Initiierung des Seitentransfers durch Systemprogramm (ZP)
 - Z8 Seitentransfer (KW)
 - Z9 Warten auf Unterbrechungsbehandlung (I/O)
 - Z10 Unterbrechungsbehandlung (I/O) durch Systemprogramm (ZP)
 - Z11 Warten auf Auswahl
 - Z12 Auswahl durch Systemprogramm (ZP)

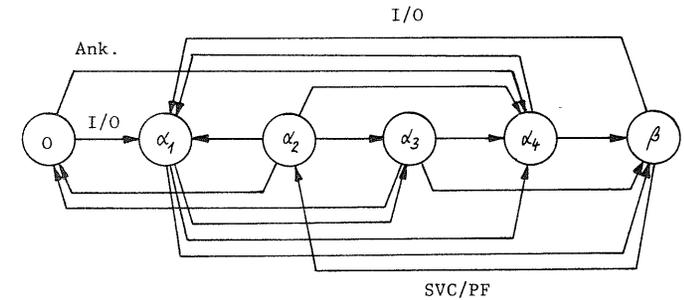
- Zustände des ZP:
- 0 ZP frei
 - α_1 Behandlung von I/O-Interrupts
 - α_2 Behandlung von SVC- bzw. PF-Interrupts
 - α_3 Auswahl eines Benutzerprogrammes für I/O
 - α_4 Auswahl eines Benutzerprogrammes für ZP-Bearbeitung
 - β Bearbeitung von Benutzerprogrammen

- Zustände des KW:
- 0 KW frei
 - γ Ausführung eines Seitentransfers

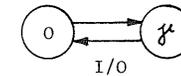
Zustandsübergänge von Benutzerprogrammen:



Zustandsübergänge des ZP nach Prioritätsregeln (Abschn. 2.3.5)



Zustandsübergänge des KW:



- Zeitliche Zusammenhänge:
- "Ankunftszeitpunkte" von Benutzerprogrammen entsprechen in dem betrachteten Teilsystem den Zeitpunkten abgeschlossener Eingabephasen (spooling)
 - α -Bedienungszeiten des ZP sind im wesentlichen systembedingt
 - β -Bedienungszeiten des ZP sind programm- und ASP-zustandsbedingt
 - γ -Bedienungszeiten sind im wesentlichen systembedingt.

b) Modellkomponenten

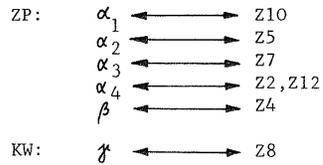
Bedienungseinheiten: ZP, KW

Warteschlangen: Listen des Betriebssystems (Pufferung von Anforderungen vor ZP)

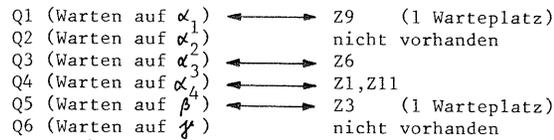
Vorrangsteuerwerk des ZP: Unterbrechungs- und Prioritätssteuerung entsprechend der Prioritätsregeln nach Abschn. 2.3.5 (Funktionsblock SCHEDULER)

c) Modell (vergl. Bild 3.4)

Zuordnung von Bedienungseinheiten zu Bedienungszuständen der Anforderungen:



Zuordnung von Warteschlangen zu Wartezuständen der Anforderungen:



Modellstruktur:

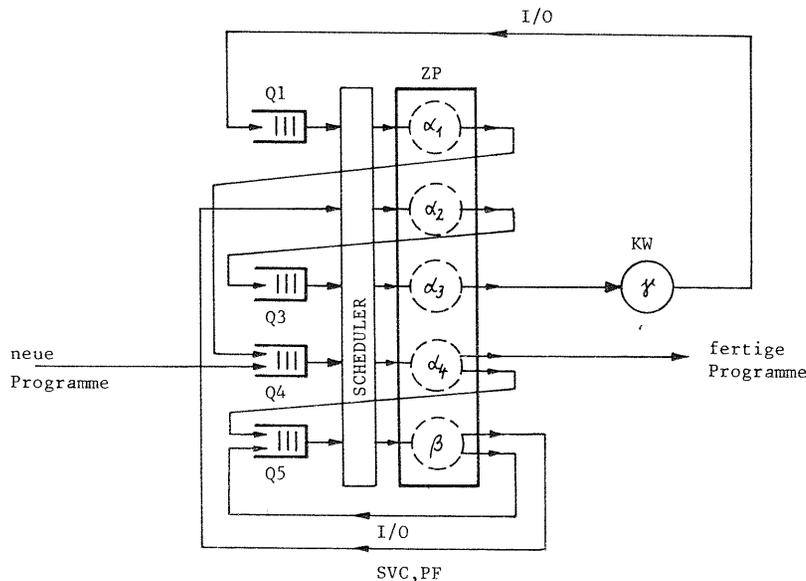


Bild 3.4. Modellstruktur zur Beschreibung des Ablaufgeschehens beim Informationsaustausch zwischen Zentralprozessor ZP und Kanalwerk KW

Zur vollständigen Charakterisierung des Modelles wären noch explizite Angaben über den Zugang von Benutzerprogrammen, die einzelnen Bedienungsphasen sowie die Gesamt-ZP-Zeit der Benutzerprogramme erforderlich (vergl. hierzu Kap. 7).

d) Vereinfachung des Modells

Die Phasen α_1 zur I/O-Interrupt-Behandlung bzw. α_3 zum Starten des Kanalprogrammes sind i.a. wesentlich kürzer als die Phase α_2 . Durch Vernachlässigung der Phase α_1 entfallen Q_1 und Q_5 . Wird auch die Phase α_3 vernachlässigt, kann die Warteschlange Q_3 direkt vor der Bedienungseinheit KW angeordnet werden. In dem verbleibenden Modell wird jede β -Phase von einer α_4 -Phase eingeleitet und - wie bisher - mit einer α_2 -Phase fortgesetzt; zur Systemanalyse können die Phasen α_2 und α_4 zu einer resultierenden Overhead-Phase α zusammengefaßt und der Phase β vorangestellt werden (die fertig bearbeiteten Programme durchlaufen dann allerdings zusätzlich eine weitere β - und α_2 -Phase). Die vereinfachte Modellstruktur zeigt Bild 3.5.

Gestrichelt ist in Bild 3.5 ferner der Fall angedeutet, bei dem ein fertiges Programm momentan durch ein neues ersetzt wird; man erhält in diesem Falle ein sog. "geschlossenes Warteschlangennetz" mit einer stets konstanten Anzahl von Anforderungen (konstanter Multiprogrammgrad).

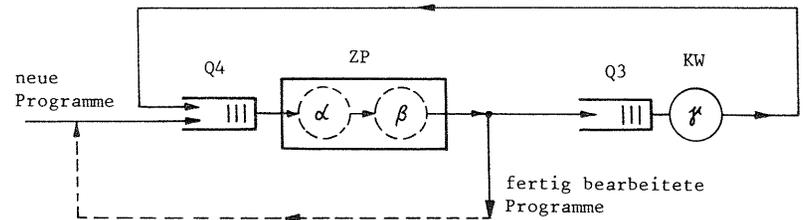


Bild 3.5. Vereinfachte Modellstruktur

3.4 Systematik verkehrstheoretischer Modelle

Verkehrstheoretische Modelle lassen sich in systematischer Weise hinsichtlich der verschiedenen Modellkomponenten klassifizieren. Dieser Abschnitt gibt einen kurzgefaßten Überblick über Klassifizierungsmerkmale und Modellvarianten.

3.4.1 Klassifizierungsmerkmale und Kurznotation

Merkmale zur Klassifizierung verkehrstheoretischer Modelle sind

- Struktur des Modelles
- Betriebsart des Modelles
- Ankunfts- und Bedienungsprozesse des Modelles.

Die bekannteste Grundform eines Modelles besitzt nur eine "Stufe" der Bedienung, nach deren Durchlaufen das System verlassen wird. Für derartige Systeme soll eine Kurznotation verwendet werden, welche durch eine leichte Verallgemeinerung gegenüber der von Kendall [21] eingeführten Kurznotation entsteht:

$$X_A/X_H/n(k)-s/Y,Z,\dots$$

mit folgender Bedeutung:

X_A	Kurzbezeichnung des Ankunftsprozesses, z.B.
M	Markoff'scher Prozeß (exponentiell verteilte Ankunftsabstände)
D	Determinierter Prozeß (konstante Ankunftsabstände)
E_k	Erlang-verteilte Ankunftsabstände k-ter Ordnung
H_k	Hyperexponentiell verteilte Ankunftsabstände k-ter Ordnung
GI	Allgemein verteilte und voneinander unabhängige Ankunftsabstände (general independent)
CB	Getaktete Gruppenankünfte (clocked batch input)
X_H	Kurzbezeichnung des Bedienungsprozesses, z.B.
M	Exponentiell verteilte Bedienungsdauern
D	Konstante Bedienungsdauern
E_k	Erlang-verteilte Bedienungsdauern k-ter Ordnung
H_k	Hyperexponentiell verteilte Bedienungsdauern k-ter Ordnung
G	Allgemein verteilte Bedienungsdauern
C	Getaktete Abfertigung (clocked service)
n	Anzahl der Bedienungseinheiten in dieser Stufe
k	Erreichbarkeit (Anzahl der von einem Eingang her zugänglichen Bedienungseinheiten). Die Angabe erfolgt nur, falls $k < n$.
s	Anzahl der Wartplätze. Die Angabe erfolgt nur, falls s beschränkt ist.
Y,Z,...	Weitere Angaben zum Modell, z.B. Anzahl der Verkehrsquellen, Betriebsweise etc.

Beispiel: Kurznotation des Modelles für ein Warteverlustsystem aus Abschnitt 3.2.2.d):

$$GI/G/n-s/FIFO.$$

3.4.2 Modelle mit einstufiger Bedienung

Modelle mit einstufiger Bedienung sind dadurch gekennzeichnet, daß eintreffende Anforderungen eine Bedienstufe durchlaufen und danach das System verlassen. Dabei kann die Bedienstufe mehrfach durchlaufen werden (z.B. bei Rückkopplungen oder bei Unterbrechungen). Die wichtigsten Modellstrukturen einstufiger Bedienungssysteme sind in Bild 3.6 zusammengestellt und weiter unten kommentiert.

Die in Bild 3.6 nur nach groben Strukturkriterien unterschiedenen Modelle werden durch weitere Angaben spezifiziert hinsichtlich

- spezieller struktureller Merkmale (z.B. Gruppierungstypen bei mehrstufigen Koppelanordnungen etc.)
- Zuteilungs-, Abfertigungs- und Prioritätsstrategien sowie hinsichtlich des Blockierungsverhaltens
- Ankunfts- und Bedienungsprozesse (vergl. hierzu Abschnitt 3.2.2c)

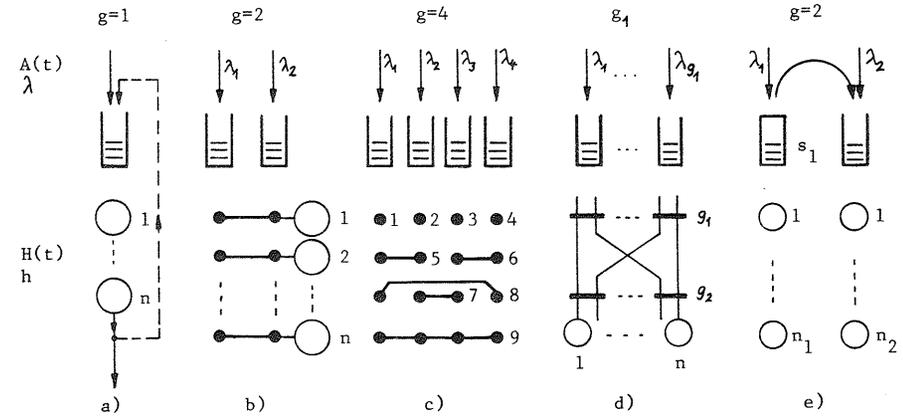


Bild 3.6. Strukturen einstufiger Bedienungssysteme

- a) Modelle mit einer Eingangs-Warteschlange ohne/mit Rückkopplung
- b) Modelle mit parallelen Eingangs-Warteschlangen und vollkommener Erreichbarkeit
- c) Modelle mit parallelen Eingangs-Warteschlangen und unvollkommener Erreichbarkeit (Mischungen)
- d) Modelle mit parallelen Eingangs-Warteschlangen und mehrstufigen Koppelanordnungen (Linksysteme)
- e) Modelle mit Überlaufmöglichkeit

a) Modelle mit einer Eingangs-Warteschlange ohne/mit Rückkopplung

Das wohl am häufigsten angewandte Grundmodell besitzt n vollkommen erreichbare Bedienungseinheiten, welche von einem Eingang her abgesehen werden. Wartende Anforderungen werden in einer Eingangs-Warteschlange zwischengespeichert. Nach Durchlaufen der Bedienungseinheit wird das System verlassen oder eine erneute Bedienung angefordert (Rückkopplung).

b) Modelle mit parallelen Eingangs-Warteschlangen und vollkommener Erreichbarkeit

In Erweiterung zu a) werden Anforderungen klassenweise gruppiert (nach Herkunft, Dringlichkeit etc.) und entsprechend abgefertigt (interqueue discipline). Die Anzahl der Gruppen (Klassen) sei $g > 1$.

c) Modelle mit parallelen Eingangs-Warteschlangen und unvollkommener Erreichbarkeit (Mischungen)

Gegenüber Modell b) können von einem Eingang her nur k der n Bedienungseinheiten (k < n) abgesucht (erreicht) werden, z.B. die Bedienungseinheiten Nr. 2, 5, 7 und 9 von Eingang 2 aus in Bild 3.6c). Derartige Anordnungen entstehen durch verschiedenartiges Verschalten der Ausgänge von einstufigen Koppelanordnungen. Die Art und Weise der Verschaltung heißt "Mischung".

d) Modelle mit parallelen Eingangs-Warteschlangen und mehrstufigen Koppelanordnungen (Linkssysteme)

Bei diesen Modellen werden die Bedienungseinheiten von den Eingängen her über ein mehrstufiges Koppelnetzwerk angeschaltet. Das Koppelnetzwerk besteht aus mehreren hintereinander angeordneten Koppelvielfachen, welche netzartig über Zwischenleitungen (Links) miteinander verbunden sind. Ein Weg wird nur dann durchgeschaltet, wenn ein freier Pfad von dem Eingang bis zu einem freien Bedienungsorgan besteht (konjugierte Durchschaltung). In Bild 3.6d) ist schematisch ein 2-stufiges Linkssystem eingezeichnet.

e) Modelle mit Überlaufmöglichkeit

Eine eintreffende Anforderung wird, abhängig vom angetroffenen Belegungszustand, entweder in der Ankunftsgruppe bedient oder umgelenkt ("Überlauf") auf eine Folgegruppe. Diese Umlenkung auf ein "Sekundärsystem" kann bei vollbelegten Bedienungseinheiten bzw. bei vollbelegten Bedienungseinheiten und Warteplätzen des "Primärsystems" erfolgen (alternative Wegelenkung).

3.4.3 Modelle mit mehrstufiger Bedienung (Netzmodelle)

Netzmodelle entstehen durch "Zusammenschalten" von einstufigen Bedienungssystemen, vergl. Beispiele zur Modellbildung in Abschn. 3.3. Bild 3.7 zeigt zwei prinzipielle Netztypen:

- a) Offene Warteschlangennetze
- b) Geschlossene Warteschlangennetze

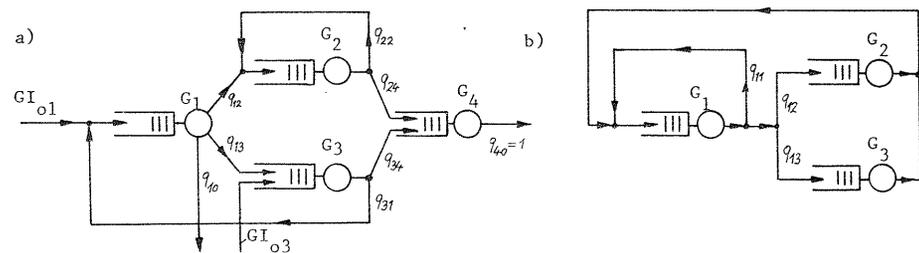


Bild 3.7. Strukturen mehrstufiger Bedienungssysteme
 a) Offenes Warteschlangenmodell
 b) Geschlossenes Warteschlangenmodell

a) Offene Warteschlangennetze

"Offene" Netze sind charakterisiert durch das Vorhandensein von mindestens einem Eingang und mindestens einem Ausgang von bzw. zur "Außenwelt" des Netzmodelles. Die Struktur des Netzes kann mittels einer Strukturmatrix beschrieben werden:

$$\underline{S} = (s_{ij}) \text{ mit } s_{ij} = 1(0) \text{ falls ein(kein) Weg zwischen Knoten } i \text{ und } j \text{ existiert, } i = 0, 1, \dots, N, j = 0, 1, 2, \dots, N$$

N = Anzahl der Netzknoten (Bedienungssysteme)
 (Knoten "0" entspricht der "Außenwelt").

Zusätzlich müssen Angaben über Ankunfts- und Bedienungsprozesse, Zuteilungs-, Abfertigungs-, Prioritäts- und Verzweigungsstrategien gemacht werden. Die "Wegelenkung" der einzelnen Anforderungen durch das Netz kann dabei z.B. fest, alternativ (abhängig vom Knotenzustand) oder adaptiv erfolgen. In vielen Fällen kann die Wegelenkung mit Hilfe unabhängiger Wahrscheinlichkeiten in Form einer sog. Verzweigungsmatrix beschrieben werden:

$$\underline{Q} = (q_{ij}) \text{ mit } q_{ij} = \text{Wahrscheinlichkeit für Übergang einer Anforderung nach Knoten } j \text{ bei Verlassen des Knotens } i, i = 1, 2, \dots, N, j = 0, 1, \dots, N.$$

Werden Klassen von Anforderungen unterschieden, so können Ankunfts- und Bedienungsprozesse sowie Verzweigungsstrategien klassenabhängig definiert werden. Ferner ist anzugeben, ob bzw. nach welcher Vorschrift Anforderungen ihre Klassenzugehörigkeit wechseln.

Ein wichtiger Sonderfall der offenen Warteschlangennetze sind serielle Warteschlangennetze, welche durch Reihenschaltung einzelner einstufiger Warteschlangennetze entstehen.

b) Geschlossene Warteschlangennetze

Geschlossene Netze besitzen weder Eingänge noch Ausgänge zur Außenwelt. Sie sind ferner dadurch gekennzeichnet, daß innerhalb des Netzes stets eine konstante Anzahl von Anforderungen zirkuliert.

Wie bei offenen Netzen können die Anforderungen zusätzlich nach Klassen eingeteilt sein. Ferner können auf dieser Basis Mischformen aus offenen und geschlossenen Netzen aufgebaut werden. Ein spezieller Sonderfall der geschlossenen Warteschlangennetze sind serielle Warteschlangennetze mit einer Rückkopplung vom Ausgang auf den Eingang; man spricht hierbei auch von zyklischen Warteschlangenmodellen.

3.4.4 Allgemeine Bemerkungen

Die in diesem Kapitel beschriebene Methodik der Modellbildung und Zusammenstellung der wichtigsten Verkehrsmodelltypen stellt einen Versuch dar, dieses weitläufige Gebiet zu systematisieren und zu klassifizieren. Die hier angestellten Überlegungen erheben nicht den Anspruch auf Vollständigkeit, es sollte vielmehr das Charakteristische zum Ausdruck gebracht werden. Eine detailliertere Darstellung ist nur hinsichtlich speziellerer Teilgebiete sinnvoll durchführbar und würde den Rahmen dieser Arbeit erheblich übersteigen.

Das Gebiet der Modellbildung und Modellanalyse erfährt gegenwärtig eine stürmische Entwicklung und eine dementsprechend große Beachtung in der Fachwelt. Innerhalb der Nachrichtenvermittlungs- und Rechnertechnik gehört das Gebiet als fester Bestandteil zur Systementwicklung und Planung. Die wissenschaftlichen Beiträge überspannen ein beachtliches Spektrum von der Komponententwicklung (Hardware) bis zu analytischen Methoden des Operations Research. Einzelbeiträge können aus diesem Grunde nur exemplarisch gewürdigt werden. Stattdessen sei hier auf Schriftumsreihen einiger für dieses Gebiet richtungsweisender Institutionen verwiesen, deren Arbeiten einen stetigen Beitrag zur Entwicklung dieses Gebietes darstellen, vgl. [22 - 31].

Nach der Behandlung der Modellbildung wird im nachfolgenden Kapitel versucht, das Gebiet der Modellanalyse entsprechend zu umreißen.

4. METHODEN DER MODELLANALYSE

Aufgabe dieses Kapitels ist eine kurze Darstellung der wichtigsten Methoden der Modellanalyse. Neben experimentellen Methoden wird im wesentlichen auf exakte und approximative analytische Methoden eingegangen. Zu den exakten analytischen Methoden kann auf umfangreiche Darstellungen in der einschlägigen neueren Literatur verwiesen werden; sie sollen aus diesem Grunde nur von ihrer prinzipiellen Seite her betrachtet werden. Auf dem oft von den Anwendungen her geprägten und deshalb überwiegend nur in der Quell-Literatur behandelten Gebiet der approximativen Verfahren soll hier der Versuch einer Klassifizierung unternommen werden, welche sich an grundsätzlichen Prinzipien orientiert. Das Kapitel wird beschlossen mit einem Überblick über die bekanntesten Ergebnisse der Bedienungstheorie.

4.1 Experimentelle Methoden

Unter "experimentellen" Methoden sollen Verfahren verstanden werden, bei denen die Analyse durch Messungen⁺⁾ am realen bzw. einem simulierten Ablaufgeschehen erfolgt.

4.1.1 Messung

Die Leistung einzelner Betriebsmittel oder eines ganzen Systems kann meßtechnisch auf 2 prinzipiell verschiedene Weisen erfolgen [10,32,33]:

- a) Hardware-Messungen
- b) Software-Messungen

a) Hardware-Messungen

Hardware-Messungen werden i.a. immer dann angewandt, wenn sich die Meßgröße eines Betriebsmittels durch festes Anschalten von "Meßfühlern" parallel zum realen Ablaufgeschehen aufzeichnen läßt. Bild 4.1a) zeigt als Beispiel die Ermittlung von Befehlshäufigkeiten durch Anschalten einer Decodier- und Aufzeichnungseinrichtung parallel (on-line) am Zentralprozessor. Die Befehlsstatistik kann durch separate (off-line) Auswertung der aufgezeichneten Daten erfolgen.

^{+) An realen Systemen vorgenommene Messungen werden hier unter "Modellanalyse" mit behandelt. Tatsächlich ist es bei vielen Messungen erforderlich, ein "Modell" einer repräsentativen Last zugrunde zu legen. Schließlich gehört auch die Messung als Hilfsmittel bei der Modellbestätigung zur "Modellanalyse".}

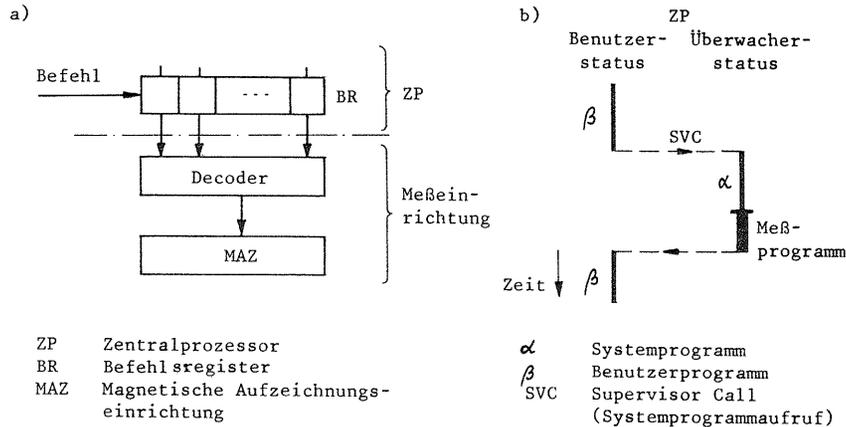


Bild 4.1. Prinzip von Hardware- und Software-Messungen
 a) Hardware-Messung von Befehlsstatistiken
 b) Software-Messung von I/O-Vorgängen

b) Software-Messungen

Programmgesteuerte Abläufe sind häufig nur mit erheblichem Aufwand meßtechnisch erfaßbar. Hier kann die Meßgröße oft schneller durch programmgesteuerte Meßprogramm-Aufrufe ermittelt werden, indem an bestimmten Unterbrechungszeitpunkten ein Meßprogramm eingeschleift wird. Das Beispiel Bild 4.1b) zeigt, wie man beispielsweise I/O-Vorgänge hinsichtlich Typ, Kanal, Umfang etc. erfassen kann; hierzu wird nach jedem Systemprogramm-Aufruf (SVC) zusätzlich ein Meßprogramm eingeschleift, welches die Zustandsverhältnisse auswertet. Die Aufzeichnung so gewonnener Programmeigenschaften ist unter dem Begriff "tracing" bekannt. Ähnliche Methoden werden angewandt zur Bestimmung von Ausführungszeiten typischer Routinen (kernels) oder Programme (benchmarks). Ein dem Hardware-Meßverfahren ähnlicheres Verfahren ist das "sampling"-Verfahren, wobei in konstanten Zeitabständen eine künstliche Unterbrechung zwecks Meßprogrammaufruf eingeleitet wird.

Die Problematik bei Messungen liegt einmal in der möglichen Beeinflussung des realen Ablaufgeschehens durch die Messung selbst. Messungen sind ferner i.a. auf ein bestimmtes Objekt beschränkt, aufwendig und wenig flexibel. Für Projektierungszwecke sind daher Modelluntersuchungen geeigneter; die Genauigkeit der Modell-Aussagen kann jedoch nur mittels Messungen kontrolliert werden.

4.1.2 Simulation

Unter Simulation wird die Nachbildung des dynamischen Ablaufgeschehens auf einem Rechner mittels eines Simulationsprogrammes verstanden. Die Simulation stützt sich dabei i.a. wieder auf Modellvorstellungen von der Realität. Struktur und Betriebsart des Modelles werden durch eine entsprechende Programmstruktur festgelegt. Stochastische Größen, wie Ankunftsabstände und Bedienungsdauern, werden künstlich mit Hilfe gleichverteilter Zufallszahlen nach Mittelwert und Typ einer vorgegebenen VF erzeugt.

a) Prinzip der zeitreuen Simulation

Die Simulation erfolgt i.a. nach einem zeitreuen Verfahren, bei welchem das Ablaufgeschehen durch Ereignisse (Zustandsänderungen) zu diskreten Zeitpunkten charakterisiert wird. Hierzu wird eine Variable ZEIT eingeführt und ein "Kalender" geführt für die Eintragung von Ereignissen. Zeitspannen zur Bestimmung von zukünftigen Ereignissen werden, wie oben schon angedeutet, nach vorgegebenen VF mittels gleichverteilter Zufallszahlen ermittelt (bei zufallsabhängigen Ereignissen) bzw. sind determiniert festgelegt. Während der Simulation "springt" das Programm von einem Ereigniszeitpunkt zum nächsten (event by event simulation), wobei zu jedem Ereigniszeitpunkt der Zustand des Systems auf den neuesten Stand gebracht wird sowie Zählgrößen zur statistischen Auswertung (Messungen von Wartezeiten, Auslastungen etc.) erneuert werden.

In dem beschriebenen Verfahren treten nur zeitdiskrete Zustandsänderungen auf zu den Zeitpunkten t_0, t_1, \dots . Das Programm springt von Ereigniszeitpunkt zu Ereigniszeitpunkt ("event by event" simulation), wobei die dazwischenliegenden Zeiten nur "simuliert" werden. Auf diese Weise verläuft die Simulation des Ablaufgeschehens auf dem Rechner i.a. zeitlich sehr stark gerafft ab. Die Laufzeit des Simulationsprogrammes hängt nur ab von der Anzahl der simulierten Ereignisse sowie dem Organisationsaufwand für Zustands- und Zählgrößenberichtigung.

Organisatorisch besteht ein Simulationsprogramm aus 2 Hauptteilen, dem Kernprogramm und einem Rahmenprogramm. Das Kernprogramm bildet das eigentliche Modell nach und übernimmt sämtliche Organisations- und Meßaufgaben, welche bei jedem Ereigniszeitpunkt anfallen. Nach der Abarbeitung eines jeden Ereignisses wird in das übergeordnete Rahmenprogramm gesprungen, welches die Überwachung der Ereignisanzahl sowie Meßauswertungen übernimmt. Zur Überwachung der Ereignisanzahl gehört u.a. auch die zeitliche Unterteilung eines Simulationslaufes in einen Vorlauf und einzelne Teiltests. Der Vorlauf dient zum Einschwingen des Systems (bei Simulation von stationären Prozessen). Um eine statistische Aussage-sicherheit ermitteln zu können, wird i.a. der Simulationslauf in eine Anzahl

von Teiltests unterteilt, z.B. 10 Teiltests mit je 10 000 Ereignissen. Aus den Schwankungen der Teiltestergebnisse kann ein Vertrauensintervall für eine vorgegebene Aussagesicherheit ermittelt werden.

Zur übersichtlicheren und schnelleren Simulationsprogramm-Erstellung wurden auch verschiedene Simulationssprachen entwickelt wie GPSS, SIMULA, SIMSCRIPT, TETRASIM u.a.m., welche aber i.a. wesentlich größere Simulationszeiten erfordern im Vergleich zu normalen problemorientierten Sprachen wie ALGOL und FORTRAN. Häufig wird aus diesem Grunde ein Zwischenweg beschritten, indem für den jeweiligen Anwendungsfall oft wiederkehrende Vorgänge in Form von Codeprozeduren realisiert werden, welche in ein ALGOL- oder FORTRAN-Programm eingebunden werden.

Zur vertiefenden Diskussion von Simulationstechniken sei auf [34-38] verwiesen.

b) Prinzip der Ruf-Löschzahlenmethode

Im speziellen Falle exponentiell verteilter Ankunfts- und Bedienungsprozesse kann ein geringfügig vereinfachtes Simulationsverfahren angewandt werden [34,35]. In diesen Fällen ist der Zustandsprozeß durch exponentiell verteilte Aufenthaltszeiten in den einzelnen Zuständen ausgezeichnet (wobei der jeweilige Mittelwert dieser zufallsabhängigen Aufenthaltszeiten in beliebiger Weise vom Systemzustand selbst abhängen kann).

Die Ruf-Löschzahlenmethode (auch Roulette-Methode genannt) nutzt explizit die Markoffsche Eigenschaft dieser Prozesse aus, indem in der Simulation anstelle von Zeiten die Übergangswahrscheinlichkeiten realisiert werden. Von einem neu erreichten Zustand werden dabei die Übergangswahrscheinlichkeiten für alle möglichen Übergänge betrachtet, und es wird mit Hilfe gleichverteilter Zufallszahlen einer dieser Übergänge realisiert unter Berücksichtigung einer den Übergangswahrscheinlichkeiten entsprechenden Längenaufteilung des Zufallszahlenintervalls (0,1). Die Dauer einer bis zu dieser Zustandsänderung noch nicht abgeschlossenen Teilphase (z.B. belegte Bedienungseinheit) beeinflusst wegen der Markoffschen Eigenschaft (vergl. hierzu Abschn. 4.2.2) den zukünftigen Verlauf des Prozesses nicht; ihre "Enderate" kann zur Bestimmung der Übergangswahrscheinlichkeit für die nächste Zustandsänderung wieder "voll" berücksichtigt werden.

Der Name dieser Methode rührt daher, daß Zustandsänderungen i.a. nur durch eintreffende Anforderungen ("Rufe") bzw. endigende Bedienungseinheiten ("Löschchen") ausgelöst werden können. Man teilt daher das Zufallszahlenintervall in einen "Rufzahlenteil" und einen "Löschzahlenteil" entsprechend der momentanen Übergangswahrscheinlichkeiten ein.

c) Beispiel

In Bild 4.2 ist das Prinzip der Kalenderorganisation für ein Wartesystem des Typs GI/G/1-s/FIFO entsprechend Bild 3.1 angegeben (s=3). Für jedes Betriebsmittel (Bedienungseinheit, Warteplatz) wird eine Speicherzelle reserviert. In die Speicherzelle der belegten Bedienungseinheit wird der Zeitpunkt des Bedienungsendes, in eine Speicherzelle eines belegten Warteplatzes der Zeitpunkt des Wartebeginns eingeschrieben. Aus organisationstechnischen Gründen werden die Speicherinhalte für belegte Warteplätze nicht nach jedem Bedienungsende umorganisiert entsprechend dem Vorrücken wartender Anforderungen; es wird vielmehr durch eine Verzögerung die Zuordnung zwischen tatsächlich belegtem Warteplatz und Speicherzelle im Simulationsprogramm auf den neuesten Stand gebracht. Nichtbelegte Einheiten werden besonders gekennzeichnet, z.B. durch eine negative Zahl.

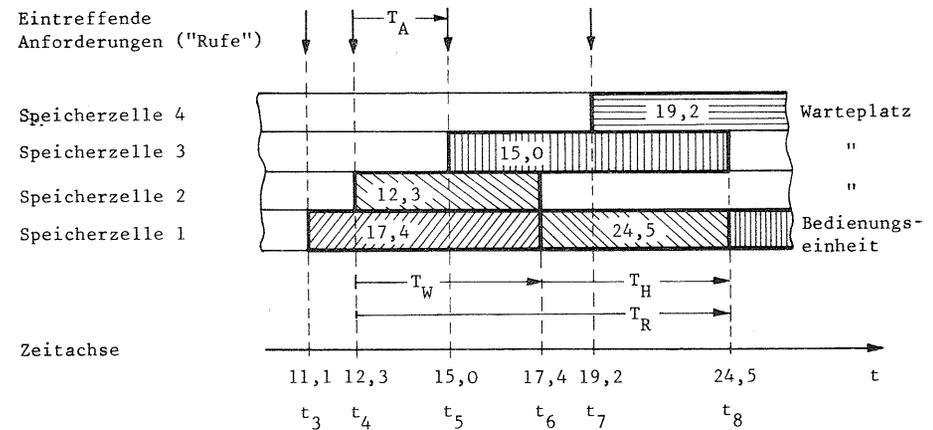


Bild 4.2. Beispiel für eine Kalenderorganisation bei zeittreuer Simulation

Zum Zeitpunkt t₃ = 11,1 treffe eine Anforderung ein, welche eine zufällige Bedienungsdauer T_H = 6,3 benötige. Da das System zu diesem Zeitpunkt gerade leer ist, belegt sie die Bedienungseinheit, indem in die Speicherzelle 1 der Zeitpunkt des Bedienungsendes t₃+T_H = 17,4 eingeschrieben wird. Der nächste Ankunftsabstand werde ermittelt zu T_A = 1,2, d.h. es trifft die nächste Anforderung zum Zeitpunkt t₃+T_A = 12,3 ein. Dies ist auch zugleich der nächste Ereigniszeitpunkt, da die Bedienungseinheit über den Zeitpunkt t₄ = 12,3 hinaus belegt bleibt. Im Simulationsprogramm wird nun zum nächsten Ereigniszeitpunkt t₄ gesprungen. Die eintreffende Anforderung muß warten. In die Speicherzelle 2 wird der Zeitpunkt des Wartebeginns t₄ = 12,3 eingetragen. Es wird der nächste Ankunftsabstand T_A ermittelt, z.B. T_A = 2,7. Das nächste Ereignis ist die Ankunft dieser Anforderung zum Zeitpunkt t₅ = 15,0. Da sie ebenfalls warten muß, wird in Speicherzelle 3 der Zeitpunkt t₅ = 15,0 eingeschrieben. Es wird der nächste Ankunftsabstand T_A = 4,2 ermittelt. Das nächste Ereignis ist nun der Zeitpunkt t₆ = 17,4 für das Belegungsende der Bedienungseinheit. Zum Zeitpunkt t₆ wird die Speicherzelle 1 freigegeben und sofort belegt, bei FIFO also durch die wartende Anforderung

aus Speicherzelle 2. Mit der zufälligen Bedienungszeit dieser Anforderung von $T_H = 7,1$ wird die Bedienungseinheit bis zur Zeit 24,5 belegt. Die zufällige Wartezeit der Anforderung aus Speicherzelle 2 betrug $T_W = t_6 - t_4 = 5,1$, usw.

Die einzeln gemessenen (realisierten) Zufallsgrößen (Meßwerte) werden während der Simulation akkumuliert und am Ende jedes Simulations-Teilabschnittes ("Teilttest", s.o.) ausgewertet. Als Beispiele seien folgende charakteristische Verkehrsgrößen betrachtet:

$E[T_W]$	Mittlere Wartezeit eintreffender Anforderungen
$VAR[T_W]$	Varianz der Wartezeiten eintreffender Anforderungen
$W = P\{T_W > 0\}$	Wartewahrscheinlichkeit eintreffender Anforderungen
B	Verlustwahrscheinlichkeit eintreffender Anforderungen
Y	Auslastung der Bedienungseinheit (Verkehrswert)
Ω	Mittlere Warteschlangenlänge
$w(\leq t) = P\{T_W \leq t\}$	Wartezeit-VF eintreffender Anforderungen

Hierzu werden folgende Meßzähler geführt:

CA	Anzahl eintreffender Anforderungen im Teilttest
CY	Anzahl abgeschlossener Bedienungen im Teilttest
CW	Anzahl der Warteereignisse im Teilttest
CV	Anzahl der Verlustereignisse im Teilttest
$C(\tau_j)$	Anzahl eintreffender Anforderungen mit einer Wartezeit $\leq \tau_j$
T	Teilttestdauer

Die charakteristischen Verkehrsgrößen ergeben sich zu

$$E[T_W] = \frac{\sum_{i=1}^{CA} T_{Wi}}{CA} \quad (4.1) \quad VAR[T_W] = \frac{\sum_{i=1}^{CA} T_{Wi}^2}{CA} - E[T_W]^2 \quad (4.2)$$

$$W = \frac{CW}{CA} \quad (4.3) \quad B = \frac{CV}{CA} \quad (4.4)$$

$$Y = \frac{\sum_{i=1}^{CA} T_{Hi}}{T} \quad (4.5) \quad \Omega = \frac{\sum_{i=1}^{CA} T_{Wi}}{T} \quad (4.6)$$

$$w(\leq \tau_j) = \frac{C(\tau_j)}{CA} \quad , j = 0, 1, \dots \quad (4.7)$$

(Der Index i bezieht sich auf die einzelnen Realisationen von Zufallsvariablen)

4.2 Analytische Methoden

Die quantitative Untersuchung des dynamischen Ablaufgeschehens mit analytischen Methoden ist Gegenstand der Bedienungstheorie (Verkehrstheorie, Warteschlangentheorie). Die analytischen Methoden können von der prinzipiellen Betrachtungsweise her unterteilt werden in

- deterministisch-kombinatorische Methoden
- wahrscheinlichkeitstheoretische Methoden (stochastische Prozesse).

4.2.1 Deterministisch-kombinatorische Methoden

Unter den deterministisch-kombinatorischen Methoden werden Verfahren verstanden, mit denen das Systemverhalten analysiert bzw. optimiert wird unter bekanntem Voraussetzungen hinsichtlich des "Verkehrs". Durch dieses Vorauswissen bedingt, können oft sehr allgemeine Aussagen über Systemeigenschaften bewiesen werden. Da auf diesem Gebiet eine sehr ausführliche Literatur besteht [39,40,41], soll hier lediglich mit zwei Beispielen die Denkweise verdeutlicht werden.

a) Beispiel 1: Deterministische Prozessorzuteilung

Ein klassisches Problem der Betriebsmittelverwaltung ist die Zuteilung von Prozessoren an eine bekannte Auftragsmenge mit dem Ziel einer möglichst schnellen Abarbeitung. Aus der Zuteilungstheorie (theory of scheduling) ist eine Vielzahl von Lösungen bekannt, welche optimale Zuteilungsstrategien bzw. Grenzwerte von Bearbeitungszeiten angeben unter den unterschiedlichsten Voraussetzungen wie z.B.

- Vorrangbeziehungen hinsichtlich der Abarbeitbarkeit (precedence constraints)
- Unterbrechende bzw. nichtunterbrechende Prioritäten etc.

Als Beispiel werde die Zuteilung voneinander unabhängig bearbeitbarer Aufgaben ("tasks") A_1, \dots, A_N mit bekannten Bedienungszeiten h_1, \dots, h_N auf n gleichartige Zentralprozessoren ZP_1, \dots, ZP_n betrachtet. Es ist hierfür bekannt, daß die Zuteilungsstrategie LPT (Reihenfolge entsprechend "largest processing time first") suboptimal ist, wobei die Gesamtbearbeitungszeit höchstens $(\frac{4}{3} - \frac{1}{3n})$ -mal größer ist als bei der optimalen Strategie (OPT) [40]. Bild 4.3 zeigt beide Zuteilungszeitdiagramme im Falle $N = 5, n = 2$.

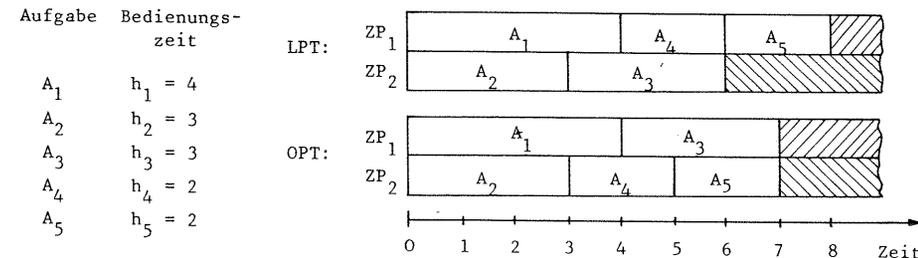
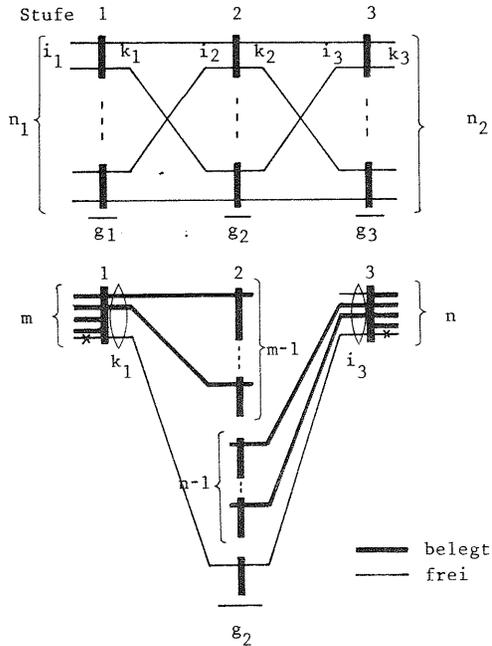


Bild 4.3. Beispiele für Mehrprozessor-Zuteilungen für unabhängig bearbeitbare Aufgaben

b) Beispiel 2: Blockierungsfreie mehrstufige Koppelanordnungen (Linksysteme)

Ein klassisches Problem der Vermittlungstechnik ist die Frage ideal blockierungsfreier mehrstufiger Koppelanordnungen. Eine Koppelanordnung heißt "ideal blockierungsfrei", wenn zwischen einem beliebigen freien Eingang und einem beliebigen freien Ausgang in jedem Belastungsfalle ein freier Weg durch das Koppelnetz geschaltet werden kann. Nach Clos [42] ist es stets möglich, Koppelanordnungen ungerader Stufenzahl ideal blockierungsfrei zu gestalten. Dies ist am Beispiel einer 3-stufigen Koppelanordnung in Bild 4.4 dargestellt.



Bedingungen für Blockierungsfreiheit:

$$\begin{aligned}
 i_1 &= m, & k_3 &= n \\
 k_1 &= i_3 = g_2 = m+n-1 \\
 g_1 &= i_2 = n_1/i_1 = n_1/m \\
 g_3 &= k_2 = n_2/k_3 = n_2/n
 \end{aligned}$$

Zum Beweis betrachte man den Fall, daß in einem Koppelvielfach der ersten bzw. dritten Stufe gerade noch ein Eingang bzw. Ausgang frei ist.
 Im ungünstigsten Falle führen die $(m-1)+(n-1) = m+n-2$ belegten Zwischenleitungen auf unterschiedliche Koppelvielfache der zweiten Stufe. Durch Hinzufügen eines weiteren Koppelvielfachs und der zugehörigen Zwischenleitungen, also $g_2 = m+n-1$, wird die Koppelanordnung blockierungsfrei.

Bild 4.4. Beispiel für eine ideal blockierungsfreie mehrstufige Koppelanordnung

Obwohl viele der durch deterministische bzw. kombinatorische Methoden gewonnenen Ergebnisse von großem theoretischen Interesse sind, gibt die Praxis Anlaß zu einer erweiterten Betrachtungsweise. Die Prozessor-Auftragslast ist beispielsweise nur in den seltensten Fällen im voraus bekannt. Blockierungsfreie Koppelanordnungen benötigen i.a. einen unvergleichbar hohen Aufwand an "Koppelpunkten" und sind daher in vielen praktischen Fällen nicht wirtschaftlich. Die praktisch orientierte Verkehrstheorie muß aus diesem Grunde nichtideale Voraussetzungen hinsichtlich Struktur, Betrieb oder Auftragslast einbeziehen, welche mit einer deterministisch-kombinatorischen Analysenmethode i.a. nicht befriedigend behandelt werden können, vergl. hierzu [68-69, 82-86].

4.2.2 Wahrscheinlichkeitstheoretische Methoden (Stochastische Prozesse)

Ausgangspunkt dieser Betrachtung ist die Tatsache, daß die in vielen technischen Systemen ablaufenden Vorgänge hinsichtlich der Eintreffzeitpunkte von Anforderungen bzw. Belegungszeiten von Betriebsmitteln so komplex sind, daß sie als zufallsabhängige (stochastische) Prozesse aufgefaßt und mit diesen Mitteln beschrieben werden können. Im folgenden werden die in einem Bedienungsmodell auftretenden Zufallsprozesse zunächst an einem Beispiel aufgezeigt und anschließend formal definiert bzw. klassifiziert.

a) Stochastische Prozesse in Bedienungssystemen

Betrachtet werde ein einstufiges Wartesystem des Typs GI/G/1 mit einer Bedienungseinheit, einer Warteschlange und der Abfertigungsdisziplin FIFO, vergl. Bild 4.5a):

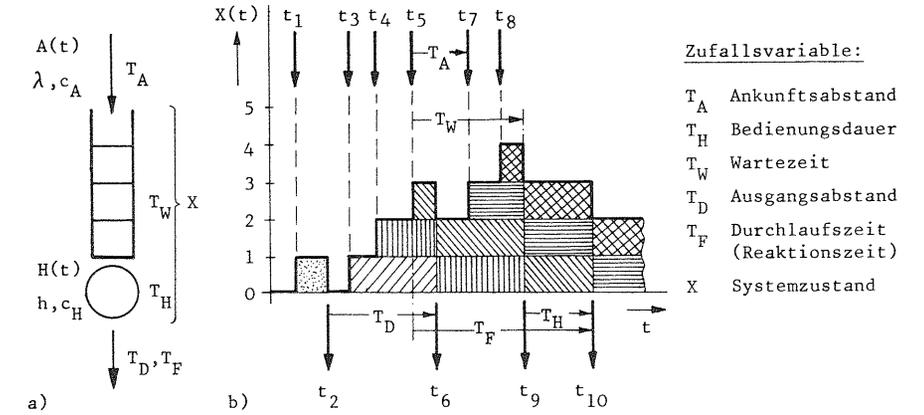


Bild 4.5. Ablaufgeschehen in einem einstufigen Wartesystem
 a) Modell
 b) Zeitdiagramm

Anforderungen werden von einem Ankunftsprozeß erzeugt. Der Ankunftsprozeß wird beschrieben durch die VF $A(t)$ der zufallsabhängigen Ankunftsabstände T_A mit Ankunftsrate λ und Variationskoeffizient c_A . Der Bedienungsprozeß ist charakterisiert durch die zufallsabhängigen Bedienungsdauern T_H mit VF $H(t)$, Mittelwert h und Variationskoeffizient c_H . Das i.a. nur durch die vorgegebenen Ankunfts- und Bedienungsprozesse erzeugte Ablaufgeschehen kann durch weitere zufallsabhängige Prozesse beschrieben werden, vergl. dazu das Zeitdiagramm in Bild 4.5b):

- Warteprozess; charakterisiert durch die zufallsabhängigen Wartezeiten T_W
- Ausgangsprozess; charakterisiert durch die zufallsabhängigen Ausgangsabstände T_D
- Aufenthaltsprozess; charakterisiert durch die zufallsabhängigen Durchlaufzeiten T_F
- Systemzustandsprozess; charakterisiert durch die zufallsabhängige Anzahl X von Anforderungen im System

Die Analyse dieser Prozesse und die daraus abgeleiteten charakteristischen Verkehrsgrößen sind Hauptgegenstand der Verkehrstheorie.

b) Definition Stochastischer Prozesse

Unter einem Stochastischen Prozeß $\{Z(t), t \geq 0\}$ versteht man eine Familie von Zufallsvariablen (ZV) Z , welche von einem Parameter t abhängen. Ein Stochastischer Prozeß ist durch drei Komponenten gekennzeichnet:

- Zustandsraum $\{Z(t)\}$
- Parameter t (Zeit)
- Abhängigkeit zwischen den ZV $Z(t)$ für verschiedene t .

