

Annette Merkel

---

**Modellgestützte Untersuchung  
konkurrierender Zugriffe bei leitungs-  
vermittelnden Kommunikationssystemen**

---



**Cuvillier Verlag Göttingen**  
Internationaler wissenschaftlicher Fachverlag

Modellgestützte Untersuchung  
konkurrierender Zugriffe  
bei leitungsvermittelnden  
Kommunikationssystemen

Der Technischen Fakultät der  
Universität Erlangen-Nürnberg  
zur Erlangung des Grades

D O K T O R - I N G E N I E U R

vorgelegt von  
**Annette Merkel**  
Erlangen - 2010

## **Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2010

Zugl.: Erlangen-Nürnberg, Univ., Diss., 2009

978-3-86955-255-2

Als Dissertation genehmigt von  
der Technischen Fakultät der  
Universität Erlangen-Nürnberg

Tag der Einreichung: ..... 30.06.2009

Tag der Promotion: ..... 27.11.2009

Dekan: ..... Prof. Dr.-Ing. Reinhard German

Berichterstatter: ..... Prof. Dr.-Ing. Reinhard German  
Prof. Dr.-Ing. Bernhard Schmauß

© CUVILLIER VERLAG, Göttingen 2010

Nonnenstieg 8, 37075 Göttingen

Telefon: 0551-54724-0

Telefax: 0551-54724-21

[www.cuvillier.de](http://www.cuvillier.de)

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage 2010

Gedruckt auf säurefreiem Papier.

978-3-86955-255-2

# Danksagung

Für die Bereitschaft, die Aufgabe des Erstberichterstatters kurzfristig zu übernehmen, möchte ich mich bei Herrn Prof. Dr.-Ing. Reinhard German bedanken. Mein Dank gilt Herrn Prof. Dr.-Ing. Bernhard Schmauß für seine Unterstützung und das Interesse am Fortgang der Arbeit. Herrn Prof. Dr.-Ing. Herbert Haunstein spreche ich meinen Dank aus, dass er sich als weiterer Prüfer zur Verfügung gestellt hat. Ebenso danke ich Herrn Prof. Dr.-Ing. Joachim Hornegger für die Übernahme des Prüfungsvorsitzes.

Bei meinen ehemaligen Kollegen Frau Monika Dietz und Herrn Heinz Keilholz möchte ich mich bedanken, dass sie mir durch langjährige Durchführung von Lasttests bei Kommunikationssystemen die Mächtigkeit dieses Testansatzes aufzeigten. Ebenso danke ich Herrn Dr. Rainer Fechner von Alcatel-Lucent. Er ermöglichte mir, aufgetretene Versagen beim Test von Kommunikationssystemen zu analysieren und als Basis für diese Arbeit zu verwenden.

Mein besonderer Dank gilt Frau Dr. Gabriele Eichfelder für ihre Unterstützung.



# Zusammenfassung

Kommunikationssysteme müssen eine sehr hohe Verfügbarkeit haben. Fehler in der Software dieser Systeme können zum Ausfall des gesamten Telekommunikationsnetzes führen. Aus diesem Grund bedarf es einer hohen Qualität beim Test derartiger Systeme.

Der Ausgangspunkt dieser Arbeit waren Fehlerbeschreibungen aufgetretener Versagen der Fehlerklasse „konkurrierende Zugriffe“ beim Test eines neu entwickelten 40 GBit Übertragungssystems in der Industrie. Es wird untersucht, wie die Auftrittshäufigkeit solcher Versagen mittels Lasttests erhöht werden kann. Die hier vorgestellte neuartige Untersuchung basiert auf einem stochastischen Simulationsmodell. Die daraus gewonnenen Ergebnisse werden mit den in einem Wahrscheinlichkeitsmodell analytisch ermittelten Auftrittshäufigkeiten verglichen. Dabei ist eine der zentralen Fragestellungen die Abhängigkeit der Auftrittshäufigkeit von verschiedenen Parametern wie z.B. der Anzahl der geschalteten Verbindungen. Weitere neue Erkenntnisse zur Auftrittshäufigkeit in Abhängigkeit von Betriebssystemeigenschaften werden gewonnen. Potentielle Varianzmöglichkeiten bei der Durchführung von Lasttests zur Steigerung der Auftrittshäufigkeit und deren Relevanz werden bestimmt.

Mit dieser Arbeit steht erstmals eine Beschreibung einer systematischen Vorgehensweise bei der Durchführung von Lasttests an Kommunikationssystemen in der Industrie zur Verfügung. Dadurch ist eine verbesserte Analyse dieser Art von Softwarefehlern möglich. Somit wird eine höhere Verfügbarkeit der Kommunikationssysteme im Betrieb beim Kunden erreicht.



# Abstract

The availability of communication systems has to be very high. Software faults in these systems can lead to a breakdown of the whole network. For this reason a high quality of the tests of these systems is important.