$Z(t) = z$ heißt Zustand des Stochastischen Prozesses zur Zeit t . Jenachdem ob z nur diskrete oder kontinuierliche Werte annehmen kann, spricht man von einem Stochastischen Prozeß mit diskreten oder kontinuierlichen Zuständen. $\{Z(t)\}$ umfaßt die Menge aller Zustände, welche zu einem Zeitpunkt t auftreten können und wird "Zustandsraum" genannt.

Der Parameter t (Zeit) kann diskreter oder kontinuierlicher Werte fähig sein. Dementsprechend werden Stochastische Prozesse mit diskretem Parameter ($t = t_i, i = 0, 1, \dots$, "Stochastische Kette") oder kontinuierlichem Parameter (t stetig) unterschieden.

Die Abhängigkeit zwischen den ZV $Z(t)$ für verschiedene t wird i.a. durch eine Verteilung $F(z, t)$ bzw. eine Verteilungsfunktion angegeben:

$$F(z, t) = P\{Z(t) = z\} \quad \text{für alle } t \geq 0. \quad (4.8)$$

Der Stochastische Prozeß $\{Z(t), t \geq 0\}$ heißt stationär, wenn die Zustandsverteilung $F(z, t)$ invariant ist gegenüber Zeitverschiebungen:

$$F(z, t + \tau) = F(z, t) \quad (4.9)$$

Beispiel 1: Bedienungsprozess

Der Bedienungsprozess einer Anforderung beschreibt die i.a. zufallsabhängige Dauer der Belegung einer Bedienungseinheit durch eine Anforderung. Der Zustand $Z(t)$ eines bei $t = t_0$ begonnenen Bedienungsprozesses ist 0 oder 1 jenachdem, ob die Bedienung zur Zeit t abgeschlossen ist oder nicht. Es handelt sich hierbei um einen Prozess mit diskreten Zuständen und einem kontinuierlichen Parameter t . Beschrieben wird dieser Prozeß entweder direkt durch eine Hilfsvariable τ , welche das "Bedienungsalter" angibt, oder durch die VF $H(\tau)$ der Gesamtbedienungsdauern T_H bzw. deren Dichtefunktion $H'(\tau)$ ⁺⁾

$$H(\tau) = P\{T_H \leq \tau\}, \quad (4.10a)$$

$$H'(\tau) = \lim_{d\tau \rightarrow 0} \frac{P\{\tau < T_H \leq \tau + d\tau\}}{d\tau} = \frac{dH(\tau)}{d\tau}. \quad (4.10b)$$

Beispiel 2: Systemzustandsprozeß

Der in Bild 4.5 skizzierte Systemzustandsprozeß $\{X(t), t \geq 0\}$ beschreibt den Verlauf der Zustandsvariablen $X(t) = x$ über der Zeit. Es handelt sich hierbei um einen Stochastischen Prozeß mit diskreten Zuständen $X(t) = x, x = 0, 1, 2, \dots$ und einem kontinuierlichen Parameter t . Beschrieben wird der Prozeß durch die Verteilung der Systemzustände

$$P(x, t) = P\{X(t) = x\}, \quad x = 0, 1, \dots, \quad t \geq 0. \quad (4.11)$$

Beispiel 3: Systemzustandsprozeß zurückgelassener Anforderungen

Der in Beispiel 2 zugrundegelegte Systemzustandsprozeß werde speziell an den Zeitpunkten $t = t_i + 0$ des Abgangs einer fertig bedienten Anforderung betrachtet. Die Stochastische Kette $\{X(t_i + 0), i = 0, 1, \dots\}$ entspricht einem Stochastischen Prozeß mit diskreten Zuständen $X(t_i + 0) = x, x = 0, 1, 2, \dots$ und einem diskreten Parameter $t_i, i = 0, 1, \dots$. Beschrieben wird der Prozeß durch die Verteilung der Anzahl der jeweils zurückgelassenen Anforderungen im System

$$P(x, t_i) = P\{X(t_i + 0) = x\}, \quad x = 0, 1, \dots, \quad i = 0, 1, \dots \quad (4.12)$$

c) Klassifizierung Stochastischer Prozesse

Unterschieden werden die Stochastischen Prozesse nach Gesichtspunkten des "Gedächtnisses" hinsichtlich ihres vergangenen Prozeßverlaufes. Hier sollen unterschieden werden, vergl. z.B. [43]:

- Markoffsche Prozesse
- Geburts- und Sterbeprozesse
- Semi-Markoff-Prozesse
- Erneuerungsprozesse
- Nichtmarkoffsche Prozesse

⁺⁾ Die Variable τ wurde hier ausnahmsweise verwendet, um keine Verwechslung mit der hier verwendeten Größe t als absolutem Zeitmaßstab aufkommen zu lassen.

Markoffsche Prozesse

Ein Stochastischer Prozeß heißt Markoffscher Prozeß, wenn seine zukünftige Entwicklung nur vom gegenwärtigen Zustand abhängig ist:

$$P\{Z(t_{n+1})=z_{n+1} \mid Z(t_n)=z_n, \dots, Z(t_0)=z_0\} = P\{Z(t_{n+1})=z_{n+1} \mid Z(t_n)=z_n\} \quad (4.13)$$

für alle $t_0 < t_1 < t_2 < \dots < t_n < t_{n+1}$.

Die Eigenschaft der "Gedächtnislosigkeit" (Markoffsche Eigenschaft) drückt sich übersichtlicher aus in folgender Beziehung (vergl. Bild 4.6):

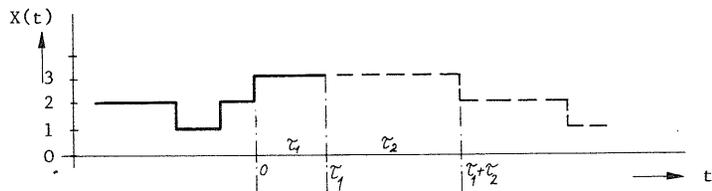


Bild 4.6. Zur Gedächtnislosigkeit Markoffscher Prozesse

$$P\{T > \tau_1 + \tau_2 \mid T > \tau_1\} = P\{T > \tau_2\} \quad (4.14)$$

Gleichung (4.14) besagt folgendes: Ein Zustand x werde zum Zeitpunkt $t = \tau_1$ betrachtet und bestehe schon für die Dauer τ_1 . Die VF der Restzeit τ_2 seines Bestehens bis zur nächsten Zustandsänderung ist identisch der VF der Gesamtzeit seines Bestehens. Unter allen kontinuierlichen VF ist die Exponentialverteilung die einzige, welche die Eigenschaft (4.14) besitzt:

$$P\{T \leq t\} = 1 - e^{-\lambda t} \quad (4.15)$$

mit $E[T] = 1/\lambda$.

Ein rein Markoffscher Prozeß ist stets dadurch gekennzeichnet, daß die Abstände zwischen zwei Zustandsänderungen exponentialverteilt sind. Der Parameter λ dieser VF kann in beliebiger Weise vom jeweiligen Systemzustand abhängen.

Geburts- und Sterbeprozesse

Ein Geburts- und Sterbeprozess ist ein spezieller Markoffscher Prozeß, in welchem nur Übergänge zwischen Nachbarzuständen möglich sind. Treten diese Übergänge nur in Richtung zunehmender Werte der Zustandsvariablen auf, so heißt der Prozeß "reiner Geburtsprozeß", im entgegengesetzten Falle "reiner Sterbeprozess".

Semi-Markoff-Prozesse

Ein Stochastischer Prozeß heißt Semi-Markoff-Prozeß, wenn seine zukünftige Entwicklung außer vom gegenwärtigen Zustand noch vom Alter des bestehenden Zustands abhängig ist. Die Zeit zwischen zwei Zustandsänderungen darf beliebig verteilt sein.

Der wichtigste Anwendungsfall innerhalb der Semi-Markoff-Prozesse sind die "eingebetteten Markoff-Prozesse" oder "eingebetteten Markoff-Ketten". Hierbei wird der Prozeß nur zu diskreten Zeitpunkten (sog. "Regenerationspunkten") betrachtet; der eingebettete Markoff-Prozeß besitzt an diesen Punkten die Markoffsche Eigenschaft, während der Abstand zwischen diesen Regenerationspunkten beliebig verteilt sein darf.

Erneuerungsprozesse

Ein Stochastischer Prozeß heißt Erneuerungsprozeß (Rekurrenter Prozeß), wenn die zeitlichen Abstände zwischen aufeinanderfolgenden Ereignissen (Zustandsänderungen) unabhängig voneinander und überdies identisch verteilt sind.

Nichtmarkoffsche Prozesse

Unter den Nichtmarkoffschen Prozessen werden alle jene Stochastischen Prozesse zusammengefaßt, welche sich nicht durch die Besonderheiten der o.a. Gedächtnislosigkeit auszeichnen.

Die eigentlichen Analysemethoden für die verschiedenen Stochastischen Prozesse werden im Zusammenhang mit ihrer speziellen Anwendung auf Bedienungssysteme in den Abschnitten 4.4 - 4.6 kurzgefaßt dargestellt.

4.3 Hybride Methoden

Experimentelle wie auch analytische Methoden sind jeweils durch spezielle Eigenschaften hinsichtlich Anwendbarkeit, Flexibilität, Aussagekraft und Handhabbarkeit ausgezeichnet und haben ihren festen Platz in der Systemanalyse. Insbesondere zeigt sich bei jedem Verfahren die Grenze der Anwendbarkeit wie etwa bei zu großer Anzahl zu messender oder zu simulierender Ereignisse bzw. bei der mathematischen Analyse zu komplexer Zusammenhänge. Aus diesem Grunde werden in jüngster Zeit sog. "hybride Verfahren" diskutiert, welche durch die kombinierte Anwendung experimenteller wie auch analytischer Verfahren den Anwendungskreis auszudehnen versuchen. Die hybriden Methoden werden in dieser Arbeit nicht weiter verfolgt und sind nur der Vollständigkeit halber angeführt.

4.4 Analyse Markoffscher Bedienungssysteme

Aufgabe dieses Abschnittes ist ein kurzgefaßter Abriß der Analysemethodik für Bedienungssysteme, welche mit Markoffschen Prozessen beschreibbar sind, vergl. hierzu [43,44,45].

4.4.1 Kolmogoroffsche Gleichungen

Betrachtet werde der Systemzustandsprozeß $\{X(t), t \geq 0\}$ am Beispiel eines Wartesystems des Typs M/M/n. Der Systemzustand $X(t)$ bezeichne die gesamte Anzahl von Anforderungen im System zur Zeit t. Die Zustandsübergänge erfolgen gemäß den Übergangswahrscheinlichkeiten

$$p_{ij}(u) = P\{X(t+u) = j \mid X(t) = i\}, \quad u \geq 0. \quad (4.16)$$

Hierbei wurde ein "homogener" Prozeß zugrundegelegt, bei welchem p_{ij} nur von der Spanne u, jedoch nicht von der absoluten Zeit t abhängt. Die Übergangswahrschein-

lichkeiten gehorchen der Chapman-Kolmogoroff-Gleichung

$$P_{ij}(t+u) = \sum_k P_{ik}(t) \cdot P_{kj}(u), \quad t, u > 0. \quad (4.17)$$

Gl.(4.17) besagt, daß sich die Wahrscheinlichkeit des Überganges von i nach j während der Summenspanne t+u zusammensetzen läßt aus den wegen der Markoffschen Eigenschaft voneinander unabhängigen Wahrscheinlichkeiten der Übergänge von i nach allen möglichen k während der 1. Teilspanne t und den Wahrscheinlichkeiten der Übergänge von diesen k in den betrachteten Zielzustand j während der 2. Teilspanne u. Durch die Grenzübergänge $u \rightarrow 0$ bzw. $t \rightarrow 0$ folgen aus Gl.(4.17) zwei Differentialgleichungen, welche den Gradienten angeben für die bedingte Zustandsänderung:

$$\frac{d}{dt} P_{ij}(t) = -q_j P_{ij}(t) + \sum_{k \neq j} q_{kj} P_{ik}(t), \quad (4.18a)$$

$$\frac{d}{du} P_{ij}(u) = -q_i P_{ij}(u) + \sum_{k \neq j} q_{ik} P_{kj}(u), \quad (4.18b)$$

wobei

$$q_{ik} = \lim_{u \rightarrow 0} \frac{P_{ik}(u)}{u}, \quad q_j = \lim_{u \rightarrow 0} \frac{1 - P_{jj}(u)}{u}. \quad (4.18c)$$

die Übergangs-Wahrscheinlichkeitsdichten oder Übergangsraten. Die Übergangsraten hängen also nur vom gegenwärtigen Zustand und nicht von seiner Vergangenheit ab (Markoffsche Eigenschaft!). Gl.(4.18a) wird als Kolmogoroffsche Vorwärtsgleichung, Gl.(4.18b) als Kolmogoroffsche Rückwärtsgleichung bezeichnet.

4.4.2 Systemzustandsprozeß

Der Prozeß der Systemzustandsfolge $\{X(t), t \geq 0\}$ wird durch die Zustandswahrscheinlichkeit

$$P_j(t) = P\{X(t) = j\} \quad (4.19)$$

charakterisiert. Entsprechend dem Gesetz von der totalen Wahrscheinlichkeit läßt sie sich aus der Anfangsverteilung $P_j(0)$, $j = 0, 1, \dots$ und den (bedingten) Übergangswahrscheinlichkeiten darstellen:

$$P_j(t) = \sum_i P\{X(0)=i\} \cdot P\{X(t)=j | X(0)=i\} = \sum_i P_i(0) \cdot p_{ij}(t), \quad (4.20)$$

$j = 0, 1, \dots$

Aus Gl.(4.18a) und Gl.(4.20) folgt die Kolmogoroffsche Vorwärtsgleichung der Zustandswahrscheinlichkeiten

$$\frac{d}{dt} P_j(t) = -q_j P_j(t) + \sum_{k \neq j} q_{kj} P_k(t), \quad j = 0, 1, \dots \quad (4.21)$$

Gl.(4.21) gibt den Gradienten der Änderung der Zustandswahrscheinlichkeit im allgemeinen nichtstationären Falle an. Zusammen mit der Normierungsbedingung

$$\sum_j P_j(t) = 1 \quad (4.22)$$

und der Anfangsverteilung $P_j(0)$, $j = 0, 1, \dots$ können aus dem Differentialgleichungssystem (4.21) sämtliche zeitabhängigen Zustandswahrscheinlichkeiten bestimmt werden.

In den meisten Anwendungsfällen interessiert das Systemverhalten im stationären Falle, d.h. im "eingeschwungenen Zustand". Für die stationären Zustandswahrscheinlichkeiten

$$p_j = \lim_{t \rightarrow \infty} P_j(t), \quad \lim_{t \rightarrow \infty} \frac{d}{dt} P_j(t) = 0 \quad (4.23)$$

folgt aus Gl.(4.21) das System der linearen Gleichungen

$$q_j p_j = \sum_{k \neq j} q_{kj} p_k, \quad j = 0, 1, \dots \quad (4.24)$$

Gl.(4.24) stellt die Bedingung für das "statistische Gleichgewicht" dar. Es besagt, daß in einem infinitesimalen Intervall der Länge dt die Wahrscheinlichkeit für das Verlassen eines Zustandes j gleich der Wahrscheinlichkeit des Entstehens dieses Zustandes aus seinen "Nachbarzuständen" ist:

$$p_j(q_j dt) = \sum_{k \neq j} p_k(q_{kj} dt), \quad j = 0, 1, \dots \quad (4.25a)$$

wobei

$$q_j dt = P\{X(t+dt) \neq j | X(t) = j\}, \quad (4.25b)$$

$$q_{kj} dt = P\{X(t+dt) = j | X(t) = k\}. \quad (4.25c)$$

Die Bestimmung der stationären Zustandswahrscheinlichkeiten erfolgt durch Auflö- sung des linearen Gleichungssystems (4.24) mit Hilfe einer Gl.(4.22) entsprechen- den Normierungsbedingung.

Beispiel: Wartesystem M/M/n als stationärer Geburts- und Sterbeprozess

Der Systemzustand $X(t) = x$, $x = 0, 1, \dots$ bezeichne die Anzahl von Anforderungen im System zu einem Zeitpunkt t . Der Prozeß entwickelt sich entsprechend der Übergangsraten, vergl. Bild 4.7:

$$q_{kj} = \begin{cases} \lambda_{x-1}, & k = x-1, j = x & x = 1, 2, \dots \\ \mu_{x+1}, & k = x+1, j = x & x = 0, 1, \dots \end{cases} \quad (4.26a)$$

$$q_j = (\lambda_x + \mu_x), \quad j = x \quad x = 0, 1, \dots \quad (\mu_0 = 0). \quad (4.26b)$$

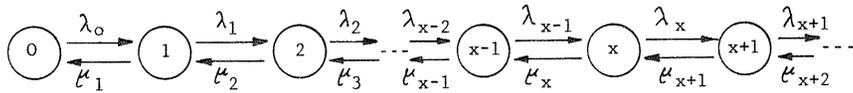


Bild 4.7. Zustände und Übergänge des allgemeinen Wartesystems M/M/n

Aus Gl.(4.24) folgt mit Gl.(4.26a,b):

$$\lambda_0 p_0 = \mu_1 p_1 \quad (4.27a)$$

$$(\lambda_x + \mu_x) p_x = \lambda_{x-1} p_{x-1} + \mu_{x+1} p_{x+1}, \quad x = 1, 2, \dots \quad (4.27b)$$

Die rekursive Auflösung ergibt

$$p_x = \frac{\lambda_0 \lambda_1 \dots \lambda_{x-1}}{\mu_1 \mu_2 \dots \mu_x} \cdot p_0 \quad x = 1, 2, \dots \quad (4.28a)$$

mit p_0 aus der Normierungsbedingung

$$p_0 = \left[1 + \sum_{x=1}^{\infty} \prod_{\nu=1}^x \frac{\lambda_{\nu-1}}{\mu_{\nu}} \right]^{-1} \quad (4.28b)$$

In der vorstehenden Ableitung können die Übergangsraten in beliebiger Weise vom Systemzustand x abhängen. Im einfachsten Falle des Poisson-Ankunftsprozesses und bei gleichbleibender Enderate jeder einzelnen belegten Bedienungseinheit ist speziell

$$\lambda_x = \lambda \quad x = 0, 1, \dots \quad (4.29a)$$

$$\mu_x = \begin{cases} x\varepsilon & x = 1, 2, \dots, n \\ n\varepsilon & x = n+1, n+2, \dots \end{cases} \quad (4.29b)$$

(λ = Ankunftsrate, ε = 1/h Enderate einer belegten Bedienungseinheit). Gl.(4.29a,b) eingesetzt in Gl.(4.28a,b) ergibt

$$p_x = \begin{cases} p_0 \cdot \frac{\lambda^x}{x!}, & x = 0, 1, \dots, n \\ p_0 \cdot \frac{\lambda^n}{n!} \cdot \left(\frac{\lambda}{n}\right)^{x-n}, & x = n+1, n+2, \dots \end{cases} \quad (4.30a)$$

mit

$$p_0 = \left[\sum_{i=0}^{n-1} \frac{\lambda^i}{i!} + \frac{\lambda^n}{n!} \cdot \frac{n}{n-\lambda} \right]^{-1}, \quad A = \frac{\lambda}{\varepsilon} \quad (\text{Verkehrsangebot}) \quad (4.30b)$$

Mit Hilfe der stationären Zustandswahrscheinlichkeiten werden weitere charakteristische Verkehrsgrößen bestimmt wie die Wartewahrscheinlichkeit W , die mittlere Warteschlangenlänge Ω und die mittlere Wartezeit w .

4.4.3 Warte-prozeß

Zur detaillierteren Aussage über die zufallsabhängigen Wartezeiten T_W ist das Schicksal der einzelnen Anforderungen nach ihrem Eintreffen zu verfolgen. Der Warte-prozeß einer Test-Anforderung ist - im Gegensatz zum Systemzustands-prozeß - ein Prozeß endlicher Dauer. Er kann formal beschrieben werden als modifizierter Zustands-prozeß $\{\zeta(t), t \geq 0\}$, wobei $\zeta(t)$ den "Wartezustand" der Test-Anforderung zur Wartezeit t charakterisiert. $\zeta(t)$ enthält alle diejenigen Anforderungen, welche mit der Test-Anforderung zur Zeit t gleichzeitig warten und welche die Wartezeit T_W der Test-Anforderung beeinflussen. Außerdem muß aus der Konstruktion von ζ die Position der Test-Anforderung im Verhältnis zu den "beeinflussenden" Anforderungen hervorgehen.

Beispiele:

- a) Abfertigungsdisziplin FIFO: ζ = Anzahl der vor der Test-Anforderung wartenden Anforderungen
- b) Abfertigungsdisziplin RANDOM: ζ = Anzahl der außer der Test-Anforderung wartenden Anforderungen.

Unter Markoffschen Voraussetzungen kann der Warte-prozeß mittels eines Differentialgleichungssystems für die bedingten Wartezeit-VF

$$w(t|i) = P\{T_W > t | \zeta(0) = i\} \quad (4.31)$$

beschrieben werden:

$$\frac{d}{dt} w(t|i) = -q_i w(t|i) + \sum_{\substack{k \neq i \\ k \notin H}} q_{ik} w(t|k). \quad (4.32)$$

In Gl.(4.31) bedeutet $\zeta(0)$ den Antreff-Wartezustand einer Testanforderung, mit dem der betrachtete Warte-prozeß beginnt. Die Übergangsraten q_i, q_{ik} in Gl.(4.32) beziehen sich auf den Warte-prozeß $\{\zeta(t), t \geq 0\}$. H bezeichnet die Menge der "absorbierenden Zustände" für den Warte-prozeß; ein "absorbierender Zustand" ist ein Zustand, welcher den Warte-prozeß einer Test-Anforderung bei Erreichen dieses Zustands sofort beendet (z.B. Bedienungsbeginn der Test-Anforderung oder Verlust). Durch Auflösung des Differentialgleichungssystems (4.32) können zunächst die

bedingten Wartezeit-VF und mit Hilfe der Antreff-Wartezustandswahrscheinlichkeiten die gesamte Wartezeit-VF gewonnen werden. Eine ausführliche Darstellung dieser Methode findet sich in [45].

4.5 Analyse Nichtmarkoffscher Bedienungssysteme

Die Verfahren zur Analyse Nichtmarkoffscher Bedienungssysteme sind weniger einheitlich und i.a. von den speziellen Eigenschaften der betrachteten Prozesse abhängig. Im folgenden werden die wichtigsten Ansätze diskutiert.

4.5.1 Phasenmethode

a) Darstellung allgemeiner VF durch Phasenmodelle

Grundlage der Phasenmethode ist die Tatsache, daß sich allgemeine, d.h. Nichtmarkoffsche, VF beliebig genau darstellen lassen aus exponentiellen (d.h. Markoffschen) Teilphasen. Das allgemeine Modell einer derart aufgebauten allgemeinen VF wurde erstmals von Cox [46,43] angegeben, vergl. Bild 4.8:

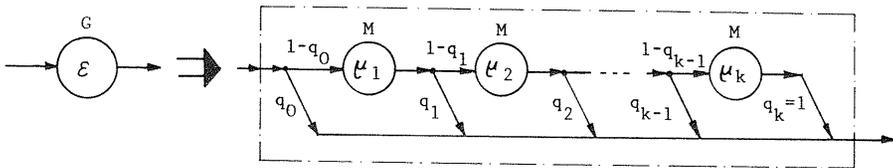


Bild 4.8. Allgemeines Phasenmodell nach Cox

In dem Modell nach Bild 4.8 besteht die VF aus

- 1 Nullphase mit Wahrscheinlichkeit q_0
- oder 1 exp. Phase (μ_1) mit Wahrscheinlichkeit $(1-q_0)q_1$
- oder 2 exp. Phasen (μ_1, μ_2) mit Wahrscheinlichkeit $(1-q_0)(1-q_1)q_2$
- ...
- oder k exp. Phasen ($\mu_1, \mu_2, \dots, \mu_k$) mit Wahrscheinlichkeit $\prod_{i=0}^{k-1} (1-q_i) q_k$.

Anstelle eines kontinuierlichen Parameters μ zur Kennzeichnung des Alters einer angefangenen allgemeinverteilten Phase genügt die Angabe eines ganzzahligen Parameters i zur Kennzeichnung der momentan erreichten exponentiellen Teilphase ($i = 1, 2, \dots, k$). Die Restzeit der Teilphase i ist wegen der Markoffschen Eigenschaft wieder exponentiell verteilt mit Mittelwert $1/\mu_i$. Die Analyse von Prozessen mit allgemein verteilten Phasen kann mit Hilfe der in Bild 4.8 angegebenen Ersetzung auf die Analyse von Prozessen mit exponentiell verteilten

Phasen (Markoffsche Prozesse) zurückgeführt werden; durch die Phasendarstellung der allgemein verteilten Prozesse nimmt die Zustandsbeschreibung allerdings um jeweils eine Komponente (erreichte Phasenzustandsnummer) zu. Dies kann unter Umständen zu sehr großen Zustandszahlen führen, welche numerisch nicht mehr beherrscht werden.

Zur praktischen Anwendung der Phasenmethode werden deshalb oft Sonderfälle der allgemeinen Phasenmethode betrachtet. Ein erster Sonderfall entsteht dadurch, daß die Mittelwerte der exponentiellen Teilphasen identisch sind und folgt aus dem Bild 4.8 durch die Einschränkung

$$\mu_i = \mu, \quad i = 1, 2, \dots, k. \tag{4.33}$$

Die zugehörige VF der Gesamtdauer T hat die Form

$$F(t) = P\{T \leq t\} = \sum_{i=0}^k a_i (1 - e^{-\mu t} \sum_{j=0}^{i-1} \frac{(\mu t)^j}{j!}), \quad t \geq 0 \tag{4.34}$$

mit

$$a_i = \left[\prod_{\nu=0}^{i-1} (1-q_\nu) \right] \cdot q_i, \quad i = 0, 1, \dots, k.$$

Dieses von Bux und Herzog vorgeschlagene Modell erlaubt ebenfalls die Approximation beliebiger VF, benötigt dafür jedoch i.a. mehr Teilphasen [47]. Die Besonderheit der gleichen Phasenmittelwerte ermöglicht dafür aber eine vorteilhaftere Modellanalyse.

b) Erlang-k-VF (E_k)

Die in Abschn. 3.2.2 angeführte Erlang-k-VF ist als weiterer Sonderfall in Gl. (4.34) enthalten für $q_i = 0, i = 0, 1, \dots, k-1$:

$$F(t) = P\{T \leq t\} = 1 - e^{-\mu t} \sum_{j=0}^{k-1} \frac{(\mu t)^j}{j!}, \quad t \geq 0. \tag{4.35}$$

Die Erlang-k-VF läßt sich deuten als VF von k hintereinandergeschalteten exponentiell verteilten Phasen des Mittelwertes $1/\mu$, vergl. Bild 4.9.

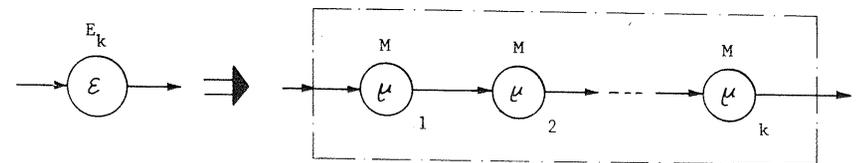


Bild 4.9. Erlang-k-Modell

c) Hyperexponentielle VF k-ter Ordnung (H_k)

Schließlich läßt sich ebenfalls zeigen, daß auch die hyperexponentielle VF k-ter Ordnung (H_k) im allgemeinen Phasenmodell enthalten ist. Hierzu betrachten wir die Laplace-Stieltjes-Transformierten der beiden VF (d.h. die Laplace-(L) Transformatierten der Verteilungsdichtefunktionen):

allgemeines Phasenmodell nach Bild 4.8:

$$f_1(s) = L\{F_1'(t)\} = q_0 + \sum_{i=1}^k \left[\prod_{r=0}^{i-1} (1-q_r) \right] q_i \cdot \prod_{j=1}^i \frac{\mu_j}{s+\mu_j}, \quad (4.36)$$

hyperexponentielles Modell nach Abschn. 3.2.2:

$$f_2(s) = L\{F_2'(t)\} = \sum_{i=1}^k p_i \cdot \frac{\epsilon_i}{s+\epsilon_i}. \quad (4.37)$$

Nach Partialbruchzerlegung von Gl.(4.36) und Gleichsetzen mit Gl.(4.37) können sämtliche Parameter bestimmt werden. Im Falle $k = 2$ erhält man zum Beispiel:

$$\begin{aligned} q_0 &= 0, & \mu_1 &= \epsilon_1 \\ q_1 &= p_1 + p_2 \frac{\epsilon_2}{\epsilon_1}, & (\epsilon_2 \leq \epsilon_1), & \mu_2 &= \epsilon_2. \end{aligned} \quad (4.38)$$

Die entsprechenden äquivalenten Phasenmodelle zeigt Bild 4.10.

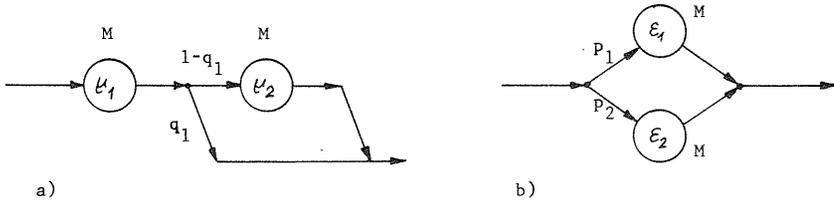


Bild 4.10. Zur Äquivalenz von Coxschem und hyperexponentiellem Phasenmodell
a) Coxsches Phasenmodell 2. Ordnung
b) Hyperexponentielles Phasenmodell 2. Ordnung (H_2)

d) Beispiel: Wartesystem $M/E_k/1$

Zur Beschreibung des stationären Systemzustandsprozesses des Wartesystems $M/E_k/1$ wird folgender 2-dimensionaler Zustand eingeführt:

(X, I) , wobei $X = 0, 1, \dots$ Anzahl von Anforderungen im System
 $I = 0, 1, \dots, k$ Phasenstand der gerade bedienten Anforderung

Das Modell und den zugehörigen Zustandsraum zeigt Bild 4.11. Die Phasendarstellung der E_k -VF ermöglicht den Ansatz der Zustandsgleichungen für die Zustandswahrscheinlichkeiten $p(x, i) = P\{X = x, I = i\}$ wie bei einem 2-dimensionalen Markoffschen Prozeß.

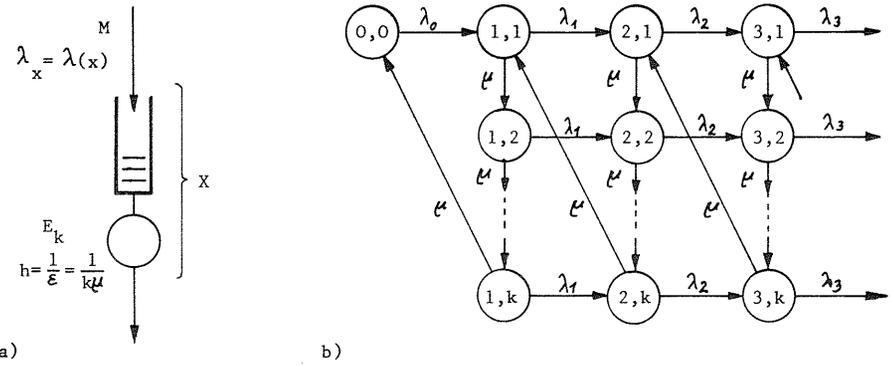


Bild 4.11. Wartesystem $M/E_k/1$ und Zustandsraum
a) Modell
b) Zustandsraum

4.5.2 Methode der Gedächtnisvariablen

Der Phasenstand im vorangegangenen Beispiel (gekennzeichnet durch die diskrete Variable $I = 1, 2, \dots, k$) stellt eine diskrete "Gedächtnisvariable" über die bereits verbrauchte Bedienungszeit im speziellen Falle einer Erlang-k-VF dar. Durch die Verallgemeinerung auf eine beliebige Gedächtnisvariable U ist es möglich, beliebig verteilte Phasen in die Analyse einzubeziehen.

Beispiel: Wartesystem $M/G/1$

Der stationäre Systemzustandsprozeß wird beschrieben durch

(X, U) , wobei $X = 0, 1, \dots$ Anzahl von Anforderungen im System
 $U = u, u \geq 0$ Bedienungsalter eines bestehenden Bedienungsprozesses.

Der Zustandsprozeß wird hierbei durch einen 2-dimensionalen aus einer diskreten sowie einer kontinuierlichen Variablen bestehenden Zustand exakt gekennzeichnet. Entsprechend der kontinuierlichen Variablen U wird eine zweidimensionale Zustandsdichte $d(x, u)$ definiert, vergl. hierzu auch Bild 4.12

$$d(x, u) = \lim_{du \rightarrow 0} \frac{P\{X=x, u \leq U \leq u+du\}}{du}, \quad x \geq 1, u \geq 0. \quad (4.39)$$

Das Verfahren der Gedächtnisvariablen wurde von Vulot, Kosten und Cox eingeführt [49, 50, 58].

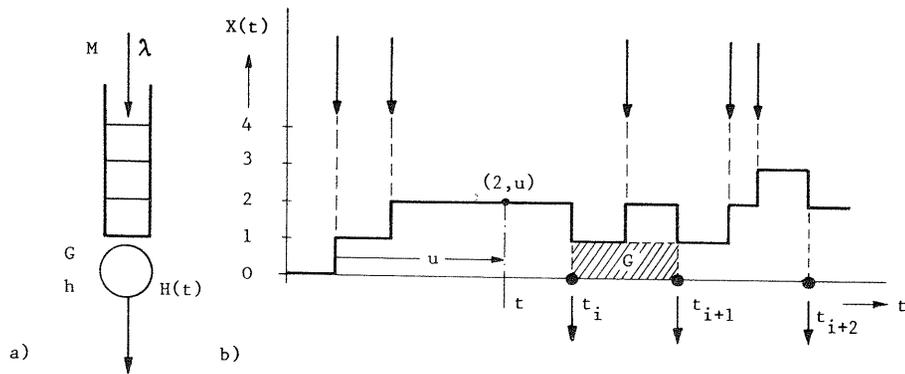


Bild 4.12. Zustandsprozeß im Wartesystem M/G/1
 a) Modell
 b) Zeitdiagramm

4.5.3 Methode der Eingebetteten Markoff-Kette

Die Phasenmethode und Methode der Gedächtnisvariablen sind dadurch gekennzeichnet, daß der gedächtnisbehaftete Prozeß zu jedem Zeitpunkt betrachtet werden kann, wobei das "Gedächtnis" in einer Zusatzvariablen festgehalten wurde. Auf diese Zusatzvariable kann verzichtet werden, wenn es gelingt, einzelne Zeitpunkte des Prozesses zu finden, welche die Markoffsche Eigenschaft besitzen. Der Zustandsprozeß verliert in diesen "Regenerationspunkten" sein Gedächtnis, d.h. er entwickelt sich von dort aus unabhängig von der Vergangenheit. In den allgemeinen Zustandsprozeß ist in den Regenerationspunkten eine Markoff-Kette "eingebettet". Das Verfahren der Eingebetteten Markoff-Kette geht auf Kendall zurück [51,43,52].

Beispiel: Wartesystem M/G/1

Betrachtet werde das Zeitdiagramm des Zustandsprozesses X(t) nach Bild 4.12. Die Abgangszeitpunkte fertig bedienter Anforderungen seien mit t_i bezeichnet, $i = 1, 2, \dots$. Diese Abgangszeitpunkte sind gleichzeitig Regenerationspunkte des Zustandsprozesses, von denen aus sich der Prozeß unabhängig von der Vergangenheit entwickelt, da gerade jeweils in diesen Punkten die gedächtnisbehaftete Bedienungsphase beendet ist. Infolge des Markoffschen Ankunftsprozesses ist das zukünftige Eintreffen einer Anforderung unabhängig von dem Zeitpunkt des Eintreffens der letzten Anforderung. An den Regenerationspunkten gilt die Markoffsche Eigenschaft

$$P\{X(t_{i+1}+0)=x_{i+1} \mid X(t_i+0)=x_i, X(t_{i-1}+0)=x_{i-1}, \dots\} \quad (4.40)$$

$$= P\{X(t_{i+1}+0)=x_{i+1} \mid X(t_i+0)=x_i\} .$$

Die eingebettete Markoff-Kette wird beschrieben durch ein Gleichungssystem, welches die Entwicklung des Zustandsprozesses zwischen zwei Regenerationspunkten angibt:

$$P\{X(t_{i+1}+0)=x_{i+1}\} = \sum_{x_i} P\{X(t_i+0)=x_i\} \cdot P\{X(t_{i+1}+0)=x_{i+1} \mid X(t_i+0)=x_i\} . \quad (4.41)$$

Die auftretende Übergangswahrscheinlichkeit in Gl.(4.41) ist im stationären Falle unabhängig vom Zeitindex i

$$q_{jk} = P\{X(t_{i+1}+0) = k \mid X(t_i+0) = j\} , \quad (4.42)$$

und läßt sich durch die Anzahl eintreffender Anforderungen während einer vollen Bedienungsdauer T_H ausdrücken. In einem Intervall der Länge u treffen r Anforderungen ein mit der Wahrscheinlichkeit

$$\frac{(\lambda u)^r}{r!} \cdot e^{-\lambda u} .$$

Eine Bedienungsdauer der Länge u tritt auf mit der Wahrscheinlichkeitsdichte

$$\frac{dH(u)}{du} .$$

Integriert über alle möglichen Bedienungsdauern u ergibt schließlich:

$$q_{jk} = \begin{cases} \int_0^\infty \frac{(\lambda u)^k}{k!} \cdot e^{-\lambda u} dH(u) , & j = 0, k \geq 0 \\ \int_0^\infty \frac{(\lambda u)^{k-j+1}}{(k-j+1)!} \cdot e^{-\lambda u} dH(u) , & j > 0, k \geq j-1. \end{cases} \quad (4.43)$$

Mit Hilfe von q_{jk} kann das Gleichungssystem (4.41) im stationären Falle aufgestellt werden:

$$P_k^* = \sum_j P_j^* \cdot q_{jk} , \quad k = 0, 1, \dots \quad (4.44)$$

wobei

$$P_k^* = \lim_{i \rightarrow \infty} P\{X(t_i+0) = k\}$$

die stationäre Zustandswahrscheinlichkeit an den speziellen Regenerationspunkten der eingebetteten Markoff-Kette. Die Auflösung des Gleichungssystems (4.44) erfolgt zweckmäßigerweise mit Hilfe der erzeugenden Funktion (siehe z.B. [52]).

4.5.4 Lindleysche Integralmethode

Wartesysteme mit allgemeinen Ankunfts- und Bedienungsprozessen des Typs GI/G/1 lassen sich i.a. nicht mit den o.a. Methoden behandeln. Lindley nutzt eine allgemeine Beziehung zwischen Wartezeiten aufeinanderfolgender Anforderungen bei FIFO aus, welche über eine Integralgleichung des Wiener-Hopf-Typs verbunden sind:

$$W(\leq t) = \int_{-\infty}^t W(\leq t-u) dC(u), \quad t \geq 0, \quad (4.45)$$

wobei

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

Die Lösung dieser Gleichung kann mit Hilfsmitteln der Funktionentheorie in Angriff genommen werden; die geschlossene Lösung steht im allgemeinen Falle noch aus. Zur weiteren Diskussion dieser Methode sei auf [43] verwiesen.

4.5.5 Momentenmethoden

Im Zusammenhang mit der Lösung spezieller Probleme haben sich Methoden als sehr sinnvoll erwiesen, welche nicht auf die Analyse des gesamten Zustandsprozesses, sondern nur auf die Erwartungswerte einiger charakteristischer Größen zielen. Charakteristische Merkmale dieser Methoden sind:

- Betrachtung von Test-Anforderungen und Aufstellung des Zusammenhangs zwischen Zufallsgrößen zur Beschreibung des Schicksals dieser Test-Anforderungen (Teilwartezeiten, Bedienungszeiten etc.)
- Übergang zu Erwartungswerten
- Anwendung von Ergebnissen der Erneuerungstheorie sowie Beziehungen zwischen Zeit- und Ensemblemittelwerten ("Little's Law").

Beispiele für die erfolgreiche Anwendung dieser Verfahren sind:

- Mittlere Wartezeit und mittlere Warteschlangenlänge im Wartesystem M/G/1 (Kendall u.a., siehe z.B. [52])
- Mittlere Wartezeiten in Wartesystemen des Typs M/G/1 mit nichtunterbrechenden Prioritäten (Cobham [53]) und Unterbrechungs-Distanz-Prioritäten (Herzog [20, 54]).

Über die im vorangehenden kurz umrissenen Methoden hinaus ist eine Reihe weiterer Methoden bekannt, auf welche hier aus Umfangsgründen nicht weiter eingegangen werden kann. Darstellungen dieser Verfahren finden sich in neueren Büchern der Warteschlangentheorie [39, 43, 48, 50, 52, 55, 56, 57, 58].

4.6 Approximative Methoden

Für eine Vielzahl verkehrstheoretischer Probleme sind exakte Lösungen entweder gar nicht bekannt oder nur sehr schwierig auswertbar. Aus beiden Gründen wurden daher Näherungsverfahren entwickelt und angewendet. Im folgenden sollen die wesentlichsten Ansätze kurz skizziert werden. Auf rein numerische Verfahren kann in diesem Zusammenhang nicht eingegangen werden.

4.6.1 Makrozustände

Eine Vielzahl verkehrstheoretischer Modelle erfordert zur exakten Analyse die Einführung vieldimensionaler Zustände, insbesondere zur Kennzeichnung der komplexen Zusammenhänge in Durchschaltenetzen der Nachrichtenvermittlungstechnik (ein- und mehrstufige Koppelanordnungen, Überlaufsysteme etc.), vergl. [59-62]. In den meisten Fällen muß die Auflösung der entsprechenden Gleichungssysteme numerisch-iterativ erfolgen. Infolge des hohen Ranges der Gleichungssysteme bleibt die exakte Analyse i.a. auf verhältnismäßig kleine Strukturen beschränkt.

Durch die Einführung von "Makrozuständen" kann die Komplexität oft sehr vereinfacht werden. Makrozustände fassen eine Untermenge von "Mikrozuständen" zu einem resultierenden Zustand zusammen, wobei in der Regel ein Verlust an Detailinformationen über den Prozeßverlauf verbunden ist und die resultierende Beschreibung des Ablaufgeschehens nur approximativ die tatsächlichen Verhältnisse wiedergibt.

Als typisches Beispiel für eine Makrozustandsbeschreibung sei die Mischung nach Bild 3.6c) betrachtet. Zur exakten Kennzeichnung des Belegungszustandes der $n=9$ Bedienungseinheiten und $g=4$ Wartespeicher müßte ein 13-dimensionaler Zustand $(x_1, x_2, \dots, x_9; z_1, z_2, z_3, z_4)$ eingeführt werden, wobei $x_i = 0(1)$ den Belegungszustand "frei" ("belegt") der Bedienungseinheit i und z_j die Anzahl der wartenden Anforderungen in Warteschlange j angibt, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, g$. Approximativ läßt sich das Modell durch den Makrozustand (x, z) beschreiben, wobei x (z) die Gesamtzahl der belegten Bedienungseinheiten (Warteplätze) angibt, vergl. [63].

Zur Auflösung der Gleichungssysteme für Makrozustandswahrscheinlichkeiten selbst müssen oft weitere Näherungsverfahren angewandt werden wie etwa Symmetrieannahmen in den Gleichgewichtsgleichungen, welche eine effektive rekursive Auflösung ermöglichen, vergl. z.B. [63-65]. Die Genauigkeit solcher Verfahren muß mit Hilfe von Simulationen auf ihre Zulässigkeit hin überprüft werden.

4.6.2 Kontinuierliche Zustände

Im Zusammenhang mit der Analyse von Wartesystemen unter starker Verkehrsbelastung wurden in den vergangenen Jahren Verfahren vorgeschlagen, bei denen diskrete Zustandsvariable durch kontinuierliche Zustandsvariable ersetzt werden. Die bekanntesten Verfahren dieser Art sind die sog. "Diffusionsapproximationen": Unter gewissen Voraussetzungen kann der Zustandsprozeß in Analogie zur Brownschen Molekularbewegung gesetzt und mit Hilfe von Diffusionsgleichungen der Physik beschrieben werden. Die Ergebnisse zeigen allgemein gute Genauigkeit bei hoher Verkehrsbelastung ("heavy traffic"), vergl. z.B. [43].

4.6.3 Unabhängigkeitsannahmen

In den meisten Fällen von komplexen Verkehrsmodellen sind Bedienungszeiten einer einzelnen Anforderung in verschiedenen Stufen bzw. Zustandsprozesse in verschiedenen Teilsystemen eines Modelles nicht unabhängig voneinander, beispielsweise etwa in seriellen Wartesystemen [66], in Warteschlangenmodellen für Datennetze mit Speichervermittlung [20, 67] oder in mehrstufigen Koppelanordnungen der Vermittlungstechnik [68, 69]. Durch die Annahme der Unabhängigkeit kann die Analyse erheblich vereinfacht werden, wodurch i.a. eine Entkopplung von Teilsystemen oder Teilzustandsprozessen gelingt.

4.6.4 Zerlegungsmethoden

Zerlegungsmethoden werden i.a. angewandt, um komplexe Zusammenhänge in Gesamtsystemen auf einfacher überschaubare Teilsysteme zurückzuführen. In den meisten Fällen wird intuitiv bei der Definition eines Modelles bereits eine Zerlegung vorgenommen, nämlich die Abtrennung des betrachteten Modelles von seiner Umwelt. Eine Zerlegung liefert i.a. umso bessere Ergebnisse, je unabhängiger die Teilsysteme über ihre Schnittstellen hinweg miteinander verknüpft sind. Typische Zerlegungsbedingungen sind:

1. Geringe Interaktionen zwischen Teilsystemen im Verhältnis zu den Interaktionen innerhalb der Teilsysteme.

Die entstehenden Teilsysteme können entweder vollkommen unabhängig (complete decomposability) oder nur bedingt unabhängig voneinander betrachtet werden (nearly complete decomposability, vergl. [70]).

2. Überlagerung einer Vielzahl von Verkehren.

Bei der Überlagerung von Teilverkehren tritt in der Regel eine Zunahme des Zufälligkeitscharakters ein, welcher eine bessere Entkopplung von Systemen erlaubt. Dieses Verfahren hat seine Begründung in einem Grenzwertsatz der Erneuerungstheorie [71], welcher besagt, daß die Überlagerung beliebiger Teilprozesse in der Grenze unendlich vieler Teilprozesse einen Poisson-Prozeß ergibt.

Es sei noch bemerkt, daß Zerlegungen unter gewissen Voraussetzungen exakt sind, z.B. bei seriellen Wartesystemen mit Markoffschen Voraussetzungen oder bei Markoffschen Warteschlangennetzen, vergl. Kap. 5; im ersten Falle sind die Ausgangsprozesse wieder Markoff-Prozesse (Output-Theorem von P.J. Burke [72]), während im zweiten Falle die Lösung "robust" gegenüber einer Zerlegung ist, selbst bei Eingangs- und Ausgangsprozessen von Teilsystemen mit Abhängigkeiten [73]. Darüberhinaus können Zerlegungen auch ohne Zutreffen obiger Voraussetzungen angewandt werden, wenn die Verkehre an den Schnittstellen hinreichend genau charakterisiert werden können, wie etwa bei Überlaufsystemen [74-81] oder bei Warteschlangennetzen (vergl. Abschnitt 5.4).

4.6.5 Ersetzungsmethoden

Zur Analyse zusammengesetzter Systeme ist es mitunter zweckmäßig, ein komplexeres Teilsystem durch ein einfacheres zu ersetzen, welches den Schnittstellenverkehr hinreichend genau "erzeugt"; z.B. exakt in den ersten beiden Momenten. Das entstehende Ersatzsystem ist dann in der Regel einfacher zu analysieren. Dieses Verfahren wurde mit großem Erfolg für Überlaufsysteme angewendet ("Equivalent Random Traffic"-Methoden), vergl. [74-81]. Bei der Analyse von mehrstufigen Koppelanordnungen mit konjugierter Durchschaltung (Linksysteme) können die verschiedenen Blockierungseffekte mit Hilfe geeigneter einstufiger Anordnungen mit unvollkommener Erreichbarkeit analysiert werden, woraus durch Zusammensetzung die Gesamtblockierung gefunden wird, vergl. [82-86].

4.6.6 Momentenapproximationen

Die mathematische Analyse von Verkehrsmodellen führt nicht selten auf komplizierte Lösungen, bei denen eine VF z.B. nur in Form ihrer Laplace-Stieltjes Transformierten oder eine Verteilung in Form ihrer erzeugenden Funktion erhalten wird, woraus die Momente erster, zweiter oder auch höherer Ordnung wesentlich leichter gefunden werden können als die ganze VF bzw. Verteilung. Da in den meisten Fällen die ersten zwei oder ersten drei Momente die wesentlichsten Eigenschaften wiedergeben, können unbekannte VF bzw. Verteilungen durch bekannte Funktionen ersetzt werden, welche in den ersten Momenten mit den exakten übereinstimmen. Dieses Verfahren wurde z.B. mit Erfolg zur Approximation von Wartezeit-VF angewendet [50, 87, 45, 65].

Eine weitere interessante Eigenschaft wurde für eine Reihe von Wartesystemen bemerkt, bei denen Proportionalitätsbeziehungen zwischen Momenten "ähnlicher" Wartesysteme bestehen. Beispielsweise verhalten sich die ersten beiden Momente der Wartezeit-VF folgender vier Wartesysteme wie folgt zueinander:

$$\frac{E\left[T_W^i\right]_{M/G/1 \text{ RANDOM}}}{E\left[T_W^i\right]_{M/G/1 \text{ FIFO}}} = \frac{E\left[T_W^i\right]_{M/M/1 \text{ RANDOM}}}{E\left[T_W^i\right]_{M/M/1 \text{ FIFO}}} = \frac{2}{2-A}, \quad i = 1, 2.$$

Durch Annahme derartiger Beziehungen kann z.B. aus drei bekannten Größen eine vierte gewonnen werden, wenn möglichst gleichartige Vergleichssysteme herangezogen werden, welche sich jeweils nur in einer Annahme unterscheiden. Dieses heuristische Approximationsverfahren wurde erfolgreich angewandt bei der Bestimmung höherer Momente von Wartezeit-VF [65].

4.6.7 Grenzwerttheorien

Für Zwecke der Anwendung ist es in vielen Fällen ausreichend, das Systemverhalten durch untere und obere Grenzwerte einzuschachteln. Zu diesem Zwecke wurde eine Reihe von Grenzwerttheorien entwickelt. Die bekannteste Theorie dieser Art ist die "heavy-traffic"-Theorie für Wartesysteme, welche nahe der Stationaritätsgrenze betrieben werden. Zur tieferen Diskussion wird auf die einschlägige Literatur verwiesen [43, 57].

4.7 Zusammenfassung

Ziel dieses Kapitels war es, einen Überblick über die wesentlichsten Methoden der Modellanalyse zu verschaffen. Die angegebene Übersicht erhebt nicht den Anspruch auf Vollständigkeit; insbesondere konnte in diesem Rahmen nicht auf speziellere Methoden eingegangen werden, welche einen wesentlich höheren Aufwand an mathematischen Voraussetzungen erfordern würden.

Die Vielfalt der bisher analytisch gelösten Probleme ist beeindruckend und heute bereits sehr schwierig in ihrer Gesamtheit zu überblicken. Mit Hilfe der o.a. Methoden sind die wichtigsten Standardergebnisse abgeleitet worden wie z.B. für Verlustsysteme des Typs M/M/n, Überlaufsysteme, Wartesysteme des Typs M/M/n, M/G/1, GI/M/n, M/D/n bzw. Warteverlustsysteme. Zahlreiche Ergebnisse liegen vor für Systeme mit spezielleren Voraussetzungen hinsichtlich der Struktur (unvollkommene Erreichbarkeit, Linkanordnungen, Überlaufanordnungen, Netze) bzw. der Betriebsart (Abfertigungsdisziplinen, Prioritäten, Berücksichtigung von Systemverwaltungszeiten u.a.m.). Darüberhinaus sind allgemein anwendbare Ergebnisse von großer Bedeutung wie Erhaltungssätze für Wartezeiten, Zusammenhänge zwischen Zeit- und Zustandswertwerten (Little's Theorem), Zusammenhänge zwischen Zustandsverteilungen zu beliebigen Zeitpunkten und Zustandsverteilungen zu speziellen Zeitpunkten, bei denen eine Anforderung eintrifft bzw. das System verläßt, Ausgangsproßtheoreme, Zerlegungstheoreme u.a.m. Zur weiteren Diskussion sei auf die einschlägige Fachliteratur verwiesen [43, 67, 50, 52, 55-58].

5. ANALYSE VON WARTESCHLANGENNETZEN

Die Beschreibung des Ablaufgeschehens in Rechnersystemen, Datennetzen sowie Steuerungsteilen von Fernsprechvermittlungssystemen führt häufig auf Warteschlangennetze, welche i.a. von komplexer Struktur sind und überdies durch Abhängigkeiten zwischen Anforderungen und/oder Betriebsmitteln gekennzeichnet sind. Ziel dieses Kapitels ist es, zunächst eine Übersicht über exakte und approximative Analyseverfahren zu geben. Nach einer kritischen Beurteilung dieser Verfahren im Hinblick auf die Anwendbarkeit wird abschließend ein neues Verfahren zur approximativen Analyse allgemeiner Warteschlangennetze beschrieben.

5.1 Exakte Ergebnisse für Warteschlangennetze

In diesem Abschnitt werden die bekannten exakten Ergebnisse kurzgefaßt zusammengestellt. Im ersten und zweiten Unterabschnitt wird dabei eine Klasse von Netzen behandelt, deren Zustandswahrscheinlichkeiten sich durch eine sog. "Produktform" auszeichnet. Im dritten Unterabschnitt wird eine Erweiterung der Gültigkeit der Produktlösungsform angegeben. Im vierten Unterabschnitt schließlich wird auf exakte Ergebnisse allgemeinerer Netze kurz eingegangen.

5.1.1 Warteschlangennetze mit Produktlösungsform und einer einzigen Klasse von Anforderungen

a) Voraussetzungen

Betrachtet werden offene bzw. geschlossene Warteschlangennetze, deren Struktur durch die Verzweigungsmatrix $Q = (q_{ij})$ festgelegt ist, vergl. Abschnitt 3.4.3 und Beispiel Bild 5.1. Es werde zunächst der Fall einer einzigen Klasse von Anforderungen betrachtet. Das Bedienungssystem der Stufe i , $i = 1, 2, \dots, N$, kann einer der folgenden 4 Kategorien angehören:

1. Eine Bedienungseinheit mit exponentieller Bedienungszeit.

Die Bedienungsrate $\mu_i = \mu_i(x_i)$ darf abhängig sein von der momentanen Anzahl x_i von Anforderungen in Stufe i .¹⁾ Die Abfertigungsdisziplin für wartende Anforderungen ist FIFO.

2. Eine Bedienungseinheit mit allgemeiner Bedienungszeit.

Alle sich momentan in Stufe i befindlichen Anforderungen teilen sich die Bedienungskapazität zu gleichen Anteilen auf nach dem Zeitscheibungsverfahren, wobei die Zeitscheiben verschwindend klein sind ("processor sharing").

¹⁾ Die zustandsabhängige Bedienungsrate schließt den Fall mehrerer Bedienungseinheiten mit ein.

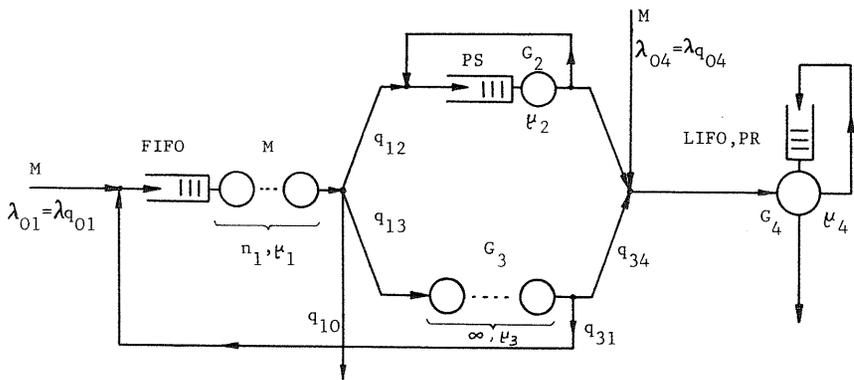


Bild 5.1. Beispiel eines offenen Warteschlangennetzes aus N = 4 Bedienungssystemen

- Stufe 1: Bedienungssystem der Kategorie 1
- Stufe 2: Bedienungssystem der Kategorie 2
- Stufe 3: Bedienungssystem der Kategorie 3
- Stufe 4: Bedienungssystem der Kategorie 4

3. Hinreichend große Anzahl von Bedienungseinheiten mit allgemeiner Bedienungszeit.

Die Stufe i hat stets so viele Bedienungseinheiten (endlich oder unendlich viele), daß jede neu ankommende Anforderung sofort bedient werden kann.

4. Eine Bedienungseinheit mit allgemeiner Bedienungszeit.

Neu ankommende Anforderungen erhalten sofort Bedienung, wobei eine eventuell sich gerade in Bedienung befindliche Anforderung sofort unterbrochen wird. Unterbrochene Anforderungen setzen später ihre Bedienung von dem erreichten Bedienungszustand aus in umgekehrter Reihenfolge ihres Eintreffens fort (LIFO preemptive-resume).

Externe Ankünfte erfolgen bei offenen Netzen entsprechend exponentiell verteilter Ankunftsabstände, wobei die Gesamtankunftsrate $\lambda = \lambda(x)$ abhängig sein darf von der Gesamtzahl x von Anforderungen im Netz. Bei geschlossenen Netzen zirkuliert eine konstante Anzahl M von Anforderungen durch das Netz.

b) Produktlösung

Bezeichnet X_i die Anzahl von Anforderungen in Stufe i, so ist der Zustand des Netzes durch den Zustandsvektor $\underline{X} = (X_1, X_2, \dots, X_N)$ gekennzeichnet. Das Auftreten des Zustandes \underline{X} im stationären Falle ist gegeben durch die Zustandswahrscheinlichkeit

$$p(x_1, x_2, \dots, x_N) = P\{X_1=x_1, X_2=x_2, \dots, X_N=x_N\}.$$

Unter obigen Voraussetzungen gilt für die Zustandswahrscheinlichkeiten die Lösung [88, 43] :

$$p(x_1, x_2, \dots, x_N) = C \cdot d(x) \cdot \prod_{i=1}^N p_i(x_i), \tag{5.1a}$$

wobei

$$p_i(x_i) = \begin{cases} \prod_{j=1}^{x_i} \frac{e_i}{\mu_i(j)} & \text{für Kategorien 1,2,4} \\ \frac{e_i^{x_i}}{(\mu_i)^{x_i} / x_i!} & \text{für Kategorie 3} \end{cases} \tag{5.1b}$$

$$d(x) = \prod_{k=0}^{x-1} \lambda(k), \quad x = x_1 + x_2 + \dots + x_N. \tag{5.1c}$$

Die hierin verwendeten Größen e_i sind ein Maß für die Anzahl von Durchläufen einer beliebigen Anforderung durch Stufe i während ihres Aufenthaltes im Netz. Die Größen e_i bestimmen sich aus folgender Bilanz:

$$e_i = q_{0i} + \sum_{j=1}^N e_j q_{ji}, \quad i = 1, 2, \dots, N. \tag{5.1d}$$

In offenen Netzen sind die Größen e_i die absoluten Mittelwerte der Anzahl von Durchläufen einer Anforderung durch Stufe i, $i = 1, 2, \dots, N$, während bei geschlossenen Netzen ($q_{0i} \equiv 0$) die Größen e_i nur bis auf einen multiplikativen Faktor bestimmt sind. Die Größe $\lambda(k)$ ist die absolute externe Ankunftsrate im Netzzustand $x = k$; bei geschlossenen Netzen ist $\lambda(k) \equiv 0$ und $d(x) \equiv 1$. Die Normierungskonstante C bestimmt sich schließlich aus der Bedingung, daß die Summe aller Zustandswahrscheinlichkeiten eins sein muß.

c) Eigenschaften

Die Produktlösungsform nach Gl.(5.1) hat eine Reihe bemerkenswerter Eigenschaften, welche im folgenden kurzgefaßt dargestellt seien:

- (1) Die Produktterme $p_i(x_i)$ sind die Lösungen von isoliert betrachteten Wartesystemen mit Markoffschem Eingangs- und Ausgangsprozeß. Eine anschaulich einfache Erklärung läßt sich im Falle offener und rückkopplungsfreier Netze mit exponentiellen Bedienungszeiten angeben (Kategorie 1): In diesem Falle kann sukzessive das Ausgangsprozeßtheorem von P.J. Burke [72] angewendet werden; dieses besagt, daß der Ausgangsprozeß eines Wartesystems M/M/n wieder einen Poisson-Prozeß ergibt. Bei Netzen mit Rückkopplungen kann diese Argumentation nicht mehr angewandt werden, dennoch gilt dieselbe Lösung wie wenn alle Prozesse Markoffschen Charakter hätten !

- (2) Die Lösung Gl.(5.1) ist "robust" gegenüber der Verteilungsfunktion-Charakteristik der Bedienungszeiten unter speziellen Voraussetzungen. Diese Robustheit hängt zusammen mit der Tatsache, daß die einzelnen Bedienungsstufen unter diesen Voraussetzungen bei Markoffschem Eingangsprozeß wieder einen Markoffschen Ausgangsprozeß erzeugen ("M ⇒ M -Eigenschaft": Markoff-Eingangsprozeß impliziert Markoff-Ausgangsprozeß, [89]) bzw. daß neu eintreffende Anforderungen sofort ihre Bedienung antreten und keine Restbedienungszeiten teilweise abgefertigter Anforderungen abwarten müssen [90].
- (3) Die Produktlösungsform ist robust gegenüber der Verzweigungsdisziplin (Wegelenkung). Die Annahme unabhängiger Wahrscheinlichkeiten für das Überwechseln einer Anforderung von einer Stufe i zu einer Stufe j kann verallgemeinert werden hinsichtlich der Abhängigkeit dieses Übergangs von den vorausgegangenen Übergängen derselben Anforderung, vergl. [96]. Als praktische Konsequenz folgt hieraus, daß sich die Zustandswahrscheinlichkeiten nicht ändern, wenn eine Anforderung eine Stufe in einer geometrisch verteilten oder konstanten Anzahl durchläuft, sofern sich die mittlere Gesamtbedienungszeit der Anforderung in dieser Stufe nicht ändert.
- (4) Die allgemeinen Gleichgewichtsgleichungen können zerlegt werden in Teilgleichgewichtsgleichungen ("Local Balance"). Diese Eigenschaft besagt, daß die Wahrscheinlichkeitsdichten für das Entstehen eines Zustandes durch die Ankunft einer Anforderung in der Stufe i und das Verschwinden dieses Zustandes durch den Abgang einer Anforderung aus Stufe i gleich sind, $i = 1, 2, \dots, N$. Im allgemeinen Falle zerfällt somit die globale Gleichgewichtsgleichung in N Teilgleichgewichtsgleichungen; insbesondere folgt hieraus auch, daß jeder Zustand sich bereits mit all seinen "oberen" bzw. "unteren" Nachbarzuständen im Gleichgewicht befindet.
- (5) Die gesamte Netzumgebung einer einzelnen Stufe kann, in Bezug auf diese Stufe, äquivalent ersetzt werden durch eine einzige Komplementärstufe ("Nortons Theorem für Warteschlangennetze mit Produktlösungsform" [91]).
- (6) Die Produktlösungsform gestattet für geschlossene Netze effektive Rekursionsalgorithmen zur Bestimmung der Konstanten C sowie der mittleren Anzahl von Anforderungen in einer Stufe ohne die Berechnung aller einzelnen Zustandswahrscheinlichkeiten [92,93]. Die Zerlegungseigenschaft (5) gestattet ferner eine sukzessive Analyse aller einzelnen Netzstufen [91].
- (7) Die Abhängigkeit der Gesamt-Ankunftsrate $\lambda = \lambda(x)$ von der Gesamtzahl x von Anforderungen im Netz erlaubt sowohl eine absolute Beschränkung der gleichzeitigen Anzahl von Anforderungen im Netz als auch "getriggerte" Ankünfte, wenn eine Anforderung das Netz verläßt. Geschlossene Netze können in diesem Sinne als Sonderfall der offenen Netze aufgefaßt werden.

Die Produktlösungsform nach Gl. (5.1) wurde bereits sehr viel früher für offene bzw. geschlossene Netze mit Bedienungsstufen der Kategorie 1 gefunden, vergl. J.R. Jackson [94]. bzw. W.J. Gordon und G.F. Newell [95]. Der Beweis der verschiedenen Produktlösungsformen erfolgt durch Nachweis der Widerspruchsfreiheit in allen Gleichgewichtsgleichungen, wobei für die allgemeine Bedienungszeit-VF das Coxsche Phasenmodell angewendet wird (vergl. Abschn. 4.5.1). In einer erweiterten Betrachtungsweise können allgemeine differenzierbare Bedienungszeit-VF zugelassen werden [90].

5.1.2 Warteschlangennetze mit Produktlösungsform und mehreren Klassen von Anforderungen

a) Voraussetzungen

Über die unter 5.1.1 a) getroffenen Voraussetzungen hinaus werde nun der Fall von R Anforderungsklassen betrachtet. Externe Ankünfte (bei offenen Netzen) haben die Markoff-Eigenschaft, wobei die Gesamt-Ankunftsrate $\lambda = \lambda(x)$ von der Gesamtzahl x von Anforderungen aller Klassen im Netz abhängt. Jede Bedienungsstation kann einer der vier Kategorien aus 5.1.1 a) angehören, wobei die mittleren Bedienungszeiten in Stufen der Kategorien 2-4 außerdem noch von der Klasse der bedienten Anforderung abhängig sein dürfen. Eine Anforderung der Klasse r wechselt beim Übergang von einer Stufe i in eine Stufe j die Klassenzugehörigkeit von r nach s mit Wahrscheinlichkeit $rs^{q_{ij}}$.

b) Produktlösung

Die Beschreibung des Systemzustandes kann in verschiedenen Ebenen hinsichtlich der "Feinstruktur" erfolgen, vergl. [88, 65]. Zum Beweis der Gültigkeit der angegebenen Lösung muß i.a. auf die unterste Ebene zurückgegangen werden, aus welcher neben der Anzahl von Anforderungen jeder Klasse auch noch die Position in der Warteschlange bzw. der Bedienungszustand jeder einzelnen Anforderung hervorgeht. Im folgenden wird die daraus abgeleitete Lösung für die nächsthöhere Beschreibungsebene angegeben.

Der Systemzustand werde durch den Vektor $\underline{Y} = (Y_1, \dots, Y_N)$ gekennzeichnet, wobei $\underline{Y}_i = (Y_{i1}, \dots, Y_{iR})$ und $Y_{ir} =$ Anzahl der Anforderungen der Klasse r in Stufe i. Die Systemzustandswahrscheinlichkeit

$$p(\underline{Y}) = p(\dots; y_{i1}, \dots, y_{iR}; \dots) = P\{\dots; Y_{i1} = y_{i1}, \dots, Y_{iR} = y_{iR}; \dots\}$$

ist gegeben durch die Produktform

$$p(\underline{Y}) = C \cdot d(x) \cdot \prod_{i=1}^N p_i(Y_i) \quad (5.2a)$$

wobei

$$p_i(Y_i) = \begin{cases} y_i! \cdot \prod_{r=1}^R \frac{e_{ir}}{y_{ir}!} \cdot \prod_{j=1}^{y_i} \frac{1}{\mu_i(j)} & , \text{Kategorie 1} \\ y_i! \cdot \prod_{r=1}^R \left(\frac{e_{ir}}{\mu_{ir}}\right)^{y_{ir}} \cdot \frac{1}{y_{ir}!} & , \text{Kategorien 2, 3} \\ \prod_{r=1}^R \left(\frac{e_{ir}}{\mu_{ir}}\right)^{y_{ir}} \cdot \frac{1}{y_{ir}!} & , \text{Kategorie 4} \end{cases} \quad (5.2b)$$

$$d(x) = \prod_{k=1}^{x-1} \lambda(k) \quad , \quad x = \sum_{i=1}^N y_i \quad , \quad y_i = \sum_{r=1}^R y_{ir} \quad (5.2c)$$

Die Größe e_{ir} bedeutet die absolute (bei offenen Netzen) bzw. relative (bei geschlossenen Netzen) mittlere Anzahl von Durchläufen von Anforderungen der Klasse r durch Stufe i . Analog zu Gl.(5.1d) bestimmen sich die Größen e_{ir} aus einem linearen Gleichungssystem:

$$e_{ir} = r^{q_{0i}} + \sum_{j=1}^N \sum_{s=1}^R e_{js} \cdot sr^{q_{ji}}, \quad i = 1, 2, \dots, N \quad (5.2d)$$

Bei geschlossenen Netzen sind die Größen e_{ir} nur bis auf einen multiplikativen Faktor bestimmt sowie $d(x) \equiv 1$. Die Normalisierungskonstante C bestimmt sich wieder aus der Bedingung, daß die Summe aller Zustandswahrscheinlichkeiten eins ist.

c) Eigenschaften

Die Produktlösung für Netze mit mehreren Klassen von Anforderungen hat entsprechende Eigenschaften wie unter 5.1.1c) angeführt. Insbesondere seien folgende allgemeine Eigenschaften festgehalten:

- (1) Die globalen Gleichgewichtsgleichungen für einen allgemeinen Netzzustand lassen sich zerlegen in Teilgleichgewichtsgleichungen. Insbesondere existieren Teilgleichgewichtsgleichungen, welche das statistische Gleichgewicht eines einzelnen Netzzustandes hinsichtlich des Verschwindens bzw. Entstehens durch Eintreffen bzw. Endigen einer Bedienung von Anforderungen jeder einzelnen Klasse beschreiben ("local balance" [88, 90, 97]).
- (2) Besitzt jede einzelne Stufe des Netzes die Eigenschaft, daß sie - isoliert betrachtet - bei einem Markoffschen Eingangsprozeß wieder einen Markoffschen Ausgangsprozeß erzeugt ("M \implies M-Eigenschaft"), so besitzt jede Stufe und das gesamte Netz die Eigenschaft der Produktlösungsform und der Teilgleichgewichte [89] und umgekehrt [90].
- (3) Für alle Arten von Warteschlangendisziplinen, welche sich nicht an der Klasse der wartenden Anforderungen orientieren (d.h. alle Klassen gleich behandeln), impliziert die Produktlösung die Eigenschaft der Teilgleichgewichte [90].

Wir bemerken abschließend, daß die Gültigkeit der Produktlösungsform darüberhinaus für Netze nachgewiesen wurde, deren Verzweigungsmatrix sich in einzelne Teilketten zerlegen läßt. Die Zustandsabhängigkeit der Ankunftsrate läßt sich dann auf jede einzelne Teilkette erweitern, siehe [88,97].

5.1.3 Eine Verallgemeinerung der Warteschlangennetze mit Produktlösungsform

In den vorausgegangenen Abschnitten wurden vier Kategorien von Bedienstationen im Netz zugelassen. In diesem Abschnitt wird gezeigt, daß sich die Kategorie 1 auf die allgemeinste klassenunabhängige Abfertigungsdisziplin verallgemeinern läßt.

a) Definition der allgemeinsten klassenunabhängigen Abfertigungsdisziplin

Betrachtet werde eine Bedienstung des Typs $. / M / n$ mit n Bedienungseinheiten und exponentiellen klassenunabhängigen Bedienungszeiten mit Mittelwert $1/\epsilon$, vergl. Bild 5.2. Es werden Anforderungen von R Klassen angeboten, welche mit Raten $\lambda_1, \dots, \lambda_R$ eintreffen. Sind $x < n$ Bedienungseinheiten belegt, so wird die neu eintreffende Anforderung sofort bedient; im Falle $x = n$ reiht sich die Anforderung in die Warteschlange ein. Bei Bedienungsende einer der Bedienungseinheiten wird eine wartende Anforderung auf Warteplatz i mit Wahrscheinlichkeit $d(i|z)$ abgefertigt, wobei z die Gesamtzahl der wartenden Anforderungen aller Klassen ist.

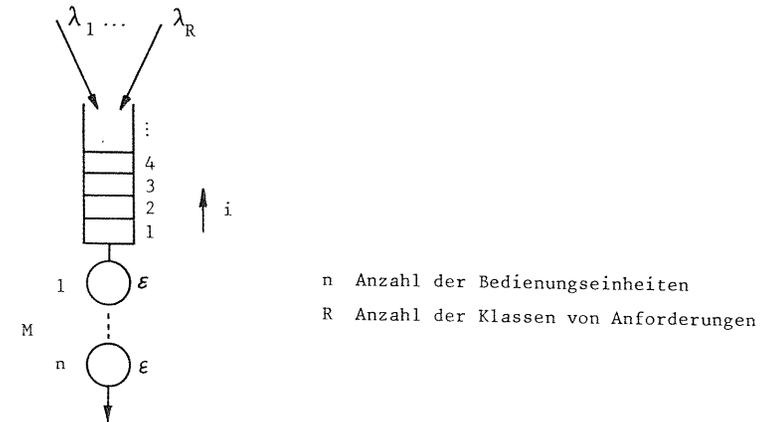


Bild 5.2. Verallgemeinerte Bedienstung $. / M / n$

Die Abfertigungswahrscheinlichkeiten $d(i|z)$ unterliegen der Vollständigkeitsbedingung

$$\sum_{i=1}^z d(i|z) = 1 \quad (5.3)$$

Spezialfälle dieser Disziplin sind die bekannten Disziplinen

$$\text{FIFO:} \quad d(i|z) = \begin{cases} 1 & \text{für } i = 1 \\ 0 & \text{für } i > 1 \end{cases} \quad (5.4a)$$

$$\text{LIFO:} \quad d(i|z) = \begin{cases} 0 & \text{für } i < z \\ 1 & \text{für } i = z \end{cases} \quad (5.4b)$$

$$\text{RANDOM:} \quad d(i|z) = \frac{1}{z} \quad \text{für } z > 0 \quad (5.4c)$$

b) Verallgemeinerung

Für Warteschlangennetze entsprechend Abschnitt 5.1.1 und 5.1.2 gilt die Produktlösungsform für Bedienungsstufen der Kategorie 1 mit der allgemeinsten klassenunabhängigen Abfertigungsdisziplin.

Beweis:

Unter Zugrundelegung der Eigenschaften (2) aus Abschnitt 5.1.2 c) genügt zu zeigen, daß eine Netzstufe des Typs M/M/n mit der allgemeinsten klassenunabhängigen Abfertigungsstrategie die Eigenschaft der Teilgleichgewichte ("local balance") besitzt.

Der Zustand der Bedienungsstufe wird beschrieben durch den Zustandsvektor

$$\underline{X} = (X_1, \dots, X_R; C_1, \dots, C_z), \text{ wobei } X_i = \text{Anzahl belegter Bedienungseinheiten durch Anforderungen der Klasse } i, i=1,2,\dots,R$$

$$C_j = \text{Klassennummer der wartenden Anforderung auf Warteplatz Nr. } j, j=1,2,\dots,z.$$

Es wird vermutet, daß die Zustandswahrscheinlichkeit der Netzstufe folgende Lösung besitzt:

$$p(\underline{X}) = p_0 \cdot \prod_{i=1}^R \frac{A_i^{X_i}}{X_i!} \cdot \prod_{j=1}^z \frac{A_j^{C_j}}{n^{C_j}}, \quad (5.5)$$

wobei

$$A_i = \lambda_i / \epsilon \quad \text{und} \quad p_0 = p(\underline{0}).$$

Der Beweis wird erbracht in folgenden Schritten:

- (1) Aufstellung aller oberen Nachbarzustände des Zustandes \underline{X} , aus welchen der Zustand \underline{X} durch Endigung einer derjenigen Bedienungseinheiten hervorgeht, welche durch eine Anforderung der Klasse i belegt sind.
- (2) Bestimmung sämtlicher Übergangsraten zwischen dem Zustand \underline{X} und seinen oberen Nachbarzuständen hinsichtlich des Eintreffens bzw. Endigens einer i -Anforderung.
- (3) Aufstellung der Gleichungen für das statistische Teilgleichgewicht hinsichtlich der Klasse i und Nachweis der vollständigen Widerspruchsfreiheit mit Gl. (5.5).

Die Schritte (1)-(3) sind elementar und sollen aus Umfangsgründen hier nicht weiter dargestellt werden. Die Gültigkeit der Lösung Gl. (5.5) sowie die Eigenschaft der Teilgleichgewichte wurde speziell für die Disziplinen FIFO, RANDOM

und LIFO für Wartesysteme wie auch für Warteverlustsysteme mit einer Beschränkung der Gesamtzahl von Warteplätzen für alle Klassen von Anforderungen gefunden [65]. Darüberhinaus wurde in [65] ein entsprechendes Theorem für Systeme mit einer endlichen Quellenzahl pro Klasse angegeben.

5.1.4 Allgemeinere Voraussetzungen

Exakte Lösungen für Warteschlangennetze unter allgemeineren Voraussetzungen sind nur in Spezialfällen bekannt bzw. nur über einen oft sehr erheblichen numerischen Aufwand zu gewinnen.

Für Warteschlangennetze mit allgemeineren Bedienungsprozessen läßt sich im Prinzip das Problem mit Hilfe der Coxschen Phasendarstellung auf die Lösung von einem linearen Gleichungssystem zurückführen. Der Rang dieser Systeme ist jedoch i.a. so hoch, daß die Anwendbarkeit auf kleine Netze beschränkt bleibt. In Kapitel 6 wird ein Beispiel im Zusammenhang mit der Leistungsanalyse von Rechnersystemen angegeben.

Für den Spezialfall eines geschlossenen Wartesystems aus 2 Stufen, von denen eine Stufe eine allgemeine und die andere eine exponentielle Bedienungszeit-VF besitzt, kann das Problem auf die Lösung eines äquivalenten Warteverlustsystems des Typs M/G/1-s zurückgeführt werden [98, 99]. Weitere Spezialfälle mit bekannten Lösungen sind serielle Wartesysteme mit konstanten Bedienungszeiten, serielle Wartesysteme mit beschränkter Anzahl von Warteplätzen (siehe [66]), und Netze mit entartet-exponentieller Bedienungszeit-VF [100].

5.2 Übersicht über approximative Ergebnisse für Warteschlangennetze

In den vergangenen Jahren ist eine Reihe von approximativen Verfahren für Netze mit allgemeineren Voraussetzungen entwickelt worden. Die wichtigsten hiervon werden im folgenden kurz charakterisiert.

5.2.1 Diffusionsapproximationen

Der Grundgedanke der Diffusionsapproximation beruht auf der Ersetzung des diskreten Zustandsprozesses durch einen kontinuierlichen Prozeß, wobei die Inkremente der Zustandsänderung so gewählt werden, daß sie denen des diskreten Prozesses entsprechen. Die Annahme normalverteilter Schwankungen der Anzahl von Anforderungen in einer Warteschlange (welche im Falle hoher Verkehrsbelastung durch den zentralen Grenzwertsatz gerechtfertigt ist) führt auf eine Differentialgleichung zur Beschreibung des Zustandsprozesses, welche als Diffusionsgleichung der Brownschen Molekularbewegung unter speziellen Randbedingungen aufgefaßt werden

kann. Die Lösung dieser Gleichung liefert eine Zustandsverteilung, in welche im wesentlichen die beiden ersten Momente des Ankunfts- und Bedienungsprozesses eingehen. Mit Hilfe dieser Lösungen und unter der Annahme einer Produktform kann eine approximative Zustandsverteilung für Warteschlangennetze abgeleitet werden, vergl. z.B. [101-103]. Das Diffusions-Approximationsverfahren liefert im Falle hochbelasteter Netzstufen gute Ergebnisse; für niedrigere Belastungen werden die Abweichungen jedoch i.a. unzulässig groß.

5.2.2 Äquivalenzverfahren

Bei diesem Verfahren wird eine Eigenschaft der Netze mit Produktlösungsform analog übertragen auf allgemeinere Netze. In Netzen mit Produktlösungsform kann die Netzumgebung einer beliebigen Stufe äquivalent ersetzt werden durch eine Komplementärstufe mit passender zustandsabhängiger Bedienungsrate [91].

In Netzen mit allgemeineren Prozessen wird nun die Netzumgebung einer bestimmten Stufe durch eine komplementäre Stufe mit exponentiellen Bedienungszeiten ersetzt mit einer angepaßten Bedienungsrate. Das so entstehende zweistufige Netz wird mit Hilfe von Zustandsgleichungen exakt analysiert [104, 105]. Für geschlossene Netze, welche vor allem als Modelle zur Beschreibung des Ablaufgeschehens von Rechnersystemen auftreten, wurde mit diesem Verfahren eine gute Genauigkeit erzielt.

5.2.3 Zerlegungsverfahren

Die Zerlegungsverfahren haben zum Ziel, das Netz nach bestimmten Kriterien in kleinere Einheiten zu zerlegen, welche entweder unabhängig voneinander untersucht werden können (vollständige Zerlegung) oder welche unter einer Zustandsbedingung zerlegt und untersucht werden können (nahezu vollständige Zerlegung).

Bei ersteren Verfahren wird auf die Beschreibung der Verkehre zwischen den Teilsystemen abgehoben, vergl. [106] und Abschnitt 5.4. Bedingte Zerlegungen beruhen auf der Betrachtung von Schnittstellen mit geringer Wechselwirkung zwischen verschiedenen Teilsystemen des Gesamtsystems. Hierbei wird von einer Eigenschaft des transienten Übergangsverhaltens Gebrauch gemacht: Betrachtet werden zwei Teilsysteme mit jeweils starken inneren Zustandsbindungen, jedoch verhältnismäßig schwachen Zustandsbindungen zwischen den Teilsystemen. Nach einem Zugang/ Abgang einer Anforderung erreicht ein Teilsystem - infolge seiner häufigeren Zustandsänderungen - wesentlich schneller ein neues dynamisches Gleichgewicht im Vergleich zum Einschwingverhalten des Gesamtsystems. Aufbauend hierauf können solche Teilsysteme für sich untersucht werden, z.B. unter der Bedingung einer

jeweils konstanten Anzahl von Anforderungen. Umgekehrt kann das Ablaufgeschehen zwischen Teilsystemen untersucht werden ohne Berücksichtigung der "Mikrozustandsänderungen" innerhalb der Teilsysteme, wobei sämtliche Mikrozustände eines Teilsystems mit gleicher Anzahl von Anforderungen innerhalb des Teilsystems zu einem Makrozustand zusammengefaßt werden, vergl. [107, 70, 108].

5.3 Schlußfolgerungen und Begründung für ein neues Zerlegungsverfahren

Der gegenwärtige Entwicklungsstand auf dem Gebiet der Warteschlangennetze bietet dem Anwender eine Reihe von Methoden und Verfahren zur Analyse an. Die Auswahl eines Verfahrens kann dabei jedoch erst nach einer sorgfältigen Modellbildung erfolgen, welche sicherstellt, daß die wesentlichsten beeinflussenden Parameter in ihrem funktionellen Zusammenhang wiedergegeben werden.

Die Tragweite der Produktlösungs-Netze erlaubt z.B.

- die Beschreibung des Ablaufgeschehens in Rechnersystemen mit mehrstufiger Speicherhierarchie und unterschiedlichen Typen von Anwenderprogrammen (vergl. z.B. [40, 88, 109, 155 - 159]).
- die Beschreibung des Datenflusses in Datennetzen unter Berücksichtigung der Quittierungssignalisierung und zusätzlichen Speicherbelegung durch nichtquittierte Nachrichten (vergl. [110, 111]).
- die Beschreibung der gegenseitigen Beeinflussung von Leitungs- und Registerbelegungen in bestimmten Arten von Vermittlungssystemen [112].

Als wesentliche Nachteile stehen demgegenüber:

- Beschränkung auf Markoffsche Ankunfts- und Bedienungsprozesse bzw. klassenunabhängige Bedienungszeiten (von speziellen Abfertigungsdisziplinen abgesehen)
- Beschränkung auf Zustandswahrscheinlichkeiten und daraus abgeleitete Mittelwerte
- Ausschluß von Prioritätsdisziplinen, zustandsabhängigen bzw. zeitabhängigen Schaltern sowie Blockierungen infolge begrenzter Speicherkapazität
- Numerische Probleme bei der Auswertung der Vektorzustandsräume.

Obwohl eine Verallgemeinerung der exakten Lösungsmethodik auf eine Vielzahl von Problemen theoretisch im Prinzip möglich ist, scheitert deren Auswertung i.a. jedoch daran, daß die entstehenden stochastischen Prozesse mathematisch nicht beherrscht werden und/oder der Unzulänglichkeit der numerischen Hilfsmittel. Sämtliche approximativen Verfahren haben zum Ziel, diese Nachteile zumindest teilweise auszuschalten.

Im folgenden Abschnitt 5.4 wird ein neues approximatives Analyseverfahren vorgeschlagen, welches auf der vollständigen Zerlegungsmethode beruht. Durch die Beschränkung in der Charakterisierung sämtlicher Prozesse durch ihre ersten beiden Momente ist eine Verallgemeinerung auf beliebige Ankunfts- und Bedienungsprozesse möglich. Die Analyse wird dabei auf eine Reihe elementarer Schritte zurückgeführt, welche durch einen stabilen und unaufwendigen Algorithmus auswertbar sind, wodurch die wirtschaftliche Analyse großer Netzstrukturen ermöglicht wird. Die Zerlegung in Teilsysteme und die Charakterisierung des Verkehrs an deren Schnittstellen erlaubt ferner die Einbeziehung von zeitabhängigen Schaltern (getaktete Systeme) sowie von beliebig komplexen Teilsystemen, sofern deren Ein/Ausgangsverhalten bekannt ist.

5.4 Ein neues Verfahren zur Analyse von Warteschlangennetzen nach der Zerlegungsmethode

In diesem Abschnitt wird ein neues Analyseverfahren beschrieben, welches auf der Zerlegungsmethode beruht [118, 119]. Nach der Beschreibung der Voraussetzungen und der Lösungsübersicht werden die einzelnen Lösungsschritte systematisch entwickelt und zu einem Lösungsalgorithmus zusammengefaßt. Das Kapitel schließt ab mit Genauigkeitsnachweisen für dieses Verfahren sowie einer kritischen Diskussion der Tragweite und der Verallgemeinerungen.

5.4.1 Voraussetzungen

a) Netzstruktur

Das Warteschlangennetz bestehe aus N Bedienungssystemen mit jeweils einer Bedienungseinheit und unbegrenzter Warteschlange (Verallgemeinerungen siehe 5.4.6). Die einzelnen Bedienstationen sind durch Pfade in beliebiger Weise zu einem offenen Netz verknüpft. Bild 5.3 zeigt (a) die Bedienstufe i und (b) ein Netzbeispiel bestehend aus N = 4 Stufen.

Die Bedienstufe i wird über den Eingang C_i betreten und über den Ausgang D_i verlassen. Am Eingangspunkt C_i überlagern sich alle Verkehre, welche der Stufe i angeboten werden (composition point); umgekehrt verzweigt sich der Verkehr der Stufe i am Ausgangspunkt D_i (decomposition point). Eine Anforderung, welche Stufe i verläßt, wechselt über zu Stufe j nach einer konstanten Wahrscheinlichkeit q_{ij} ($j = 0$: Anforderung verläßt das Netz zur "Außenwelt"). Die Matrix $Q = (q_{ij})$ der Verzweigungswahrscheinlichkeiten definiert gleichzeitig die Netzstruktur.

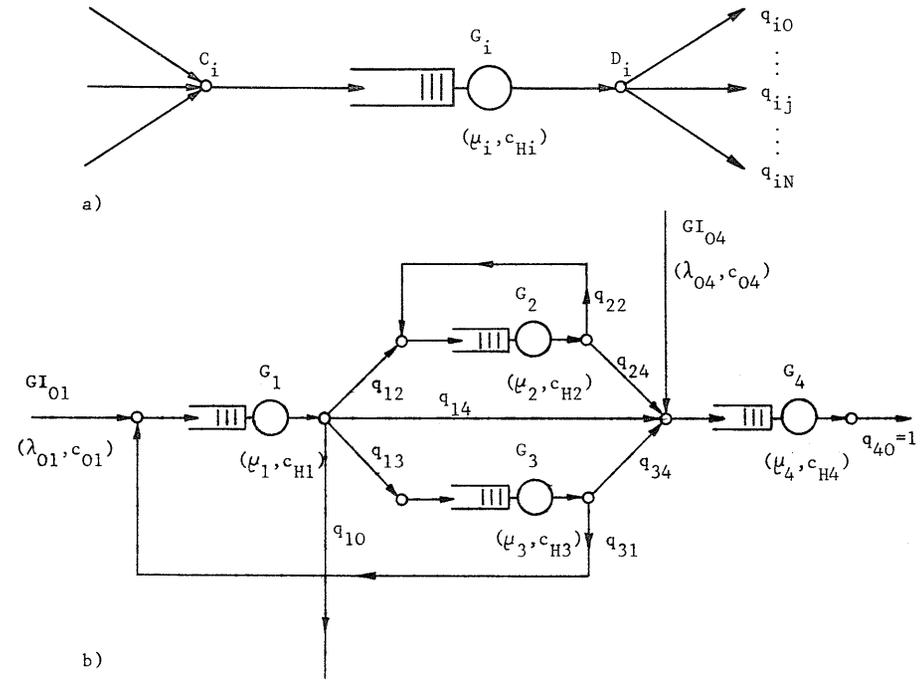


Bild 5.3. Beispiel eines Warteschlangennetzes
 a) Einzelne Bedienstufe Nr. i
 b) Netz aus N = 4 Bedienstufen

b) Ankunfts- und Bedienungsprozesse

Externe Ankunftsprozesse sind allgemeine Prozesse, deren aufeinanderfolgende Ankunftsabstände unabhängig voneinander sind (GI). Im allgemeinsten Falle können externe Anforderungen über jede Stufe des Netzes angeboten werden. Die externen Ankunftsprozesse sind gegeben durch

$\underline{GI}_0 = (GI_{0i})$ Vektor der externen Ankunftsprozeßtypen

$\underline{\lambda}_0 = (\lambda_{0i})$ Vektor der externen Ankunftsrate ($\lambda_{0i} = 1/a_{0i}$, a_{0i} = mittlerer Ankunftsabstand des externen Ankunftsprozesses an Stufe i)

$\underline{c}_0 = (c_{0i})$ Vektor der Variationskoeffizienten der externen Ankunftsprozesse.

Die Bedienungszeiten in den einzelnen Stufen sind ebenfalls allgemein und gegeben durch

$\underline{G} = (G_i)$ Vektor der Bedienungsprozeßtypen

$\underline{\mu} = (\mu_i)$ Vektor der Bedienungsraten ($\mu_i = 1/h_i$, h_i = mittlere Bedienungszeit in Stufe i)

$\underline{c}_H = (c_{Hi})$ Vektor der Variationskoeffizienten der Bedienungsprozesse.

Die externen Ankunftsabstände T_{oi} bzw. Bedienungszeiten T_{Hi} sind unabhängig voneinander und identisch verteilt entsprechend der VF $A_{oi}(t) = P\{T_{oi} \leq t\}$ bzw. $H_i(t) = P\{T_{Hi} \leq t\}$, $i = 1, 2, \dots, N$. Ferner wird angenommen, daß sich das gesamte Warteschlangennetz im eingeschwungenen Zustand befindet.

c) Betriebsweise

Es werde zunächst nur eine Klasse von Anforderungen betrachtet, welche entsprechend der unabhängigen Verzweigungswahrscheinlichkeiten q_{ij} , $i = 1, 2, \dots, N$, $j = 0, 1, \dots, N$, durch das Netz laufen. Wartende Anforderungen werden nach einer beliebigen Disziplin abgefertigt, welche jedoch von der Bedienungszeit unabhängig sein muß, wie z.B. bei FIFO, RANDOM, LIFO oder der in Abschnitt 5.1.3 a) definierten allgemeinen Abfertigungsdisziplin.

5.4.2 Lösungsübersicht

Die Lösung basiert auf den folgenden Prinzipien:

- (1) Zerlegung des Gesamtnetzes in Teilsysteme. Teilsysteme können dabei Einzelstufen oder Teilnetze mit einem Eingang und einem Ausgang sein.
- (2) Isolierte Analyse der Teilsysteme. Die Wechselwirkung zwischen den Teilsystemen und ihrer Umgebung wird durch Eingangs- und Ausgangsprozesse beschrieben.
- (3) Annahme von Erneuerungsprozessen. Alle Prozesse im Netz werden als stationäre Erneuerungsprozesse (vergl. Abschnitt 4.2.2 c)) betrachtet.
- (4) Berücksichtigung von nur 2 Momenten. Alle Prozesse im Netz werden nur durch Mittelwert und Variationskoeffizient charakterisiert.
- (5) Schneller und stabiler Lösungs-Algorithmus. Die Analyse wird auf eine Reihe elementarer Schritte zurückgeführt, welche sich in einem schnellen und stabilen Algorithmus implementieren lassen.
- (6) Ausbaufähigkeit des Verfahrens (vergl. 5.4.6).

Kernpunkte des Verfahrens sind die Prinzipien (3) und (4). Die Annahme von stationären Erneuerungsprozessen erleichtert die mathematische Beherrschung der auftretenden Prozesse. Darüberhinaus ist diese Annahme durch eine heuristische Erweiterung einer Eigenschaft für Netze mit Produktlösungsform begründet: Netze mit Produktlösungsform lassen sich exakt zerlegen in einzelne Stufen, deren Ein- und Ausgangsprozesse als Markoff-Prozesse angenommen werden können. Diese Annahme ist nur im Falle rückkopplungsfreier Netze erfüllt; bei Netzen mit Rückkopplungen sind aber die im Inneren des Netzes auftretenden Prozesse weder Markoff- noch Erneuerungsprozesse oder m.a.W. die Produktlösung ist robust gegenüber der Nichterneuerungseigenschaft dieser Prozesse. Diese Eigenschaft wird approximativ auf Netze mit allgemeineren Ankunfts- und Bedienungsprozessen übertragen.

Die Berücksichtigung von nur 2 Momenten (4) beruht auf der Beobachtung, daß in einer Reihe von Ergebnissen der Verkehrstheorie charakteristische Mittelwerte (z.B. mittlere Wartezeiten) im wesentlichen oder sogar ausschließlich von den ersten beiden Momenten der Ankunfts- bzw. Bedienungsprozesse abhängen, wie beispielsweise im Wartesystem M/G/1 ("Pollaczek-Khintchine-Formel"), vergl. [43, 50, 52]. Die nachfolgenden Operationen sind überdies besser zu beherrschen für die ersten Momente als für ganze Prozesse.

5.4.3 Lösungsschritte des Verfahrens

Nachfolgend werden die einzelnen Schritte des Verfahrens entwickelt, welche in einen Lösungsalgorithmus münden (Abschnitt 5.4.4). Die Lösungsschritte sind im einzelnen:

- a) Bestimmung der Verkehrsflüsse (1. Momente)
- b) Wartezeit und Ausgangsprozeß des Wartesystems GI/G/1
- c) Verzweigung von Verkehrsflüssen
- d) Überlagerung von Verkehrsflüssen
- e) Ersetzung von Rückkopplungen
- f) Bestimmung der Verkehrsflüsse (2. Momente)
- g) Bestimmung der Durchlaufzeiten

a) Bestimmung der Verkehrsflüsse (1. Momente)

Im stationären Zustand ist die mittlere pro Zeiteinheit eintreffende Anzahl von Anforderungen in einer Stufe des Netzes gleich der mittleren pro Zeiteinheit die Stufe verlassende Anzahl (Erhaltung des Verkehrsflusses). Die Durchsatzrate λ_i der Stufe i bilanziert sich nach

$$\lambda_i = \lambda_{oi} + \sum_{j=1}^N \lambda_j q_{ji}, \quad i = 1, 2, \dots, N. \tag{5.6}$$

Die Lösung des Gleichungssystems (5.6) ergibt sämtliche Durchsatzraten der Netzstufen. Unter der Stationaritätsbedingung muß für jede Netzstufe (eine Bedienungseinheit) gelten:

$$A_i = \lambda_i / \mu_i < 1, \quad i = 1, 2, \dots, N, \quad (5.7)$$

wobei A_i das Verkehrsangebot an Stufe i bedeutet. Da nur reine Wartesysteme betrachtet werden, ist A_i zugleich die Auslastung der Bedienungseinheit, welche im stationären Falle unterhalb von eins liegen muß. Die Verkehrsraten der einzelnen Zweige sind

$$\lambda_{ij} = \lambda_i q_{ij}, \quad \begin{matrix} i = 1, 2, \dots, N \\ j = 0, 1, \dots, N. \end{matrix} \quad (5.8)$$

Die Verkehrsraten sind exakt, solange sich das Netz im stationären Zustand befindet und insbesondere unabhängig von speziellen Voraussetzungen der Prozesse.

b) Wartezeit und Ausgangsprozeß des Wartesystems GI/G/1

Im allgemeinen Falle wird eine Netzstufe i nach Bild 5.3a) betrachtet. Entsprechend der Annahmen (3) und (4) in 5.4.2 wird der durch Superposition in C_i entstehende Eingangsprozeß als Erneuerungsprozeß (GI) betrachtet mit Ankunftsrate λ_i und Variationskoeffizient der Ankunftsabstände c_{Ai} . Der allgemeine Bedienungsprozeß (G) wird durch die Bedienungsrate $\mu_i = 1/h_i$ (h_i = mittlere Bedienungsdauer) sowie dem Variationskoeffizienten der Bedienungszeiten c_{Hi} vorgegeben. Die Eigenschaften der Stufe i werden beschrieben durch die Wartezeit T_{Wi} , die Durchlaufzeit T_{Fi} , die Anzahl X_i von Anforderungen in Stufe i sowie dem Abstand T_{Di} von fertig bedienten Anforderungen. Für die nachfolgende Netzanalyse interessieren wir uns insbesondere für

$$w_i = E[T_{Wi}] \quad (5.9a)$$

$$f_i = E[T_{Fi}] = w_i + h_i \quad (5.9b)$$

$$N_i = E[X_i] = \lambda_i f_i = \Omega_i + A_i \quad (5.9c)$$

$$c_{Di}^2 = E[T_{Di}^2] / E[T_{Di}]^2 - 1 = \lambda_i^2 E[T_{Di}^2] - 1 \quad (5.9d)$$

In Gl. (5.9.c) bedeutet Ω_i die mittlere Warteschlangenlänge der wartenden Anforderungen in Stufe i .