The starting point of this thesis were different reports of failures of the class "concurrency" which occurred during testing a new 40 GBit transmission system in industry. Thus it is investigated how the frequency of occurrence of such failures can be increased using load tests. The presented analysis is based on a stochastic simulation model. The obtained results are compared with the frequency of occurrence gained analytically with the help of a probability model. One main question is the dependence of the frequency of occurrence of different parameters like the number of switched connections. Further, new results about the dependence of the frequency of occurrence on the characteristics of the operating system were obtained. Possible variances for the load tests for increasing the frequency of occurrence and their relevance are determined.

In this thesis for the first time a description of a systematic approach for load tests of communication systems in industry is given. Based on that an improved analysis of these failures is possible. A better availability of the communication systems in operation is achieved.





# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>xi</b>
<b>Tabellenverzeichnis</b>	<b>xv</b>
<b>Pseudocodeverzeichnis</b>	<b>xvii</b>
<b>Symbolverzeichnis</b>	<b>xix</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Stand der Technik . . . . .	4
1.3 Ziel dieser Arbeit . . . . .	5
1.4 Kapitelübersicht . . . . .	5
<b>2 Beschreibung des betrachteten Systemaufbaus</b>	<b>9</b>
2.1 Leitungsvermittelnde Kommunikationssysteme . . . . .	9
2.1.1 Netzgröße von Kommunikationssystemen . . . . .	10
2.1.2 Übertragungsmedium zwischen Netzelementen . . . . .	11
2.1.3 Dynamische Verbindungen zwischen Netzelementen . . . . .	12
2.1.4 Kommunikation zwischen Netzelementen . . . . .	13
2.1.5 Arten von Netzelementen . . . . .	14
2.1.6 Aufbau eines Netzelements . . . . .	16
2.1.7 Systemfunktionen . . . . .	17
2.1.8 Betriebssystemeigenschaften . . . . .	18
2.2 Konkurrierende Zugriffe . . . . .	20
2.2.1 Aufgetretene Versagen . . . . .	21
2.2.2 Beschreibung der Ursache eines Versagens . . . . .	21
2.3 Lastgenerator . . . . .	25
2.3.1 Beschreibung der Last . . . . .	25
2.3.2 Eigenschaften eines Lastgenerators . . . . .	26
<b>3 Grundlagen der Modellierung</b>	<b>29</b>
3.1 Leitungsvermittelnde Kommunikationssysteme . . . . .	29
3.1.1 Netzgröße . . . . .	29

3.1.2	Physikalische Verbindungen zwischen den Knoten . . . . .	30
3.1.3	Dynamische Verbindungen zwischen den Knoten . . . . .	30
3.1.4	Kommunikation zwischen den Knoten . . . . .	32
3.1.5	Arten von Knoten . . . . .	33
3.1.6	Eigenschaften der Knoten und Verbindungen . . . . .	34
3.2	Konkurrierende Zugriffe . . . . .	35
3.3	Lastgenerator . . . . .	39
3.3.1	Beschreibung der Last . . . . .	39
3.3.2	Eigenschaften des Lastgenerators . . . . .	40
<b>4</b>	<b>Analytisches Modell</b>	<b>43</b>
4.1	Modelleinschränkungen . . . . .	43
4.1.1	Modelle von Kommunikationssystemen . . . . .	43
4.1.2	Konkurrierende Zugriffe . . . . .	44
4.1.3	Last . . . . .	44
4.1.4	Eigenschaften des Gesamtmodells . . . . .	45
4.2	Herleitung der Auftrittshäufigkeit . . . . .	45
<b>5</b>	<b>Simulationsmodell</b>	<b>51</b>
5.1	Eigenschaften der Schreib-Lesevorgänge . . . . .	51
5.2	Aufbau des Simulationsmodells . . . . .	55
5.3	Arten konkurrierender Zugriffe . . . . .	65
<b>6</b>	<b>Modellauswertungen</b>	<b>67</b>
6.1	Parametrisierung der Modelle . . . . .	67
6.1.1	Modelle von Kommunikationssystemen . . . . .	67
6.1.2	Konkurrierende Zugriffe . . . . .	69
6.1.3	Last zur Untersuchung von konkurrierenden Zugriffen . . .	72
6.2	Auswertung durch das analytische Modell . . . . .	74
6.2.1	Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbin- dungen . . . . .	74
6.2.2	Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Dauer der Schreib-Lesevorgänge . . . . .	84
6.2.3	Bestimmung der Auftrittshäufigkeit in Abhängigkeit von Betriebssystemeigenschaften . . . . .	84
6.3	Auswertung durch das Simulationsmodell . . . . .	88
6.3.1	Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbin- dungen . . . . .	88
6.3.2	Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Dauer der Schreib-Lesevorgänge . . . . .	92