Exakte Lösungen für Wartesysteme des Typs GI/G/1 sind nur für einige Sonderfälle bekannt wie M/G/1, GI/M/1, D/D/1, $E_k/E_1/1$. Für den allgemeinen Fall wurden daher Näherungsformeln für die mittlere Wartezeit w_i und die Wartewahrscheinlichkeit

$w_i = P\{T_{Wi} > 0\}$ entwickelt, welche nur die ersten beiden Momente der Ankunfts- und Bedienungsprozesse berücksichtigen, siehe W. Krämer und M. Langenbach-Belz [113]. Das Ergebnis für w_i lautet:

$$w_i = h_i \cdot \frac{A_i}{2(1-A_i)} \cdot (c_{Ai}^2 + c_{Hi}^2) \cdot g(A_i, c_{Ai}^2, c_{Hi}^2), \quad (5.10a)$$

wobei

$$g(A_i, c_{Ai}^2, c_{Hi}^2) = \begin{cases} \exp\left[-\frac{2(1-A_i)}{3A_i} \cdot \frac{(1-c_{Ai}^2)^2}{c_{Ai}^2 + c_{Hi}^2}\right], & c_{Ai} < 1 \\ \exp\left[-(1-A_i) \cdot \frac{c_{Ai}^2 - 1}{c_{Ai}^2 + c_{Hi}^2}\right], & c_{Ai} \geq 1 \end{cases} \quad (5.10b)$$

Wir bemerken, daß Gl. (5.10) im Falle $A_i \rightarrow 1$ mit der "heavy-traffic"-Theorie übereinstimmt sowie die exakten Ergebnisse der Systeme M/G/1 und D/D/1 als Sonderfall enthält. Die Approximationsformeln wurden durch zahlreiche Simulationen als hinreichend genau bestätigt. Mit w_i liegen gleichzeitig f_i und N_i fest.

Der Ausgangsprozeß der Netzstufe ist im wesentlichen durch die Ausgangsabstände T_{Di} gekennzeichnet. Exakte Ergebnisse über Ausgangsprozesse sind nur bekannt für Wartesysteme des Typs M/M/n [72], M/D/1 [114] sowie $H_2/M/n$ [115]. Während für M/M/n der Ausgangsprozeß wieder ein Markoff-Prozeß ist, verlieren in nahezu allen anderen Fällen die Ausgangsprozesse ihre Erneuerungseigenschaft [116].

Zwischen dem zweiten Moment der Ausgangsabstände und der mittleren Wartezeit w_i besteht jedoch eine allgemeine Beziehung (siehe [117]), welche in unserer Darstellung auf folgende Form gebracht werden kann:

$$c_{Di}^2 = c_{Ai}^2 + 2A_i^2 c_{Hi}^2 - 2A_i(1-A_i) \cdot \frac{w_i}{h_i} \quad (5.11)$$

Mit Gl. (5.10) liegt somit c_{Di} fest:

$$c_{Di}^2 = c_{Ai}^2 + 2A_i^2 c_{Hi}^2 - A_i^2 (c_{Ai}^2 + c_{Hi}^2) \cdot g(A_i, c_{Ai}^2, c_{Hi}^2) \quad (5.12)$$

Wir bemerken, daß der Ausdruck Gl. (5.12) im Falle $A_i \rightarrow 0$ bzw. $A_i \rightarrow 1$ die exakten Ergebnisse liefert ($c_{Di} \rightarrow c_{Ai}$ bzw. $c_{Di} \rightarrow c_{Hi}$). Die Ausgangsprozeßformel (5.12) hat sich als außerordentlich genau erwiesen, vergl. Abschnitt 5.4.5. Ferner sei bemerkt, daß anstelle von Gl. (5.10a,b) und (5.11-12) jede andere Lösung für Wartesysteme des Typs GI/G/1 angewendet werden kann, sofern diese auf nur zwei Momenten des Eirungsprozesses beruht und die mittlere Wartezeit liefert (z.B. Phasenmethode für GI).

c) Verzweigung von Verkehrsflüssen

Betrachtet werde ein Prozeß von Anforderungen, welcher durch die VF der Ankunftsabstände T , $F(t) = P\{T \leq t\}$, bzw. seine Ankunftsrate λ sowie den Variationskoeffizienten c der Ankunftsabstände gegeben ist (z.B. am Ausgang einer Bedienungsstufe). Die Anforderungen werden in Punkt D nach Richtung j verzweigt entsprechend der Wahrscheinlichkeit q_j , $j = 0, 1, \dots, N$, vergl. Bild 5.4. Wir interessieren uns für die Charakteristik der $(N+1)$ Komponentenprozesse, d.h. für die VF der Ankunftsabstände T_j in Richtung j , $F_j(t) = P\{T_j \leq t\}$, $j = 0, 1, \dots, N$, bzw. deren ersten beiden Momente.

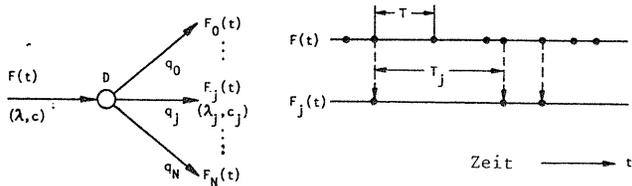


Bild 5.4. Verzweigung eines Verkehrsflusses in $(N+1)$ Komponentenprozesse

Unter der Annahme, daß $F(t)$ ein Erneuerungsprozeß ist, läßt sich T_j ausdrücken als Summe einer zufälligen Anzahl X von aufeinanderfolgenden Realisierungen der Ankunftsabstände $T^{(v)}$ des Ankunftsprozesses, d.h.

$$T_j = \sum_{v=1}^X T^{(v)} \quad (5.13)$$

Die Wahrscheinlichkeit $p_x = P\{X = x\}$, daß sich T_j aus x Ankunftsabständen T konstituiert, ist

$$p_x = \begin{cases} 0 & \text{für } x = 0 \\ q_j (1 - q_j)^{x-1} & \text{für } x \geq 1 \end{cases} \quad (5.14)$$

Die Wahrscheinlichkeiten p_x gehorchen einer geometrischen Verteilung. Die erzeugende Funktion dieser Verteilung ist

$$G(z) = \sum_{x=0}^{\infty} p_x z^x = \frac{q_j z}{1 - (1 - q_j)z}, \quad |z| < 1 \quad (5.15)$$

Es seien

$$\phi(s) = \int_{0-}^{\infty} e^{-st} dF(t) \quad (5.16a)$$

und

$$\phi_j(s) = \int_{0-}^{\infty} e^{-st} dF_j(t) \quad (5.16b)$$

die Laplace-Stieltjes-Transformierten der VF $F(t)$ bzw. $F_j(t)$ (d.h. die Laplace-Transformierten der entsprechenden Verteilungsdichtefunktionen). Die VF $F_j(t)$ kann zunächst durch eine bedingte VF $P\{T_j \leq t | X = x\}$ ausgedrückt werden:

$$F_j(t) = \sum_{x=0}^{\infty} p_x \cdot P\{T_j \leq t | X = x\} \quad (5.17a)$$

Die bedingte VF $P\{T_j \leq t | X = x\}$ ist die VF einer Summe von x unabhängigen und identisch verteilten Zufallsvariablen $T^{(v)}$, $v = 1, 2, \dots, x$, welche jeweils nach $F(t)$ verteilt sind. Die VF einer Summe von unabhängigen Zufallsvariablen führt auf die Faltung der einzelnen VF bzw. deren Laplace-Stieltjes-Transformierte auf das Produkt der einzelnen Laplace-Stieltjes-Transformierten, d.h.

$$\phi_j(s) = \sum_{x=0}^{\infty} p_x [\phi(s)]^x \quad (5.17b)$$

Durch Vergleich von (5.17b) mit (5.15) finden wir das Endergebnis

$$\phi_j(s) = G(\phi(s)) = \frac{q_j \phi(s)}{1 - (1 - q_j) \phi(s)} \quad (5.18)$$

Aus Gl. (5.18) folgen Mittelwert und Varianz von T_j :

$$E[T_j] = - \left. \frac{d\phi_j(s)}{ds} \right|_{s=1} = E[T] \cdot E[X] \quad (5.19a)$$

$$\text{VAR}[T_j] = \left. \frac{d^2\phi_j(s)}{ds^2} \right|_{s=1} - E[T_j]^2 = E[T]^2 \cdot \text{VAR}[X] + \text{VAR}[T] \cdot E[X] \quad (5.19b)$$

Mit $E[X] = 1/q_j$, $\text{VAR}[X] = (1 - q_j)/q_j^2$, $E[T] = 1/\lambda$, $\text{VAR}[T] = c^2/\lambda^2$ finden wir aus Gl.(5.19ab) schließlich

$$\lambda_j = E[T_j]^{-1} = \lambda q_j, \quad (5.20a)$$

$$c_j^2 = q_j c^2 + (1 - q_j), \quad j = 0, 1, \dots, N. \quad (5.20b)$$

Gl. (5.20a) stimmt sinngemäß mit Gl. (5.8) überein. Eine interessante und anschaulich weniger einsichtige Beziehung folgt aus Gl. 5.20b)

$$\sum_{i=0}^N c_i^2 = c^2 + N, \quad (5.20c)$$

welche die Variationskoeffizienten der Komponentenprozesse mit dem Variationskoeffizienten des Ausgangsprozesses in Verbindung setzt.

Die Ergebnisse dieses Abschnittes sind nur exakt im Falle eines rekurrenten Prozesses $F(t)$. In Abschnitt 5.4.5 wird gezeigt, daß die Ergebnisse auch für nicht-rekurrente Prozesse eine sehr gute Genauigkeit liefern. Schließlich sei noch bemerkt, daß im Falle eines Markoffschen Prozesses, d.h. $F(t) = 1 - \exp(-\lambda t)$, sämtliche Komponentenprozesse $F_j(t)$ wiederum Markoffsche Prozesse sind.

d) Überlagerung von Verkehrsflüssen

Das duale Problem zur Verzweigung von Verkehrsflüssen ist die Überlagerung von Verkehrsflüssen. Gegeben seien $(N+1)$ unabhängige und stationäre Erneuerungsprozesse mit den VFn $F_j(t) = P\{T_j \leq t\}$ der Ankunftsabstände T_j , den Ankunftsrate λ_j , sowie den Variationekoeffizienten c_j , $j = 0, 1, \dots, N$. Wir interessieren uns für den Summenprozeß nach der Überlagerung sämtlicher Komponentenprozesse bzw. der VF $F(t) = P\{T \leq t\}$ der Ankunftsabstände T . Zunächst werde das elementarere Problem der Überlagerung zweier Verkehrsflüsse betrachtet, vergl. Bild 5.5.

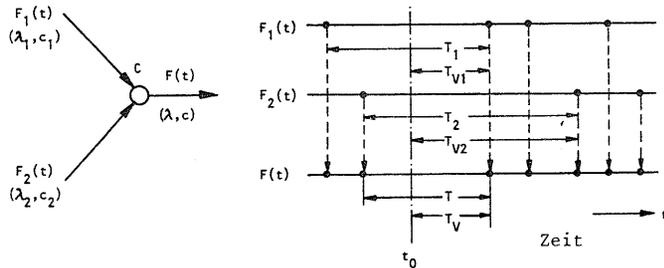


Bild 5.5. Überlagerung zweier Verkehrsflüsse

Zur Bestimmung des Summenprozesses werden zu einem beliebigen Zeitpunkt t_0 die sog. Vorwärts-Erneuerungszeiten T_{V1} , T_{V2} und T_V betrachtet, welche die Zeitspannen bis zum nächsten Ereignis der Komponentenprozesse bzw. des Summenprozesses angeben. Im Falle stationärer Erneuerungsprozesse gilt für die VF der

Vorwärts-Erneuerungszeiten $V_j(t) = P\{T_{Vj} \leq t\}$, $j = 1, 2$, nach der Erneuerungstheorie [71] :

$$\frac{d}{dt} V_j(t) = \lambda_j F_j^c(t), \quad j = 1, 2, \quad (5.21a)$$

wobei $F_j^c(t) = 1 - F_j(t)$ die komplementäre VF von $F_j(t)$. Aus (5.21a) folgt die komplementäre VF der Vorwärts-Erneuerungszeiten durch Integration zu

$$V_j^c(t) = \int_{u=t}^{\infty} \lambda_j F_j^c(u) du, \quad j = 1, 2. \quad (5.21b)$$

Die Vorwärts-Erneuerungszeit T_V des Summenprozesses ist das Minimum der beiden voneinander unabhängigen Vorwärts-Erneuerungszeiten T_{V1} und T_{V2} , d.h.

$$P\{T_V > t\} = P\{T_{V1} > t\} \cdot P\{T_{V2} > t\} \quad (5.22a)$$

oder

$$\begin{aligned} V^c(t) &= V_1^c(t) \cdot V_2^c(t) \\ &= 1 - \left[\int_{u=t}^{\infty} \lambda_1 F_1^c(u) du \right] \cdot \left[\int_{u=t}^{\infty} \lambda_2 F_2^c(u) du \right]. \end{aligned} \quad (5.22b)$$

Unter der Approximationsvoraussetzung (3) gilt für den Summenprozeß die Gl.(5.21a) entsprechende Beziehung zwischen der VF der Vorwärts-Erneuerungszeiten $V(t)$ und der VF der Ankunftsabstände $F(t)$:

$$\frac{d}{dt} V(t) = \lambda \cdot F^c(t). \quad (5.23)$$

Aus Gl. (5.22b) und (5.23) folgt schließlich das Ergebnis

$$F(t) = 1 - \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \left[F_1^c(t) \cdot \int_{u=t}^{\infty} F_2^c(u) du + F_2^c(t) \cdot \int_{u=t}^{\infty} F_1^c(u) du \right]. \quad (5.24)$$

Die Momente der Ankunftsabstände T des Summenprozesses sind definiert durch

$$E[T^k] = \int_{0-}^{\infty} t^k dF(t) = -\frac{1}{\lambda} \int_{0-}^{\infty} t^k \cdot \frac{d^2 V}{dt^2} dt. \quad (5.25)$$

Leider lassen sich aus (5.25) die Momente $E[T^k]$ nicht direkt ausdrücken in den Momenten $E[T_j^k]$, $j = 1, 2$, der Komponentenprozesse außer für $k = 1$; in diesem Falle folgt

$$\lambda = E[T]^{-1} = \lambda_1 + \lambda_2, \quad (5.26)$$

was die notwendige Beziehung zur Erhaltung des Verkehrsflusses ausdrückt. Für die

höheren Momente muß zunächst $F(t)$ nach Gl. (5.24) bestimmt werden, woraus die Momente nach Gl. (5.25) folgen. Dieses Verfahren läßt sich immer dann durchführen, wenn die VF der Komponentenprozesse gegeben sind. Das Ergebnis Gl. (5.24) kann über die Netzanwendung hinaus zur Analyse von Wartesystemen des Typs $GI_1+GI_2/M/n$ angewendet werden, bei welchen der Ankunftsprozeß durch Überlagerung zweier Erneuerungsprozesse entsteht.

Im Zusammenhang mit der zweimomentigen Netzanalyse sind die Komponentenprozesse i.a. nur in ihren ersten beiden Momenten (λ_j, c_j) vorgegeben. Um das Ergebnis (5.24) anwenden zu können, wird hier das Prinzip der Ersatzprozesse eingeführt: Aus der Vorgabe von (λ_j, c_j) wird ein möglichst einfacher Ersatzprozeß konstruiert, welcher in den ersten beiden Momenten mit dem tatsächlichen Prozeß übereinstimmt. Wir verwenden hierfür einfache Phasendarstellungen aus einer Reihenschaltung einer konstanten (D) und exponentiellen (M) Phase für $c_j \leq 1$ sowie einer H_2 -Verteilung für $c_j \geq 1$, vergl. Bild 5.6.

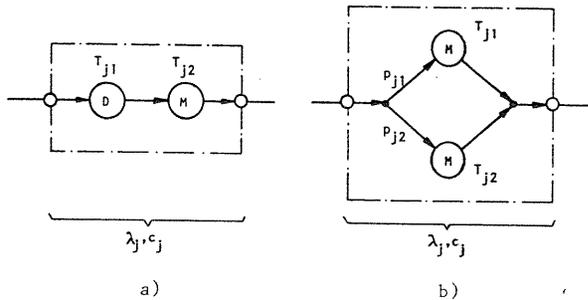


Bild 5.6. Ersatzprozesse für
 a) hypoexponentiellen Charakter ($c_j \leq 1$)
 b) hyperexponentiellen Charakter ($c_j \geq 1$)

Die Ersatzprozesse sind mathematisch beschrieben durch

$$F_j(t) = \begin{cases} \begin{cases} 0 & , 0 \leq t \leq t_{j1} \\ 1 - \exp(-\epsilon_{j2}(t-t_{j1})) & , t \geq t_{j1} \end{cases} & , c_j \leq 1 \end{cases} \quad (5.27a)$$

$$1 - p_{j1} \exp(-\epsilon_{j1}t) - p_{j2} \exp(-\epsilon_{j2}t), \quad c_j \geq 1, \quad (5.27b)$$

wobei die Parameter aus der Vorgabe von (λ_j, c_j) bestimmt werden:

$$c_j \leq 1: \quad \epsilon_{j1} = \frac{\lambda_j}{1-c_j}, \quad \epsilon_{j2} = \frac{\lambda_j}{c_j} \quad (5.27c)$$

$$c_j \geq 1: \quad \epsilon_{j1,2} = \lambda_j \left[1 \pm \sqrt{\frac{c_j^2 - 1}{c_j^2 + 1}} \right], \quad p_{j1,2} = \frac{\epsilon_{j1,2}}{2\lambda_j}, \quad (5.27d)$$

wobei $p_{j1}t_{j1} = p_{j2}t_{j2}$.

Mit Hilfe dieser Darstellung der Komponentenprozesse lassen sich aus Gl. (5.24) und (5.25) die Momente berechnen. Bei der praktischen Durchführung müssen drei Fälle von unterschiedlichen Kombinationen der beiden Ersatzprozeßtypen unterschieden werden. Die Ergebnisse sind etwas umfangreich und sollen hier nicht explizit angeführt werden, vergl. [119, 120]. Schließlich sei bemerkt, daß der resultierende Summenprozeß i.a. kein Erneuerungsprozeß mehr ist. Im Sonderfalle von Markoffschen Komponentenprozessen ergibt der Summenprozeß wiederum einen Markoff-Prozeß.

Die Verallgemeinerung der Superposition von zwei auf $(N+1)$ Prozesse erfolgt rekursiv, z.B. nach dem in Bild 5.7 angegebenen Algorithmus, wobei in N aufeinanderfolgenden Schritten jeweils zwei Prozesse überlagert werden.

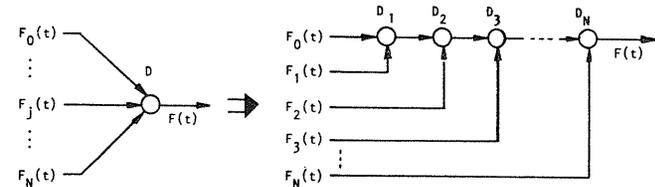


Bild 5.7. Überlagerung von $(N+1)$ Verkehrsflüssen durch N rekursive Überlagerungen von jeweils 2 Verkehrsflüssen

c) Ersetzung von Rückkopplungen

Die Annahme der Erneuerungseigenschaft ist insbesondere bei Netzen mit starken Rückkopplungen kritisch. Dies zeigte sich im Verlaufe der Entwicklung des Analyseverfahrens vor allem bei Netzstufen, deren Ausgang mit größerer Wahrscheinlichkeit auf den Eingang derselben Stufe rückgekoppelt ist. Um den Einfluß der

Abhängigkeiten von Ausgangs- und Eingangsprozess abzuschwächen, wird eine Ersetzung der rückgekoppelten Netzstufe durch eine Stufe ohne Rückkopplung vorgenommen, vergl. Bild 5.8.

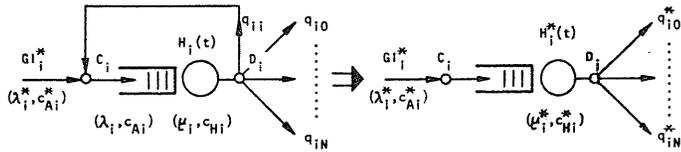


Bild 5.8. Zur Ersetzung stufeninterner Rückkopplungen

In der ursprünglichen Netzstufe i erfährt eine von außerhalb der Netzstufe eintreffende Anforderung eine zufällige, geometrisch verteilte Anzahl von Bedienungsphasen infolge der unabhängigen Rückkopplungswahrscheinlichkeit q_{ii} , welche möglicherweise durch Phasen anderer Anforderungen in dieser Stufe getrennt sind. In dem Ersatzsystem erfährt dieselbe Anforderung nur eine Bedienungsphase, welche jedoch in Mittelwert und VF exakt mit der Gesamtbedienungszeit der ursprünglichen Stufe übereinstimmt. Die äquivalente Gesamtbedienungszeit sei T_{Hi}^* und setzt sich aus einer Summe von identisch verteilten Bedienungszeiten T_{Hi} der ursprünglichen Stufe zusammen, wobei die Anzahl X der Summanden zufällig ist und einer geometrischen Verteilung genügt. Die VF von T_{Hi}^* ist analog konstruiert zu der VF der Ankunftsabstände eines Zweigprozesses wie in Abschnitt 5.4.3c) mit den Entsprechungen $T_{Hi}^* \leftrightarrow T_i$ bzw. $(1 - q_{ii}) \leftrightarrow q_j$. Zwischen den LS-Transformierten $\phi_{Hi}(s)$ und $\phi_{Hi}^*(s)$ der ursprünglichen bzw. äquivalenten Bedienungszeit-VF besteht entsprechend Gl. (5.18) die Beziehung

$$\phi_{Hi}^*(s) = \frac{(1 - q_{ii}) \phi_{Hi}(s)}{1 - q_{ii} \phi_{Hi}(s)} \quad (5.28)$$

Aus Gl. (5.18) folgen unmittelbar die neuen Parameter der Stufe ohne interne Rückkopplung:

$$\mu_i^* = \mu_i (1 - q_{ii}) \quad (5.29a)$$

$$c_{Hi}^{*2} = q_{ii} + (1 - q_{ii}) c_{Hi}^2 \quad (5.29b)$$

$$q_{ij}^* = q_{ij} / (1 - q_{ii}), \quad j \neq i \quad (q_{ii}^* \equiv 0) \quad (5.29c)$$

Durch diese Transformation, angewendet auf jede Stufe des Netzes mit stufeninterner Rückkopplung ($q_{ii} > 0, i = 1, 2, \dots, N$), wird das gesamte Netz rekonfiguriert in ein Netz ohne stufeninterne Rückkopplungen. Im weiteren Verlauf werden nur noch Netze dieser Art betrachtet und der oben eingeführte Stern zur Kennzeichnung der ersetzten Stufe weggelassen.

Die Ersetzungsmethode ist exakt in Hinblick auf isoliert betrachtete Stufen des Typs M/G/1 für die Verteilung der Anzahl von Anforderungen in der Stufe und die mittlere Durchlaufzeit [121]. Hierin drückt sich bereits eine "Robustheit" der Lösung in Bezug auf die Verzweigungswahrscheinlichkeit aus, welche später für Netze der Produktlösungsform nachgewiesen wurde [96]. Die Äquivalenz gilt jedoch nicht in Bezug auf die höheren Momente der Durchlaufzeit-VF. Die Äquivalenz wird im vorliegenden Netzverfahren auf Stufen des Typs GI/G/1 übertragen. Die Genauigkeit der Ersetzung wird in Abschnitt 5.4.5 nachgewiesen.

In Bezug auf das gesamte Netz wurde mit diesem Verfahren eine beträchtliche Genauigkeitssteigerung erzielt, wohingegen ohne die Transformation größere Abweichungen in Kauf genommen werden müssen. Wie in diesem Falle ist es allgemein von Vorteil, Netzteile mit starken "inneren Bindungen" nicht zu zerlegen, sondern möglichst zu einem Komplex zusammenzufassen und als Ganzes zu behandeln in Bezug auf die Analyse des Netzteils selbst wie auch in Bezug auf dessen Wechselwirkung mit der restlichen Netzumgebung.

f) Bestimmung der Verkehrsflüsse (2. Momente)

Bei rückkopplungsfreien Netzen (von stufeninternen Rückkopplungen abgesehen, welche mit der Transformation e) beseitigt wurden) existiert stets eine Folge von Netzstufen derart, daß Schritt für Schritt die 2. Momente der Eingangs- und Ausgangsprozesse bestimmt werden können. Bei nicht-rückkopplungsfreien Netzen kann die Überlagerungsprozedur an bestimmten Eingangspunkten C nicht direkt ausgeführt werden, da das 2. Moment eines oder mehrerer Rückkopplungs-Verkehrszweige noch nicht festliegt (vergl. Eingangspunkt der Stufe 1 in Bild 5.3b).

In dem vorliegenden Netzverfahren wird dieses Problem durch einen Iterationsalgorithmus gelöst: Alle Variationskoeffizienten c_{ij} , $i = 1, 2, \dots, N, j = 0, 1, \dots, N$, werden zunächst auf 1 festgesetzt. Die einzelnen Netzstufen werden in einer willkürlich festgelegten Folge (welche sich in der Numerierung $i = 1, 2, \dots, N$ ausdrückt) analysiert, wobei nach jeder Stufe die abzweigenden Verkehrsflüsse bezüglich ihrer Variationskoeffizienten auf den neuesten Stand gebracht werden. Der Algorithmus konvergiert in wenigen Schritten (typisch < 10) auf den stabilen Endwert aller Variationskoeffizienten.

g) Bestimmung der Durchlaufzeiten

Zur Bestimmung der Durchlaufzeiten einer Anforderung werden folgende Größen benötigt:

- $e_{i(a)}$ Mittlere Anzahl von Durchläufen einer Anforderung des externen Eingangs a durch Stufe i, $i, a = 1, 2, \dots, N$.
- e_i Mittlere Anzahl von Durchläufen einer beliebigen Anforderung durch Stufe i, $i = 1, 2, \dots, N$.

Letztere Größe liegt bereits fest durch die in a) berechneten Verkehrsflüsse:

$$e_i = \frac{\lambda_i}{\sum_{a=1}^N \lambda_{0a}}, \quad i = 1, 2, \dots, N. \quad (5.30)$$

Zur Berechnung von $e_{i(a)}$ werden sämtliche externen Ankunftsrate zu Null gesetzt außer an Eingang a. Die Ankunftsrate $\lambda_i(a)$ bestimmt sich analog zu Gl. (5.6):

$$\lambda_i(a) = \lambda_{0i} \delta_{ia} + \sum_{j=1}^N \lambda_j(a) q_{ji}, \quad i = 1, 2, \dots, N \quad (5.31)$$

wobei δ_{ia} das Kronecker-Symbol bezeichnet⁺⁾ . Die mittlere Anzahl von Durchläufen einer Anforderung des externen Eingangs a durch Stufe i ist

$$e_{i(a)} = \frac{\lambda_i(a)}{\lambda_{0a}}, \quad \begin{matrix} i = 1, 2, \dots, N \\ a = 1, 2, \dots, N. \end{matrix} \quad (5.32)$$

Im Mittel benötigt eine Anforderung $f_i = w_i + h_i$ zum Durchlaufen der Stufe i. Die Gesamt-Durchlaufzeiten f und f(a) einer beliebigen Anforderung bzw. einer Anforderung des externen Eingangs a bestimmen sich schließlich aus

$$f = \sum_{i=1}^N e_i f_i \quad (5.33a)$$

und

$$f(a) = \sum_{i=1}^N e_{i(a)} f_i, \quad a = 1, 2, \dots, N. \quad (5.33b)$$

⁺⁾ Definition von δ_{ia} : $\delta_{ia} = \begin{cases} 0, & i \neq a \\ 1, & i = a \end{cases}$

Gl.(5.33a) kann auch über Gl.(5.33b) abgeleitet werden: Sei $\lambda = \sum_{i=1}^N \lambda_{0i}$, dann gilt:

$$\begin{aligned} f &= \sum_{a=1}^N \frac{\lambda_{0a}}{\lambda} \cdot f(a) = \sum_{a=1}^N \frac{\lambda_{0a}}{\lambda} \cdot \sum_{i=1}^N e_{i(a)} f_i = \sum_{a=1}^N \frac{\lambda_{0a}}{\lambda} \cdot \sum_{i=1}^N \frac{\lambda_i(a)}{\lambda_{0a}} \cdot f_i \\ &= \frac{1}{\lambda} \cdot \sum_{i=1}^N \lambda_i f_i = \sum_{i=1}^N e_i f_i = \frac{1}{\lambda} \cdot \sum_{i=1}^N E[X_i] = \frac{1}{\lambda} \cdot E[X]. \end{aligned}$$

Der letzte Ausdruck verkörpert dabei Little's Theorem angewandt auf das gesamte Netz.

5.4.4 Algorithmus

Die im vorausgegangenen Abschnitt dargestellten Lösungsschritte des Verfahrens wurden in einem Algorithmus zusammengefaßt und durch ein ALGOL-Programm implementiert [120]. Das prinzipielle Ablaufdiagramm des Algorithmus ist in Bild 5.9 dargestellt. Der Algorithmus ist rechenzeitarm (wenige Sekunden CPU-Zeit) und benötigt ca. $5N^2$ Speicherplätze für Daten. Der Algorithmus liefert die exakten Ergebnisse für Markoffsche Warteschlangennetze mit konstanten externen Ankunftsrate.

5.4.5 Genauigkeitsnachweise

In diesem Abschnitt werden die approximativen Lösungsschritte des Verfahrens sowie die erzielten Gesamtergebnisse auf ihre Genauigkeit hin überprüft.

a) Elementaroperationen

a1) Das Wartesystem GI/G/1

Die approximative Formel Gl.(5.10) für die mittlere Wartezeit sowie eine entsprechende Formel für die Wartewahrscheinlichkeit wurden durch intensive Verkehrssimulationen bestätigt [113]. An dieser Stelle soll deshalb nur die Genauigkeit des Variationskoeffizienten des Ausgangsprozesses c_D behandelt werden. Bild 5.10 zeigt die Ergebnisse (aufgetragen für c_D^2) im Falle der Systemtypen $E_2/G/1$ und $H_2/G/1$ ($c_A = 1.5$) verglichen zur Simulation (100 000 Ereignisse, 95%-Vertrauensintervalle). Die Ergebnisse zeigen eine sehr gute Genauigkeit. Wir bemerken nochmals, daß für das Angebot $A \rightarrow 0$ und $A \rightarrow 1$ die Ergebnisse exakt sind.

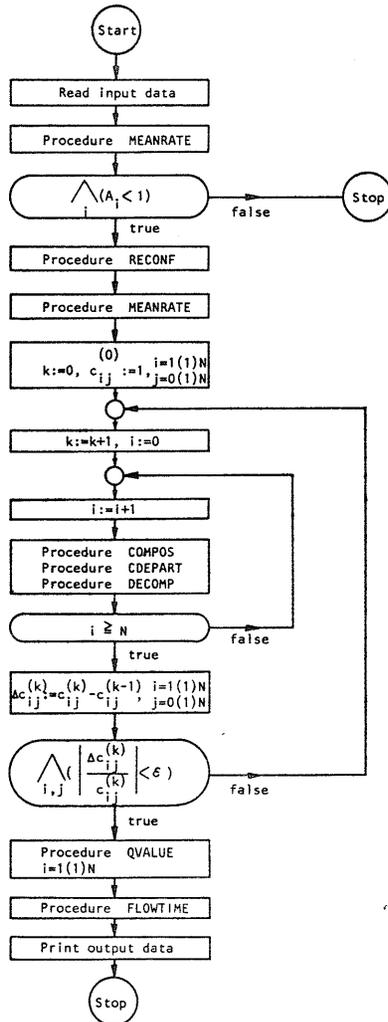


Bild 5.9. Ablaufdiagramm des Analyseverfahrens für allgemeine Warteschlangennetze

Prozeduren:

- MEANRATE: Bestimmung der Ankunftsraten λ_i , $i = 1, 2, \dots, N$
- RECONF: Rekonfiguration durch Transformation sämtlicher stufeninterner Rückkopplungen
- COMPOS: Überlagerung von $(N+1)$ Komponentenprozessen
- DECOMP: Verzweigung eines Ausgangsprozesses in $(N+1)$ Richtungen
- CDEPART: Ausgangsprozeß-Variationskoeffizient einer Stufe GI/G/1
- QVALUE: Charakteristische Größen des Wartesystems GI/G/1
- FLOWTIME: Berechnung der mittleren Durchlaufzeiten

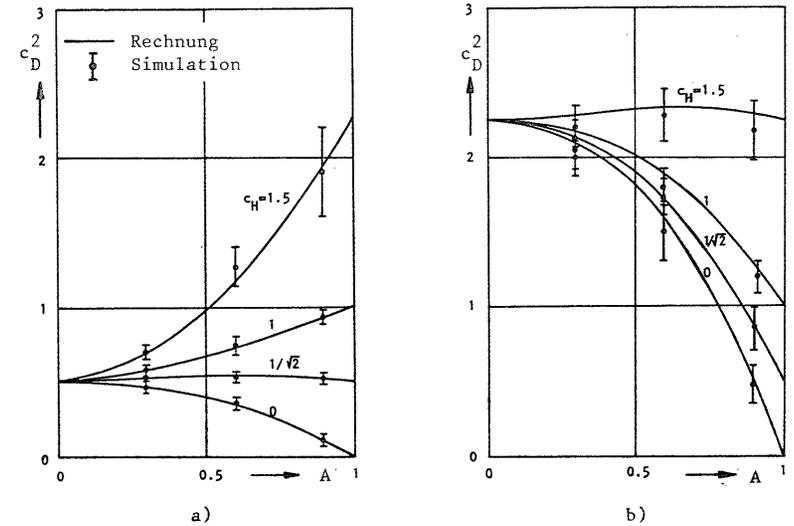


Bild 5.10. Variationskoeffizienten c_D des Ausgangsprozesses in Abhängigkeit des Verkehrsangebots A
 a) Wartesystem $E_2/G/1$
 b) Wartesystem $H_2/G/1$ ($c_A = 1.50$)

a2) Verzweigung von Verkehrsflüssen

Bild 5.11 zeigt die Abhängigkeiten $c_j^2(c^2)$ im Falle der Verzweigung eines rekurrenten Prozesses (Bild 5.11a; dieses Ergebnis ist exakt) sowie eines nichtrekurrenten Prozesses (Bild 5.11b) mit dem Variationskoeffizienten c sowie den Werten der Verzweigungswahrscheinlichkeit $q_j = 0, 1, 0, 5$ und $0, 9$. Der nichtrekurrente Prozeß wurde dabei erzeugt als Ausgangsprozeß eines $M/G/1$ -Wartesystems bei der Belastung $A = 0, 6$ (Hinweis: für $A \rightarrow 0$ und $A \rightarrow 1$ sind die Ausgangsprozesse wiederum rekurrent!). Bild 5.11 zeigt, daß die Verzweigungsoperation bei nichtrekurrenten Prozessen außerordentlich genau ist.

a3) Überlagerung von Verkehrsflüssen

Die Ergebnisse des Variationskoeffizienten c zweier überlagerter Prozesse (λ, c_1) bzw. (λ, c_2) zeigt Bild 5.12 im Falle zweier rekurrenter (5.12a) bzw. nichtrekurrenter Komponentenprozesse (5.12b). Der quadrierte Variationskoeffizient des ersten Komponentenprozesses, c_1^2 , bildet die Abszisse, während c_2^2 als Parameter in die Kurven eingeht. Die nichtrekurrenten Prozesse wurden wiederum als Ausgangsprozesse von Wartesystemen des Typs $M/G/1$ bei $A = 0, 6$ erzeugt. Während die Überlagerung rekurrenter Prozesse außerordentlich gute Ergebnisse liefert, beobachten wir im Falle nichtrekurrenter Komponentenprozesse Abweichungen bis

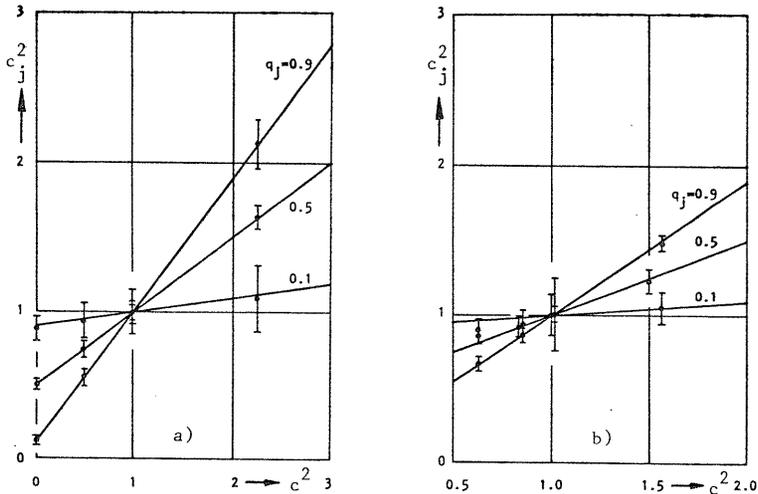


Bild 5.11. Variationskoeffizient des verzweigten Prozesses c_j in Abhängigkeit des Variationskoeffizienten c des ursprünglichen Prozesses (Parameter q_j)
 a) Verzweigung eines rekurrenten Prozesses
 b) Verzweigung eines nichtrekurrenten Prozesses

zu 15% (im Falle der Überlagerung zweier Ausgangsprozesse von M/D/1-Stufen). Die Überlagerung zweier deterministischer Ströme ($c_1=c_2=0$) ist im Falle von Bild 5.12a ausgeklammert; die unter 5.4.3d) entwickelte Theorie liefert den Wert $c^2 = \frac{1}{3}$, welcher exakt dem Erwartungswert entspricht, wenn die beiden Komponentenströme eine zufällige Phase untereinander haben. In der Simulation liegt dagegen die Phase von der Initialisierung her fest und müßte z.B. bei jedem Teilttest zufallsabhängig geändert werden.

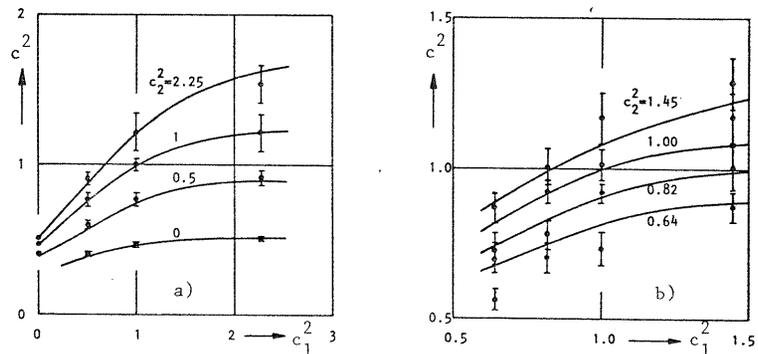


Bild 5.12. Variationskoeffizient c des überlagerten Prozesses in Abhängigkeit der Variationskoeffizienten c_1, c_2 der Komponentenprozesse
 a) Überlagerung zweier rekurrenter Prozesse
 b) Überlagerung zweier nichtrekurrenter Prozesse

a4) Ersetzung von Rückkopplungen

In Bild 5.13 sind schließlich die Ergebnisse zur Transformation von stufeninternen Rückkopplungen dargestellt. Betrachtet wird eine Stufe GI/G/1 mit Rückkopplung nach Bild 5.8, wobei $\lambda^* = 0,3$ die externe Ankunftsrate, $h = 1$ die mittlere Bedienungsdauer und $q = 0,5$ die Rückkopplungswahrscheinlichkeit (Index i weggelassen). Bild 5.13a) zeigt zunächst den Ausgangsprozess-Variationskoeffizienten c_D^* in Abhängigkeit der Variationskoeffizienten der Bedienungsdauern c_H (Abszisse) und der externen Ankunftsabstände c_A^* (Parameter) für die transformierte Stufe entsprechend Abschnitt 5.4.3e). Die Ergebnisse sind verglichen mit Simulationsergebnissen des ursprünglichen, rückgekoppelten Systems. In Bild 5.13b) sind die entsprechenden Ergebnisse für die mittlere Gesamt-Durchlaufzeit f dargestellt. Beide Ergebnisse zeigen eine gute Übereinstimmung mit der Simulation, obwohl die Äquivalenz nur im Falle exponentieller Ankunftsabstände nachgewiesen ist [121].

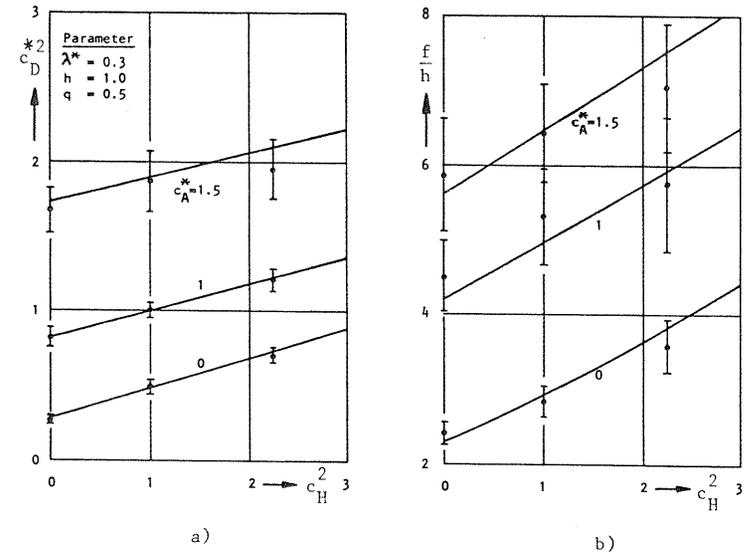


Bild 5.13. Zur Äquivalenz von Stufen mit und ohne Rückkopplungen
 a) Ausgangsprozess-Variationskoeffizient c_D^* in Abhängigkeit der Variationskoeffizienten c_A^* und c_H des Ankunfts- bzw. Bedienungsprozesses
 b) Mittlere Gesamt-Durchlaufzeit in Abhängigkeit von c_A^* und c_H

b) Gesamtergebnisse

Die Genauigkeit des Verfahrens für ganze Netze wurde an einer Reihe von Beispielen nachgewiesen. Betrachtet werde im folgenden ein Netzbeispiel nach Bild 5.14, welches aus $N = 9$ Bedienstufen besteht. Das Netz besitzt drei externe Markoffsche Ankunftsprozesse und wurde untersucht für homogene Bedienstufen ($h_{1-9} = 1, c_{H1-9}$ alle gleich) wie auch für inhomogene Bedienstufen ($h_{1-9} = 1, c_{H1-3}$ unterschiedlich zu c_{H4-9}).

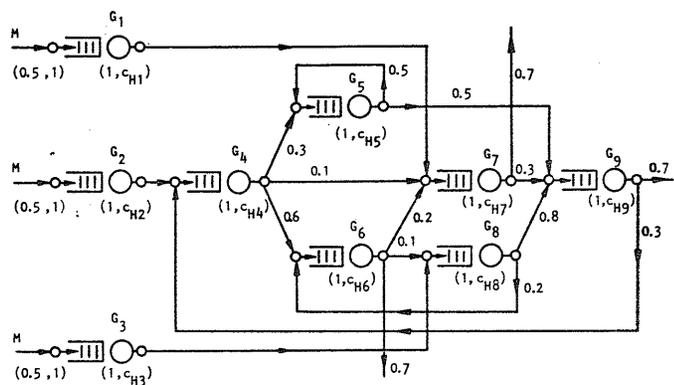


Bild 5.14. Warteschlangennetzbeispiel aus 9 Stufen

Bild 5.15 zeigt die mittlere Gesamtdurchlaufzeit f sowie die mittlere Durchlaufzeit f_4 der Stufe 4 im Inneren des Netzes in Abhängigkeit der Variationskoeffizienten c_{H1-9} des homogenen Netzes (Bild 5.15a) bzw. der Variationskoeffizienten c_{H4-9} mit c_{H1-3} als Parameter (Bild 5.15b). Bild 5.15a zeigt zwei weitere Kurven: die gestrichelt eingezeichneten Kurven sind die Ergebnisse, wenn alle Eingangsprozesse als Markoffsche Prozesse angenommen werden, während die punktierten Linien die Ergebnisse im Falle exponentiell verteilter Bedienstufen darstellen, d.h. wenn die Produktlösungsform angenommen wird.

Die Gesamtergebnisse werden von der Simulation her als hinreichend genau bestätigt. Der relativ geringe Unterschied bei Annahme von Markoffschen Eingangsprozessen rührt daher, daß das Netz nach Bild 5.14 infolge seiner Komplexität bereits nahezu Markoffsche Zweigprozesse liefert. Bei Vergleich mit den punktierten Kurven wird der Einfluß der Bedienstufen-Zeicharakteristik sehr deutlich.

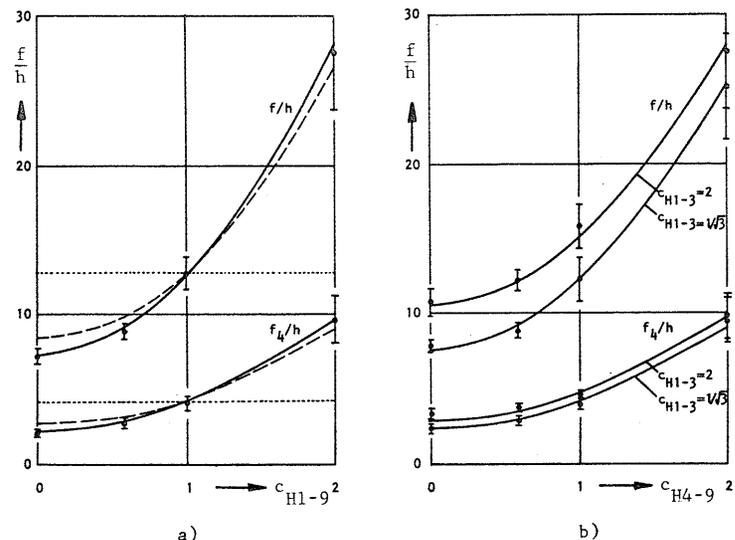


Bild 5.15. Mittlere Durchlaufzeiten f und f_4 in Abhängigkeit der Variationskoeffizienten c_H der Bedienstufen des Netzes nach Bild 5.14
 a) homogene Bedienstufen
 b) inhomogene Bedienstufen

Außer diesen Genauigkeitsnachweisen wurde das Verfahren ferner mit zwei verschiedenen Verfahren von Diffusionsapproximationen [102, 103] verglichen, wobei eine bessere Gesamtgenauigkeit erzielt wurde. Weitere Genauigkeitsnachweise finden sich in Abschnitt 6.4.4.