6.3.3	Bestimmung der Auftrittshäufigkeit in Abhängigkeit von Betriebssystemeigenschaften . . . . .	93
6.4	Vergleich der Ergebnisse . . . . .	99
6.4.1	Vergleich der Ergebnisse bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbindungen . . . . .	99
6.4.2	Betrachtung des Einflusses der Betriebssystemeigenschaften	106
6.4.3	Zusammenfassung . . . . .	107
<b>7</b>	<b>Leitlinien zur Durchführung von Lasttests</b>	<b>111</b>
7.1	Eigenschaften der Modellierungen und des leitungsvermittelnden Systems . . . . .	111
7.2	Systematische Vorgehensweise bei der Durchführung von Lasttests	113
7.3	Zusammenfassung . . . . .	117
	<b>Literaturverzeichnis</b>	<b>119</b>



# Abbildungsverzeichnis

2.1	Optisches Übertragungssystem des Netzbetreibers Global Crossing (vgl. [22]) . . . . .	11
2.2	Topologie eines Kommunikationssystems für die Durchführung von Tests beim Systementwickler . . . . .	12
2.3	Aufbau einer ISDN-Verbindung . . . . .	15
2.4	Aufbau eines Netzelements . . . . .	16
2.5	Ausführungsreihenfolgen zweier nebenläufiger Anweisungen, (a) parallel, (b) verzahnt . . . . .	19
2.6	Verkettete Liste . . . . .	23
2.7	Prozess A setzt Zeiger von neuem Element 27 auf die Liste . . . .	23
2.8	Prozess B setzt Zeiger von neuem Element 28 auf die Liste . . . .	23
2.9	Kopf der Liste wird durch Prozess B auf Listenelement 28 geändert	23
2.10	Kopf der Liste wird durch Prozess A auf Listenelement 27 geändert	24
2.11	Element 28 von Prozess B ist nicht mehr Teil der Liste . . . . .	24
2.12	Verkehrsaufkommen eines realen Netzwerks (nach [16]) . . . . .	27
3.1	Aufbau und Abbau einer Verbindung des Modells eines Kommunikationssystems . . . . .	32
3.2	Zeitlicher Ablauf des Verbindungsaufbaus und -abbaus . . . . .	33
3.3	Gerichtete Verbindung $V_{1_s}$ mit $r_s = 2 - 3 - 5 - 4$ . . . . .	34
3.4	Zeitlicher Ablauf des Verbindungsaufbaus und Verbindungsabbaus in einem Vermittlungssystem . . . . .	35
3.5	Zeitlicher Ablauf des Verbindungsaufbaus und Verbindungsabbaus in einem Übertragungssystem . . . . .	35
3.6	Prozessorbelegung des A-Knotens während mehrerer Schreib-Lesevorgänge . . . . .	37
3.7	Aufbau eines Schreib-Lesevorgangs . . . . .	38
4.1	Zwei Verbindungen mit je zwei Schreib-Lesevorgängen . . . . .	45
4.2	Zwei Verbindungen mit je vier Schreib-Lesevorgängen . . . . .	46
4.3	Vier Verbindungen mit je zwei Schreib-Lesevorgängen . . . . .	46
5.1	Schreib-Lesevorgänge von Threads . . . . .	52
5.2	Theoretische Betrachtung von Schreib-Lesevorgängen zweier Threads	53

5.3	Messergebnis von Schreib-Lesevorgängen zweier Threads . . . . .	53
5.4	Schreib-Lesevorgänge dreier Threads . . . . .	54
5.5	Sequenz von vier Messblöcken . . . . .	56
5.6	Aufbau eines Messblockes . . . . .	57
5.7	Aufbau einer Messung . . . . .	57
5.8	Prozessorbelegung der Threads während zweier Schreib-Lesevorgänge	58
5.9	Prozessorbelegung aller beteiligten Threads während des Verbindungs- aufbaus und -abbaus . . . . .	59
5.10	Arten konkurrierender Zugriffe . . . . .	65
6.1	Topologie des komplexen Modells . . . . .	69
6.2	Ergebnisse des einfachen Modells eines Vermittlungssystems — Lastvarianten A1 – A5 . . . . .	77
6.3	Ergebnisse des einfachen Modells eines Übertragungssystems — Lastvarianten B1 – B5 . . . . .	79
6.4	Ergebnisse des einfachen Modells eines Vermittlungssystems — Lastvariante A2 — und eines Übertragungssystems — Lastvari- ante B2 . . . . .	80
6.5	Ergebnisse des komplexen Modells eines Vermittlungssystems — Lastvariante C1 . . . . .	81
6.6	Ergebnisse des komplexen Modells eines Vermittlungssystems — Lastvariante C2 . . . . .	81
6.7	Ergebnisse des komplexen Modells eines Übertragungssystems — Lastvariante D1 — für die verschiedenen Speicherfelder der Knoten	82
6.8	Ergebnisse des komplexen Modells eines Vermittlungssystems — Lastvariante E1 . . . . .	83
6.9	Ergebnisse des komplexen Modells eines Übertragungssystems — Lastvariante D1 — bei unterschiedlicher Dauer der Schreib-Lese- vorgänge mit $\delta = 32 \text{ ms}$ . . . . .	84
6.10	Prozessorswitch während eines Schreib-Lesevorgangs . . . . .	86
6.11	Ergebnisse des einfachen Modells eines Übertragungssystems B2 mit Verdopplung der Dauer des Timeslots $t$ bei einer Rate von 100 Schreib-Lesevorgängen im Knoten 1 . . . . .	87
6.12	Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante E1 versus Lastvariante E2 versus Lastvariante E3 . .	95
6.13	Ergebnisse bzgl. geplanter und nicht geplanter Timeslots für das komplexe Modell eines Vermittlungssystems — Lastvariante E1 versus Lastvariante E2 versus Lastvariante E3 . . . . .	96
6.14	Zeitliche Streckung einer Haltephase mit einem Schreib-Lesevorgang	96
6.15	Ergebnisse für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor — Lastvariante E1 versus Lastvariante E1 mit Streckfaktor $q = 1.2$ . . . . .	98

6.16	Vergleich der Ergebnisse für das einfache Modell eines Übertragungssystems — Lastvarianten B1 – B5 . . . . .	100
6.17	Vergleich der Ergebnisse für das komplexe Modell eines Übertragungssystems — Lastvariante D1 . . . . .	100
6.18	Vergleich der Ergebnisse für das komplexe Modell eines Übertragungssystems — Lastvariante D2 . . . . .	101
6.19	Vergleich der Ergebnisse für das einfache Modell eines Vermittlungssystems — Lastvarianten A1 bis A5 . . . . .	102
6.20	Vergleich der Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante C1 . . . . .	102
6.21	Vergleich der Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante E1 . . . . .	103
6.22	Ergebnisse des einfachen Modells eines Vermittlungssystems für die Lastvarianten A1 bis A3 mit Einfluss der Threadverdrängung durch den Verbindungsaufbau und -abbau . . . . .	104
6.23	Ergebnisse des komplexen Modells eines Vermittlungssystems für die Lastvariante C1 mit Einfluss der Threadverdrängung durch den Verbindungsaufbau und -abbau . . . . .	105





# Tabellenverzeichnis

6.1	Einfaches Modell eines Kommunikationssystems . . . . .	68
6.2	Komplexes Modell eines Kommunikationssystems . . . . .	70
6.3	Definition der Routen und der Wahrscheinlichkeit je Route für die komplexen Modelle von Kommunikationssystemen . . . . .	71
6.4	Parameter der Schreib-Lesevorgänge für einfache Modelle von Kommunikationssystemen . . . . .	71
6.5	Parameter der Schreib-Lesevorgänge für komplexe Modelle von Kommunikationssystemen . . . . .	71
6.6	Lastparameter für einfache Modelle von Kommunikationssystemen	73
6.7	Lastparameter für komplexe Modelle von Kommunikationssystemen	73
6.8	Wahrscheinlichkeit mindestens eines Prozessorswitches während eines Schreib-Lesevorgangs . . . . .	85
6.9	Auftrittshäufigkeit konkurrierender Zugriffe für das einfache Modell eines Vermittlungssystems (Lastvarianten A1 bis A5) . . . . .	88
6.10	95%-Konfidenzintervalle für das einfache Modell eines Vermittlungssystems (Lastvarianten A1 bis A5) . . . . .	89
6.11	Auftrittshäufigkeit konkurrierender Zugriffe für das einfache Modell eines Übertragungssystems (Lastvarianten B1 bis B5) . . . . .	90
6.12	95%-Konfidenzintervalle für das einfache Modell eines Übertragungssystems (Lastvarianten B2 bis B5) . . . . .	90
6.13	Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems (Lastvarianten C1 und C2) . . . . .	91
6.14	95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems (Lastvarianten C1 und C2) . . . . .	91
6.15	Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Übertragungssystems (Lastvarianten D1 und D2) . . . . .	92
6.16	95%-Konfidenzintervalle für das komplexe Modell eines Übertragungssystems (Lastvarianten D1 und D2) . . . . .	92
6.17	Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems (Lastvarianten E1, E2 und E3) . . . . .	93
6.18	95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems (Lastvarianten E1, E2 und E3) . . . . .	94

6.19	Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor $q = 1.2$ (Lastvariante E1) . . . . .	97
6.20	95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor $q = 1.2$ (Lastvariante E1)	97
6.21	Vorteile und Grenzen der Auswertungsmethoden . . . . .	107
6.22	Einfluss verschiedener Aspekte auf die Auftrittshäufigkeit konkurrierender Zugriffe . . . . .	108
6.22	Einfluss verschiedener Aspekte auf die Auftrittshäufigkeit konkurrierender Zugriffe . . . . .	109
6.23	Vergleich der Ergebnisse bzgl. Variation der Raten . . . . .	110
7.1	Vergleich der Modellierungen und des leitungsvermittelnden Systems	112
7.2	Potentielle Varianzmöglichkeiten bei der Durchführung von Lasttests zur Steigerung der Auftrittshäufigkeit . . . . .	113
7.2	Potentielle Varianzmöglichkeiten bei der Durchführung von Lasttests zur Steigerung der Auftrittshäufigkeit . . . . .	114
7.3	Optimale Parameter bei der Durchführung von Lasttests . . . . .	115
7.4	Ablauf der Planung von Lasttests . . . . .	116
7.5	Ablauf der Durchführung von Lasttests . . . . .	116