5.4.6 Kritische Diskussion und Verallgemeinerungen

a) Diskussion der Ergebnisse

Das vorgeschlagene Zerlegungsverfahren ist anwendbar auf offene Warteschlangennetze mit allgemeinen Ankunfts- und Bedienstufenprozessen. Die konsequente Berücksichtigung der ersten beiden Momente aller Prozesse scheint in den meisten Anwendungsfällen den wesentlichsten Prozesseinfluß wiederzugeben. Der mathematische Aufwand zur Auswertung ist durch die Modularisierung des Lösungsalgorithmus überschaubar, relativ einfach und rechenzeitarm.

Das Verfahren ist im Prinzip auf geschlossene Netze übertragbar. Erste Ergebnisse haben jedoch Ungenauigkeiten bis zu 30% ergeben infolge der stärkeren Abhängigkeiten zwischen den Stufen durch die konstante Anzahl von Anforderungen im Netz.

Ferner erlaubt das Verfahren in dieser Form nur zustandsunabhängige Ankunfts-raten. Die Zerlegungsmethode und die Charakterisierung der Verkehre an den Zerlegungsschnittstellen lassen dieses Verfahren jedoch für eine Reihe von Verallgemeinerungen prädestiniert erscheinen.

b) Verallgemeinerungen

b1) Mehrere Bedienungseinheiten pro Stufe

Jede GI/G/1-Stufe kann ersetzt werden durch eine entsprechende GI/G/n-Stufe, vergl. Bild 5.16a). In dem Lösungsalgorithmus sind lediglich die GI/G/1-Ergebnisse (Abschnitt 5.4.3b) durch entsprechende Ergebnisse für GI/G/n-Stufen zu ersetzen.

b2) Mehrere Klassen von Anforderungen

Eine weitere Verallgemeinerung auf R unterschiedliche Klassen von Anforderungen wie in Abschnitt 5.1.2 ist prinzipiell möglich. Die einzelnen Bedienstufen werden durch Stufen mit mehreren Eingangswarteschlangen nach Bild 5.16b) ersetzt, solange die Bedienungszeiten nicht von der Klasse der Anforderungen abhängen. In diesem Falle können sogar beliebige nichtunterbrechende Prioritäten zwischen den einzelnen Klassen zugelassen werden. Das Klassenkonzept erlaubt u.a. auch klassenindividuelle Verzweigungsmatrizen zur Beschreibung der einzelnen klassenabhängigen Verkehrsströme.

b3) Teilnetze

Im Zusammenhang mit der äquivalenten Ersetzung von stufeninternen Rückkopplungen wurde bereits darauf hingewiesen, Netzteile mit starken "inneren Bindungen" nicht zu zerlegen, sondern getrennt zu untersuchen und nur bezüglich ihres Ein/Ausgangsverhaltens in das Analyseverfahren einzubeziehen, vergl. Bild 5.16c). Teilnetze mit starken inneren Bindungen treten z.B. auf bei der Modellbildung von Rechnernetzen, welche sich logisch wie auch verkehrsmäßig in hierarchische Ebenen unterschiedlichen Grades hinsichtlich der Wechselwirkungen zwischen diesen Ebenen unterteilen lassen.

b4) Getaktete Schalter und Verzögerungsglieder

In allen vorausgegangenen Beispielen wurde der Verkehrsfluß zwischen verschiedenen Teilen des Netzes nicht gehemmt; eine Anforderung wechselte bei Verlassen der Stufe i unmittelbar und ohne Zeitverlust zur nächsten Stufe über. Das vorliegende Zerlegungsverfahren erlaubt auf einfache Weise den Einschluß von Verzögerungsgliedern sowie getakteten Schaltern.

Verzögerungsglieder, welche z.B. die Laufzeit einer Nachricht auf einer Leitung nachbilden, haben keinen Einfluß auf die Charakteristik der Verkehre; sie werden lediglich bei der Durchlaufzeitberechnung berücksichtigt. Getaktete Schalter treten auf bei der periodischen Ein/Ausgabe von Meldungen bzw. Befehlen von Steuerrechnern. Während der Phase eines geöffneten Schalters werden eintreffende Anforderungen in einem Pufferspeicher zwischengespeichert; zum Zeitpunkt des Schließens werden die zwischengespeicherten Anforderungen gleichzeitig in die nächste Stufe übertragen, vergl. Bild 5.16d). Bei bekanntem Ankunftsprozeß und bekannter Taktzeit kann die Verteilung der Gruppengröße bestimmt werden, welche auf die nachfolgende Stufe trifft. Zur Anwendung in dem oben beschriebenen Netzanalyseverfahren gehen ferner die mittlere Wartezeit und die Ausgangsprozeßcharakteristik der nachfolgenden Stufe (getaktete Gruppenankunft/G/1) ein. Eine Anwendung dieser Erweiterung mit Einschluß von Wartesystemen mit zyklischer Abfertigung und Prioritäten wird in Abschnitt 6.5 behandelt.

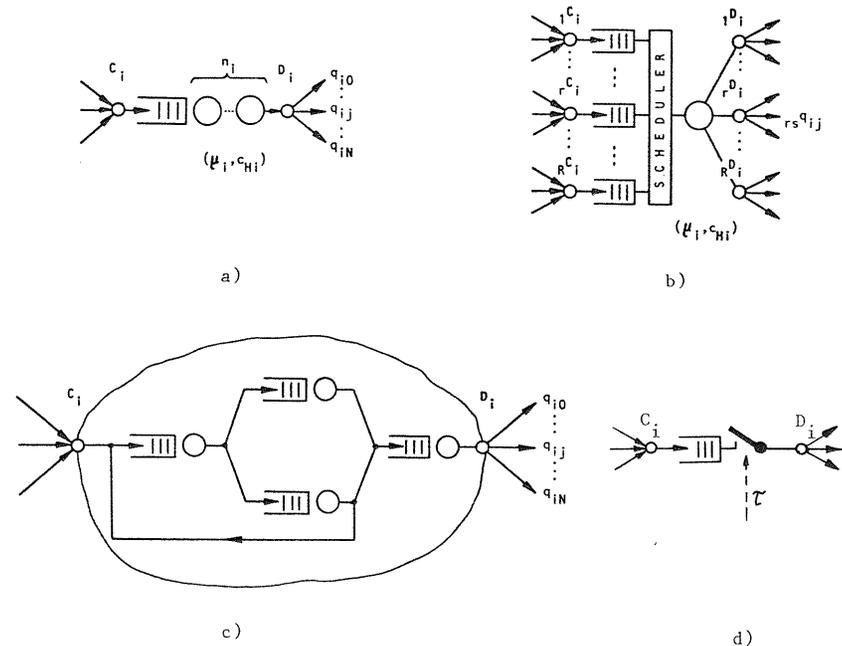


Bild 5.16. Netzelemente des verallgemeinerten Analyseverfahrens

- a) Bedienstufe /G/n
- b) Bedienstufe /G/1 mit mehreren Klassen von Anforderungen
- c) Teilnetze
- d) Getaktete Schalter

6. AUSGEWÄHLTE BEISPIELE ZUR SYSTEMANALYSE UND OPTIMIERUNG

In diesem Kapitel wird über die Anwendung verschiedener Verfahren zur Systemanalyse und Optimierung von rechnergesteuerten Systemen berichtet. In einem ersten Abschnitt wird ein kurzer Überblick über die wichtigsten Modelle und Ergebnisse der Analyse von Rechnersystemen und rechnergesteuerten Vermittlungssystemen gegeben. In den folgenden Abschnitten werden exakte und approximative Verfahren exemplarisch angewandt zur Bestimmung der Leistungsfähigkeit von rechnergesteuerten Systemen.

6.1 Überblick über Modelle und Ergebnisse

In den vergangenen Jahren ist eine Vielzahl von Modellen vorgeschlagen und untersucht worden. Eine umfassende Klassifizierung würde über den Rahmen dieser Arbeit hinausgehen. Ausführlichere Darstellungen wurden wiederholt unternommen und können dem Schrifttum entnommen werden, vergl. hierzu z.B. [18, 40, 43, 122-124]. In diesem Rahmen sollen nur einige wesentliche Klassifizierungsmerkmale hervorgehoben werden mit entsprechenden Hinweisen auf die Quell-Literatur oder auf andere zusammenfassende Darstellungen.

6.1.1 Prozessormodelle

Unter "Prozessormodellen" sollen alle jene Modelle verstanden werden, welche im wesentlichen das Betriebsmittel "Prozessor" beschreiben. Sie können in zwei Kategorien unterteilt werden:

- Prozessor-interne Modelle
- Prozessor-Zuteilungsmodelle.

Prozessor-interne Modelle beschreiben das dynamische Ablaufgeschehen zwischen den Baugruppen innerhalb von Prozessoren wie Rechenwerken, Befehlswerken und lokalen Pufferspeichern. Typische "Anforderungen" für diese Betriebsmittel sind Befehle oder daraus abgeleitete Teil-Operationen. Verkehrsmodelle zur Analyse der Leistungsfähigkeit beschreiben Einzeloperationen wie "Befehl-Holen", "Adreßrechnung", "Operand-Holen" und "Rechenwerksoperation" durch mehrstufige Warteschlangenarrangements [125] bzw. die Wirksamkeit von lokalen Pufferspeichern mit vorausschauender Befehls- und Datenversorgung [149].

Prozessor-Zuteilungsmodelle haben bisher die größte Beachtung erfahren. Sie beschreiben die Zuteilung des Betriebsmittels "Prozessor" an anfordernde Quellen nach fest vorgegebenen (statischen) Strategien oder nach dynamischen Strategien:

Modelle mit statischen Zuteilungs-/Abfertigungsstrategien

- Reihenfolge nach Ankunftszeitpunkten (FIFO,RANDOM,LIFO) [126-128,52,55,65]u.a.m.
- Nichtunterbrechende Prioritäten [43, 53, 87, 129]
- Unterbrechende Prioritäten [129-131]
- Kombination aus unterbrechenden und nichtunterbrechenden Prioritäten [20, 54, 132, 133]
- Unterbrechungs-Verzögerungs-Prioritäten [134, 122, 135]
- Zyklische Abfertigung bzw. zyklische Prioritäten [136-140]
- Getaktete Ankünfte bzw. getaktete Abfertigung [141-145].

Modelle mit dynamischen Zuteilungs-/Abfertigungsstrategien

- Systemzustandsabhängige Prioritäten [45, 146-147]
- Fertigstellungstermin-abhängige Prioritäten [147-148]
- Bedienungszeitabhängige Prioritäten [40, 43, 122, 147]
 - z.B. nach kürzester Gesamtbedienungszeit
 - nach kürzester Restbedienungszeit
 - nach bereits verbrauchter Bedienungszeit.

6.1.2 Speicherzugriffsmodelle

Speicherzugriffsmodelle beschreiben das Ablaufgeschehen beim Zugriff mehrerer Anforderungen auf physikalische Speicher wie

- Zugriff auf modulare Hauptspeicherblöcke
- Zugriff auf rotierende Speichermedien (Trommel, Platte, Band).

Derartige Modelle sind i.a. gekennzeichnet durch

- Zugriffsmechanismus (z.B.mechanische Positionierung von Schreib/Leseköpfen)
- Transportkanal zur Übertragung der Informationen
- Zugriffsstrategie (z.B. FIFO, SCAN, etc.).

Aus der umfangreichen Literatur sei auf [40, 149-154] verwiesen.

6.1.3 Gesamtmodelle

Unter Gesamtmodellen sollen jene Modelle verstanden werden, welche die Wechselwirkung zwischen verschiedenen Betriebsmitteln beschreiben wie Prozessoren, Speichern, Ein/Ausgabekanälen und Bedienungsstationen. Gesamtmodelle für Rechnersysteme sind i.a. gekennzeichnet durch eine Warteschlangennetzstruktur ("Central Server Model"), ferner durch die Annahme eines Modelles zur Beschreibung des Auftragsprofils sowie durch Wechselwirkungen wie zwischen Anwenderprogrammen und Systemprogrammen bzw. zwischen Prozessor-Belegungsphasen und Speicherorganisation

[18, 40, 43, 122, 155-171]. Gesamtmodelle für rechnergesteuerte Vermittlungssysteme enthalten im wesentlichen die zentrale Steuerung, die Ein/Ausgabe, das Signalsammel- und Verteilsystem; das Ablaufgeschehen ist ferner beeinflusst von der Rückwirkung des Nutzwegenetzes (Blockierungszustände des Koppelnetzes u.ä.) sowie dem Teilnehmerverhalten [110-111, 141, 144-146, 172-173].

6.1.4 Standardisierte Modelle, Tabellen

Die wirklichkeitsgetreue Modellierung des Ablaufgeschehens führt in den meisten Fällen auf spezielle Modelle, welche durch die Gegebenheiten der jeweiligen Anwendung gekennzeichnet sind. Teilmodelle oder vereinfachte Versionen führen jedoch oft auf einstufige Grundmodelle, welche einen breiten Anwendungskreis besitzen. Typische Standardmodelle sind (siehe Kurznotation aus Abschnitt 3.4.1):

Wartesysteme	M/G/1, GI/M/1, GI/G/1
	CB /G/1, M/C/n
	M/M/n, M/D/n, GI/M/n
	M/M/n mit endlicher Quellenzahl
	M/M/n(k), M/D/n(k)
Warteverlustsysteme	M/G/1-s, M/M/n-s, M/C/n-s.

Die Analyse derartiger Modelle erfordert ein weites Spektrum an exakten wie auch approximativen Methoden, welche dem Anwender in der Regel nicht bekannt sind.

Für die bekanntesten Standardmodelle von Verlust- und Wartesystemen des Typs M/M/n mit unendlicher/endlicher Quellenzahl sowie Wartesystemen des Typs M/D/1 und M/D/n stehen schon seit längerem Tabellen zur Verfügung (Übersicht siehe [174]). Um dem Anwender darüberhinaus eine größere Auswahl von Tabellen zu geben, wurde ein neues Tabellenwerk entwickelt, welches mehr als 25 verschiedene Verkehrsmodelle umfaßt und die wichtigsten Leistungsgrößen in Abhängigkeit von Angebot, System- und Betriebsparametern angibt [174, 175].

Netzmodelle sind infolge der Vielzahl von Parametern i.a. nicht geeignet für die Tabellierung. Hier liegt der Schwerpunkt auf der Entwicklung von allgemein anwendbaren Prozeduren. Beispiele hierfür sind die numerischen Rekursionsalgorithmen für Warteschlangennetze mit Produktlösungsform [92-93, 109] oder allgemein anwendbare Prozeduren für Näherungslösungen [118-120].

6.2 Durchsatzoptimierung in Stapelverarbeitungssystemen mit Virtuellem Speicher

6.2.1 Problemstellung

Mittlere und größere Rechnersysteme für den kommerziellen wie auch technisch-wissenschaftlichen Aufgabenbereich sind durch den Aufbau einer Speicherhierarchie gekennzeichnet. In der untersten Ebene (Prozessorebene) werden schnelle Speicher eingesetzt, welche an die Verarbeitungsgeschwindigkeit der Prozessoren angepaßt, in ihrer Speicherkapazität jedoch begrenzt sind. Programme und Daten befinden sich in darüberliegenden Ebenen und werden stückweise zur Verarbeitung in die unterste Speicherebene befördert. Diese Konfiguration verbindet kleine Zugriffs- und Verarbeitungszeiten mit großer Speicherkapazität; dieser Vorteil wird jedoch erkauft mit dem Preis eines umfangreichen Datenaustausches zwischen den Ebenen der Speicherhierarchie, welcher unter Umständen zu Engpässen und damit zu einem drastischen Leistungsabfall führen kann.

In dem folgenden Beispiel wird ein Stapelverarbeitungssystem betrachtet, welches nach dem Virtuellen Speicherprinzip mit Demand-Paging organisiert ist. Die Speicherhierarchie besteht aus 2 Ebenen, dem Arbeitsspeicher und einem Hintergrundspeicher mit Direktzugriff (Platten- oder Trommelspeicher). Es wird der Einfluß von verschiedenen Systemparametern wie Programmeigenschaften, Systemverwaltungszeit, Multiprogrammgrad und E/A-Häufigkeit auf den Durchsatz und andere Größen untersucht. Aus den Ergebnissen, welche unter gewissen idealisierenden Annahmen gefunden werden, wird ein Kriterium zur Durchsatz-optimalen Steuerung des Multiprogrammgrades gefunden.

Das angeführte Problem ist in der Literatur wiederholt untersucht worden unter verschiedenen Voraussetzungen, vergl. [155-170]. Das nachfolgend beschriebene Verfahren basiert auf [168] und unterscheidet sich von anderen Verfahren in der Einbeziehung von

- Systemverwaltungszeiten
- nichtexponentiell verteilten Prozessor- und Speicherzugriffszeiten
- Daten-Ein/Ausgabe (File-I/O).

Die genauere Charakterisierung der Bedienungsprozesse führt auf ein allgemeineres Warteschlangennetz, welches nicht mehr der Produktlösungsform genügt; die Anwendung der Phasenmethode erlaubt dennoch eine exakte Lösung.

6.2.2 Voraussetzungen

a) Rechnerstruktur

Die Grundkonfiguration des Rechnersystems umfaßt einen Rechnerkern (CPU) und eine zweistufige Speicherhierarchie (ASP, HSP), vergl. Bild 6.1.

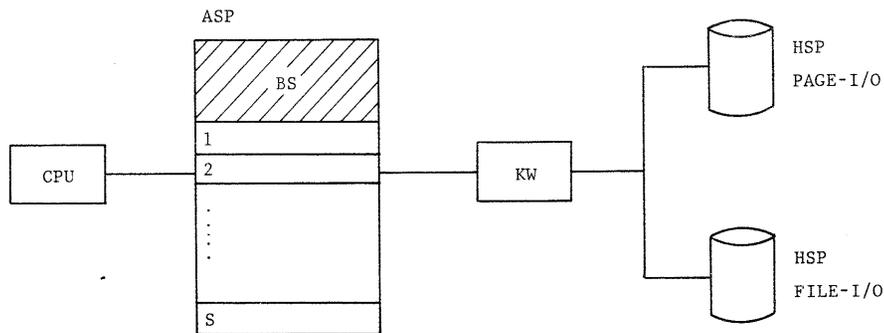


Bild 6.1. Systemkonfiguration eines Rechnersystems mit zweistufiger Speicherhierarchie

CPU	Rechnerkern (Prozessor)	KW	Kanalwerk
ASP	Arbeitsspeicher	BS	Betriebssystem
HSP	Hintergrundspeicher	S	Anzahl der Seitenrahmen für Benutzerprogramme

Der Arbeitsspeicher enthält neben den residenten Systemprogrammteilen (BS) S gleichgroße Seitenrahmen für Benutzerprogramme. Die Hintergrundspeicher sind getrennt angenommen für Seitentransporte (Virtueller Speicher, Page-I/O) und Ein/Ausgabe von Massendaten (File-I/O). Der Datenaustausch mit der Zentraleinheit erfolgt über ein Kanalwerk KW (Schnellkanäle SK und E/A-Prozessor).

b) Betriebsstrategien

Die Betriebsstrategien sind im wesentlichen gekennzeichnet durch

- Multiprogrammingsgrad
- Speicherplatzverwaltung
- Seitenholstrategie
- Seiteneretzungsstrategie.

b1) Multiprogrammingsgrad

Der Multiprogrammingsgrad bezeichnet die Anzahl gleichzeitig in den ASP geladener Benutzerprogramme. Es wird bei den Untersuchungen von einem beliebigen, doch jeweils konstanten Multiprogrammingsgrad M ausgegangen.

b2) Speicherplatzverwaltung (memory management policy)

Die Platzaufteilung des Arbeitsspeichers unter die am Multiprogramm beteiligten Programme sei fest (fixed partitioning), d.h. jedes aktivierte Programm erhält einen festen Teil des Arbeitsspeichers für sich. Darüberhinaus wird vereinfachend angenommen, daß alle Benutzerprogramme denselben ASP-Anteil S/M erhalten (balanced partitioning).

b3) Seitenholstrategie (page fetch strategy)

Als Seitenholstrategie wird demand paging vorausgesetzt, d.h. eine Seitennachladung wird nur nach dem Auftreten eines page faults initiiert (Page-I/O). Ferner wird angenommen, daß die Benutzerprogramme vor Beginn und nach Beendigung ihrer Bearbeitung auf dem HSP abgelegt sind. Die Ein/Ausgabe von Massendaten (File-I/O) ist durch die Benutzerprogramme selbst festgelegt. Es wird angenommen, daß diejenigen Datenseiten, welche an einem File-I/O beteiligt sind und sich zum Zeitpunkt des I/O-Interrupts nicht im ASP befinden, vorher zusätzliche "page-faults" auslösen, bevor die eigentliche Ein/Ausgabe gestartet werden kann.

b4) Seiteneretzungsstrategie (page replacement strategy)

Die Seiteneretzungsstrategie bestimmt diejenige Seite, welche bei Seitennachladung aus dem ASP verdrängt wird, falls kein freier Seitenrahmen verfügbar ist. Es werden lokal wirkende Strategien zugrundegelegt, welche sich nur auf Seiten des betreffenden Benutzerprogrammes beziehen. Ferner wird vorausgesetzt, daß für die Untersuchungen nur Eretzungsstrategien für feste Speicherplatzaufteilung zugelassen werden sollen wie z.B. LRU (least recently used: es wird diejenige Seite verdrängt, deren letzter Zugriff am weitesten zurück liegt), FIFO (first-in, first-out: es wird diejenige Seite verdrängt, welche zuerst in den ASP geladen wurde) oder RANDOM (es wird eine zufällig bestimmte Seite verdrängt).

c) Programmeigenschaften (Auftragsprofil)

Es wird vereinfachend angenommen, daß sämtliche Benutzerprogramme - statistisch gesehen - gleichartiges Verhalten aufweisen. Dieses Verhalten ist im wesentlichen charakterisierbar durch

- Größe der Benutzerprogramme
- Rechenzeitbedarf
- Lokalitätseigenschaften der Seitenreferenzen
- Ein/Ausgabeverhalten.

c1) Größe der Benutzerprogramme

Benutzerprogramme haben i. a. eine schwankende Anzahl von Seiten, welche jeweils durch die Anzahl der Programmbefehle, die Datenfelder sowie die von den Systemprogrammen eingebundenen Hilfsprogramme (z.B. für Ein/Ausgabe) festgelegt ist. Es wird vereinfachend angenommen, daß alle Benutzerprogramme gleich groß sind und L Seiten besitzen (konstante Programmgröße).

c2) Rechenzeitbedarf

Unter dem Rechenzeitbedarf eines Benutzerprogrammes soll die Gesamt-CPU-Zeit verstanden sein, welche nötig ist zur kompletten Bearbeitung des Programmes ohne Einschluß der CPU-Systemverwaltungszeiten. Der Rechenzeitbedarf läßt sich durch eine Wahrscheinlichkeits-VF beschreiben. Infolge der speziellen Annahmen (konstanter Multiprogrammgrad und gleiche statistische Eigenschaften der Benutzerprogramme) genügt im wesentlichen die Festlegung des mittleren Gesamt-Rechenzeitbedarfs h_{GR} .

Die Gesamt-Rechenzeit setzt sich aus einzelnen CPU-Phasen zusammen entsprechend der Häufigkeit von page fault-Unterbrechungen bzw. Ein/Ausgabe-Unterbrechungen. Messungen an realen Systemen haben eine starke Schwankung dieser CPU-Phasen gezeigt. Diesem Effekt wird durch die Annahme hyperexponentiell verteilter CPU-Phasen Rechnung getragen; die CPU-Rechenphasen werden beschrieben durch ihren Mittelwert h_{1R} und den Variationskoeffizienten c_{1R} . Der Mittelwert h_{1R} ist festgelegt durch die mittlere Anzahl von Seitenreferenzen zwischen zwei Unterbrechungen (Page- bzw. File-I/O) sowie die mittlere CPU-Rechenzeit h_{Ref} pro Seitenreferenz. Der Variationskoeffizient c_{1R} ist eine vorgebbare Größe.

c3) Lokalitätseigenschaften der Seitenreferenzen

Benutzerprogramme weisen eine mehr oder weniger stark ausgeprägte "Lokalität" bezüglich der zeitlichen Häufung ihrer Zugriffe zu einzelnen Seiten auf. Dieses Lokalitätsverhalten kann meßtechnisch ermittelt werden in Form spezieller Charakteristiken [40, 176-179]:

- Working Set-Charakteristik

Relative Häufigkeit der Zugriffe auf Seiten außerhalb des momentanen Working Set in Abhängigkeit einer "Fensterbreite" τ . Der Working Set ist die Menge derjenigen Seiten, auf welche während der letzten τ Seitenreferenzen zugegriffen wurde.

- Fehlseiten-Charakteristik (page fault rate function)

Relative Häufigkeit f_p von page faults in Abhängigkeit des relativen Anteils x von ASP-Seiten eines Programmes bezogen auf dessen Gesamt-Programmgröße L sowie der Seiteneretzungsstrategie.

Für die Untersuchungen werden gemessene Fehlseiten-Charakteristiken zugrundegelegt, welche für verschiedene Programmtypen relativ gut bekannt sind. Bild 6.2 zeigt sieben Beispiele für verschiedene technisch-wissenschaftliche und kommerzielle Programmtypen [176-178]. Zur Untersuchung wird vereinfachend angenommen, daß alle Benutzerprogramme jeweils dieselbe Fehlseiten-Charakteristik besitzen.

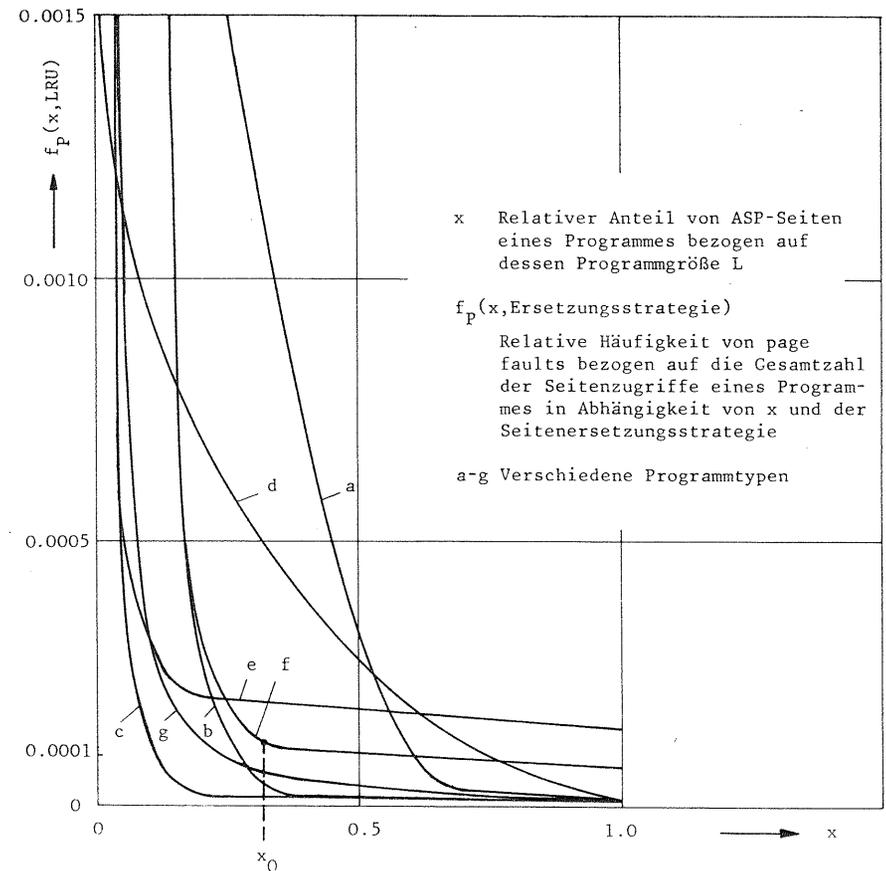


Bild 6.2. Fehlseiten-Charakteristiken verschiedener Programmtypen

c4) Ein/Ausgabeverhalten

Unter dem Ein/Ausgabeverhalten sollen die statistischen Eigenschaften der File-I/O Vorgänge verstanden werden. Im allgemeinen kann das Ein/Ausgabeverhalten charakterisiert werden durch

- die absolute Häufigkeit der File-I/O Unterbrechungen
- ein Maß für die Abstandsschwankungen (z.B. Variationskoeffizient)
- ein Maß für den Umfang der ein- bzw. ausgegebenen Daten (Cluster).

Im Rahmen der analytischen Modelluntersuchungen nehmen wir eine gleichwahrscheinliche Verteilung von insgesamt n_p File-I/O Unterbrechungen über der CPU-Gesamt-rechenzeit an (die zusammenhängend betrachtete CPU-Zeit eines Programmes wird oft auch als "virtuelle Zeit" bezeichnet, um den Unterschied zwischen dem Programmzustand hinsichtlich der Abarbeitung und der unabhängigen Systemzeit auszudrücken).

6.2.3 Systemmodell

Das Systemmodell besteht aus folgenden drei Teilmodellen

- Warteschlangenmodell
- Speichermodell
- Benutzerprogramm-Modell.

Die Teilmodelle beschreiben sehr unterschiedliche Aspekte des Rechnersystems. Wesentlich ist der parametrische Zusammenhang zwischen diesen Teilmodellen, welcher die drei Teilmodelle zu einem Systemmodell vereinigt.

a) Warteschlangenmodell

Unter Zugrundelegung der in Abschnitt 3.3.2 getroffenen Vereinfachungen kann die logische Zustandsfolge der Benutzerprogramme in einem dreistufigen Warteschlangennetz-Modell beschrieben werden, vergl. Bild 6.3.

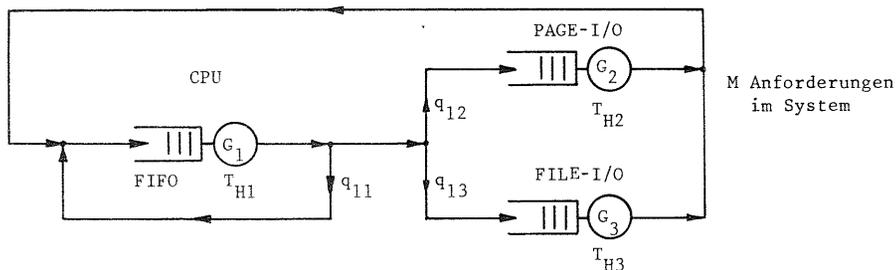


Bild 6.3. Warteschlangenmodell

Die drei Bedienungsstationen beschreiben die CPU-Rechenphasen bzw. CPU-Systemverwaltungsphasen, den Seitentransfer zwischen ASP und HSP (Page-I/O) sowie die Ein/Ausgabe von Massendaten (File-I/O). In letzteren Fällen beschreibt die "Bedienungszeit" näherungsweise die Zugriffs- und Übertragungszeit; eine genaue Modellierung des I/O-Systems führt auf ein zweistufiges Teilmodell, dessen Details in diesem Zusammenhang hier nicht weiter verfolgt werden sollen.

Das Modell nach Bild 6.3 ist ein geschlossenes Warteschlangenmodell, welches jeweils unter der Annahme von M Benutzerprogrammen (M = Multiprogrammgrad) untersucht wird. Die Rückkopplungsschleife in Stufe 1 charakterisiert das Bearbeitungsende eines Benutzerprogrammes, welches augenblicklich ein neues, unbearbeitetes Benutzerprogramm startet; dies entspricht i.a. dem Fall von Stapelverarbeitungssystemen mit größerer Auftragslast (z.B. Nachtschichtbetrieb). (Je nach Interpretation der Ablauffolge kann die Rückkopplungsschleife auch in Stufe 2 angenommen werden). Die Verzweigungswahrscheinlichkeiten q_{11} , q_{12} bzw. q_{13} drücken die rel. Häufigkeit eines Überganges nach

- Stufe 1 (Programmende, Initiierung eines neuen Benutzerprogrammes),
- Stufe 2 (Page-I/O) bzw.
- Stufe 3 (File-I/O)

bei Beendigung einer CPU-Belegungsphase aus.

Die VF der zufallsabhängigen Bedienungszeiten T_{H1} bzw. T_{H2} für die CPU-Belegungsphasen bzw. für Page-I/O sind in Bild 6.4a,b skizziert.

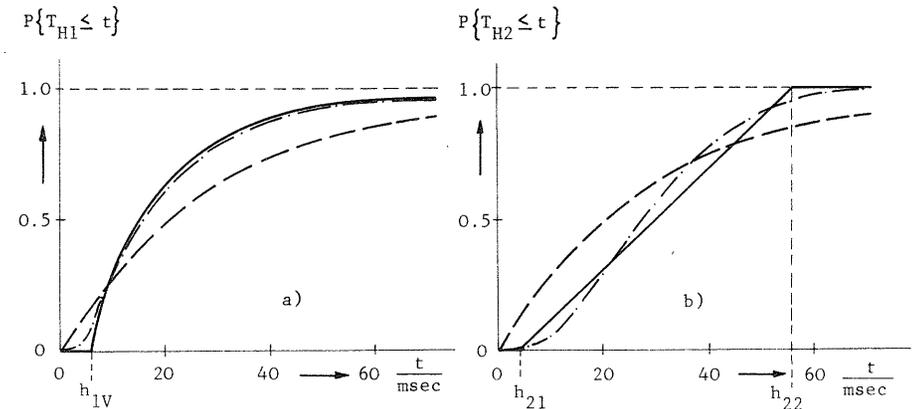


Bild 6.4. Bedienungszeit-Verteilungsfunktionen
 a) CPU b) Page-I/O (Beispiel: Trommelzugriff)
 --- exponentielle VF -.-.- Phasenapproximation

Die CPU-Belegungsphasen (Bild 6.4a) setzen sich aus einem konstanten Anteil für das Betriebssystem (Systemverwaltungszeit h_{1V}) und einem stark streuenden Anteil für die Rechenzeitphasen zusammen. Eine einfache Phasenapproximation hierfür ist in Bild 6.5a gezeigt. Die VF nach Bild 6.4a wird durch eine Serienschaltung einer H_2 -VF (Rechenphase) mit einer exponentiellen VF (Verwaltungsphase) approximiert. Die Güte der Approximation ist in Bild 6.4a angegeben für den Fall $h_{1V} = 5$ msec, $h_{1R} = 25$ msec, $h_1 = 30$ msec, $c_{1R} = \sqrt{10}$. Der Verlauf der exponentiellen VF mit gleichem Mittelwert $h_1 = 30$ msec ist ebenfalls zum Vergleich eingezeichnet.

Die Page-I/O- bzw. File-I/O-Bedienungszeiten (Beispiel Bild 6.4b für Trommelzugriffe) setzen sich zusammen aus der konstanten Übertragungszeit h_{21} sowie einem linearen Anteil, welcher die Rotations-Latenzzeit des Trommelspeichers beschreibt (ein ähnlicher Verlauf, jedoch mit unterschiedlichen Parametern, läßt sich im Falle eines Plattenzugriffes angeben). Die VF nach Bild 6.4b wird durch eine E_k -VF derart approximiert, daß die ersten beiden Momente übereinstimmen. Beispiele von realen Zugriffszeiten ergaben $k = 3...4$ (Trommel) und $k = 5...6$ (Platte). Die Phasenapproximation ist in Bild 6.5b skizziert. In Bild 6.4b ist die Güte der Approximation angegeben für den Fall $h_{21} = 4$ msec, $h_{22} = 56$ msec, $h_2 = 30$ msec ($c_2 = 0.5$). Die entsprechende exponentielle VF ist zum Vergleich ebenfalls eingezeichnet.

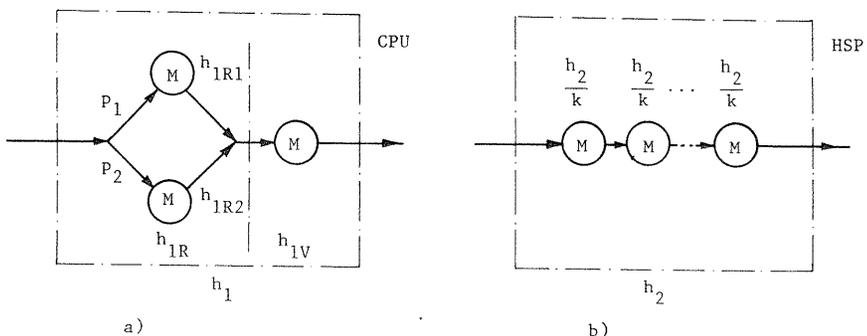


Bild 6.5. Phasendarstellungen für CPU-Belegungsphasen und HSP-Zugriffe
 a) CPU-Belegungsphasen b) HSP-Zugriffe.

Es sei bemerkt, daß sich die Güte der Approximation durch die Berücksichtigung weiterer höherer Momente oder spezieller Stützstellen beliebig genau verfeinern läßt, vgl. hierzu [47]. Die Approximation benötigt dann allerdings i.a. wesentlich mehr Phasen, wodurch die numerische Auswertung der Netz-Zustandsgleichungen erschwert wird. Von der praktischen Anwendung her gesehen, ergeben

bereits wenige Phasen eine gute Genauigkeit, wenn es gelingt, hiermit das Wesentliche der approximierten VF (z.B. 2 Momente oder eine spezielle Charakteristik) wiederzugeben.

b) Speichermodell

Entsprechend der getroffenen Voraussetzungen stehen jedem in den ASP geladenen Benutzerprogramm S/M Seitenrahmen zur Verfügung. Der relative Anteil von in den ASP geladenen Seiten eines Benutzerprogrammes ist demnach

$$x = \frac{S}{ML} \quad (6.1)$$

Die Größe x bestimmt die Fehlseiten-Häufigkeit und damit die mittleren CPU-Rechenphasen des Warteschlangenmodelles. Die ASP-Seiten eines jeden Benutzerprogrammes seien nach der LRU-Strategie geordnet. Alle restlichen Programm- und Datenseiten befinden sich auf dem Hintergrundspeicher. Bei Seitenersetzung wird stets diejenige ASP-Seite verdrängt, deren letzter Zugriff am weitesten zurückliegt (LRU).

In realen Systemen wird für jede neu in den ASP geladene Seite ein Kennbit (change bit) geführt, welches angibt, ob die Seite seit dem Ladevorgang verändert wurde oder nicht (vergl. Abschnitt 2.3.4). Bei Seitenersetzung muß eine veränderte Seite erst auf den HSP geschrieben werden, bevor die neue Seite geladen werden kann, wohingegen eine nichtveränderte Seite einfach überschrieben werden kann. Dementsprechend sind entweder zwei Zugriffe oder nur ein Zugriff auf den HSP erforderlich. Diese Eigenschaft läßt sich in einem modifizierten Modell für die Bedienungszeiten des Kanals für Page-I/O ausdrücken.

c) Benutzerprogramm-Modell

Benutzerprogramme lassen sich allgemein kennzeichnen durch ihre Befehlsfolge. In der betrachteten Modellierungsebene wird anstelle der Befehlsfolge die Folge der referierten Seitennummern verwendet ("Seitenzugriffskette" ω). Um das Ein/Ausgabeverhalten auf einfache Weise einzubeziehen, schließen wir die File-I/O Befehle in die Seitenzugriffskette mit ein. Die gesamte Anzahl von Seitenzugriffen innerhalb von ω sei R .

Beispiel: $\omega = \{1,9,2,9,1,7,3,9, \text{FILE-I/O}, 9,2,7,8,6,8,5, \text{FILE-I/O}, 7,8,\dots\}$.

Die absolute page fault-Häufigkeit eines Benutzerprogrammes mit einem ASP-Ladeteil x und der Ersetzungsstrategie ES sei $n_p = n_p(x, ES)$. Entsprechend sei die absolute File-I/O-Häufigkeit n_F . Die n_p page faults unterteilen die Seitenzugriffskette in n_p Teile (es wird angenommen, daß ein Programm mit einem page

fault beginnt) oder m.a.W., die mittlere Anzahl von Seitenzugriffen zwischen zwei page faults ist R/n_p . Neben den absoluten Häufigkeiten n_p und n_f werden weiterhin die relativen Häufigkeiten verwendet:

$$f_p = f_p(x, ES) = n_p/R \quad (6.2a)$$

$$f_f = n_f/R. \quad (6.2b)$$

f_p wird dabei durch die vorgegebene Fehlseitencharakteristik, den Ladeanteil x und die Ersetzungsstrategie ES festgelegt (vergl. Bild 6.2). Die Streuung der Anzahl von Seitenzugriffen zwischen zwei Unterbrechungen (Page-I/O bzw. File-I/O) wird durch den Variationskoeffizienten c_{1R} erfaßt.

d) Festlegung der Parameter des Systemmodelles

Im folgenden werden die Parameter des Systemmodelles im einzelnen festgelegt aus Vorgaben über Systemeigenschaften und Programmeigenschaften.

Festzulegende Parameter:

h_{1R}	Mittlere CPU-Rechenphasen für Benutzerprogramme
$\left. \begin{matrix} h_{1R1}, h_{1R2} \\ p_1, p_2 \end{matrix} \right\}$	Parameter der Phasenapproximation der CPU-Rechenphasen
h_{1V}	Mittlere CPU-Verwaltungsphase
h_2, h_3	Mittlere Zugriffszeiten auf HSP
k	Phasenzahl der E_k -VF
q_{11}, q_{12}, q_{13}	Verzweigungswahrscheinlichkeiten
R	Länge der Seitenzugriffskette
f_f	Rel. Ein/Ausgabe-Häufigkeit (File-I/O)
x	Ladeanteil

Vorgaben:

M	Multiprogrammingsgrad
S	Anzahl von ASP-Seitenrahmen
L	Programmlänge in Seiten
h_{GR}	Gesamt-Rechenzeitbedarf eines Benutzerprogrammes
h_{Ref}	CPU-Rechenzeit/Seitenzugriff
c_{1R}	Variationskoeffizient der CPU-Rechenphasen
$f_p(x, ES)$	Fehlseiten-Charakteristik
n_f	Anzahl von File-I/O Vorgängen eines Benutzerprogrammes

Die Parameter werden wie folgt festgelegt:

- 1) Mit der Bilanz $h_{GR} = R \cdot h_{Ref}$ liegt R fest.
- 2) Mit n_f und R liegt f_f nach Gl. (6.2b) fest.
- 3) Mit M, S, L wird x nach Gl. (6.1) festgelegt. Mit x und der Ersetzungsstrategie ES ist gleichzeitig f_p bestimmt.
- 4) Mit f_p, f_f und h_{GR} folgt für die mittlere CPU-Rechenphase

$$h_{1R} = \frac{h_{GR}}{n_p + n_f} = \frac{h_{GR}}{R \cdot (f_p + f_f)} \quad (6.3)$$

- 5) Mit h_{1R} und c_{1R} und der Symmetrieannahme $p_1 h_{1R1} = p_2 h_{1R2}$ können die Parameter der Phasenapproximation für die CPU-Rechenphasen bestimmt werden:

$$h_{1R1,2} = \frac{h_{1R}}{1 \pm \sqrt{1 - \frac{2}{1 + c_{1R}^2}}}, \quad (c_{1R} \geq 1) \quad (6.4a)$$

$$p_{1,2} = \frac{h_{1R}}{2h_{1R1,2}} \quad (6.4b)$$

- 6) Infolge der Annahme unabhängiger Verzweigungswahrscheinlichkeiten in Bild 6.3 durchläuft ein Benutzerprogramm die CPU-Stufe genau n mal mit Wahrscheinlichkeit $(1 - q_{11})^{n-1} \cdot q_{11}$. Mit der mittleren Anzahl $1/q_{11}$ der CPU-Durchläufe (von denen nur $\frac{1}{q_{11}} - 1$ eigentliche Rechenphasen sind), dem vorgegebenen Gesamt-Rechenzeitbedarf h_{GR} und der mittleren CPU-Rechenphase h_{1R} nach Gl. (6.3) kann eine Rechenzeitbilanz aufgestellt werden, aus welcher q_{11} bestimmt wird:

$$h_{GR} = \left(\frac{1}{q_{11}} - 1\right) h_{1R} \quad (6.5a)$$

oder

$$q_{11} = \frac{h_{1R}}{h_{GR} + h_{1R}} = \frac{1}{R(f_p + f_f) + 1} = \frac{1}{n_p + n_f + 1}$$

Von den $n_p + n_f + 1$ CPU-Durchläufen eines Benutzerprogrammes entfallen n_p auf Page-I/O und n_f auf File-I/O; dementsprechend folgt für q_{12} und q_{13}

$$q_{12} = \frac{n_p}{n_p + n_f + 1} = \frac{R f_p}{R(f_p + f_f) + 1} \quad (6.5b)$$

$$q_{13} = \frac{n_f}{n_p + n_f + 1} = \frac{R f_f}{R(f_p + f_f) + 1} \quad (6.5c)$$

7) die restlichen Parameter h_{1V}, h_2, h_3 liegen durch das Rechnersystem fest. Für den Parameter der E_k -Approximation für die HSP-Zugriffe wird $k = 3$ angenommen (Trommel- oder Festkopflattensysteme).

Durch die Festlegung der Parameter des Systemmodelles - welches die drei Teilmodelle miteinander verknüpft - sind auch die Parameter des Warteschlangenmodells festgelegt, auf welchem die Modellanalyse beruht.

6.2.4 Modellanalyse

a) Zustandsdefinition und Zustandsraum

Unter Zugrundelegung der Phasenmethode zur Darstellung der CPU- und HSP- Bedienzeiten kann das Ablaufgeschehen mit Hilfe eines mehrdimensionalen Markoffschen Zustandsprozesses beschrieben und analysiert werden. Der Systemzustand ist eindeutig gekennzeichnet durch das Tupel

$$(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}),$$

wobei

x_i Anzahl der Anforderungen (Programme) in Stufe $i, i = 1, 2, 3$

x_{iP} Phasenzustand der Bedienung in Stufe $i, i = 1, 2, 3$

$$x_{1P} = \begin{cases} 1 & \text{Benutzerprogrammphase (Mittelwert } h_{1R1}) \\ 2 & \text{Benutzerprogrammphase (Mittelwert } h_{1R2}) \\ 3 & \text{Systemprogrammphase (Mittelwert } h_{1V}) \end{cases}$$

$x_{iP} = 1, 2, \dots, k$ Phasenzustand des Kanals für Page-I/O ($i = 2$) bzw. File-I/O ($i = 3$).

Die Zustände $x_{iP} = 0$ kennzeichnen die Stufe i als "frei" und können nur auftreten, wenn gleichzeitig $x_i = 0$. Da das Warteschlangenmodell ein geschlossenes Netzmodell mit insgesamt M Anforderungen ist, gelten ferner die Einschränkungen für jeden Systemzustand

$$0 \leq x_i \leq M, \quad i = 1, 2, 3$$

$$\sum_{i=1}^3 x_i = M \quad (6.6)$$

Der Zustandsraum besteht aus allen logisch möglichen Variablenkombinationen. Für einen allgemeinen Fall ist der Ausschnitt des Zustandsraumes für den Zustand $(x_1, 1, x_2, x_{2P}, x_3, k)$, seine Nachbarzustände sowie die entsprechenden Übergänge in Bild 6.6 angegeben (Die Randzustände unterscheiden sich von dem allgemeinen Fall durch das Fehlen einiger Übergänge).

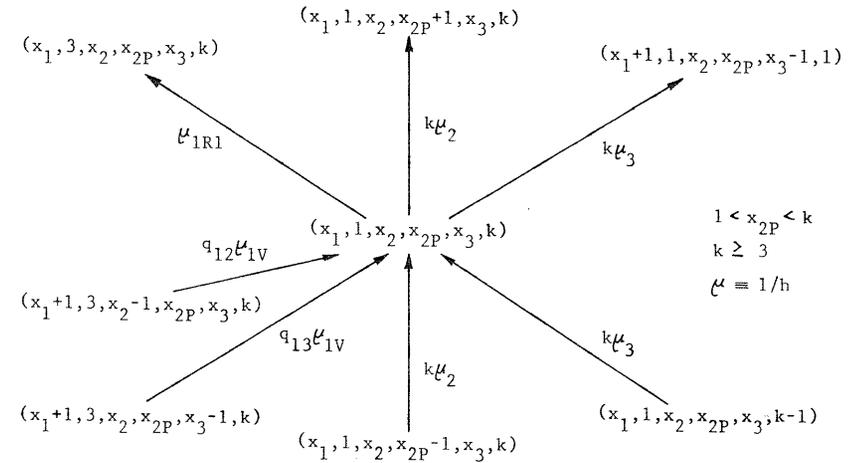


Bild 6.6. Ausschnitt aus dem Zustandsraum

Eine ausführliche Darstellung dieser Methode und dem Lösungsverfahren findet sich in [120].

b) Zustandsgleichungen und Zustandswahrscheinlichkeiten

Die Zustandsgleichungen für die stationären Zustandswahrscheinlichkeiten $p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P})$ erhält man durch Ansetzen der Gleichungen für das statistische Gleichgewicht eines jeden Zustands. Für das Beispiel des Zustands $(x_1, 1, x_2, x_{2P}, x_3, k)$ in Bild 6.6 lautet die Gleichung

$$(\mu_{121} + k\mu_2 + k\mu_3)p(x_1, 1, x_2, x_{2P}, x_3, k) = q_{12}\mu_{1V}p(x_1+1, 3, x_2-1, x_{2P}, x_3, k) + q_{13}\mu_{1V}p(x_1+1, 3, x_2, x_{2P}, x_3-1, k) + k\mu_2 p(x_1, 1, x_2, x_{2P}-1, x_3, k) + k\mu_3 p(x_1, 1, x_2, x_{2P}, x_3, k-1) \quad (6.7a)$$

Zusammen mit der Normierungsbedingung

$$\sum_{\text{alle Zustände}} p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}) = 1 \quad (6.7b)$$

liegen sämtliche Zustandswahrscheinlichkeiten fest. Die numerische Auswertung kann mit dem iterativen Verfahren der sukzessiven Überrelaxation [120] erfolgen.

c) Charakteristische Größen

Aus den Zustandswahrscheinlichkeiten werden charakteristische Größen ermittelt, welche die Leistung des Systems ausdrücken wie

- Y_{1R} CPU-Auslastung durch Benutzerprogramme
- Y_{1V} CPU-Auslastung durch Systemprogramme
- Y_2 Auslastung des Kanals für Page-I/O
- Y_3 Auslastung des Kanals für File-I/O
- n_i Mittlere Anzahl von Anforderungen in Stufe i , $i = 1, 2, 3$
- D Durchsatz (Anzahl fertig bearbeiteter Programme/Zeiteinheit)
- t_F Mittlere Durchlaufzeit eines Anwenderprogrammes.