# Pseudocodeverzeichnis

2.1	Fehlerhafter Prozedur zum Eintrag eines neuen Listenelements . . .	25
2.2	Fehlerfreie Prozedur zum Eintrag eines neuen Listenelements . . .	25
5.1	Ablaufdiagramm des Lastgenerators pro Messung . . . . .	60
5.2	Ablaufdiagramm einer Verbindung . . . . .	61
5.3	Ablaufdiagramm eines Knotens . . . . .	62
5.4	Ablauf eines Schreib-Lesevorgangs . . . . .	63
5.5	Ablauf eines Schreibzugriffs auf ein Speicherfeld eines Knotens . .	63
5.6	Ablauf der Überprüfung des Inhalts des Speicherfelds . . . . .	64



# Symbolverzeichnis

$A$	Verteilung der Zwischenankunftszeiten,
$B$	Verteilung der Bedienzeiten,
$c_{n_1 n_2}$	Leitungskapazität an der Kante von Knoten $n_1$ zu Knoten $n_2$ des Modells des Kommunikationssystems, $c_{n_1 n_2}$ $N \times N$ -Matrix, $c_{n_1 n_2} \geq 0$ , $1 \leq n_1 \leq N$ , $1 \leq n_2 \leq N$ , $n_1, n_2 \in \mathbb{N}_+$
CCSN	Common Channel Signaling Network,
CPU	Central Processing Unit,
$D_{f_{n,m}} = 1$	Schreib-Lesevorgang findet auf dem Speicherfeld $f_{n,m}$ des Knotens $n$ innerhalb einer Zeiteinheit statt, $n \in \{1, \dots, N\}$ , $m \in \{1, \dots, F_n\}$ ,
DWDM	Dense Wavelength Division Multiplex,
$E[K_{f_{n,m}}]$	Erwartete Anzahl konkurrierender Zugriffe auf das Speicherfeld $f_{n,m}$ innerhalb von $t$ Zeiteinheiten, $n \in \{1, \dots, N\}$ , $m \in \{1, \dots, F_n\}$ , $t \in \mathbb{N}_+$ ,
$E[K_n]$	Erwartete Anzahl konkurrierender Zugriffe im Knoten $n$ innerhalb von $t$ Zeiteinheiten, $n \in \{1, \dots, N\}$ , $t \in \mathbb{N}_+$
$E[K_N]$	Erwartete Anzahl konkurrierender Zugriffe im Modell des Kommunikationssystems innerhalb von $t$ Zeiteinheiten, $N \in \mathbb{N}$ , $t \in \mathbb{N}_+$ ,
$F_n$	Anzahl der Speicherfelder des Knotens $n$ , $n \in \{1, \dots, N\}$ ,
FIFO	First-In-First-Out,
$G$	Verteilungsfunktion der Schreib-Lesevorgänge,
GSM	Group Special Mobile,
gTS	geplanter Timeslot,
$H_i$	Dauer der Haltephase der Verbindung $i$ , $i \in \{1, \dots, I\}$ ,
$H_M$	Mittelwert der Dauer der Haltephase,
$I$	Anzahl der Verbindungen
$I_M$	Anzahl der zu einem Zeitpunkt durchschnittlich existierenden Verbindungen in der Messphase

ISDN	Integrated Services Digital Network,
$k$	maximal mögliche Anzahl der Nachrichten je Knoten in Bedienstation und Warteschlange,
$K_-$	untere Grenze eines Konfidenzintervalls,
$K_+$	obere Grenze eines Konfidenzintervalls,
$K_{f_{n,m}} = 1$	zwei oder mehr Schreib-Lesevorgänge von Verbindungen greifen innerhalb einer Zeiteinheit auf das Speicherfeld $f_{n,m}$ zu, $n \in \{1, \dots, N\}, m \in \{1, \dots, F_n\}$ ,
$L$	Last,
L	Lesevorgang,
$M$	Exponentialverteilung,
$M/M/1/k$	Warteschlangensystem mit Exponential-Verteilung der Zwischenankunftszeit, Exponential-Verteilung der Bedienzeit, einer Bedienstation und Warteschlange der Länge $k - 1, k \in \mathbb{N}_+$ ,
$N$	Anzahl der Knoten des Modells des Kommunikationssystems, $N \in \mathbb{N}_+$ ,
NCSL	Non Commentary Source Line, Anzahl nicht-kommentierter Quellcodezeilen,
ngTS	nicht geplanter Timeslot,
$p$	Wahrscheinlichkeit eines Schreib-Lesevorgangs,
$p_{f_{n,m}}$	Wahrscheinlichkeit, dass bei einem Schreibzugriff im Knoten $n$ das Speicherfeld $f_{n,m}$ ausgewählt wird, $n \in \{1, \dots, N\}, m \in \{1, \dots, F_n\}$ ,
$p_n$	Wahrscheinlichkeit, dass Verbindung den Knoten $n$ als A-Knoten oder Z-Knoten hat, $n \in \{1, \dots, N\}$ ,
$p_{r_s}$	Wahrscheinlichkeit einer Route $r_s, s \in \{1, \dots, S\}$ ,
$p(K_{f_{n,m}} = 1)$	Wahrscheinlichkeit, dass es einen Konflikt zweier oder mehrerer Schreibzugriffe von Verbindungen innerhalb einer Zeiteinheit auf das Speicherfeld $f_{n,m}$ gibt, $n \in \{1, \dots, N\}, m \in \{1, \dots, F_n\}$ ,
$p_{switch}$	Wahrscheinlichkeit mindestens eines Prozessorswitches während eines Schreib-Lesevorgangs,
$p(X_{r_s;A;Z=n} = 1)$	Wahrscheinlichkeit, dass eine Verbindung mit der Route $r_s$ den Knoten $n$ als A-Knoten oder Z-Knoten hat,
$q$	zeitlicher Streckfaktor, $q > 1$ ,
$q_{f_{n,m}}$	Wahrscheinlichkeit, dass bei einem Schreibzugriff der Verbindung, welche den Knoten $n$ als Start- oder Zielknoten hat, innerhalb einer Zeiteinheit das Speicherfeld $f_{n,m}$ , beschrieben wird, $n \in \{1, \dots, N\}, m \in \{1, \dots, F_n\}$ ,

$r_s$	s-te Route, $s \in \{1, \dots, S\}$ ,
$r_{s;A}$	A-Knoten der Route $r_s$ , $s \in \{1, \dots, S\}$ ,
$r_{s;Z}$	Z-Knoten der Route $r_s$ , $s \in \{1, \dots, S\}$ ,
$S$	Anzahl der Routen des Modells des Kommunikationssystems, $S \in \mathbb{N}_+$ ,
S	Schreibzugriff,
SDH	Synchrone Digitale Hierarchie,
SLV	Schreib-Lesevorgang,
SONET	Synchronous Optical Network,
SS7	Signaling System Number 7,
$T$	Dauer der Messphase, $T > 0$ ,
$t$	Anzahl der Zeiteinheiten eines Untersuchungsintervalls, $t \in \mathbb{N}_+$ ,
$u$	Anzahl der Messungen in einem Messblock, $u \in \mathbb{N}_+$ ,
Ü	Modell eines Übertragungssystems,
V	Modell eines Vermittlungssystems,
$V_{i,r_s}$	geschaltete Verbindung $i$ , $i \in \{1, \dots, I\}$ , mit der Route $r_s$ , $s \in \{1, \dots, S\}$ ,
WAN	Wide Area Network,
$X_{r_s;A;Z=n} = 1$	Verbindung hat Route $r_s$ und Knoten $n$ als A-Knoten oder Z-Knoten, $n \in \{1, \dots, N\}$ , $s \in \{1, \dots, S\}$ ,
$Y_{i,f_{n,m}} = 1$	Schreibzugriff einer Verbindung $i$ greift auf das Speicherfeld $f_{n,m}$ des Knotens $n$ innerhalb einer Zeiteinheit zu, $i \in \{1, \dots, I\}$ , $n \in \{1, \dots, N\}$ , $m \in \{1, \dots, F_n\}$ ,
$z$	Dauer des Schreib-Lesevorgangs,
$z'$	Dauer der Ruhephase,
$\gamma$	Rate der Haltephase,
$\lambda$	Ankunftsrate der Verbindungsanforderungen des Lastgenerators,
$\mu$	Ankunftsrate der Anforderungen für Schreib-Lesevorgänge des Lastgenerators an eine Verbindung,
$\sigma$	Standardnormalverteilung,
$\nu$	Verteilung der Bedienzeiten des Warteschlangensystems.





# Kapitel 1

## Einleitung

E.W. Dijkstra *"Program testing can be used  
to show the presence of bugs,  
but never to show their absence!"*

Die größte und leistungsfähigste nachrichtenverarbeitende Maschine der Welt ist das weltweite Telekommunikationsnetz mit mehr als 1,3 Milliarden Telefonanschlüssen im Festnetz. Auch das Mobilfunknetz (auch als Mobilfunksystem bezeichnet) umfasste im Jahr 2007 ca. 3,3 Milliarden Teilnehmer (Stand: 2007, vgl. [32]). Nicht nur der Telefonanschluss gehört heute zu der Grundausstattung für die meisten Bewohner Europas oder der USA, auch das Internet ist mittlerweile eine der wichtigsten Informationsquellen im täglichen Leben. 1,5 Milliarden Internetnutzer gab es im Jahr 2007 und die Anzahl wird weiter steigen (vgl. [32]). Die wirtschaftliche Bedeutung der Verfügbarkeit von Kommunikationssystemen ist durch die Globalisierung der Märkte sehr groß. So gab es im Jahr 1999 weltweit ca. 25 berichtete Fehler pro 100 Anschlüssen pro Jahr (vgl. [33]). Dabei ist die Definition eines berichteten Fehlers von Land zu Land unterschiedlich. Manche Betreiber erfassen Betriebsstörungen beim Kunden auch als Fehler, manche nur technische Fehler.