Die Größen Y_i und n_i sind Erwartungswerte und können aus den Zustandswahrscheinlichkeiten gewonnen werden:

$$Y_{1R} = \sum_{\substack{\text{alle Zustände} \\ \text{mit } x_{1P} = 1, 2}} p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}) \quad (6.8a)$$

$$Y_{1V} = \sum_{\substack{\text{alle Zustände} \\ \text{mit } x_{1P} = 3}} p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}) \quad (6.8b)$$

$$Y_{2(3)} = \sum_{\substack{\text{alle Zustände} \\ \text{mit } x_2(x_3) > 0}} p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}) \quad (6.9)$$

$$n_i = \sum_{\text{alle Zustände}} x_i \cdot p(x_1, x_{1P}, x_2, x_{2P}, x_3, x_{3P}) \quad , \quad i = 1, 2, 3. \quad (6.10)$$

Der Durchsatz ist proportional zur Nutz-Auslastung der CPU. Betrachtet werde hierzu ein langes Systemzeit-Intervall $(0, t)$, $t \gg h_{GR}$. Die CPU ist in diesem Intervall für die Dauer $Y_{1R} \cdot t$ durch Benutzerprogramme belegt. Die mittlere Anzahl fertig bearbeiteter Programme ist $Y_{1R} \cdot t / h_{GR}$. Der Durchsatz D ist die pro Zeiteinheit abgefertigte Anzahl von Programmen, d.h.

$$D = \frac{Y_{1R}}{h_{GR}} \quad (6.11)$$

Zur Bestimmung der mittleren Durchlaufzeit eines Benutzerprogrammes wenden wir Littles Gesetz an. Hierzu schneiden wir (gedanklich) die Rückkopplungsschleife der CPU-Stufe auf und ersetzen den Eingang in das Netz durch einen zustandsabhängigen Verkehrsfluß mit gleicher Eigenschaft, insbesondere gleicher Verkehrsrate λ_{11} . Nach Littles Gesetz gilt $\lambda_{11} \cdot t_F = n_1 + n_2 + n_3 = M$, wobei $\lambda_{11} = \lambda_1 \cdot q_{11}$ und λ_1 die gesamte durchgesetzte Rate von CPU-Anforderungen. Eine Anforderung verläßt die CPU-Stufe nach jeder Beendigung der Systemverwaltungsphase, d.h. $\lambda_1 = Y_{1V} \cdot \mu_{1V}$, ($\mu_{1V} = 1/h_{1V}$). Damit ergibt sich schließlich

$$t_F = \frac{M}{Y_{1V} \cdot q_{11}} \cdot h_{1V} \quad (6.12)$$

6.2.5 Ergebnisse und Simulationsvergleiche

Im folgenden wird über eine Reihe von Ergebnissen berichtet, welche den Einfluß der wesentlichsten Parameter widerspiegeln. Die idealisiert gefundenen analytischen Ergebnisse werden dann Simulationsergebnissen gegenübergestellt, welche auf einem detaillierteren System- und Programm-Modell beruhen.

a) Rechenergebnisse

Beispiel 1: Multiprogrammgrad und Durchsatz

Betrachtet werde zunächst der Fall rechenzeitintensiver Programme, welche (idealisiert) keine Ein/Ausgabe erfordern. Bild 6.7 zeigt die Ergebnisse für die Nutzauslastung der CPU Y_{1R} (womit nach Gl. (6.11) gleichzeitig auch D festliegt), die Auslastung der CPU durch Systemprogramme Y_{1V} , die Gesamtauslastung $Y_1 = Y_{1R} + Y_{1V}$ sowie die Kanalauslastung Y_2 (Page-I/O) für den Programmtyp f nach Bild 6.2.

Die Ergebnisse haben folgende Interpretation: Für $M = 1$ besitzt das Programm eine gewisse page fault-Häufigkeit (Laden in den ASP, Übersetzen etc.). Die Auslastung der CPU ist jedoch kleiner als 100% infolge dieser Page-I/O Vorgänge, da bei $M = 1$ keine Parallelarbeit zwischen CPU und Page-I/O Kanal möglich ist. Für $M = 2$ und 3 kann die CPU-Nutzauslastung gesteigert werden durch Parallelarbeit von CPU und Page-I/O Kanal. Für $M > 3$ fällt jedoch die Nutzauslastung der CPU mit zunehmendem M drastisch ab: Es teilen sich zu viele Programme den ASP, die effektive Laufzeit zwischen zwei page faults sinkt ab und die Programme erzeugen eine hohe page fault-Rate ("thrashing"-Effekt), wodurch der Page-I/O Kanal zunehmend zum Systemengpaß wird; gleichzeitig steigt der Systemverwaltungsaufwand der CPU infolge häufiger Unterbrechungen. Im Grenzfall $M = S$ (jedes Programm hat nur 1 Seitenrahmen) sinkt die CPU-Nutzauslastung praktisch auf

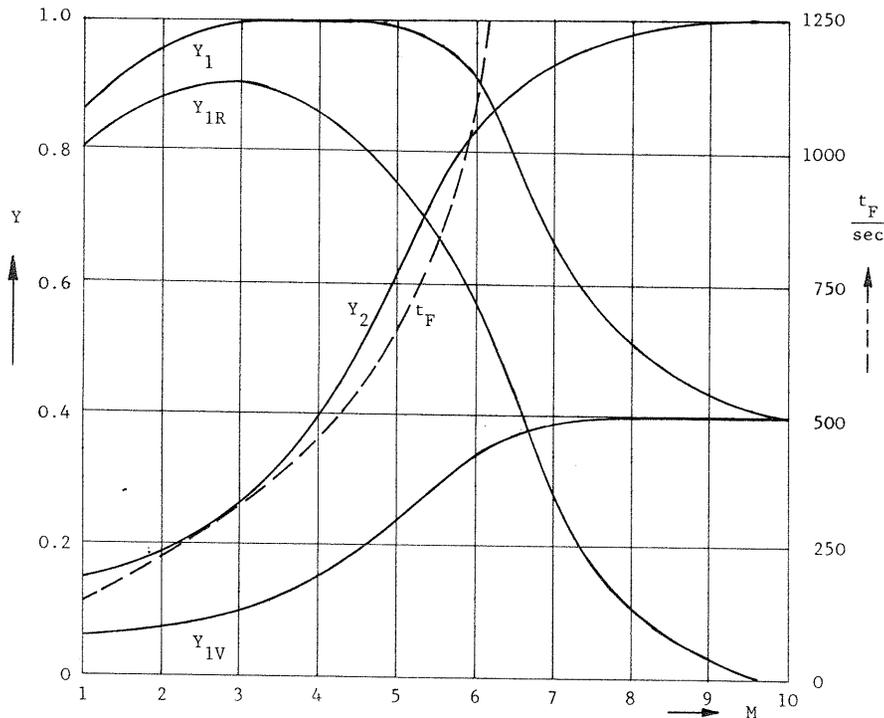


Bild 6.7. Belastungen der CPU und des Page-I/O Kanals sowie mittlere Durchlaufzeit in Abhängigkeit des Multiprogrammingsgrades

Parameter:

$L = S = 50$	$h_{1V} = 10 \text{ msec}$	Rechenphasen: $H_2(c_{1R}=2)$
$h_{GR} = 100 \text{ sec}$	$h_2 = 25 \text{ msec}$	Page-I/O Kanal: $E_3(k=3)$
$h_{Ref} = 10 \mu\text{sec}$	Programmtyp f	(Bild 6.2)

Null, während die CPU-Systemverwaltung bis auf 40% steigt entsprechend der Sättigungsrate des Page-I/O Kanals von 40 Zugriffen/sec.

Offensichtlich arbeitet das System Durchsatz-optimal für $M = 3$. Der Optimalitätspunkt stimmt i.a. mit der Lage eines charakteristischen "Knickes" ("Parachore" x_0 in Bild 6.2) in der Fehlseiten-Charakteristik überein. Wir beobachten ferner, daß die mittlere Durchlaufzeit eines Programmes zunächst etwa linear mit dem Multiprogrammingsgrad ansteigt, jedoch im thrashing-Bereich überproportional zunimmt.

Beispiel 2: Einfluß des Programmtyps

In Bild 6.8 sind die Funktionen $Y_{1R} = Y_{1R}(M)$ dargestellt für die 6 Programmtypen a-f nach Bild 6.2. Wie man sieht, eignen sich nur die Programme b, c, f, e zum Multiprogrammbetrieb unter den gegebenen Voraussetzungen.

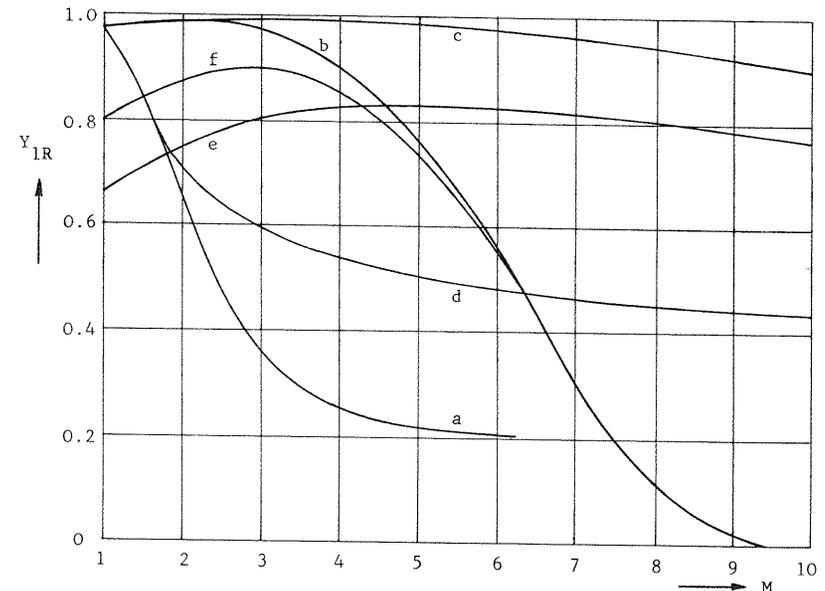


Bild 6.8. CPU-Nutzauslastung für verschiedene Programmtypen

Parameter:

$L = S = 50$	$h_{1V} = 10 \text{ msec}$	Rechenphasen: $H_2(c_{1R}=2)$
$h_{GR} = 100 \text{ sec}$	$h_2 = 25 \text{ msec}$	Page-I/O Kanal: $E_3(k=3)$
$h_{Ref} = 10 \mu\text{sec}$	Programmtypen a-f	(Bild 6.2)

Beispiel 3: Benutzerprogramme mit Ein/Ausgabe (File-I/O)

In diesem Beispiel werde der allgemeine Fall betrachtet mit zwei Kanälen für Page-I/O und File-I/O. Bild 6.9 zeigt die Ergebnisse der Auslastungen Y_{1R} , Y_2 (Page-I/O) und Y_3 (File-I/O) in Abhängigkeit des Multiprogrammingsgrades M und zwei Ausbaustufen des Arbeitsspeichers ($S = 30$ und $S = 90$).

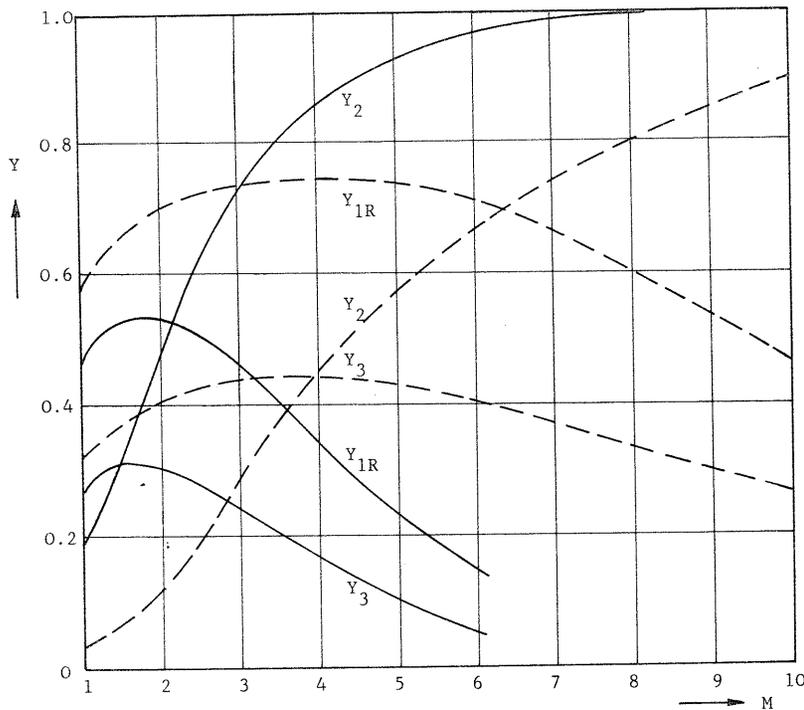


Bild 6.9. Auslastungen von CPU, Page-I/O und File-I/O Kanälen in Abhängigkeit des Multiprogrammgrades und der ASP-Kapazität

Parameter:

$L = 60$	$h_{GR} = 100 \text{ sec}$	Programmtyp g (Bild 6.2)
$S = 30$ (—)	$h_{Ref} = 2.5 \mu\text{sec}$	$n_F = 2000$
$S = 90$ (---)	$h_{1V} = 5 \text{ msec}$	Rechenphasen: H_2 ($c_{1R}=2$)
	$h_2=h_3 = 30 \text{ msec}$	I/O-Kanäle: E_3 ($k = 3$)

Bei den Berechnungen wurde angenommen, daß jeweils einem Page-I/O bzw. einem File-I/O Vorgang eine Systemverwaltungsphase mit Mittelwert h_{1V} vorausgeht.

Die Ergebnisse für $S = 30$ zeigen zunächst den prinzipiellen Verlauf wie in Bild 6.7. Die Größe Y_3 (File-I/O Kanalauslastung) ist proportional zu Y_{1R} infolge des konstanten (d.h. systemunabhängigen) Daten-Ein/Ausgabeverhaltens der Benutzerprogramme oder m.a.W., eine Optimierung des Durchsatzes garantiert automatisch die maximale Auslastung des File-I/O Kanals. Die max. CPU-Nutzauslastung liegt in diesem Beispiel bei etwa 52% ($M = 2$); die entsprechende Auslastung des Page-I/O Kanals liegt bei 48%. Zur Steigerung des Durchsatzes können prinzipiell zwei Maßnahmen ergriffen werden:

- schnelleres Medium für Page-I/O
- Ausbau des Arbeitsspeichers.

Die Entscheidung muß im einzelnen Falle nach sorgfältiger Analyse des Nutzens und der Kosten erfolgen. In Bild 6.9 sind gestrichelt die Ergebnisse für einen ausgebauten ASP ($S = 90$) eingezeichnet: die Nutzauslastung konnte in diesem Falle auf ca. 75% gesteigert werden; gleichzeitig bemerken wir ein wesentlich flacheres Maximum, d.h. das System ist unempfindlicher gegenüber Schwankungen des Multiprogrammgrades.

b) Simulationsvergleiche

Jede Modellanalyse, welche unter idealisierten Voraussetzungen durchgeführt wird, muß hinsichtlich ihres Gültigkeitsbereiches überprüft werden. Rechneranalysen können letztlich nur durch Messungen an realen Systemen überprüft werden, welche schwierig durchzuführen sind. Aus diesem Grunde wurden umfangreichere Simulationsstudien angestellt. Hierzu sind Simulationsmodelle entwickelt worden, welche - in der Hierarchie der Modellierungsgenauigkeit - zwischen den analytischen Modellen und realen Systemen liegen [180-181]. Die wesentlichsten Merkmale dieser Simulationsmodelle sind:

- künstliche Erzeugung von Seitenreferenzketten einschließlich des Ein/Ausgabeverhaltens, welche unabhängig von der eigentlichen Simulation erzeugt, getestet, gespeichert und mit gemessenen Charakteristiken verglichen werden können [180].
- Rechnermodell, welches die prinzipielle Warteschlangenstruktur mit einer beliebigen Anzahl von Kanälen für Page-I/O und File-I/O, den realen Arbeitsspeicher mit der zugrundegelegten Speicherorganisation sowie die wesentlichen Eigenschaften des Betriebssystems umfaßt. Das Rechnermodell wird in seinem Zeitablauf simuliert, wobei die vorher erzeugten und abgespeicherten Seitenreferenzketten entsprechend des Multiprogrammgrades simultan abgearbeitet werden und damit die Zustandsänderungen auslösen, d.h. das eigentliche Ablaufgeschehen steuern [181].

Diese Modell-Zwischenebene zeichnet sich durch wesentlich weniger Idealisierungen aus im Vergleich zur analytischen Modellebene; sie ist andererseits flexibler zur Anpassung an verschiedene Systemstrukturen, Betriebsstrategien oder Auftragsprofile im Vergleich zu realen Systemen. Allerdings erfordert diese Technik bereits beträchtliche Rechnersimulations-Laufzeiten und ist wahrscheinlich nahe der äußeren Grenze der Simulationsmöglichkeiten.

Bild 6.10 zeigt einen Vergleich zwischen den analytischen Ergebnissen und den Simulationsergebnissen, welche mit dem Simulationsprogramm [180-181] gewonnen wurden. Die Voraussetzungen sind im wesentlichen gleich wie im vorausgegangenen Beispiel 3. Die Seitenreferenzketten, welche für die Simulation verwendet

Das Betriebssystem trägt dieser Forderung Rechnung durch

- ein effektives Zeitscheibenverfahren ("time sharing"), welches die längere und ausschließliche Belegung des Prozessors durch ein langes Benutzerprogramm verhindert
- die zeitweilige Auslagerung von momentan inaktiven Benutzerprogrammen aus dem Arbeitsspeicher ("Swapping"), welches eine zu starke Programmzerstückelung im Arbeitsspeicher unterbindet.

Die Problematik besteht hierbei in einer effektiven Steuerung des Multiprogrammgrades derart, daß

- vorgeschriebene Antwortzeiten eingehalten werden
- die Betriebsmittel möglichst optimal genutzt werden
- keine Systemengpässe auftreten (z.B. durch Swapping).

b) Voraussetzungen

Vorausgesetzt werde eine Rechnerstruktur ähnlich wie in Bild 6.1 mit zusätzlichen Ein/Ausgabekanälen für den Terminalverkehr. Der Prozessor (CPU) wird nach einem verwaltungszeitarmen Zeitscheibenverfahren betrieben, sodaß wir die idealisierte Betriebsart "processor sharing" annehmen können (vergl. 2.3.3 e). Die typische Zustandsfolge eines Benutzerprogrammes ist CPU-Bearbeitung - Terminal ("Denkzeit" des Benutzers) - CPU-Bearbeitung usw., wobei die CPU-Phasen - außer der zyklischen Unterbrechung durch andere aktive Benutzerprogramme infolge processor sharing - durch notwendige Page-I/O bzw. File-I/O Vorgänge vorübergehend unterbrochen sein können. Wir nehmen im folgenden an, daß ein Benutzerprogramm nach Übergabe der Kontrolle an das Terminal (Benutzer-Denkphase) grundsätzlich aus dem Arbeitsspeicher (ASP) ausgelagert wird. Umgekehrt muß das Benutzerprogramm zu seiner weiteren Bearbeitung nach Abschluß einer Terminalphase jeweils wieder in den ASP transportiert werden. Nach Beendigung einer Terminalphase, welche dem System durch einen kurzen Interrupt mitgeteilt wird, wird das Benutzerprogramm zunächst in eine Eingangs-Warteschlange eingereiht; wartende Benutzerprogramme in dieser Warteschlange werden nur dann wieder aktiviert, wenn der Multiprogrammgrad unterhalb einer oberen Schranke M_{max} liegt. Die übrigen Voraussetzungen entsprechen denen aus Abschnitt 6.2. Zur Kennzeichnung des Benutzerprogramm-Verhaltens wird eine zusätzliche Größe n_T eingeführt (n_T = mittlere Anzahl von "Transaktionen" oder Terminal Ein/Ausgabevorgängen pro Benutzerprogramm, kurz "Terminal-I/O").

6.3.2 Systemmodell

Wie unter Abschnitt 6.2.3 besteht das Systemmodell aus einem Warteschlangenmodell, einem Speichermodell und einem Benutzerprogramm-Modell.

a) Warteschlangenmodell

Bild 6.11 zeigt ein einfaches Warteschlangenmodell, welches in Erweiterung zu

Bild 6.3 durch N Terminals (Bedienungsstufe 2 aus N "Bedienungseinheiten", welche die Denkzeiten beschreiben), die Eingangs-Warteschlange und die durch einen zustandsabhängigen Schalter symbolisierte Zulassungsstrategie ausgezeichnet ist.

Die Bedienungsstufe 1 (CPU) arbeitet nach der Strategie "processor sharing". Die Bedienungszeiten (Rechenphasen) seien allgemein verteilt. Die Bedienungsstufe 2 besitzt keine Warteschlange, da jedes Terminal typischerweise nur von einem Benutzer bedient wird. Es wird ferner angenommen, daß alle N Terminals aktiv sind, d.h. sich N Benutzerprogramme im System befinden. Ferner wird angenommen, daß die Benutzer-Denkzeiten negativ exponentiell verteilt sind. Die Kanal- und Zugriffssysteme für Page-I/O und File-I/O sind als einfache Bedienungssysteme (FIFO, eine Bedienungseinheit)modelliert mit exponentiellen Bedienungszeiten.

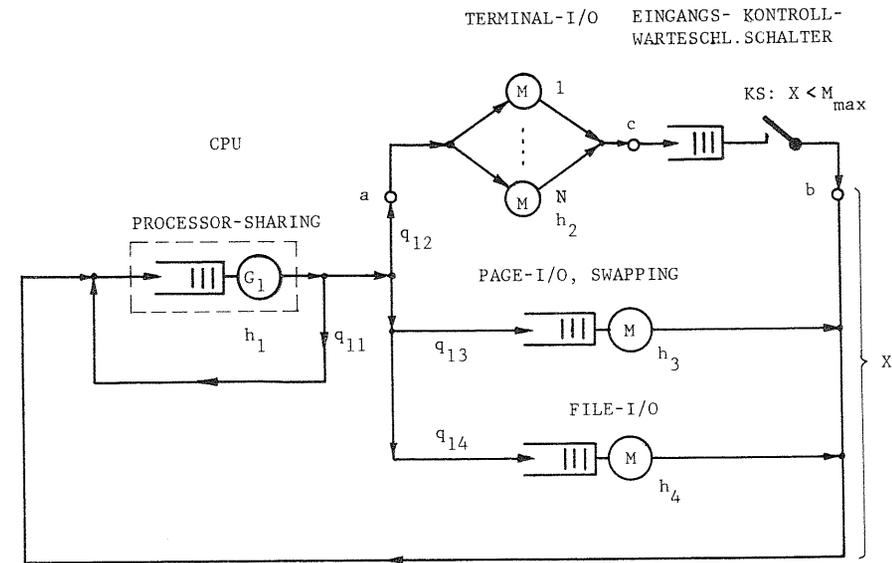


Bild 6.11. Warteschlangenmodell für Teilnehmerrechnersystem

Die Verzweigungswahrscheinlichkeiten haben folgende Bedeutung:

- q_{11} Häufigkeit von Ereignissen für Programmende und folgendem Programmstart
- q_{12} Häufigkeit von Terminal-I/O Vorgängen
- q_{13} Häufigkeit von Page-I/O Vorgängen einschließlich der Ein/Auslagerung von Benutzerprogrammen (Swapping)
- q_{14} Häufigkeit von File-I/O Vorgängen.

Von den N Programmen im System mögen sich X im Systemkern (CPU,Page-I/O,File-I/O) befinden und N-X in Stufe 2 (Terminal) bzw. der Eingangs-Warteschlange. Der Kontrollschalter KS wird zustandsabhängig betätigt:

KS geschlossen, falls $X < M_{\max}$
 KS geöffnet, falls $X = M_{\max}$.

b) Speichermodell

Das Ziel der Untersuchung ist eine optimale Festlegung von M_{\max} . Es wird erwartet, daß das System einen Arbeitspunkt X nahe M_{\max} besitzt. Aus diesem Grunde wird angenommen, daß jedes am Multiprogramming teilhabende Programm genau S/M_{\max} Seitenrahmen erhält selbst in Zuständen $X < M_{\max}$; dies hat gleichzeitig den Vorteil, daß das nächste in den ASP zu ladende Benutzerprogramm keine Seiten anderer Benutzerprogramme verdrängen muß.

c) Benutzerprogramm-Modell

Das Benutzerprogramm ist gekennzeichnet durch die Parameter

L	Programmlänge in Seiten
h_{GR}	Gesamt-Rechenzeitbedarf
$f_P(x, ES)$	Fehlseiten-Charakteristik (nur für Paging, nicht für Swapping) mit $x = S/(L \cdot M_{\max})$
n_T	Mittlere Anzahl von Terminal-I/O Vorgängen
n_F	Mittlere Anzahl von File-I/O Vorgängen.

Die page fault-Unterbrechungen seien gleichwahrscheinlich über die Seitenreferenzkette verteilt mit Mittelwert $n_p = R \cdot f_P(x, ES)$. Ferner wird angenommen, daß die Terminal-I/O und File-I/O Vorgänge sich ebenfalls gleichwahrscheinlich über die Seitenreferenzkette verteilen.

d) Festlegung der Parameter des Systemmodells

Infolge zweier Swapping-I/O Vorgänge bei jedem Terminal-I/O Vorgang durchläuft ein Benutzerprogramm insgesamt im Mittel $n_G = n_p + n_F + 3n_T$ mal die CPU, wovon $n_p + n_F + n_T$ auf echte Rechenphasen entfallen. Eine Rechenphase dauert im Mittel

$$h_{1R} = \frac{h_{GR}}{n_p + n_F + n_T} \quad (6.13)$$

Die restlichen $2n_T$ CPU-Phasen sind reine Systemverwaltungsphasen mit der mittleren Dauer h_{1V} . Die mittlere CPU-Phase einschließlich Systemverwaltungszeit ist somit

$$h_1 = h_{1V} + \frac{n_p + n_F + n_T}{n_G} h_{1R} \quad (6.14)$$

Die Verzweigungswahrscheinlichkeiten bestimmen sich zu

$$q_{11} = \frac{1}{n_G + 1} \quad (6.15a)$$

$$q_{12} = \frac{n_T}{n_G + 1} \quad (6.15b)$$

$$q_{13} = \frac{n_p + 2n_T}{n_G + 1} \quad (6.15c)$$

$$q_{14} = \frac{n_F}{n_G + 1} \quad (6.15d)$$

Ferner seien vorgegeben die mittlere Denkzeit h_2 , die mittlere Zeit für Page-I/O bzw. für Swapping h_3 sowie die mittlere Zeit für File-I/O h_4 .

Bemerkung: Die vorübergehende Auslagerung von Programmen aus dem Arbeitsspeicher bei Terminal-I/O ist nur sinnvoll, solange die Anzahl aktiver Terminals größer ist als der Multiprogramminggrad, d.h. $N > M$. In realen Systemen ist diese Bedingung i.a. erfüllt (typischerweise $N = 10 \dots 500$, während i.a. $M < 10$).

6.3.3 Modellanalyse

Die exakte Modellanalyse wird durch den zustandsabhängigen Kontrollschalter erheblich erschwert. Ähnliche Modelle ohne Kontrollschalter wurden untersucht in [164-166]. Eine neuere Veröffentlichung [171] befaßt sich mit der Analyse verschiedener Swapping-Strategien. Das Problem einer optimalen Steuerung des Multiprogramminggrades wurde in [183] behandelt auf der Grundlage eines Zerlegungsverfahrens.

Infolge der Schwierigkeiten der mathematischen Modellanalyse gehen wir hier folgenden Weg:

- Es werden zwei Grenzmodelle entwickelt, welche das tatsächliche Modell von oben und von unten her einschachteln.
- Die Voraussetzungen der beiden Grenzmodelle werden so gewählt, daß sie der Produktlösungsform genügen; hiermit ist ferner eine relativ einfache numerische Auswertung möglich.

a) Zwei analytische Grenzmodelle

Beide Grenzmodelle bestehen aus einem geschlossenen Warteschlangennetz mit M_{\max} Benutzerprogrammen entsprechend dem maximal zugelassenen Multiprogramminggrad. Der Einfluß der Terminal-Bedienstufe wird hierbei in zwei verschiedenen Grenz-betrachtungen einbezogen.

Grenzmodell A:

Grenzmodell A entsteht aus dem Modell Bild 6.11 durch die Annahme, daß stets $X = M_{\max}$ ist. Dies entspricht dem Fall, daß die Eingangs-Warteschlange nie leer ist oder m.a.W., eine in die Terminalstufe überwechselnde Anforderung wird im Systemkern (Stufen 1,3 und 4) momentan durch eine in der Eingangs-Warteschlange wartende Anforderung ersetzt. Physikalisch bedeutet dies, daß in diesem Grenzfall die Denkzeiten verschwindend klein sind ($h_2 \rightarrow 0$), womit der Systemkern maximal ausgelastet wird. In der Eingangs-Warteschlange warten konstant $N - M_{\max}$ Benutzerprogramme auf die Zulassung zur Fortsetzung ihrer Bearbeitung. Zur Beschreibung des Ablaufgeschehens im Systemkern wird die Terminalstufe 2 über die Punkte a-b kurzgeschlossen und das verbleibende geschlossene Warteschlangennetz aus den Stufen 1, 3 und 4 mit konstant $X = M_{\max}$ Anforderungen betrachtet.

Grenzmodell A liefert den maximal möglichen Durchsatz bei M_{\max} zugelassenen Programmen im Arbeitsspeicher. Gleichzeitig wird damit die Durchlaufzeit einer Terminal-Transaktion, gemessen von Punkt b zu Punkt a, maximal. Grenzmodell A ist exakt für die Übergänge

- 1) $N \rightarrow \infty$ bzw. $h_2 \rightarrow 0$
- 2) $n_T \rightarrow 0$.

Im ersten Falle ist die Eingangs-Warteschlange nicht leer, während im zweiten Falle die Terminalstufe einen verschwindenden Verkehr besitzt ($q_{12} \rightarrow 0$). Für die praktisch bedeutsamen Fälle $N \gg M_{\max}$ bzw. $n_T \ll (n_P + n_F)$ können wir daher erwarten, daß das Grenzmodell A Ergebnisse liefert, welche sehr nahe am tatsächlichen Modell Bild 6.11 liegen.

Grenzmodell B:

Ein zweites Grenzmodell B entsteht aus Bild 6.11, indem die Eingangs-Warteschlange und der Kontrollschalter über die Punkte b-c kurzgeschlossen werden und das verbleibende geschlossene vierstufige Warteschlangennetz mit konstant M_{\max} Anforderungen betrachtet wird. In diesem Falle ist zunächst die Bedingung $X \leq M_{\max}$ stets erfüllt. Der Zugangsprozeß wartender bzw. die Terminalstufe verlassender Anforderungen zum Systemkern über Punkt b wird in diesem Grenzmodell durch eine zustandsabhängige Bedienungsrate $\mu_2 = \mu_2(x_2)$ beschrieben, wobei $x_2 = M_{\max} - X$ die Anzahl der Anforderungen in der Stufe 2:

$$\mu_2(x_2) = \begin{cases} 0 & \text{für } x_2 = 0 \\ (N-X)\epsilon_2 & \text{für } x_2 > 0, \quad \epsilon_2 = 1/h_2 \end{cases} \quad (6.16)$$

Gl.(6.16) ist wie folgt zu interpretieren: Sind $X = M_{\max}$ Anforderungen im Systemkern (d.h. $x_2 = 0$), so ist der Kontrollschalter in Bild 6.11 geöffnet; dies wird

durch die Enderate $\mu_2(0) = 0$ exakt beschrieben. Sind dagegen $X < M_{\max}$ Anforderungen im Systemkern (d.h. $x_2 > 0$), so sind im tatsächlichen Modell Bild 6.11 $(N-X)$ Anforderungen in einer Terminal-Denkphase; dies wird durch die Enderate $\mu_2(x_2) = (N-X)\epsilon_2$ ebenfalls exakt beschrieben für $X = 0, 1, \dots, M_{\max} - 1$, d.h. $x_2 = M_{\max}, M_{\max} - 1, \dots, 1$. Der einzige Fehler dieses Modelles entsteht beim Übergang vom Zustand $X = M_{\max}$ zum Zustand $X = M_{\max} - 1$: Im tatsächlichen Modell 6.11 wird der Zustand $X = M_{\max}$ nur dann beendet, wenn eine der M_{\max} Anforderungen den Systemkern verläßt und inzwischen keine Anforderung die Terminal-Denkphase abgeschlossen hat; hat dagegen eine oder haben mehrere Anforderungen die Denkphase abgeschlossen, so bleibt der Systemzustand $X = M_{\max}$ bestehen, da sofort eine der wartenden Anforderungen in den Systemkern übernommen wird. Im Grenzmodell B hingegen wechselt der Zustand $X = M_{\max}$ stets in den Zustand $X = M_{\max} - 1$, wenn eine Anforderung den Systemkern verläßt.

Verglichen mit dem tatsächlichen Modell befinden sich in Grenzmodell B im Mittel weniger Anforderungen im Systemkern, d.h. wir erhalten einen unteren Grenzwert für Durchsatz und Durchlaufzeit b-a. Grenzmodell B ist ebenfalls exakt für die Übergänge

- 1) $N \rightarrow \infty$ bzw. $h_2 \rightarrow 0$
- 2) $n_T \rightarrow 0$.

Im ersten Falle gilt $\mu_2(x_2) \rightarrow \infty$ für $x_2 > 0$, während im zweiten Falle der Terminalverkehr verschwindend gering ist, wodurch ebenfalls stets $X = M_{\max}$ Anforderungen im Systemkern garantiert sind. Diese beiden Bedingungen sind identisch mit denen des Grenzmodells A; da das wirkliche Modell von beiden Grenzmodellen hinsichtlich Durchsatz und Durchlaufzeit b-a eingeschachtelt wird, beide Grenzmodelle aber für die beiden Übergänge mit dem tatsächlichen Modell Bild 6.11 übereinstimmen, können wir diesbezüglich eine sehr gute Genauigkeit erwarten.

b) Zustandswahrscheinlichkeiten und charakteristische Größen

Beide Grenzmodelle A und B genügen den Voraussetzungen von Warteschlangennetzen mit Produktlösungsform. Die Zustandswahrscheinlichkeiten lauten entsprechend Abschnitt 5.1.1

$$p(x_1, x_2, x_3) = C_A \cdot \prod_{i=1,3,4} \frac{e_i^{x_i}}{\mu_i^{x_i}}, \quad (6.17a)$$

$$p(x_1, x_2, x_3, x_4) = C_B \cdot \prod_{i=1,3,4} \frac{e_i^{x_i}}{\mu_i^{x_i}} \cdot \prod_{j=1}^{x_2} \frac{e_2}{\mu_2(j)}, \quad (6.17b)$$

wobei $\mu_i = 1/h_i$, $i = 1, 3, 4$, und $e_i = e_1 q_{1i}$, $i = 2, 3, 4$.

Die charakteristischen Größen werden aus den Zustandswahrscheinlichkeiten gewonnen, wobei zweckmäßigerweise einer der Rekursionsalgorithmen [92, 93] angewendet wird, welche direkt die Auslastungen der Bedienungseinheiten Y_1, \dots, Y_4 sowie die mittlere Anzahl von Anforderungen in den Stufen 1 bis 4, n_1, \dots, n_4 , liefern [183]. Hieraus folgen schließlich

$$Y_{1V} = Y_1 \cdot \frac{h_{1V}}{h_1} \quad , \quad (6.18a)$$

$$Y_{1R} = Y_1 - Y_{1V} \quad , \quad (6.18b)$$

$$D = \frac{Y_{1R}}{h_{GR}} \quad . \quad (6.19)$$

Die Terminal-Antwortzeit für eine Transaktion setzt sich aus zwei Anteilen zusammen:

$$t_R = t_{EW} + t_F \quad , \quad (6.20)$$

wobei t_{EW} = mittlere Wartezeit in der Eingangs-Warteschlange und t_F = mittlere Durchlaufzeit einer Anforderung von Punkt b nach Punkt a in Bild 6.11. In beiden Grenzmodellen berechnen wir hierzu jeweils die Rate λ_2 von Anforderungen zwischen den Punkten a und b:

$$\lambda_2 = Y_1 \cdot \mu_1 \cdot q_{12} \quad . \quad (6.21)$$

Durch Anwendung von Littles Gesetz auf die beiden Teilsysteme "Eingangs-Warteschlange" und "Systemkern" finden wir

$$t_{EW} = \frac{N-M_{\max}}{\lambda_2} \quad (6.22a)$$

$$t_F = \begin{cases} \frac{M_{\max}}{\lambda_2} & \text{für Grenzmodell A} \\ \frac{M_{\max} - n_2}{\lambda_2} & \text{für Grenzmodell B.} \end{cases} \quad (6.22b)$$

Mit 6.20) folgt schließlich

$$t_R = \begin{cases} \frac{N}{\lambda_2} & \text{für Grenzmodell A} \\ \frac{N-n_2}{\lambda_2} & \text{für Grenzmodell B.} \end{cases} \quad (6.23)$$

6.3.4 Ergebnisse

Die beschriebene Einschachtelungsmethode ist umso genauer

- je größer die Anzahl der Terminals N bzw. je kleiner die "Denkzeit"
- je geringer die Terminal-I/O Häufigkeit.

Wir beschränken uns daher auf den ungünstigeren Fall einer relativ kleinen Anzahl von Terminals (N = 10) mit einer kleinen sowie einer größeren Terminal-I/O Häufigkeit, um den Einfluß der Approximation sowie der Wahl des Multiprogramminggrades zu studieren.

Die Bilder 6.12a und 6.12b zeigen die CPU-Nutzauslastung Y_{1R} (und damit den Durchsatz) sowie die Terminal-Antwortzeit t_R in Abhängigkeit des maximalen Multiprogramminggrades M_{\max} für die zwei Fälle $n_T = 10$ und $n_T = 100$ (weitere Parameter siehe Legende zu Bild 6.12). Wie zu erwarten war, ergeben sich im Falle $n_T = 10$ nur sehr geringe Unterschiede zwischen den beiden Grenzmodellen, während sich im Falle $n_T = 100$ für kleine M_{\max} größere Unterschiede einstellen können (Bei Erhöhung der Anzahl von Terminals geht dieser Unterschied rasch zurück). Ferner vermindert sich der Unterschied mit ansteigendem Multiprogramminggrad: Mit zunehmendem M_{\max} steigt die page fault-Häufigkeit n_p , wodurch das Verhältnis n_T/n_p verringert wird. Dies hat denselben Effekt zur Folge wie im Falle geringer Terminal-I/O Häufigkeit. Unter Zugrundelegung realer Verhältnisse ($N \gg M_{\max}$, $n_T \ll n_p + n_F$) können wir folgern, daß das beschriebene Einschachtelungsverfahren eine gute Abschätzung des Einflusses der zustandsabhängigen Multiprogramminggrad-Steuerung erlaubt.

Abgesehen von der Approximationsgüte können wir folgende Schlußfolgerungen aus den Ergebnissen der Bilder 6.12a,b ziehen:

- In Teilnehmerrechnersystemen mit Virtuellem Speicher kann der Durchsatz optimiert werden durch eine entsprechende Steuerung des Multiprogramminggrades, indem jene Benutzerprogramme, welche sich in einer Terminal-Denkphase befinden, zeitweilig aus dem Arbeitsspeicher ausgelagert werden (Swapping). Die Lage und Höhe des Optimums hängen von den Benutzerprogramm-Eigenschaften und den Systemparametern ab.
- Die Gesamtantwortzeit auf eine Terminal-Transaktion einschließlich der Eingangs-Warteschlange wird minimal, wenn das System im Durchsatzoptimum betrieben wird.

Wir bemerken abschließend, daß die vorgestellten Ergebnisse nur im Rahmen der getroffenen Voraussetzungen zu sehen sind, insbesondere hinsichtlich der Voraussetzungen über das Swapping. Bei größeren Zeitscheiben kann es unter Umständen günstig sein, Benutzerprogramme bereits nach Ablauf eines Zeitscheibenabschnittes

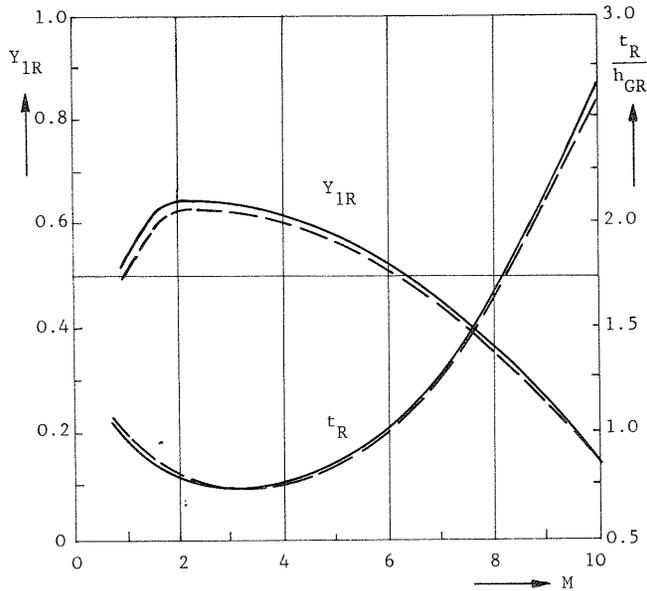


Bild 6.12a.

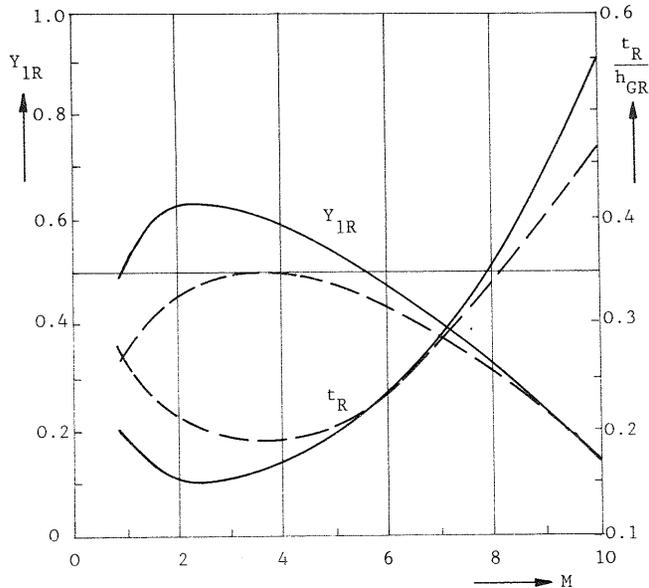


Bild 6.12b.

Bild 6.12.

CPU-Nutzauslastung Y_{1R} und Terminal-Antwortzeit t_R in Abhängigkeit des Multiprogramminggrades M

Parameter:

- $N = 10$ Terminals
- $L = 60$
- $S = 60$
- $h_{GR} = 100$ sec
- $h_{Ref} = 2.5 \mu\text{sec}$
- $h_{1V} = 5$ msec
- $h_2 = 10$ sec
- $h_3 = 30$ msec
- $h_4 = 30$ msec
- $n_F = 2000$ FILE-I/O Zugr.
- $n_T = 10$ (Bild 6.12a)
- $n_T = 100$ (Bild 6.12b)
- Programmtyp g (Bild 6.2)

- Grenzmodell A
- - - Grenzmodell B

auszulagern; die Entscheidung über eine solche Swapping-Strategie muß nach sorgfältiger Abwägung des Nutzens und der durch die zusätzlichen Transporte bedingten Mehrarbeit erfolgen.

6.4 Antwortzeiten in Interaktiven Systemen

Im Zusammenhang mit der rechnergestützten Automatisierung vieler betrieblicher Vorgänge gewinnen interaktiv betriebene Rechnersysteme zunehmende Bedeutung. Typische Beispiele sind Auskunft- und Datenbanksysteme, Datensammelsysteme zur Datenerfassung sowie Systeme zur rechnergestützten Buchung, Auftragsabwicklung, Lagerhaltung, Ersatzteil- und Reparaturplanung und dergleichen mehr. Im folgenden wird an einem Beispiel die Analyse derartiger Systeme durchgeführt.

6.4.1 Problemstellung und Voraussetzungen

Interaktiv betriebene Rechnersysteme in den oben angeführten Einsatzbereichen sind oftmals als Kleinrechnersysteme realisiert, welche mit einer größeren Anzahl von Terminals verbunden sind. Zur Verwaltung größerer Datenbestände sowie zur Abfrage und/oder Veränderung von Daten sind diese Systeme i.a. mit einem Massenspeichermedium mit Direktzugriff (Platte) ausgerüstet. Im Arbeitsspeicher befinden sich neben dem Betriebssystem bestimmte Arbeitsprogramme, welche den Datenaustausch des Rechnersystems mit den Terminals wie auch den Datenaustausch mit dem Hintergrundspeicher steuern. Sämtliche zwischen den Eingabe/Ausgabestationen und dem Hintergrundspeicher ausgetauschten Daten werden vorübergehend im Arbeitsspeicher abgelegt, eventuell modifiziert, ergänzt und formatiert.

Bei dem interaktiven und simultanen Betrieb des Systems mit einer Vielzahl von Benutzern entstehen u.a. folgende Problemstellungen:

- Dimensionierung von Arbeits- und Hintergrundspeichern
- Organisation des Datentransportes
- Einhaltung vorgeschriebener Reaktionszeiten.

Zur Beantwortung dieser Fragen ist die Kenntnis der Leistungsfähigkeit des Systems von großer Bedeutung, welche im wesentlichen von den Parametern des Anforderungsprozesses und des Systems abhängt. Die Ergebnisse einer solchen Leistungsanalyse können zur Planung, Dimensionierung sowie zur Auffindung und Beseitigung von Systemengpässen benutzt werden.

In dem folgenden Beispiel setzen wir ein einfaches System voraus mit einem Prozessor (CPU) sowie einem Ein/Ausgabekanal für einen Plattenspeicher (Disk-I/O). Der Verkehr zwischen Rechnersystem und Terminals wird nur global beschrieben. Wir setzen ferner voraus, daß das Ein/Ausgabesystem des Rechners für den

Terminalverkehr keinen Systemengpaß darstellt und keine wesentlichen verkehrsbedingten Verzögerungen verursacht.

6.4.2 Systemmodell

Das Systemmodell besteht aus einem zweistufigen Warteschlangennetz entsprechend Bild 6.13.

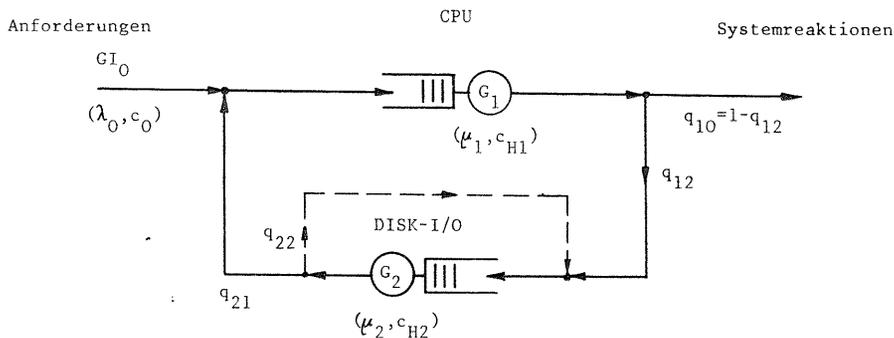


Bild 6.13. Systemmodell eines interaktiven Rechnersystems

Externe Anforderungen der Terminals warten in der CPU-Stufe auf eine Bedienungsphase. Eine abgeschlossene Bedienungsphase führt entweder direkt zu einer Systemreaktion (z.B. bei der Aktivierung eines Arbeitsprogrammes) oder zu einem oder mehreren Plattenzugriffen (Abspeichern/Verändern oder Bereitstellen von Daten). Der Ankunftsprozeß sowie die "Bedienungszeiten" für CPU-Bearbeitung bzw. Plattenzugriffe werde durch einen allgemeinen Prozeß GI_0 bzw. allgemeine Prozesse G_1, G_2 beschrieben. Alle Prozesse seien durch 2 Parameter (Rate, Variationskoeffizient) charakterisiert. Die Häufigkeit der Inanspruchnahme der Betriebsmittel "CPU" und "Platte" durch eine externe Anforderung wird global für sämtliche Anforderungen durch die Verzweigungswahrscheinlichkeiten q_{12} und q_{22} beschrieben.