Abhängig von der Sicherheitsrelevanz der Systeme muss die Verfügbarkeit auch bei den entsprechenden Genehmigungsverfahren wie z.B. des TÜVs oder von Behörden nachgewiesen werden. Zuverlässigkeitsanforderungen bedingen somit eine hohe Qualität der Systeme.

Wie groß die Auswirkungen eines Softwarefehlers in einem Vermittlungssystem sein können, wurde im Jahr 1990 in den USA deutlich. Durch einen Fehler in einem Vermittlungsrechner 4ESS der Firma AT&T kam es zu einem ca. neunstündigen Ausfall des Telefonnetzes. Es waren 114 Vermittlungsstellen betroffen, welche für die Abwicklung des Fernverkehrs zuständig waren. 70 Millionen Telefongespräche konnten nicht vollständig aufgebaut werden. Es entstand ein geschätzter Schaden von 60 Millionen Dollar. Dieser Softwarefehler trat vier Wochen nach der Installation einer neuen Softwareversion auf, d.h. die Software war

bereits vier Wochen lang ohne Auftreten dieses Softwarefehlers in zahlreichen Vermittlungsstellen in Betrieb.

Beachtet man, dass die Welt in den letzten 18 Jahren durch Kommunikationssysteme noch stärker vernetzt wurde, so wird deutlich, welchen wirtschaftlichen Schaden und sicherheitspolitische Auswirkungen ein Ausfall eines Kommunikationssystems heute haben kann.

Bei Kommunikationssystemen sind die verschiedenen Prozesse auf den Systemen voneinander abhängig, d.h. sie operieren im gleichen Adressraum auf gemeinsamen Daten. Dies erfolgt durch verschiedene Threads, welche nebenläufig sind. Kommunikationssysteme gehören somit zu den eingebetteten Echtzeitsystemen und können einige Tausend Threads haben. Die Durchführung verschiedener Threads kann desweiteren auch gekoppelt sein. D.h. nach der Durchführung einer bestimmten Funktionalität wird anschließend eine bestimmte andere Funktionalität durch einen anderen Thread durchgeführt.

Eine formale Verifikation derartiger Systeme ist nur sehr schwer möglich oder durch die Anzahl der in Frage kommenden Ausführungsreihenfolgen sogar unmöglich. Ebenso ist die vollständige Validierung bei der Kürze der heutigen Entwicklungszyklen vor der Auslieferung des Systems an den Kunden nicht möglich. Die Entwicklung effizienter Testkonzepte für derartige Systeme ist folglich von sehr großer Bedeutung.

## 1.1 Motivation

Wegen der weitreichenden Konsequenzen von Softwarefehlern wird bei der Entwicklung von Kommunikationssystemen ein großer Augenmerk auf die Qualität der Tests gelegt. Ein Kommunikationssystem besteht aus mehreren beliebig miteinander verbundenen Knoten. Diese Knoten werden auch als Netzelemente bezeichnet. Netzelemente können jedoch auch als sog. Repeater verwendet werden. Kommunikationssysteme bestehen aus asynchron miteinander kommunizierenden endlichen Automaten. Sie sind jederzeit spontan auftretenden Signalfehlern im Netz (wie z.B. dem Ausfall eines Signals) ausgesetzt. Netzelemente erhalten Dienste anderer Netzelemente und leisten ebenso Dienste für andere Netzelemente. Fehler und Alarme können sich jederzeit im Netz ausbreiten.

Wird das Telekommunikationsnetz in Deutschland betrachtet, so besteht es aus ca. 2400 Vermittlungsstellen.

Systeme aus dem Bereich der optischen Übertragungstechnik haben sehr hohe Übertragungsraten und sind häufig über den Kontinent verteilt. Sie sind Teil des Wide Area Networks (WAN) und verbinden große Netzwerke untereinander, die autark innerhalb von Städten oder Regionen aufgebaut sind. Ein Ausfall einer einzigen geschalteten Verbindung mit einer Übertragungskapazität von 40 GBit/s kann den Abbruch von bis zu 500 000 Telefongesprächen bewirken. Jedes dieser Netzelemente kann bis zu 500 000 € kosten. Deshalb ist es zu teuer, Net-

ze, wie sie bei Netzbetreibern vorhanden sind, im Testlabor zu installieren, um den realen Betrieb nachzubilden und anschließend vollständig zu testen. Ebenso sind die Entwicklungszyklen zu kurz, um den Langzeitbetrieb, wie er im realen Netz stattfindet, beim Test derartiger Systeme nachzubilden. Deshalb werden im Labor Testszenarien durchgeführt, welche jedoch nur eine betrieblich repräsentative Stichprobe umfassen können. Ebenso wird der zeitliche Ablauf komprimiert, um in einem realistischen Zeitfenster die Chancen der Aufdeckung von Fehlern stark zu erhöhen. Es wird das Verhalten des Systems geprüft, wie es auf Ausfälle von Hardware- und Softwarekomponenten, Mehrbenutzerbetrieb mit maximaler gleichzeitiger Benutzerzahl und das Eintreffen ungewöhnlicher bzw. widersprüchlicher Daten reagiert.