6.4.3 Modellanalyse

Zur Analyse des Warteschlangennetzes nach Bild 6.13 wird das in Abschnitt 5.4 beschriebene Zerlegungsverfahren für allgemeine Warteschlangennetze angewendet. Infolge der Rückkopplungen und der allgemeinen Prozeßvoraussetzungen gehört das Modell nach Bild 6.13 bereits zur allgemeinsten Klasse des in 5.4 vorausgesetzten Netztyps. Das Lösungsverfahren ist ausführlich in Abschnitt 5.4 beschrieben und in einem allgemein anwendbaren Analyseprogramm implementiert [120].

6.4.4 Ergebnisse

Zugrundegelegt sei ein interaktives System zur Datenerfassung entsprechend Bild 6.13. In den Beispielen gehen wir von folgenden Voraussetzungen aus:

- Eintreffende Anforderungen bilden einen externen Ankunftsprozeß mit exponentiell (M) oder hyperexponentiell (H_2) verteilten Ankunftsabständen
- Bedienungszeiten der CPU-Stufe sind hyperexponentiell (H_2) oder hypoexponentiell (E_4)
- Bedienungszeiten der Plattenzugriffe (Disk-I/O) sind hypoexponentiell (E_4)
- Mittlere Antwortzeit $t_R \leq 2$ sec.

Kurve ① in Bild 6.14a zeigt die Antwortzeit des Systems im Falle eines langsameren Disk-I/O Systems. Jede Anforderung benötigt im Mittel 4 Plattenzugriffe, wodurch das Plattenzugriffssystem zum Engpaß wird. Der maximale Durchsatz ist $\lambda_0 = 0.125/\text{sec}$; zur Einhaltung der Antwortzeit $t_R \leq 2$ sec muß $\lambda_0 \leq 0.10/\text{sec}$ sein. Das System soll nun durch Verbesserungsmaßnahmen auf eine höhere Leistung gebracht werden.

Kurve ② zeigt eine erste Verbesserung mit einem doppelt so schnellen Plattenzugriffssystem; der max. Durchsatz liegt nun bei $\lambda_0 = 0.20/\text{sec}$ und es kann $\lambda_0 \leq 0.15/\text{sec}$ sein zur Einhaltung der Antwortzeitforderung. Zur weiteren Verbesserung werde nun eine strukturelle Maßnahme ergriffen: durch eine verbesserte Datenorganisation sei es möglich, die gesamten abzuspeichernden Daten im Arbeitsspeicher zu sammeln und mit einem Plattenzugriff abzulegen. Die Kurven ③ und ④ sind die neuen Ergebnisse für die beiden Plattenzugriffssysteme; der max. Durchsatz konnte auf $\lambda_0 = 0.50/\text{sec}$ gesteigert werden und es können Anforderungen mit $\lambda_0 \leq 0.44/\text{sec}$ bzw. $\lambda_0 \leq 0.46/\text{sec}$ zugelassen werden. Kurve ⑤ schließlich zeigt eine Modellvariante, bei welcher im Mittel 2 Plattenzugriffe benötigt werden, welche jedoch keine zusätzliche CPU-Phase benötigen (modelliert durch die gestrichelte Rückkopplungsschleife in Bild 6.13).

Die Kurven ① bis ④ in Bild 6.14b geben Aufschluß über den Einfluß verschiedener Prozeßcharakteristiken hinsichtlich stärker streuender Ankunftsabstände bzw. weniger stark streuender CPU-Bedienungszeiten im Vergleich zu rein Markoffschen Prozessen ④, bei welchen die Produktlösungsform anwendbar ist.

In den Bildern 6.14a,b sind ferner einige Simulationsergebnisse eingetragen, welche die Approximationsgüte des Analyseverfahrens charakterisieren. Die Abweichungen rühren größtenteils von der Annahme der Erneuerungseigenschaft her, welche bei kleinen Netzen mit Rückkopplungszweigen besonders kritisch ist (vergl.

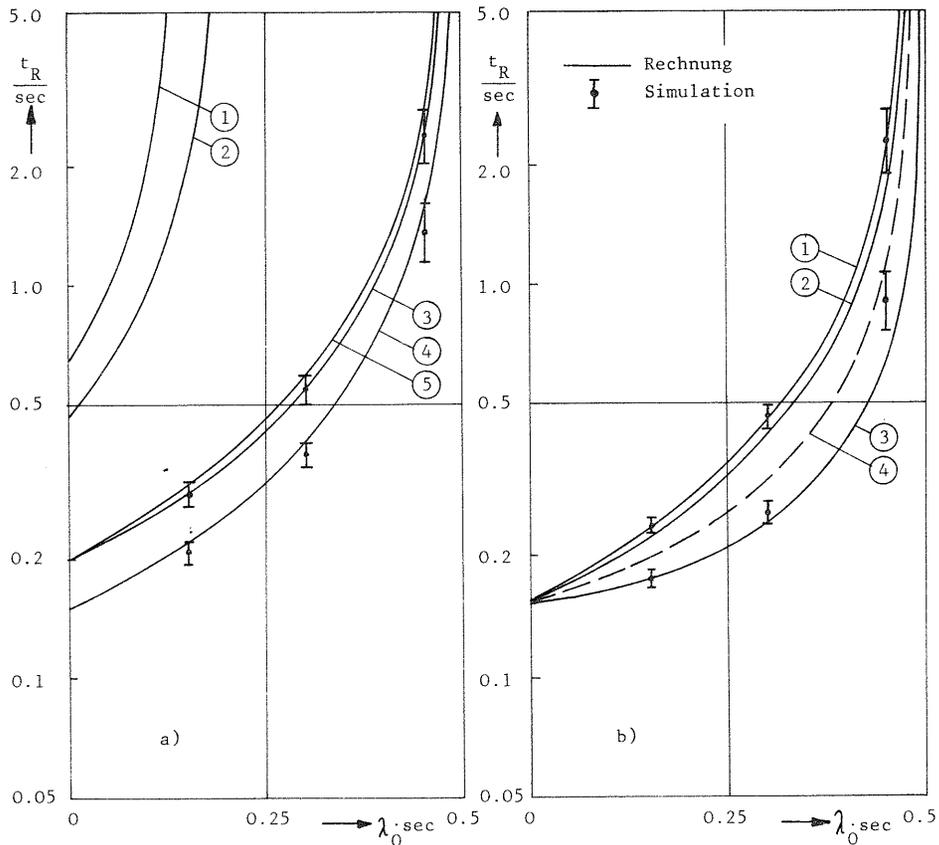


Bild 6.14. Mittlere Antwortzeit t_R in Abhängigkeit der Ankunftsrate λ_0

Parameter Bild 6.14a:

Parameter Bild 6.14b:

	$GI_0/G_1/G_2$	c_{H1}	$\frac{h_1}{msec}$	$\frac{h_2}{msec}$	q_{12}	q_{22}	Sim.
①	M/H ₂ /E ₄	1.5	50	100	0.8	0	
②	M/H ₂ /E ₄	1.5	50	50	0.8	0	
③	M/H ₂ /E ₄	1.5	50	100	0.5	0	*
④	M/H ₂ /E ₄	1.5	50	50	0.5	0	*
⑤	M/H ₂ /E ₄	1.5	50	50	0.5	0.5	

	$GI_0/G_1/G_2$	c_0	c_{H1}	$\frac{h_1}{msec}$	$\frac{h_2}{msec}$	q_{12}	q_{22}	Sim.
①	H ₂ /H ₂ /E ₄	1.5	1.5	50	50	0.5	0	*
②	M/H ₂ /E ₄	1.0	1.5	50	50	0.5	0	(*)
③	M/E ₄ /E ₄	1.0	0.5	50	50	0.5	0	*
④	M/M/M	1.0	1.0	50	50	0.5	0	

hierzu auch Abschnitt 5.4.6). Dennoch scheint die Genauigkeit selbst in diesen Fällen für die praktische Anwendung ausreichend zu sein.

6.5 Analyse des Steuerdatenflusses in rechnergesteuerten Nachrichtenvermittlungssystemen

Das abschließende Beispiel behandelt die Analyse hierarchisch gegliederter Steuerungsstrukturen, wie sie in rechnergesteuerten Nachrichtenvermittlungssystemen auftreten. Zur Analyse wird ein Zerlegungsverfahren angewandt, welches analog zu Abschnitt 5.4 auf der Charakterisierung der Schnittstellenprozesse durch zwei Parameter basiert.

6.5.1 Problemstellung

Neuere Systeme zur Nachrichtenvermittlung verwenden ausnahmslos das Prinzip der Rechnersteuerung (vergl. Abschnitt 2.2). Hierbei müssen die in der weitverbreiteten Peripherie (Teilnehmersätze, Wahlsätze, Koppelnetze, etc.) auftretenden Steuerinformationsanreize (Signale, Meldungen) über ein Signalsammelsystem erkannt, zwischengepuffert, modifiziert und in die Zentralsteuerung zur Verarbeitung übertragen werden. Umgekehrt müssen die Ergebnisse der Verarbeitung in Form von Befehlen von der Zentralsteuerung zu den peripheren Steuereinrichtungen transportiert werden. Dieser Steuerinformationsaustausch unterliegt außerordentlich strengen Forderungen hinsichtlich

- der Reaktions- oder Antwortzeiten
- des Durchsatzes
- des dynamischen Überlastverhaltens

und wird von einer Reihe von System- und Teilnehmereigenschaften stark beeinflusst (Steuerungsstruktur, Betriebsart, wiederholte Anrufversuche). Der relativ hohe Grundkostenanteil rechnergesteuerter Systeme hat außerdem zu einer Reihe unterschiedlicher Steuerungssysteme geführt, welche sich in Kosten, Leistung und Flexibilität recht deutlich unterscheiden.

Aus den vorgenannten Gründen, welche nur andeutungsweise die Komplexität der Problemstellung wiedergeben sollen, gewinnt die Systemanalyse eine zentrale Bedeutung in der Planung und Entwicklung derartiger Systeme. Zur Beantwortung der damit verbundenen Fragestellungen hat sich die Simulationstechnik sehr bewährt, da dies zur Zeit noch die beste Möglichkeit ist, komplexe Vorgänge verhältnismäßig wirklichkeitsnah zu beschreiben und zu analysieren. Analytische Methoden wurden und werden i.a. auf abgegrenztere Teilaspekte angewandt, wie etwa das Zentralsteuerwerk selbst oder die Ein/Ausgabe. Zur weiteren Diskussion sei auf das Schrifttum verwiesen [124, 139-145, 172-174].

Im folgenden wird ein Modell für den Steuerdatenfluß behandelt, welches dem in Abschnitt 3.3.1 beschriebenen Gesamtmodell im Prinzip entspricht. Zur Analyse wird hierbei ein erweitertes Zerlegungsverfahren angewandt, welches auf der

Charakterisierung der Schnittstellenprozesse zwischen den einzelnen Ebenen der hierarchischen Steuerungsstruktur beruht. Ziel der Untersuchungen ist es, die Verkehrsabläufe in einem Modell zu erfassen und zu analysieren, um den Zusammenhang zwischen System- und Verkehrsparametern herzustellen und daraus Erkenntnisse für den Systementwurf und die Planung abzuleiten.

6.5.2 Voraussetzungen

a) Struktur der Steuerung

Das vorausgesetzte rechnergesteuerte Vermittlungssystem habe prinzipiell dieselbe Struktur wie in Abschnitt 2.2 zugrundegelegt. Betrachtet werden für die spätere Modellbildung

- Periphere Signalquellen (Teilnehmer-, Wahl- und Verbindungssätze)
- Peripheres Leitungssystem
- Teilzentrale Steuer- und Informationsübertragungseinrichtungen
- Zentrale Steuereinrichtungen.

Die peripheren Signalquellen wie auch die teilzentralen Einrichtungen seien in Gruppen unterteilt.

b) Betriebsarten

In den verschiedenen Ebenen der Steuerung ist die Festlegung der Betriebsart von der jeweiligen Funktion her bestimmt. Im einzelnen werden folgende Voraussetzungen getroffen, welche überwiegend heutigen rechnergesteuerten Großvermittlungsstellen entsprechen:

- Zyklische Abfrage (Scan) der peripheren Signalquellen
- Zwischenpufferung von Signalen in der teilzentralen Ebene
- Taktgesteuerte Ein/Ausgabe (Clocked I/O) von Meldungen bzw. Befehlen zwischen teilzentralen Einrichtungen und dem Zentralsteuerwerk
- Ununterbrochene Verarbeitung von Meldungen zwischen den Ein/Ausgabephasen im Zentralsteuerwerk
- Abfertigung wartender Eingabemeldungen in der Zentralsteuerung nach unterschiedlichen Prioritäten.

Die angeführten Betriebsmerkmale gelten für den ungestörten Betrieb. Aus funktio-nellen Gründen (z.B. zur Bearbeitung von Alarmen, welche eine Fehlfunktion anzei-gen u.ä.m.) sind i.a. asynchrone Unterbrechungen zur Ein/Ausgabe oder unterbrechen-de Prioritätsdisziplinen zur schnellen Abwicklung von Fehleranalyseprogrammen vor-gesehen, diese Fälle treten im Verhältnis zum Normalbetrieb außergewöhnlich selten auf und sind nicht Gegenstand der Verkehrsanalyse.

c) Steuerinformations-Anreize

Der Verkehr innerhalb der Steuerung wird im wesentlichen durch Anreize peripherer Einrichtungen erzeugt (Teilnehmer-Zustandsänderungen, Wählinformationen, Zustands-änderungen von Leitungen, Steuerdaten zentraler Signalisierungskanäle). Analytisch kann die Gesamtheit der Steuerinformationsanreize einer Gruppe von peripheren Einrichtungen durch einen zufallsabhängigen Ankunftsprozeß beschrieben werden. Diese Beschreibung ist i.a. eine Idealisierung, indem die Abhängigkeiten zwischen einzelnen Anreizen (z.B. den Wählziffern eines bestimmten Anrufversuches) entweder vernachlässigt oder nur pauschal berücksichtigt werden. Mit zunehmender Anzahl tatsächlich unabhängiger Teilnehmer geht der Einfluß dieser Abhängigkeiten jedoch wieder zurück. Die Zulässigkeit muß im Zweifelsfalle durch wirklichkeitsnähere Modelle zur Erzeugung der Steuerdatenreize mit Hilfe der Simulation oder durch Messungen überprüft werden.

6.5.3 Systemmodell

Das System- oder Gesamtmodell besteht aus mehreren Teilmodellen, welche spezifische Funktionen ausführen. Betrachtet werde im folgenden der Steuerdatenfluß von der Peripherie zur Zentralsteuerung. Der Steuerdatenfluß der von der Zentralsteuerung ausgehenden Befehle verläuft weitgehendst parallel und unabhängig und habe kei-nen Einfluß auf das betrachtete Ablaufgeschehen; dieser kann in ähnlicher Weise behandelt werden wie im folgenden Beispiel für die Eingabe.

a) Gesamtmodell

Das Gesamtmodell (Eingabe) besteht aus drei Gruppen von Teilmodellen, vergl. Bild 6.15. Anreize der peripheren Signalquellen werden durch zyklische Abfrage erkannt und über das periphere Leitungssystem in Zwischenpuffer übertragen. War-tende Anforderungen in den Zwischenpuffern werden während der taktgesteuerten Ein/Ausgabephase des Zentralprozessors gruppenweise in die Zentralsteuerung über-tragen. Gleichzeitig erfolge hierbei eine Analyse des Meldungstyps hinsichtlich der Priorität und eine dementsprechende Einreihung in Prioritäts-Warteschlangen. Die drei unterschiedlichen Teilmodelle werden nachfolgend im einzelnen näher be-schrieben; zur Vereinfachung der Bezeichnung werden dabei die in Bild 6.15 ein-geführten Indices nur soweit benutzt wie sie zur eindeutigen Unterscheidung er-forderlich sind.

b) Teilmodell Signalsammelsystem

Jede periphere Gruppe ist durch ein Wartesystem mit mehreren Eingangs-Warteschlan-gen, einer Bedienungseinheit und zyklischer Abfertigung modelliert. Die "Bedie-nungszeiten" für Anforderungen der Warteschlange i bestehen aus zwei Teilen, der

Abfrage- und Umschaltzeit (Mittelwert u_j) sowie der Übertragungszeit im Falle einer Meldung (Mittelwert h_j), $j = 1, 2, \dots, g$. Bei jedem Abfragezeitpunkt einer Warteschlange werde höchstens eine Anforderung übertragen. Die Ankunftsprozesse der Anforderungen werden als Markoff-Prozesse (M) oder als Markoffsche Gruppenankunftsprozesse (MB) mit allgemeiner Verteilung der Gruppengröße beschrieben. Letztere Annahme erlaubt die Einbeziehung von gruppenförmig eintreffenden Meldungen bzw. eine Charakterisierung "spitzigerer" Ankunftsprozesse. Die Umschalt- bzw. Übertragungszeiten seien allgemeine Prozesse (G_{Uj} bzw. G_j). Das Teilmodell Signalsammelsystem ist somit ein Wartesystem vom Typ M/G/1 bzw. MB/G/1 mit zyklischer Abfertigung und Umschaltzeiten.

Es sei bemerkt, daß das Signalsammelsystem in größeren Systemen oft zweistufig realisiert ist; in diesen Fällen stellt das in Bild 6.15 skizzierte Teilmodell für das Signalsammelsystem die zweite (zentralisiertere) Stufe dar.

c) Teilmodell Signalpuffer- und Ein/Ausgabesystem

Meldungen der Peripherie werden in dieser Stufe zwischengepuffert und formatiert zur Übertragung in die Zentralsteuerung. Jede Gruppe dieser Zwischenebene läßt sich als Warteschlange (i.a. mit begrenzter Kapazität) darstellen, welche taktmäßig bedient wird. Bei jedem Taktzeitpunkt werden bis zu n Meldungen entnommen. Der Informationsaustausch mit der Zentralsteuerung erfolgt i.a. so schnell, daß die Übertragungszeit beim Abfragevorgang vernachlässigt werden kann. Das Teilmodell Signalpuffer- und Ein/Ausgabesystem ist somit ein getaktet abgefertigtes Warteverlustsystem des Typs GI/C/n-s. (Tatsächlich gehen i.a. keine Meldungen verloren, sondern werden im Blockierungsfalle zunächst wiederholt angeboten; die Modellierung als Warteverlustsystem erlaubt jedoch die Berechnung der Verlustwahrscheinlichkeit, welche bei kleinen Werten ($< 10^{-3}$) praktisch gleich der Blockierungswahrscheinlichkeit ist und somit zur passenden Dimensionierung der Pufferspeicherkapazität dient).

d) Teilmodell Zentralsteuerung

Die Zentralsteuerung stellt im allgemeinsten Falle ein Wartesystem mit getaktetem Eingangsprozeß und mehreren Ebenen unterbrechender und nichtunterbrechender Prioritäten dar. Je nach Arbeitsweise des Ein/Ausgabesystems muß die Ein/Ausgabezeit berücksichtigt werden. Bei kleineren Systemen läuft in der Ein/Ausgabephase ein Ein/Ausgabeprogramm ab, welches Laufzeiten in der Größenordnung der Vermittlungsprogramme hat und entsprechend berücksichtigt werden muß oder die Eingabe erfolgt überhaupt asynchron unterbrechungsgesteuert. Bei größeren Systemen kann die eigentliche Eingabe parallel zur Arbeit des Zentralprozessors erfolgen; in diesem

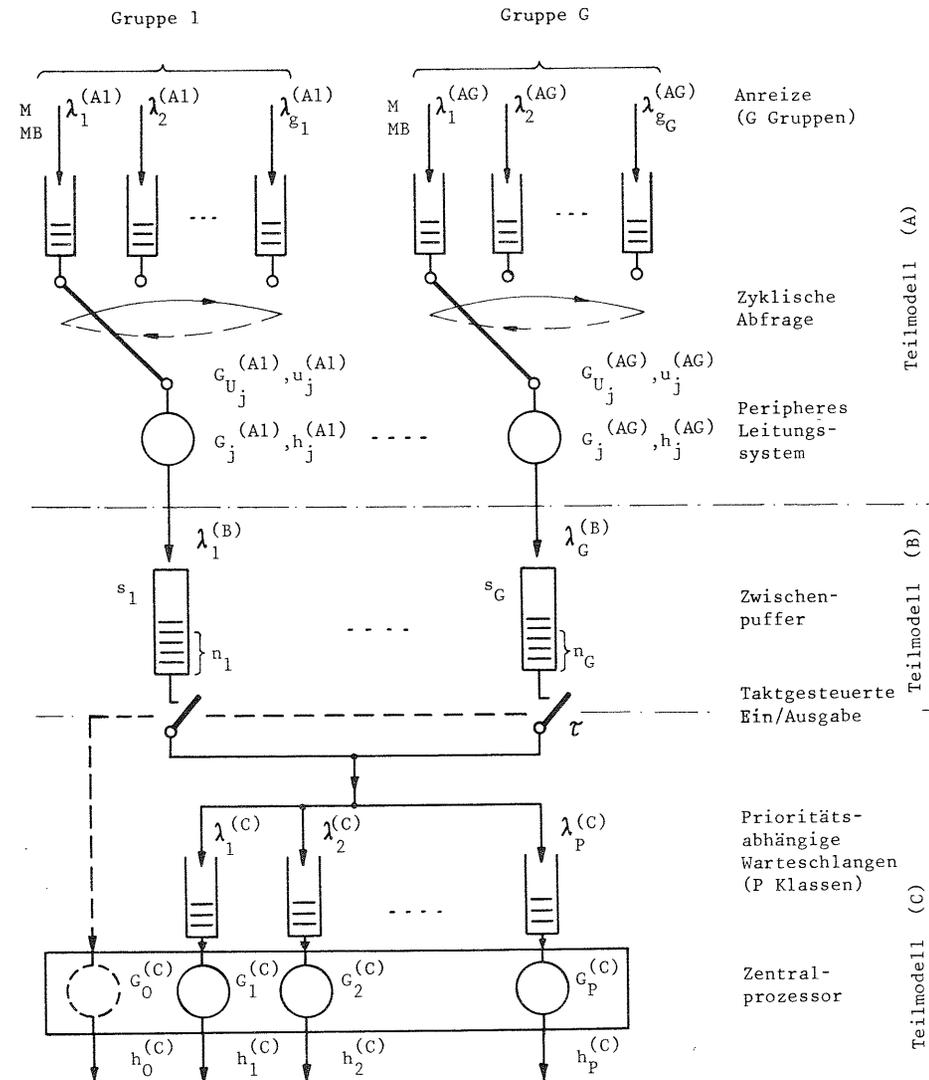


Bild 6.15. Gesamtmodell der Steuerung eines rechnergesteuerten Nachrichtensystems (Eingabe)

- Teilmodell (A): Signalsammelsystem
- Teilmodell (B): Signalpuffer- und Ein/Ausgabesystem
- Teilmodell (C): Zentralsteuerung

Fälle besitzen die teilzentralen Einrichtungen Direktzugriff zum Arbeitsspeicher (direct memory access), wobei die Zentralsteuerung lediglich bei Beginn und Ende der Eingabephase kurz unterbrochen wird.

Unter Annahme des letzteren Falles kann die Zentralsteuerung modelliert werden als Wartesystem CB/G/1 mit P Klassen nichtunterbrechender, unterbrechender oder einer Kombination aus beiden Arten von Prioritäten sowie Ein/Ausgabeverwaltung.

6.5.4 Modellanalyse

a) Methodik und Übersicht

Wie eingangs schon erwähnt, stellt die Simulation ein wichtiges Hilfsmittel dar zur Analyse komplexer Steuerungsvorgänge (Anwendungen vergl. [140, 144, 173]). Für umfangreichere Modelle kann die Simulation allerdings so aufwendig werden, daß entweder nur ein globaleres Modell oder nur Teilmodelle simuliert werden können. Analytische Verfahren sind hingegen i.a. auf einfachere Teilmodelle beschränkt, vermitteln dafür jedoch mehr prinzipielle Einsichten, lassen sich wirtschaftlicher auswerten und sind besser geeignet für die Optimierung.

Im folgenden wird der Versuch unternommen, mit Hilfe der Zerlegungsmethode das Gesamtmodell zu analysieren, wobei an den Schnittstellen der Teilmodelle die Verkehre in möglichst vielen Charakteristiken in Übereinstimmung gebracht werden. Die resultierenden Teilmodelle selbst sind i.a. bereits so komplex, so daß zu deren mathematischen Analyse neben exakten auch approximative Verfahren angewendet werden müssen. Die Ergebnisse werden schließlich überprüft mit Simulationen für Teilmodelle bzw. Aggregaten aus Teilmodellen.

Die mathematische Analyse der entstehenden Teilmodelle ist sehr umfangreich. Wir beschränken uns hier deshalb auf die Skizzierung des jeweiligen Lösungsweges, die prinzipiellen Ergebnisse (soweit sie für das vorliegende Problem benötigt werden) und Hinweise auf ausführlichere Darstellungen.

b) Teilmodell Signalsammelsystem

Zyklische Warteschlangenmodelle des Typs M/G/1 ohne und mit Umschaltzeiten sind ausführlich untersucht worden im Falle der vollständigen Abfertigung jeder Warteschlange, vergl. hierzu [136-140] und dort angegebene Literatur. Die Analyse bei beschränkter Anzahl von Abfertigungen je Warteschlange und Zyklus erweist sich als außerordentlich schwierig. Die einzige exakte Lösung wurde von Eisenberg [184] für den Fall zweier Warteschlangen ohne Umschaltzeiten und ≤ 1 Abfertigung je Warteschlange und Zyklus angegeben. Eine approximative Lösung auf der Basis einer

Unabhängigkeitsannahme wurde von Hashida/Ohara [138] für M/G/1-Systeme abgeleitet; diese Lösung unterschätzt jedoch prinzipiell die Varianz der Zykluszeiten und damit die mittleren Wartezeiten und liefert nur für größere Umschaltzeit/Bedienungszeit-Verhältnisse genauere Ergebnisse.

Durch eine verbesserte Zykluszeit-Analyse auf der Basis bedingter Zyklen wurde für den allgemeineren Fall von Wartesystemen des Typs MB/G/1 mit zyklischer Abfertigung und Umschaltzeiten mit Hilfe der Eingebetteten Markoff-Kette eine neue Lösung abgeleitet [140]. Die Analyse in [140] liefert u.a.

- (a) Allgemeine Stabilitätskriterien für Wartesysteme des Typs GI/G/1 mit allgemeiner zyklischer Abfertigungsdisziplin.
- (b) Mittelwert, Varianz und Laplace-Stieltjes Transformierte (LST) der Zykluszeiten für Wartesysteme des Typs GI/G/1 mit zyklischer Abfertigungsdisziplin und ≤ 1 Abfertigung pro Warteschlange und Zyklus.
- (c) Erzeugende Funktionen der Zustandswahrscheinlichkeiten zu Beginn der zyklischen Abfrage einer Warteschlange und bei Bedienungsende sowie die daraus abgeleitete mittlere Warteschlangenlänge für Wartesysteme BM/G/1 mit zyklischer Abfertigungsdisziplin und ≤ 1 Abfertigung pro Warteschlange und Zyklus.
- (d) Mittelwerte und LST der Wartezeiten im Falle (c).

Die Ergebnisse für die mittleren Wartezeiten sind:

$$w_{j1} = \frac{c_j^{(2)}}{2c_j} + \frac{\lambda_j c_j^{(2)}}{2(1-\lambda_j c_j)} + \frac{\lambda_j c_j^{(2)}}{2(1-\lambda_j c_j)} \left[\frac{E[K_j^2]}{E[K_j]} - 1 \right] \quad (6.24a)$$

$$w_{jk} = w_{j1} + (k-1) \cdot c_j, \quad k = 1, 2, \dots \quad (6.24b)$$

$$w_j = \frac{c_j^{(2)}}{2c_j} + \frac{\lambda_j c_j^{(2)}}{2(1-\lambda_j c_j)} + \frac{c_j^{(2)}}{2(1-\lambda_j c_j)} \left[\frac{E[K_j^2]}{E[K_j]} - 1 \right], \quad (6.24c)$$

wobei

- w_{jk} mittlere Wartezeit der k-ten Meldung innerhalb einer Gruppe
- w_j mittlere Wartezeit einer Meldung
- $c_j^{(n)}$ n-tes Moment der bedingten Zykluszeit, wenn keine Meldung in Warteschlange j abgefertigt wurde ($c_j^{(1)} = c_j$)
- $c_j^{(n)}$ n-tes Moment der bedingten Zykluszeit, wenn eine Meldung in Warteschlange j abgefertigt wurde ($c_j^{(1)} = c_j$)

λ_j Ankunftsrate von Meldungen in Warteschlange j
 $E[K_j^n]$ n-tes Moment der Gruppengröße in Warteschlange j,
 j = 1, 2, ..., g, n = 1, 2, ...

Die Berechnung der Zykluszeiten erfolgt approximativ unter einer Unabhängigkeitsannahme und führt auf geschlossene Lösungen bei symmetrischen Verkehrsparametern sowie unsymmetrischen Verkehrsparametern unterhalb einer Belastungsschranke [140].

Mit Hilfe der Zykluszeiten können bei symmetrischen Verkehrsparametern unter derselben o.a. Unabhängigkeitsannahme die höheren Momente des Ausgangsprozesses bestimmt werden. Die mittlere Zykluszeit für Wartesysteme GI/G/1 mit allgemeiner zyklischer Abfertigung ist [140]

$$c = \frac{c_0}{1-A}, \quad (6.25)$$

wobei

$c_0 = u_1 + u_2 + \dots + u_g$ minimale mittlere Zykluszeit
 u_j mittlere Umschaltzeit von Warteschlange j zur nachfolgenden Warteschlange
 g Anzahl der Warteschlangen
 $A = A_1 + A_2 + \dots + A_g$ Gesamtauslastung der Bedienungseinheit
 $A_j = \lambda_j \cdot h_j$ Auslastung der Bedienungseinheit durch Meldungen der Warteschlange (Gruppe) j
 h_j mittlere Bedienungszeit für Meldungen aus Warteschlange j.

Im Falle ≤ 1 Abfertigung pro Warteschlange und Zyklus trifft die Bedienungseinheit im Mittel mit Wahrscheinlichkeit

$$\alpha_j = \lambda_j c \quad (6.26)$$

eine nichtleere Warteschlange j an. In dem für die praktischen Anwendungen wichtigen Sonderfälle konstanter und gleicher Bedienungszeiten $h_j = h$ bzw. Umschaltzeiten $u_j = u$ bestimmt sich das k-te Moment der Ausgangsabstände T_D zu

$$E[T_D^k] = \sum_{i=1}^{\infty} \alpha (1-\alpha)^{i-1} \cdot (iu+h)^k, \quad k = 1, 2, \dots \quad (6.27)$$

wobei $\alpha = \frac{\lambda}{g} c$, λ = Gesamtankunftsrate. Insbesondere folgt hieraus für den Variationskoeffizienten c_D des Ausgangsprozesses

$$c_D^2 = (1-A) \cdot (1-A-\lambda u) \quad (6.28)$$

Gl.(6.28) gilt unter den gegebenen Voraussetzungen für symmetrische Wartesysteme des Typs GI/D/1 mit zyklischer Abfertigungsdisziplin und ≤ 1 Abfertigung pro Warteschlange und Zyklus. (Eine Verallgemeinerung auf den Fall GI/G/1 läßt sich in ähnlicher Weise durchführen). Da Gl.(6.28) für $u = 0$ das exakte Ergebnis für Wartesysteme des Typs M/D/1 einschließt, erwarten wir eine sehr gute Genauigkeit für Ankunftsprozesse mit $c_A \approx 1$. Ferner sind in den Grenzfällen $\lambda \rightarrow 0$ bzw. $\lambda = \lambda_{\max} = 1/(h+u)$ die richtigen Ergebnisse $c_D = 1$ bzw. $c_D = 0$ für Wartesysteme des Typs M/D/1 bzw. GI/D/1 mit zyklischer Abfertigungsdisziplin und konstanter Umschaltzeit enthalten. Das Ergebnis Gl.(6.28) wird in Abschnitt 6.5.5 mit Simulationen verglichen.

Abschließend sei nochmals daran erinnert, daß der Ausgangsprozeß i.a. nicht vollständig beschrieben ist durch die Ausgangsabstands-Charakteristik infolge der Abhängigkeiten zwischen aufeinanderfolgenden Ausgangsabständen. Die Annahme der Unabhängigkeit ist jedoch mit zunehmender Umschaltzeit besser gerechtfertigt. Mit der Gesamtankunftsrate λ und dem Variationskoeffizienten c_D liegen damit zwei Parameter des Eingangsprozesses für das Teilmodell Signalpuffer- und Ein/Ausgabesystem fest.

c) Teilmodell Signalpuffer- und Ein/Ausgabesystem

Für Warteverlustsysteme des Typs GI/C/n-s liegen analytische Ergebnisse in einer Reihe von Spezialfällen vor. Für reine Wartesysteme M/C/n wurden von Langenbach-Belz [141] Zustandswahrscheinlichkeiten, Mittelwerte und VF der Wartezeiten abgeleitet. Für Warteverlustsysteme des Typs M/C/1-s bzw. M/C/n-s wurden von Dor [185] bzw. Chu [186] die Zustandswahrscheinlichkeiten zu den Taktzeitpunkten und die Verlustwahrscheinlichkeit berechnet. Diese Lösungen wurden erweitert auf das Modell $E_k/C/n-s$ von Maritsas und Hartley [187]. Eine von Chu [186] angegebene Näherungsformel für die mittlere Wartezeit im Falle von M/C/n-s wurde in eigenen Untersuchungen [188] durch exakte Berechnung der Zustandswahrscheinlichkeiten zu beliebigen Zeitpunkten für den praktisch interessierenden Bereich der Verlustwahrscheinlichkeit $B < 0.01$ als sehr genau nachgewiesen. Eine Erweiterung der Analyse auf die Fälle $H_2/C/n-s$ bzw. MB/C/n-s ist prinzipiell möglich, so daß für das Modell GI/C/n-s ein hinreichendes Spektrum von Lösungen vorliegt. Das für die Anwendungen am wichtigsten erscheinende Grundmodell M/C/n-s wurde außerdem in umfangreichen Parameterkombinationen tabelliert [174, 175].

In allen Fällen der getakteten Abfertigung wurde die Analyse durch die Beschreibung mit Hilfe einer Eingebetteten Markoff-Kette durchgeführt. Regenerationspunkte des Zustandsprozesses sind die Taktzeitpunkte selbst (t_{i-0} bzw. t_{i+0}) [141, 185-188] oder verallgemeinert Zeitpunkte t_{i-t_V} mit einer festen Verschiebung

$t_v \leq \tau$ (τ = Taktdauer) zu den Taktzeitpunkten [141, 188], $i = 0, 1, 2, \dots$. Letztere Betrachtungsweise, welche von Langenbach-Belz [141] eingeführt wurde, erlaubt u.a. die exakte Berechnung der mittleren Warteschlangenlänge und wurde in [188] auf Warteverlustsysteme übertragen zur Nachprüfung der Näherungsformel von Chu, welche auf den Zustandswahrscheinlichkeiten zu den Taktzeitpunkten beruht. Für Warteverlustsysteme mit $1 < n < s$ erfordert die Auflösung der Zustandsgleichungssysteme die numerische Auswertung eines linearen Gleichungssystems. Aus den Zustandswahrscheinlichkeiten

$$p_k^* = \lim_{i \rightarrow \infty} P\{X(t_i - 0) = k\} \quad (6.29)$$

für den Prozeß X der Anzahl von Anforderungen im System kurz vor dem Taktzeitpunkt werden folgende charakteristische Größen gewonnen:

- Verlustwahrscheinlichkeit B
- mittlere Wartezeit w
- mittlere Warteschlangenlänge Ω
- mittlere Anzahl übertragener Meldungen pro Takt Y.

Aus der Zustandsverteilung p_k^* , $k = 0, 1, 2, \dots$, läßt sich ferner die Verteilung der Gruppengröße der pro Takt übertragenen Meldungen ableiten, welche den Ausgangsprozeß des Teilmodelles im wesentlichen charakterisiert (abgesehen von Abhängigkeiten zwischen aufeinanderfolgenden Gruppen). Die Verteilung der pro Takt übertragenen Meldungen ist

$$q_k = \begin{cases} p_k^* & , k = 0, 1, 2, \dots, n-1 \\ \sum_{i=n}^s p_i^* & , k = n \end{cases} \quad (6.30)$$

Mit der Verteilung der Gruppengröße und dem Taktabstand τ ist der Ausgangsprozeß wiederum durch zwei Merkmale charakterisiert. Die Approximation besteht hier in der Unabhängigkeitsannahme hinsichtlich aufeinanderfolgender Gruppengrößen.

d) Teilmodell Zentralsteuerung

Für das allgemeine Modell CB/G/1 mit Prioritäten ist keine analytische Lösung bekannt. Wir verweisen daher auf eine Reihe von vereinfachten oder ähnlichen Modellen, die exakt oder approximativ behandelt werden.

Eine erste Gruppe von Modellen des Typs CB/G/1 entsteht dadurch, wenn alle eintreffenden Meldungen als eine einzige Klasse behandelt werden. Exakte analytische Lösungen sind abgeleitet worden für CB/D/1 von Langenbach-Belz [141], für CB/M/1 von Schwärzel [143] sowie für CB/E_k/1 mit unbegrenztem oder begrenztem Warteraum unter Berücksichtigung der Systemverwaltungszeiten von Weisschuh [144, 145]. Eine approximative Lösung wurde von Krämer [113] entwickelt, welche auf einer bekannten exakten Beziehung zwischen Systemen mit und ohne Gruppenankunft beruht und die in Abschnitt 5.4 beschriebene Näherungslösung für Wartesysteme des Typs GI/G/1 verwendet. Die Ergebnisse dieses Modelles sind für den Fall CB/G/1 mit Poisson-verteilten Gruppengrößen ebenfalls tabelliert [174, 175].

Ein Modell des Typs CB/D/1 mit mehreren Klassen von Meldungen und gleicher Bedienungszeit pro Klasse wurde von Langenbach-Belz [141] behandelt. In diesem Modell werden die Meldungen innerhalb einer gesamten ankommenden Gruppe entsprechend ihrer Herkunftsrichtung geordnet und am Ende der Warteschlange vor dem Zentralprozessor eingereiht, welches einer gewissen Priorisierung der Herkunftsrichtungen entspricht. Ferner werden in dieser Arbeit außer den oben erwähnten einstufigen Grundmodellen M/C/n und CB/D/1 zweistufige Gesamtmodelle ausführlich behandelt.

Für den allgemeinen Fall des Modelles CB/G/1 mit Prioritäten steht die Lösung noch aus. Im folgenden wird ein modifiziertes Modell untersucht, welches sich von diesem nur dadurch unterscheidet, daß die synchronen Ankunftszeitpunkte aller Meldungsklassen nicht äquidistant sind, sondern in exponentiell verteilten Abständen erfolgen mit gleichem Mittelwert. Ferner sollen im allgemeinen Falle unterbrechende bzw. nichtunterbrechende Ein/Ausgabeverwaltungsphasen berücksichtigt werden. Das entstehende Modell $BM_{\text{SYNCHRON}}/G/1$ hat zusammengefaßt folgende Voraussetzungen:

- (1) P Prioritätsklassen von Meldungen
- (2) Synchrone Ankunftszeitpunkte aller Meldungsklassen
- (3) Allgemein verteilte Gruppengröße individuell je Meldungsklasse
- (4) Exponentiell verteilte Abstände zwischen den Ankunftszeitpunkten
- (5) Allgemein verteilte Bedienungszeiten individuell je Meldungsklasse
- (6) Endlich große und allgemein verteilte Ein/Ausgabeverwaltungszeiten
- (7) Unterbrechende oder nichtunterbrechende Prioritäten der Ein/Ausgabeverwaltungszeiten gegenüber den P Klassen von Meldungen
- (8) Nichtunterbrechende Prioritäten der Meldungsklassen.

Wir bemerken ergänzend zu diesen Voraussetzungen:

- Die Annahme exponentiell verteilter synchroner Ankunftsabstände resultiert in einer Überschätzung der Wartezeiten im Vergleich zum tatsächlichen Modell CB/G/1. Die Größenordnung dieser überschätzten Wartezeiten kann ermittelt werden durch Vergleiche der bekannten Modelle CB/G/1 und MB/G/1 ohne Prioritäten bzw. durch Simulationen. Da die sonstigen Voraussetzungen gleich sind, können wir im wesentlichen

dieselben Eigenschaften erwarten, insbesondere die Benachteiligung der niedrigeren Prioritätsklassen infolge der synchronen Ankunftszeitpunkte.

- Die beiden Fälle unterbrechender bzw. nichtunterbrechender Priorität für die Ein/Ausgabeverwaltung unterscheiden sich lediglich in den mittleren Wartezeiten der Ein/Ausgabe-"Unterbrechungsmeldungen" selbst sowie den mittleren Reaktionszeiten der P Meldungsklassen, jedoch nicht in den mittleren Wartezeiten der P Meldungsklassen.
- Die nachfolgend abgeleiteten Ergebnisse lassen sich verhältnismäßig einfach erweitern auf den allgemeineren Fall einer Kombination aus unterbrechenden und nichtunterbrechenden Prioritäten zwischen den P Meldungsklassen.

Zur Analyse des o.a. Modelles $BM_{\text{SYNCHRON}}/G/1$ mit Prioritäten und Ein/Ausgabeverwaltung wird im folgenden die Momentenmethode angewandt in analoger Weise wie für die Fälle von $M/G/1$ -Wartesystemen mit asynchronen Ankünften und nichtunterbrechenden Prioritäten (Cobham [53], Kosten [58]) oder Unterbrechungs-Distanz-Prioritäten (Herzog [20]). Aus der erhaltenen Lösung wird weiter unten eine approximative Lösung für den tatsächlichen Fall $CB/G/1$ angegeben. Bei der Analyse der Wartezeiten der beiden unterschiedlichen Modelle können wir uns auf den Fall unterbrechender Ein/Ausgabe beschränken (siehe obige Bemerkung). Der Fall nichtunterbrechender Ein/Ausgabe kann entweder direkt in analoger Weise abgeleitet werden oder wird erhalten als Sonderfall des Falles unterbrechender Ein/Ausgabe; hierzu wird $h_0 = 0$ gesetzt und es werden $P+1$ Klassen betrachtet, wobei die Meldungen der ersten Klasse mit konstanter Gruppengröße 1 eintreffen und als Ein/Ausgabe-Anforderungen interpretiert werden. Ferner ist der Fall verschwindender Ein/Ausgabezeiten enthalten ($h_0 = 0$).

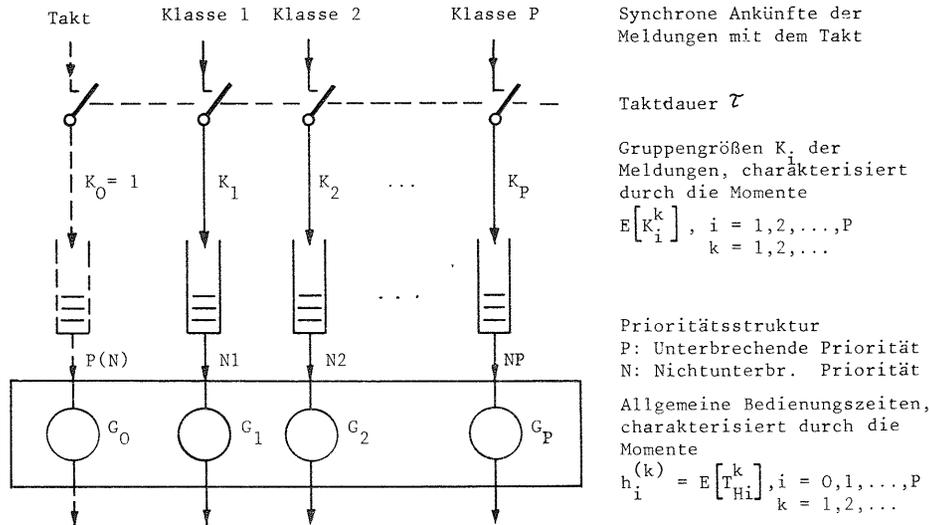


Bild 6.16. Modell der Zentralsteuerung

Die Analyse erfolgt in folgenden Schritten:

- 1) Bestimmung der mittleren Wartezeit w_0 der Klasse 0
- 2) Bestimmung der mittleren Wartezeit w_{i1} des jeweils Gruppenersten der allgemeinen Klasse i ($i \geq 1$)
- 3) Bestimmung der mittleren Wartezeit w_{ik} der k -ten Meldung innerhalb einer Gruppe der Klasse i
- 4) Bestimmung der mittleren Wartezeit w_i einer beliebigen Meldung der Klasse i .

In den Schritten 2-4 wird jeweils eine Wartezeitbilanz aufgestellt durch Betrachtung aller Möglichkeiten des Schicksals einer Meldung. Die entstehenden Bilanzgleichungen werden rekursiv für $i = 1, 2, \dots, P$ aufgelöst, woraus eine geschlossene Lösung für die jeweiligen mittleren Wartezeiten folgt.

Schritt 1:

Klasse 0 hat unterbrechende Priorität und kann deshalb unabhängig von allen anderen Klassen betrachtet werden. Die mittlere Wartezeit w_0 bestimmt sich aus dem entstehenden $M/G_0/1$ Modell zu

$$w_0 = \frac{A_0 \cdot h_0^{(2)}}{1 - A_0} \quad (6.31)$$

mit $A_0 = \lambda_0 h_0$, $\lambda_0 = 1/\tau$.

Schritt 2:

Die mittlere Wartezeit w_{i1} des Gruppenersten bis zum ersten Bedienungsbeginn setzt sich wie folgt zusammen aus fünf Teilen:

$$w_{i1} = \left\{ \begin{array}{l} \sum_{\nu=0}^P A_\nu \frac{h_\nu^{(2)}}{2h_\nu} \\ + \sum_{\nu=1}^P \left(\frac{A_0}{1-A_0} A_\nu \right) \cdot \frac{h_\nu^{(2)}}{2h_\nu} \\ + (\lambda_0 w_0 + 1) h_0 \\ + \sum_{\nu=1}^i (\lambda_\nu w_\nu h_\nu) \\ + \sum_{\nu=1}^{i-1} E[K_\nu] h_\nu \end{array} \right. \left\{ \begin{array}{l} \text{Restbedienungszeit einer zum Ankunftszeitpunkt bestehenden} \\ \text{Bedienung nach der Erneuerungstheorie, wobei} \\ A_\nu = \lambda_\nu h_\nu, \lambda_0 = 1/\tau, \lambda_\nu = \lambda_0 E[K_\nu] \cdot h_\nu, \nu = 0, 1, \dots, P. \\ \text{Restbedienungszeit für die mittlere Anzahl } \frac{A_0}{1-A_0} \cdot A_\nu \text{ unter-} \\ \text{brochener Meldungen der Klassen } \nu = 1, 2, \dots, P. \\ \text{Gesamtbedienungszeit für die mittlere Anzahl angetroffener} \\ \text{wartender "Unterbrechungsmeldungen" } \lambda_0 w_0 \text{ nach Little's Ge-} \\ \text{setz sowie der mit dem Gruppenersten eintreffenden Unter-} \\ \text{brechungsmeldung.} \\ \text{Gesamtbedienungszeit für die mittlere Anzahl angetroffener} \\ \text{wartender Meldungen der Klassen } 1, 2, \dots, i \text{ sowie der mit dem} \\ \text{Gruppenersten eintreffenden Meldungen der Klassen } 1, 2, \dots, i-1 \end{array} \right.$$

$$+ \sum_{\nu=0}^{i-1} (\lambda_{\nu} w_{i1}) h_{\nu} \left\{ \begin{array}{l} \text{Gesamtbedienungszeit aller nachfolgend w\u0304hrend } w_{i1} \text{ ein-} \\ \text{treffenden Unterbrechungsmeldungen bzw. Meldungen der} \\ \text{Klassen } 1, 2, \dots, i-1, \text{ welche vor der betrachteten Meldung} \\ \text{abgefertigt werden.} \end{array} \right.$$

Hieraus folgt schlie\u00dflich die Beziehung

$$w_{i1} (1 - A_{\leq i-1}) = \sum_{\nu=0}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}} + (\lambda_0 w_0 + 1) h_0 + \sum_{\nu=1}^{i-1} (A_{\nu} w_{\nu} + h_{\nu} E[K_{\nu}]) + A_i w_i + \frac{A_0}{1 - A_0} \cdot \sum_{\nu=1}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}}, \quad (6.32)$$

wobei $A_{\leq i-1} = A_0 + A_1 + \dots + A_{i-1}$.

Zur weiteren Bestimmung von w_{i1} mu\u00df zun\u00e4chst der Zusammenhang zwischen den mittleren Wartezeiten des Gruppenersten und einer beliebigen Meldung hergestellt werden.

Schritt 3:

Die k-te Meldung innerhalb einer Gruppe der Klasse i erleidet die mittlere Wartezeit

$$w_{ik} = w_{i1} \left\{ \begin{array}{l} \text{mittlere Wartezeit des Gruppenersten} \\ \\ \sum_{\nu=0}^{i-1} (w_{ik} - w_{i1}) A_{\nu} \left\{ \begin{array}{l} \text{Gesamtbedienungszeit f\u00fcr die mittlere Anzahl eintreffender} \\ \text{Unterbrechungsmeldungen bzw. Meldungen der Klassen } \leq i-1 \\ \text{w\u0304hrend der Zeit zwischen dem Bedienungsbeginn des Gruppen-} \\ \text{ersten und dem der betrachteten Meldung.} \end{array} \right. \end{array} \right.$$

Hieraus folgt:

$$w_{ik} = w_{i1} + \frac{(k-1)h_i}{1 - A_{\leq i-1}} \quad (6.33)$$

Schritt 4:

Die mittlere Wartezeit w_i einer beliebigen Meldung der Klasse i folgt durch Mittelung \u00fcber alle w_{ik} unter Ber\u00fccksichtigung der Verteilung der Gruppengr\u00f6\u00dfe f\u00fcr Klasse i:

$$w_i = \sum_{k=1}^{\infty} \frac{k q_{ik}}{E[K_i]} \cdot \frac{1}{k} \cdot \sum_{\nu=1}^k w_{i\nu} = w_{i1} + \frac{h_i}{2(1 - A_{\leq i-1})} \cdot \left[\frac{E[K_i^2]}{E[K_i]} - 1 \right]. \quad (6.34)$$

In Gl.(6.34) ist zu beachten, da\u00df die betrachtete Meldung mit Wahrscheinlichkeit $k \cdot q_{ik} / E[K_i]$ in einer Gruppe der Gr\u00f6\u00dfe k eintraf (siehe hierzu z.B. [189]), wobei q_{ik} , $k = 0, 1, 2, \dots$ die Verteilung der Gruppengr\u00f6\u00dfe f\u00fcr Klasse i. Die betrachtete

Meldung ist gleichwahrscheinlich an erster, zweiter, ..., k-ter Stelle innerhalb der Gruppe eingetroffen.

Mit Gl.(6.34) ist zugleich der Zusammenhang zwischen den mittleren Wartezeiten w_{i1} und w_i hergestellt. Die Gleichungen (6.32) und (6.34) werden rekursiv gel\u00f6st f\u00fcr $i = 1, 2, \dots, P$, woraus schlie\u00dflich mit Gl.(6.31) folgt:

$$w_{i1} = \frac{\sum_{\nu=0}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}} + \sum_{\nu=0}^{i-1} (1 - A_{\leq \nu}) E[K_{\nu}] h_{\nu} + \sum_{\nu=1}^i A_{\nu} \frac{h_{\nu}}{2} \left[\frac{E[K_{\nu}^2]}{E[K_{\nu}]} - 1 \right]}{(1 - A_{\leq i-1})(1 - A_{\leq i})}, \quad i = 1, 2, \dots, P. \quad (6.35)$$

Mit Gl.(6.35) und Gl.(6.33), (6.34) liegen somit auch die mittleren Wartezeiten w_{ik} und w_i fest. Das Ergebnis f\u00fcr w_i lautet

$$w_i = \frac{\sum_{\nu=0}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}} + \sum_{\nu=0}^{i-1} (1 - A_{\leq \nu}) E[K_{\nu}] h_{\nu} + \sum_{\nu=0}^{i-1} A_{\nu} \frac{h_{\nu}}{2} \left[\frac{E[K_{\nu}^2]}{E[K_{\nu}]} - 1 \right] + (1 - A_{\leq i-1}) \frac{h_i}{2} \left[\frac{E[K_i^2]}{E[K_i]} - 1 \right]}{(1 - A_{\leq i-1})(1 - A_{\leq i})}, \quad i = 1, 2, \dots, P. \quad (6.36)$$

mit $E[K_0^k] = 1$, $k = 1, 2, \dots$. Die Ergebnisse nach Gl.(6.35) und Gl.(6.36) gelten sowohl f\u00fcr unterbrechende als auch f\u00fcr nichtunterbrechende Ein/Ausgabe.

Im Falle nichtunterbrechender Ein/Ausgabe erhalten wir anstelle von Gl.(6.31)

$$w_0 = \frac{\sum_{\nu=0}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}}}{1 - A_0} \quad (6.37)$$

Die mittleren Antwort- oder Reaktionszeiten f\u00fcr Meldungen der Klasse i bestimmen sich schlie\u00dflich zu

$$r_i = w_i + h_i \quad (\text{nichtunterbrechende Ein/Ausgabe}) \quad (6.38a)$$

$$r_i = w_i + \frac{h_i}{1 - A_0} \quad (\text{unterbrechende Ein/Ausgabe}), \quad (6.38b)$$

$i = 1, 2, \dots, P$.

Eine weitere Modellvariante entsteht dadurch, da\u00df die Synchronisationsbedingung zwischen den Ankunftszeitpunkten fallengelassen wird; das entstehende Modell MB/G/1 mit nichtunterbrechenden Priorit\u00e4ten liefert generell niedrigere Wartezeiten im Vergleich zu synchronisierten Ankunftszeitpunkten und verliert den

funktionellen Zusammenhang zwischen den einzelnen Meldungsklassen und dem E/A-Takt. Für den speziellen Fall BM/D/1 wurde von Herzog [20] im Zusammenhang mit der Paketierung von Nachrichten in Datenvermittlungen eine Lösung abgeleitet. Durch eine leichte Verallgemeinerung dieser Lösung erhalten wir für die mittlere Wartezeit von Meldungen der Klasse i

$$w_i = \frac{\sum_{\nu=0}^P A_{\nu} \frac{h_{\nu}^{(2)}}{2h_{\nu}} + \sum_{\nu=0}^{i-1} A_{\nu} \frac{h_{\nu}}{2} \left[\frac{E[K_{\nu}^2]}{E[K_{\nu}]} - 1 \right] + (1 - A_{\leq i-1}) \frac{h_i}{2} \left[\frac{E[K_i^2]}{E[K_i]} - 1 \right]}{(1 - A_{\leq i-1})(1 - A_{\leq i})}, \quad (6.39)$$

$i = 1, 2, \dots, P.$

Die Differenz in den mittleren Wartezeiten zwischen diesem Modell und dem Modell mit synchronisierten Ankunftszeitpunkten beträgt

$$\Delta w_i = w_i \Big|_{N \text{ SYN}} - w_i \Big|_{N \text{ ASYN}} = \frac{\sum_{\nu=0}^{i-1} (1 - A_{\leq \nu}) E[K_{\nu}] h_{\nu}}{(1 - A_{\leq i-1})(1 - A_{\leq i})} > 0, \quad (6.40)$$

$i = 1, 2, \dots, P.$

Zur Anwendung der abgeleiteten Ergebnisse verbleibt schließlich noch die Bestimmung der Verteilung der eintreffenden Gruppengrößen insgesamt und je Prioritätsklasse. Voraussetzung hierfür ist die Kenntnis der Verteilung der Anzahl übertragener Meldungen je Pufferspeicher, welche mit Gl. (6.30) festliegt. Die Verteilung der Gruppengröße von Pufferspeicher j sei $q_{jk}^{(B)}$, $k = 0, 1, 2, \dots$, $j = 1, 2, \dots, G.$

In einem ersten Schritt wird die Verteilung $q_r^{(C)}$ der resultierenden Gesamtgruppengröße K der je Taktzeitpunkt in die Zentralsteuerung übertragenen Meldungen bestimmt. Die Gesamtverteilung bestimmt sich durch eine Faltungsoperation aus den Einzelverteilungen entsprechend

$$P\{K = r\} = q_r^{(C)} = \sum_{i_1+i_2+\dots+i_G=r} q_{1i_1}^{(B)} q_{2i_2}^{(B)} \dots q_{Gi_G}^{(B)}. \quad (6.41)$$

Die Faltungsoperation wird im allgemeinen Falle zweckmäßigerweise rekursiv in G-1 Einzelschritten durch Zusammenfassung jeweils zweier Verteilungen durchgeführt (siehe [141]). Je nach Struktur der Einzelverteilungen kann es unter Umständen günstiger sein, eine Transformationsmethode anzuwenden. Betrachtet werde hierzu die erzeugende Funktion der Verteilung $q_{jk}^{(B)}$:

$$G_j^{(B)}(z) = \sum_{k=0}^{\infty} q_{jk}^{(B)} \cdot z^k, \quad j = 1, 2, \dots, G. \quad (6.42)$$

Die erzeugende Funktion der Gesamtverteilung ist

$$G^{(C)}(z) = \sum_{r=0}^{\infty} q_r^{(C)} \cdot z^r = \prod_{j=1}^G G_j^{(B)}(z), \quad (6.43)$$

woraus die Verteilung und deren ersten beiden Momente abgeleitet werden können entsprechend

$$q_r^{(C)} = \frac{1}{r!} \cdot \frac{d^r}{dz^r} G^{(C)}(z) \Big|_{z=0}, \quad r = 0, 1, 2, \dots \quad (6.44)$$

$$E[K] = G'(z) \Big|_{z=1}^{(C)} \quad (6.45a)$$

$$E[K^2] = G''(z) \Big|_{z=1}^{(C)} + E[K]^2. \quad (6.45b)$$

Mit der Gesamtverteilung der Gruppengröße und der Taktdauer τ liegt somit der Eingangsprozeß für das Modell der Zentralsteuerung im Falle ohne Prioritäten fest.

Im Falle von P Klassen unterschiedlicher Prioritäten muß nun noch die Verteilung der Gruppengröße je Prioritätsklasse bestimmt werden. Je nachdem, ob die Häufigkeitsanteile der einzelnen Klassen unterschiedlich je Pufferspeichergruppe sind oder nicht, muß die folgende Operation auf den Ausgangsprozeß jedes einzelnen Pufferspeichers angewandt werden oder es genügt, diese Operation nur auf den Gesamt-Eingangsprozeß, d.h. auf die Verteilung $q_r^{(C)}$, anzuwenden. Die Zerlegungsoperation in einzelne Prioritätsklassen werde im folgenden am Beispiel des Gesamt-Eingangsprozesses durchgeführt.

Der Häufigkeitsanteil der Prioritätsklasse i sei vorgegeben:

$$y_i = \frac{\lambda_i}{\lambda}, \quad i = 1, 2, \dots, P, \quad (6.46)$$

wobei $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_P.$ Unter der Annahme der Unabhängigkeit folgt zunächst die bedingte Verteilung der Gruppengröße K_i für Klasse i

$$P\{K_i = k | K = r\} = \binom{r}{k} y_i^k (1 - y_i)^{r-k}, \quad k = 0, 1, \dots, r. \quad (6.47)$$

Mit Gl. (6.47) und der Verteilung der Gesamtgruppengröße folgt schließlich

$$q_{ik} = P\{K_i = k\} = \sum_{r=0}^{\infty} q_r^{(C)} \cdot \binom{r}{k} \rho_i^k (1-\rho_i)^{r-k} \quad (6.48)$$

Aus Gl. (6.48) bestimmen sich die Momente der Gruppengröße entsprechend

$$E[K_i^y] = \sum_{k=0}^{\infty} k^y \cdot q_{ik} \quad (6.49)$$

Insbesondere folgt für die ersten beiden Momente

$$E[K_i] = \rho_i E[K] = \frac{\lambda_i}{\lambda} E[K] = \lambda_i \tau, \quad (6.50a)$$

$$E[K_i^2] = \rho_i (1-\rho_i) E[K] + \rho_i^2 E[K^2]. \quad (6.50b)$$

Damit liegen sämtliche Parameter des Modelles Zentralsteuerung fest.

6.5.5 Ergebnisse und Simulationsvergleiche

Im folgenden werden zwei einfache Beispiele der Analyse des Steuerdatenflusses durchgeführt. Teilsystemergebnisse werden abschließend durch Simulationsvergleiche in ihrer Güte überprüft. Ferner werden die Frage der Optimierung der E/A-Taktdauer τ sowie Möglichkeiten einer Überlaststeuerung diskutiert.

a) Steuerdatenflüsse und Durchlaufzeiten

Betrachtet werde das Modell nach Bild 6.15 mit insgesamt $G = 10$ peripheren Gruppen. Jedes der zyklischen Signalsammelsysteme bediene $g = 20$ Teilgruppen. Die Parameter des Modelles seien wie folgt festgelegt:

(A) Signalsammelsystem:

Markoffsche Ankünfte, Rate je Teilgruppe	$\lambda_j^{(A)} = 0 \dots 2/\text{sec}, j = 1, 2, \dots, 20$
konstante Abfrage- und Umschaltzeit	$u^{(A)} = 2.5 \text{ msec}$
konstante Übertragungszeit	$h^{(A)} = 2.5 \text{ msec}$

(B) Signalpuffer- und Ein/Ausgabesystem:

Ankünfte je Gruppe	$\lambda_j^{(B)} = 0 \dots 40/\text{sec}$
max. Anzahl übertragener Meldungen/Takt	$n_j = 1, j = 1, 2, \dots, 10$
Taktdauer	$\tau = 10 \text{ msec}$

(C) Zentralsteuerung

	<u>Beispiel 1</u>	<u>Beispiel 2</u>
Konstante Ein/Ausgabebehandlungszeit	$h_0 = 0$	$h_0 = 2 \text{ msec}$
Anzahl der Prioritätsklassen	$P = 1$	$P = 2$
Aufteilung der Meldungen auf Klassen	-	$\rho_1 = \rho_2 = 0.5$
mittl. Bedienungszeiten für Meldungen	$h_1 = 2 \text{ msec}$	$h_i = 2 \text{ msec}, i = 1, 2$
Bedienungszeit-Charakteristiken	D	E_2

Beispiel 1

In Beispiel 1 wird nur eine gemeinsame Klasse von Meldungen zugrundegelegt. Ferner benötigt die Ein/Ausgabe (idealisiert) keine Systemverwaltungszeit. Durch Anwendung der im vorausgehenden Abschnitt entwickelten Methodik erhalten wir die Durchlaufzeitanteile der drei Teilmodelle (A), (B) und (C) sowie die Gesamtdurchlaufzeit einer Meldung gemessen von ihrem Antreffzeitpunkt in der Warteschlange des Teilmodelles (A) bis zum Bedienungsende in Teilmodell (C), vergl. Bild 6.17. Wir bemerken, daß unter den zugrundegelegten Parametern die einzelnen Teilmodelle unterschiedliche Sättigungspunkte erreichen:

Teilmodell (A)	bei $\lambda_m^{(A)} = 0.2$ Meldungen/msec und Gruppe
Teilmodell (B)	bei $\lambda_m^{(B)} = 0.1$ Meldungen/msec und Gruppe
Teilmodell (C)	bei $\lambda_m^{(C)} = 0.05$ Meldungen/msec und Gruppe

Durch die unterschiedlichen Sättigungspunkte bedingt, ergibt sich eine nahezu ideale Gesamtdurchlaufzeit-Charakteristik. Das System kann theoretisch bis zu $\lambda = 0.48$ Meldungen je msec verarbeiten, wobei die Gesamtdurchlaufzeit unterhalb von 60 msec bleibt. Bevor dieser Punkt erreicht wird (entweder durch dauernde oder statistisch schwankende Überlast), muß eine Überlaststeuerungsmaßnahme ergriffen werden, welche die Zentralsteuerung vor einer weiteren Zunahme der Meldungsrate und der damit verbundenen stark anwachsenden Wartezeiten schützt (s.u.).

Zur Analyse der Teilmodelle sei ergänzend bemerkt: Das Teilmodell (A) ist nur sehr schwach ausgelastet bei Berücksichtigung der Kapazitätsgrenze für die Zentralsteuerung. Der Ausgangsprozess des Teilmodelles (A) ist aus diesem Grunde nur leicht hypoexponentiell ($0.85 \leq c_p \leq 1.0$) und wurde vereinfachend als Markoff-Prozeß approximiert, wodurch für Teilmodell (B) tabellierte Ergebnisse des

Modelles M/C/n-s verwendet werden konnten [175]. Unter Verwendung der Tabellen kann gleichzeitig die Signalpufferspeicherkapazität dimensioniert werden: Zur Einhaltung einer Überlaufwahrscheinlichkeit $B \leq 10^{-6}$ sind $s \geq 9$ Pufferspeicherplätze erforderlich. Die Überlagerung der einzelnen Ausgangsprozesse der Pufferspeicher führt auf eine abgeschnittene Gruppenverteilung $q_r^{(C)}$, $r = 0, 1, \dots, 10$. Das Ergebnis des Teilmodelles (C) wurde mit Hilfe dieser Verteilung und den Ergebnissen des Modelles CB/G/1 (für $G = D$) nach [113] ermittelt. Gestrichelt eingezeichnet ist das Ergebnis, wenn anstelle dieser Verteilung eine Poisson-Verteilung angenommen wird; in diesem Falle konnten wiederum tabellierte Ergebnisse [175] angewandt werden. Die Poisson-Verteilung hat keine Begrenzung der Gruppengröße; durch die größere Streuung der Gruppengröße bedingt, stellt sich eine leicht überhöhte Wartezeit ein.

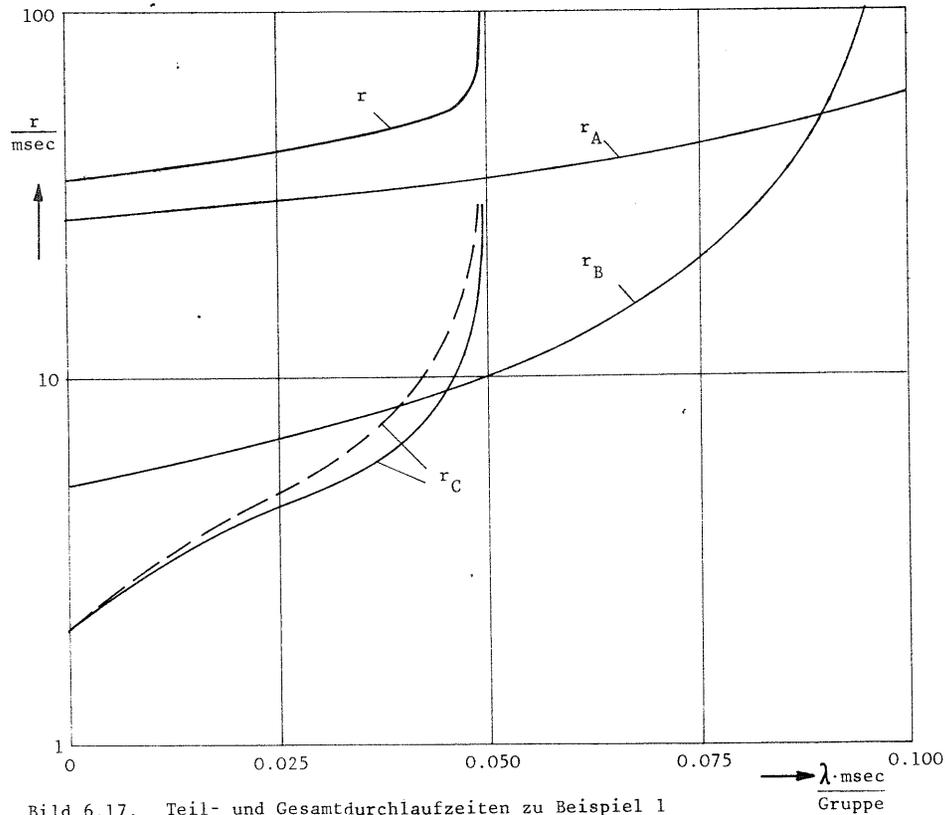


Bild 6.17. Teil- und Gesamtdurchlaufzeiten zu Beispiel 1
 r_A, r_B, r_C Durchlaufzeiten der Teilmodelle (A), (B) und (C)
 r Gesamtdurchlaufzeit

Beispiel 2

Beispiel 2 unterscheidet sich von Beispiel 1 dadurch, daß die Ein/Ausgabe mit einem Verwaltungsaufwand behaftet ist sowie 2 Prioritätsklassen von Meldungen unterschieden werden. Wir können uns hier daher auf die Analyse des Teilmodells (C) beschränken. Teilmodell (C) ist ein Wartesystem des Typs CB/G/1 mit 2 Klassen nichtunterbrechender Prioritäten und konstanter Ein/Ausgabeverwaltungszeit. Da für dieses Modell keine exakten Ergebnisse bekannt sind, verwenden wir das in Abschnitt 6.5.4 untersuchte Modell $MB_{\text{SYNCHRON}}/G/1$, welches sich nur in der Verteilungsfunktion der synchronen E/A-Takte unterscheidet. Wir erwarten von diesem Modell dieselben Eigenschaften mit dem einzigen Unterschied höherer absoluter Werte für die Wartezeiten.

Bild 6.18 zeigt die mittleren Wartezeiten für Meldungen beider Prioritätsklassen sowie im Vergleich dazu das entsprechende Ergebnis ohne Unterscheidung von Prioritäten. Zunächst bemerken wir, daß die Zentralsteuerung infolge der E/A-Verwaltungszeiten bereits bei $\lambda_m^{(C)} = 0.4$ Meldungen/msec die Sättigung erreicht. Während die Wartezeiten für die höhere Priorität beschränkt bleiben, steigen die Wartezeiten der niedrigeren Priorität stark an. Der Vorteil der Prioritätsteilung wird insbesondere deutlich im Hinblick auf die Reaktionszeiten auf dringlichere Meldungen im Überlastfalle.

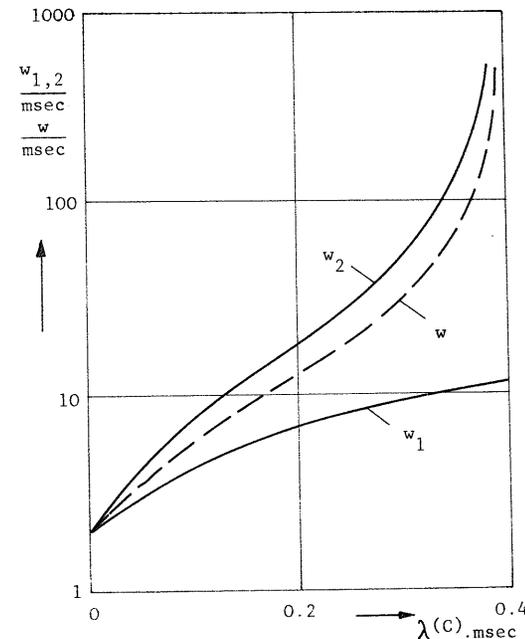


Bild 6.18.
 Mittlere Wartezeiten des Teilmodelles (C) Zentralsteuerung mit Ein/Ausgabeverwaltung
 — 2 Prioritätsklassen
 - - - 1 Prioritätsklasse

Zur Analyse sei ergänzend bemerkt, daß in allen Fällen die sich tatsächlich ergebende Ausgangsverteilung der Gruppengröße aus Teilmodell (B) zugrundegelegt wurde entsprechend den in Abschnitt 6.5.4 beschriebenen Überlagerungs- und Prioritätsaufteilungsverfahren. Die überhöhten Wartezeiten des Modelles $MB_{\text{SYNCHRON}}/G/1$ können approximativ korrigiert werden durch analoge Anwendung der in Abschnitt 4.6.6 beschriebenen heuristischen "Ähnlichkeitseigenschaft" verschiedener Wartesysteme, welche sich jeweils nur in einem Parameter unterscheiden. Wir betrachten hierzu die mittleren Wartezeiten folgender vier Systeme durch Ansatz der Gleichung

$$\frac{w_i \left| \begin{array}{l} \text{CB/G/1} \\ \text{nichtunterbr.Pr.} \end{array} \right.}{w_i \left| \begin{array}{l} \text{MB}_{\text{SYNCHRON}}/G/1 \\ \text{nichtunterbr.Pr.} \end{array} \right.} = \frac{w \left| \begin{array}{l} \text{CB/G/1, } A = A_{\leq i} \end{array} \right.}{w \left| \begin{array}{l} \text{MB/G/1, } A = A_{\leq i} \end{array} \right.}, \quad i = 1, 2, \dots, P. \quad (6.51)$$

Die Ergebnisse der drei Vergleichssysteme $CB/G/1$, $MB/G/1$ und $MB_{\text{SYNCHRON}}/G/1$ mit nichtunterbrechenden Prioritäten sind approximativ bzw. exakt bekannt. Es sei ausserdem bemerkt, daß in allen 4 Fällen dieselbe Gruppengrößenverteilung anzuwenden ist sowie ferner, daß zur Berücksichtigung des Einflusses der Klassen $< i$ auf die Klasse i das Angebot $A_{\leq i}$ in den entsprechenden Systemen ohne Prioritäten einzusetzen ist. Das Verfahren liefert erstaunlich gute Ergebnisse, vergl. hierzu Abschnitt 6.5.5c).

b) Optimierung und Überlaststeuerung

Bei nichtverschwindender Ein/Ausgabeverwaltungszeit entsteht ein Optimierungsproblem im Zusammenhang mit der Festlegung der Ein/Ausgabetaktdauer τ : Ist τ nahe der Ein/Ausgabeverwaltungszeit h_0 , so steht nur ein geringer Anteil der Realzeit-Kapazität zur Verfügung, wodurch der Durchsatz drastisch abfällt bzw. die Wartezeiten stark ansteigen; ist umgekehrt $\tau \gg h_0$, so wird zwar der bestmögliche Durchsatz erzielt, die Reaktionszeiten der Zentralsteuerung steigen jedoch trotzdem an infolge der größeren Pufferzeit und der zunehmenden Gruppengröße. Ein Problem des Systementwurfs besteht in einem ausgewogenen Kompromiß zwischen Reaktionszeit und Durchsatz.

Probleme der Optimierung unter Einbeziehung des Ein/Ausgabeverwaltungsaufwandes wurden von Langenbach-Belz [142] und Weisschuh [145] behandelt. Die Untersuchungen in [142] bzw. [145] beruhen auf einem Modell des Typs $CB/D/1$ bzw. $CB/E_k/1$ -s. In [142] wird insbesondere auf den Vergleich zwischen Unterbrechungs- und taktgesteuerter Ein/Ausgabe abgehoben, während in [145] die Optimierung der Taktdauer τ im Falle einer Meldungsklasse ohne Prioritäten behandelt wird.

Im folgenden wird die Systemoptimierung im Falle mehrerer Meldungsklassen mit Hilfe des Modelles $MB_{\text{SYNCHRON}}/G/1$ mit nichtunterbrechenden Prioritäten und E/A-Verwaltungszeiten vorgenommen. (Die Ergebnisse sind - mit Ausnahme absoluter Werte für die Wartezeiten - prinzipiell gleich wie im entsprechenden Falle $CB/G/1$).

Betrachtet werde hierzu folgendes System:

(B) Signalpuffer- und Ein/Ausgabesystem:

Anzahl der Gruppen	$G = 10$
max. Anzahl übertragener Meldungen/Takt	$n_i \gg 1$
Markoffsche Ankünfte mit Raten	$\lambda_j^{(B)} = 20 \dots 40/\text{sec}, j=1,2,\dots,10$
variable Taktdauer	$\tau = 3 \dots 40 \text{ msec}$

(C) Zentralsteuerung:

konstante Ein/Ausgabeverwaltungszeit	$h_0 = 2 \text{ msec}$
Anzahl der Prioritätsklassen	$P = 2$
Aufteilung der Meldungen auf Klassen	$\gamma_i = 0.5$
mittlere Bedienungszeiten für Meldungen	$h_i = 2 \text{ msec}, i = 1, 2$
Bedienungszeit-Charakteristiken	E_2

In Bild 6.19 sind die Reaktionszeiten für beide Prioritätsklassen dargestellt in Abhängigkeit der Taktdauer τ für die zwei Belastungsfälle $A = 0.4$ und $A = 0.8$. Die mittlere Reaktionszeit r setzt sich in diesem Falle aus den beiden Teilmodellen (B) und (C) zusammen.

Aus Bild 6.19 wird erkenntlich, daß sich die Lage des Optimums mit zunehmender Belastung zu höheren Werten τ verschiebt, insbesondere für die niedrigere Priorität. Während für $A = 0.4$ der Bereich $\tau = 7 \dots 9 \text{ msec}$ optimal für beide Klassen ist, ergibt sich für $A = 0.8$ eine größere Diskrepanz in der Lage des Optimums individuell für jede Klasse ($\tau = 10 \text{ msec}$ für Klasse 1 bzw. $\tau = 22 \text{ msec}$ für Klasse 2). Ein Kompromiß von $\tau = 15 \dots 25 \text{ msec}$ ermöglicht noch kleine Reaktionszeiten für Klasse 1 bei nahezu optimaler Behandlung von Klasse 2.

Im Hinblick auf Fragen der Überlaststeuerung können folgende Maßnahmen ergriffen werden:

(a) Erhöhung der Taktdauer.

Durch diese Maßnahme wird Realzeit-Kapazität zurückgewonnen infolge des geringeren E/A-Verwaltungsanteiles. Wie aus Bild 6.19 ersichtlich ist, können für die dringlicheren Meldungen in einem weiten Bereich die Reaktionszeitforderungen erfüllt werden bei gleichzeitiger stationärer Bearbeitung der

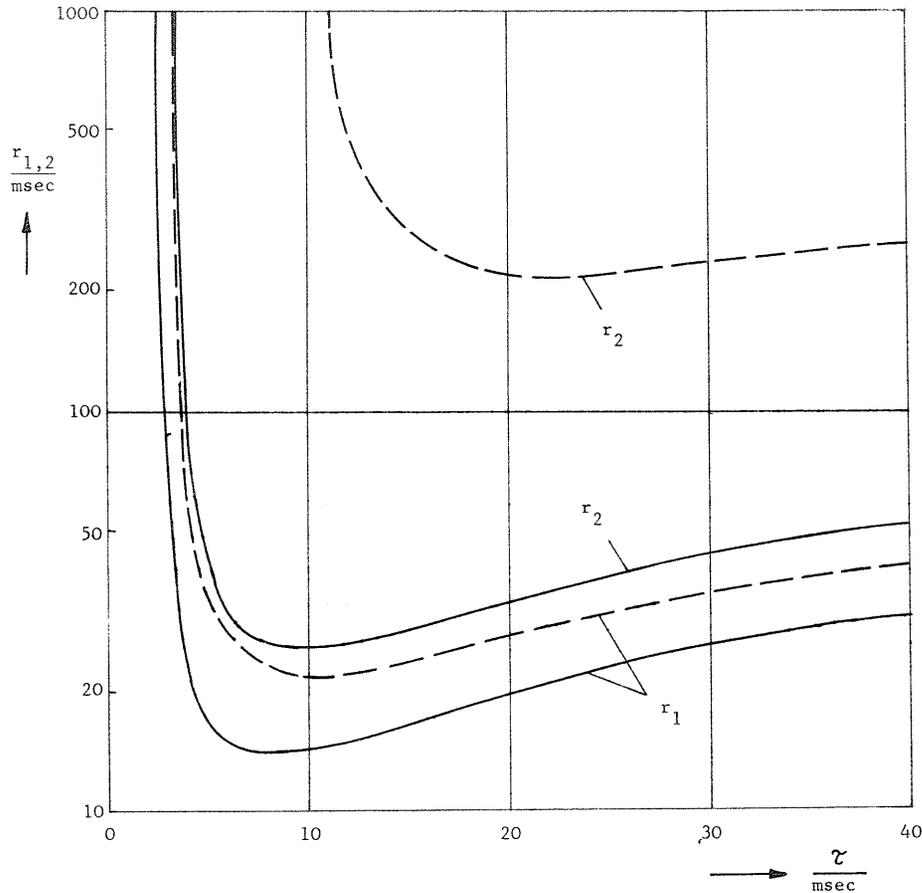


Bild 6.19. Zur Optimierung der Taktdauer τ
 Mittlere Reaktionszeiten r_1 und r_2 in Abhängigkeit der Taktdauer τ
 $P = 2$ Prioritätsklassen
 $h_0 = 2$ msec Ein/Ausgabeverwaltungszeit
 ——— Angebot $A = 0.4$ ($A_1 = A_2 = 0.2$)
 - - - - - Angebot $A = 0.8$ ($A_1 = A_2 = 0.4$)

niedrigeren Meldungsklasse. Die Reaktionszeitforderungen hinsichtlich der niedrigeren Klasse lassen sich im allgemeinen nicht mehr oder nur eingeschränkt erfüllen.

- (b) Bei stärkerer Überlast kann die Zentralsteuerung durch eine Änderung der Takt-dauer τ allein nicht den Überlastbereich abwenden. Es müssen hier drastischere Lastbegrenzungsmaßnahmen ergriffen werden. Eine relativ einfache Maßnahme, welche in realen Systemen mit zyklischer Abfrage in mehr oder weniger modifizierter Form angewandt wird, ist die Begrenzung der Anzahl übertragener Meldungen je Zyklus. Z.B. würde eine Beschränkung von ≤ 2 Meldungen/Zyklus bei den in den Beispielen 1 und 2 zugrundegelegten Parametern eine Begrenzung der Meldungsrate auf 36.36/sec und Gruppe zur Folge haben, was einer maximalen Prozessorauslastung von 72.72% entspräche.

Weitere Maßnahmen zur Optimierung und Überlaststeuerung sind diskutiert und zum Teil implementiert worden, deren Diskussion jedoch nur sinnvoll ist im Zusammenhang mit spezielleren Systemvoraussetzungen. Die Bemerkungen zu diesem Punkte sollen hier lediglich dazu dienen, die Nützlichkeit verkehrstheoretischer Methoden zur Beantwortung derartiger Fragen zu unterstreichen bzw. die notwendige Analysemethodik bereitzustellen.

c) Simulationsvergleiche

Abschließend werden einige Simulationsvergleiche angegeben, welche über die Genauigkeit approximativer Ergebnisse Aufschluss geben sollen.

Die Ergebnisse des allgemeinen Teilmodelles (A) sind approximativ und wurden in [140] ausführlichen Simulationsvergleichen unterzogen. Wir können uns hier deshalb auf den betrachteten Anwendungsfall beschränken. Bild 6.20a zeigt zunächst die mittleren Wartezeiten für ein Wartesystem mit $g = 20$ Eingangswarteschlangen des Typs M/D/1 bzw. MB/D/1 mit konstanter Ankunftsgruppengröße 4. Die Bedienungszeit- und Umschaltzeitparameter sind $h_j = u_j = 2.5$ msec. Die theoretischen Ergebnisse liegen durchweg im Vertrauensintervall der Simulation (100 000 "Rufe", 10 Teiltests). Ferner ist in Bild 6.20b der theoretische Verlauf des Quadrates des Variationskoeffizienten c_D des Ausgangsprozesses angegeben mit Simulationsergebnissen im Falle von M/D/1.

Die Empfindlichkeit der approximativen Berechnung von c_D nach Gl. (6.28) im Hinblick auf verschiedene Ankunftsprozeßcharakteristiken zeigt folgender Vergleich für die Systeme $E_4/D/1$, M/D/1, $H_2/D/1$ ($c_A = 1.5$) und MB/D/1 (konstante Gruppengröße $K_j = 4$, $j = 1, 2, \dots, 20$), siehe Tabelle 4.

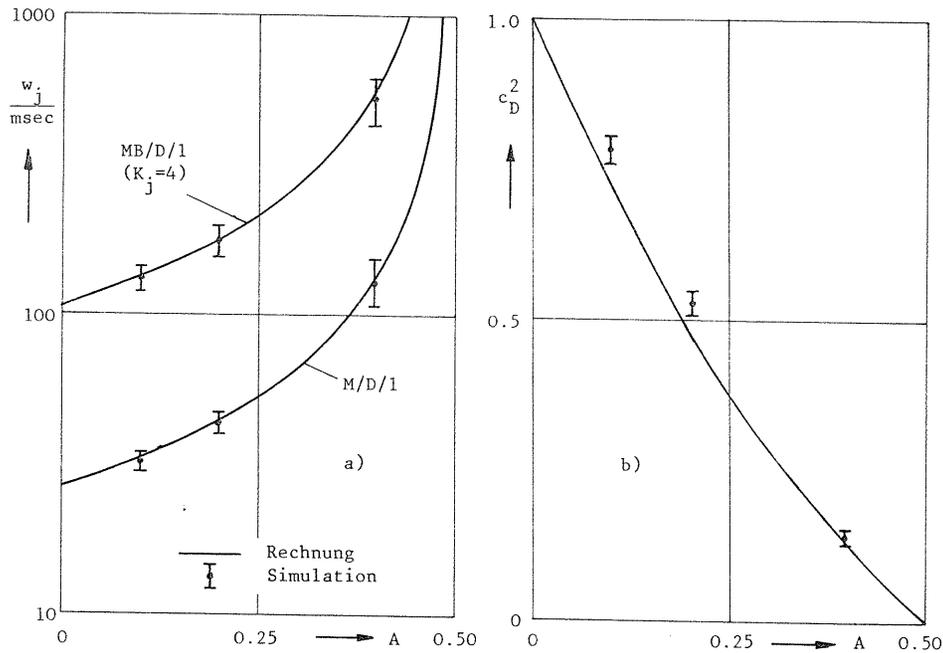


Bild 6.20. Wartezeiten und Ausgangsprozeß von Wartesystemen mit zyklischer Abfertigungsstrategie

- a) Mittlere Wartezeit w_j in Abhängigkeit des Angebots A
 - b) Variationskoeffizient c_D^2 in Abhängigkeit des Angebots A
- $g = 20$ Gruppen

Angebot A	0	0.10	0.20	0.40	0.50
c_D^2 nach Gl.(6.28)	1.000	0.720	0.480	0.120	0
Simulation $F_4/D/1$		0.761±0.018	0.517±0.015	0.125±0.004	0
Simulation M/D/1		0.788±0.015	0.530±0.014	0.128±0.009	0
Simulation $H_2/D/1$ ($c_A = 1.5$)		0.805±0.018	0.536±0.016	0.128±0.011	0
Simulation MB/D/1 ($K_j = 4$)		1.370±0.144	0.535±0.036	0.133±0.020	0

Tabelle 4. Ausgangsprozeß-Variationskoeffizient von Wartesystemen des Typs GI/D/1 mit zyklischer Abfertigungsstrategie

Parameter: $G = 20$ Gruppen

$h_j = u_j = 2.5$ msec, $j = 1, 2, \dots, 20$

symmetrische Last

Die Ergebnisse des verwendeten Teilmodelles (B), M/C/n-s, sind exakt bis auf die Ausnahme der mittleren Wartezeit. Simulationsvergleiche hierzu sind in [175] angegeben.

Die Ergebnisse der Teilmodelle (C) mit synchroner Ankunft sind exakt im Falle MB SYNCHRON/G/1. Für getaktete Ankünfte (Modelle CB/G/1 ohne bzw. mit Verwaltungszeiten und Prioritäten) sind die Ergebnisse approximativ. Simulationsvergleiche für den Fall CB/G/1 mit nur einer Klasse von Meldungen wurden in [113] und [175] angegeben, sodaß wir uns hier auf den allgemeineren Fall mit Prioritäten beschränken können.

Bild 6.21 zeigt Ergebnisse für die Systeme $CB/E_2/1$ und $CB/H_2/1$ mit zwei Klassen von nichtunterbrechenden Prioritäten. In beiden Fällen sind die exakten Ergebnisse für die entsprechenden Systeme mit exponentialverteilten Taktabständen sowie die durch Gl. (6.51) ermittelten approximativen Ergebnisse zusammen mit Simulationsergebnissen angegeben.

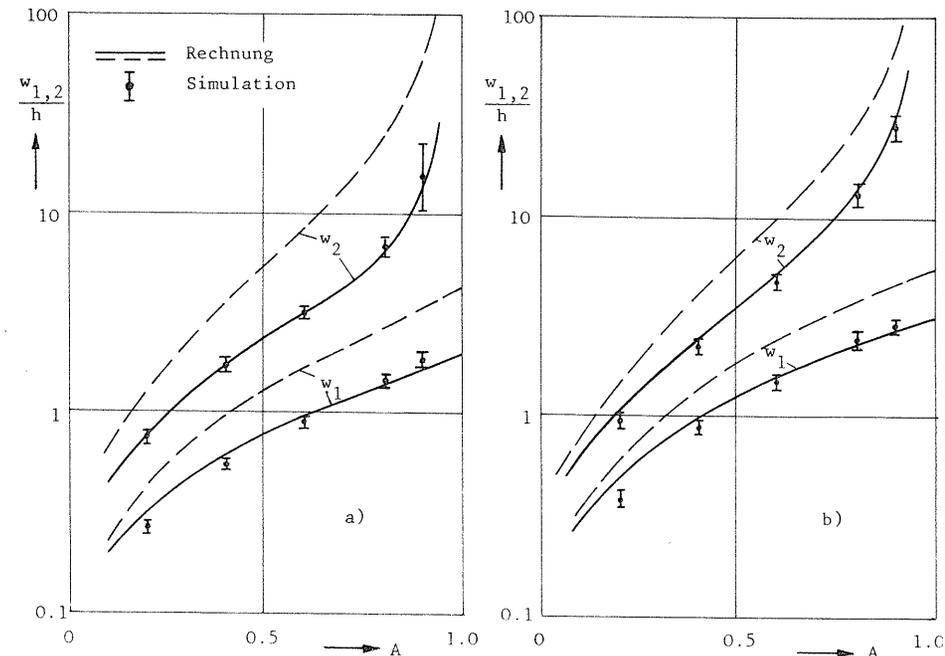


Bild 6.21. Simulationsvergleiche für Wartesysteme mit getakteter Ankunft und Prioritäten

$h_1 = h_2 = h$, $\tau/h = 5$, $h_0 = 0$. Poisson verteilte Gruppengrößen

- a) ——— $CB/E_2/1$
- $MB/E_2/1$
- b) ——— $CB/H_2/1$ ($c_H = 1.5$)
- $MB/H_2/1$ ($c_H = 1.5$)

Wir bemerken abschließend, daß Simulationsvergleiche für Teilmodelle nur einen unvollständigen Genauigkeitsnachweis darstellen. Einflüsse der Unabhängigkeitsannahmen bzw. der Beschränkung auf nur zwei Parameter zur Kennzeichnung der Schnittstellenverkehre bei dem Zerlegungsverfahren können nur durch die Simulation des Gesamtmodelles beurteilt werden. Simulationen für derartige Gesamtmodelle sind im allgemeinen sehr aufwendig und wurden im Zusammenhang mit speziellen Steuerungsstrukturen durchgeführt, vergl. hierzu etwa [173], bzw. für standardisierte Netze, vergl. hierzu Abschnitt 5.4.

Im Zusammenhang mit o.a. Beispiel 1 wurde das aus den $G = 10$ Pufferspeichern und der Zentralsteuerung bestehende zweistufige Teilmodell mit einem im Rahmen der Arbeit [141] entwickelten Simulationsprogramm untersucht. Für niedrige Verkehrsraten ist das Zerlegungsverfahren sehr genau, da die Annahme der Unabhängigkeit zwischen aufeinanderfolgend übertragenen Gruppen praktisch erfüllt ist. Für den Fall eines verhältnismäßig hohen Angebotes von $A = 0.8$ an die Zentralsteuerung ergab sich folgender Vergleich zwischen Rechnung und Simulation, vergl. Tabelle 5.

	Rechnung	Simulation
(B) Signalpuffer- und E/A-System M/C/1-s ($s = 5$) Überlaufwahrsch. Mittlere Wartezeit	$B = 0.00027$ $w^{(B)} = 8.310 \text{ msec}$	$B = 0.00036 \pm 0.00012$ $w^{(B)} = 8.308 \pm 0.005 \text{ msec}$
(C) Zentralsteuerung CB/D/1 Mittlere Wartezeit	$w^{(C)} = 4.41 \text{ msec}$	$w^{(C)} = 5.09 \pm 0.153 \text{ msec}$
Gesamt-Reaktionszeit	$r = 14.72 \text{ msec}$	$r = 15.40 \pm 0.159 \text{ msec}$

Tabelle 5. Vergleich von Teil- und Gesamtergebnissen

Parameter: $G = 10$ Signalpuffer
 $\lambda = 40/\text{sec}$ Meldungen je Puffer (M)
 $\tau = 10 \text{ msec}$ Taktdauer
 $n_j = 1$ Meldung abgefertigt/Takt, $j = 1, 2, \dots, 10$
 $h = 2 \text{ msec}$ konstante Bedienungszeit pro Meldung in der Zentralsteuerung ($\hat{A} = 0.8$)

Die Diskrepanz in $w^{(C)}$ wird verursacht durch

- Annahme unabhängiger Gruppengrößen
- Näherungslösung für das Modell CB/D/1.

Die Gesamtreaktionszeit wurde in diesem Beispiel mit etwa 4.5% Ungenauigkeit ermittelt. Selbst bei Annahme Poisson-verteilter Gruppengrößen liegt die Ungenauigkeit nur bei etwa 6.5%. Diese Genauigkeit ist i.a. tragbar für Zwecke der Anwendung.

7. ZUSAMMENFASSUNG

Das Ablaufgeschehen in Systemen zur Nachrichtenvermittlung und Nachrichtenverarbeitung ist neben dem funktionellen Zusammenspiel einzelner Systemkomponenten durch die Eigenschaften der auftretenden Verkehrsflüsse gekennzeichnet. Die Kenntnis dieser Verkehrsflüsse, ihrer Eigenschaften und Auswirkungen auf Leistungskriterien wie Durchsatz und Reaktionszeiten ist von großer Wichtigkeit für den Systementwurf und die Planung. Zur Analyse der auftretenden Verkehrsflüsse hat sich insbesondere die Betrachtungsweise der zufallsabhängigen Prozesse bewährt.

Die vorliegende Arbeit versucht, den logischen Kreis zwischen funktioneller und verkehrsbezogener Systembetrachtung zu schließen. Ausgehend von der funktionellen Systembeschreibung, welche anhand zweier Beispiele aus den Gebieten der rechnergesteuerten Nachrichtenvermittlungssysteme sowie der Rechnersysteme mit virtuellem Speicherprinzip vorgenommen wird, werden zunächst die Zusammenhänge zwischen Systemfunktion und Verkehr aufgezeigt. In den nachfolgenden Kapiteln werden dann Methoden der Modellbildung sowie der Modellanalyse in Form einer allgemeinen Übersicht und klassifizierenden Darstellung beschrieben.

Die abschließenden Kapitel sind verschiedenen Methoden zur Analyse und Optimierung von Systemen gewidmet, welche durch komplexere Strukturen und Betriebsweisen gekennzeichnet sind. Hierzu wird zunächst ein neues Analyseverfahren vorgestellt, welches auf einer Zerlegungsmethode beruht, wobei Eigenschaften und gegenseitige Beeinflussung der entstehenden Teilsysteme im Sinne einer systemtheoretischen Betrachtungsweise im wesentlichen durch ihr Ein/Ausgangsverhalten bzw. ihre Schnittstellenverkehre charakterisiert werden. Dieses Verfahren wird ausführlich im Zusammenhang mit der Analyse allgemeiner Warteschlangennetze behandelt.

Das abschließende Kapitel ist der Anwendung gewidmet. Anhand von vier ausführlicheren Beispielen von Rechnersystemen mit virtuellem Speicherprinzip, interaktiv betriebenen Rechnersystemen sowie hierarchisch gegliederten Steuerungsstrukturen von zentralgesteuerten Nachrichtenvermittlungssystemen werden Fragen der Modellierung, der Systemanalyse und der Systemoptimierung behandelt. Zur Analyse der komplexen Systemmodelle werden verschiedene exakte und approximative Verfahren angewandt bzw. entwickelt. Ferner werden approximativ gewonnene Ergebnisse einem Vergleich mit der Simulation unterzogen. Die verschiedenen Anwendungsbeispiele werden jeweils durch eine Reihe von numerischen Ergebnissen begleitet, welche den Einfluß der wesentlichsten System- und Verkehrsparameter auf die charakteristischen Leistungsgrößen aufzeigen bzw. als Grundlage zur Systemoptimierung dienen.