Der Test derartiger Systeme im Labor zeigt jedoch, dass es Fehler gibt, die nur sehr selten, d.h. nur unter ganz bestimmten Bedingungen, oder erst nach langer Betriebsdauer zum Versagen des Systems führen.

Aufgrund der hohen Kosten der Fehlerbeseitigung (vgl. *Thaller* in [74]) im realen Netz ist es die Intention von Lasttests, derartige Fehler vor dem Einsatz im realen Netz zu finden. Dies bedeutet, dass diese Fehler spätestens im Systemtest gefunden werden sollen.

Die typische Ursache für solche selten auftretende Fehler sind fehlerhafte Implementierungen nebenläufiger Prozesse (z.B. Lese- und Schreibvorgänge in einem Speicher) bei denen unter selten vorkommenden zeitlichen Bedingungen die parallele Ausführung der Prozesse zu Konflikten führen kann. Die Echtzeitanforderungen dieser Systeme und die Anzahl verschiedenartiger Prozesse führen zu einer hohen Komplexität und zu hohen Qualitätsanforderungen an die Software. Derartige Systeme haben bis zu 3 Millionen unkommentierte Netto-Codezeilen (NCSL). Durch die Größe von Entwicklungsprojekten und durch die Anzahl der beteiligten Entwickler kann es zu einer fehlerhaften Implementierung der nebenläufigen Prozesse kommen. Ebenso wird versucht, die Anzahl der verwendeten Semaphore möglichst gering zu halten, da diese das Reaktionsverhalten des Systems negativ beeinflussen. Durch den fehlerhaften Zustand kann es zu einem Versagen des Systems kommen. In diesem Fall stimmt die erbrachte Funktionalität nicht mit der erwünschten Funktionalität überein. Das Versagen des Systems kann auch erst nach einer Latenzzeit von einigen Wochen auftreten. Ist ein Versagen aufgetreten, muss die Fehlerursache gefunden werden.

Software-Analysewerkzeuge erleichtern die Analyse der Fehlerursache. Es ist jedoch nicht möglich, bei jedem durchgeführten Test stets alle Analysewerkzeuge zu nutzen, da Software-Analysewerkzeugen i.A. Einfluss auf das Zeitverhalten des Systems haben. Ebenso ist die Verwendung von Software-Analysewerkzeuge auch nicht selektiv möglich, da der Tester nicht weiß, bei welcher Systemfunktion als nächstes ein Versagen auftritt (vgl. auch *Yu et al.* in [82]).

Die Fehlerursache kann jedoch umso leichter gefunden werden, je öfter das Versagen auftritt oder je besser es reproduziert werden kann. Aus diesem Grund ist es von sehr großer Bedeutung, das Versagen möglichst schnell bzw. effizient repro-

duzieren zu können. Auch eine Verringerung der Latenzzeit ist sehr hilfreich. Bei den üblichen Testszenarien können derartige Fehler jedoch nur unter hohem materiellen oder zeitlichen Aufwand reproduziert werden. Folglich ist es besonders schwierig, durch die üblichen Softwaretests eine hohe Qualität dieser Systeme zu erreichen.

Die Durchführung von Lasttests in der Industrie beim Test von Kommunikationssystemen zeigt jedoch, dass bestimmte Fehler unter erhöhter Systemlast häufiger zu einem Versagen führen. Als Lasttest wird eine Teststrategie bezeichnet, bei der sich Systemfunktionen besonders häufig abwechseln und zeitlich komprimiert ausgeführt werden. Lasttests werden i.A. mit dem integrierten System, d.h. auf der Zielhardware, durchgeführt. Sie sind Teil des Systemtests. Der Lasttest hat das Ziel, das System im erlaubten Grenzbereich auf Zuverlässigkeit zu testen. Er unterscheidet sich vom Stresstest (vgl. *Balzert* in [5]), bei dem die vom Kunden oder von den Telekommunikationsbehörde definierten Grenzen bewusst überschritten werden.

In dieser Arbeit wird untersucht, wie die Auftrittshäufigkeit konkurrierender Zugriffe mittels Lasttests bestmöglich erhöht werden kann.

Zur Untersuchung von Computersystemen und von Kommunikationssystemen gibt es drei verschiedene Techniken der Evaluierung (vgl. *Jain* in [34]):

- Messung
- Simulation
- Analyse

In dieser Arbeit wird ein Simulationsmodell und ein analytisches Modell angewendet. Beim analytischen Modell wird ein Wahrscheinlichkeitsmodell zur Bestimmung der Auftrittshäufigkeit verwendet. Es werden die Ergebnisse verglichen, welche durch das Simulationsmodell und die Anwendung des analytischen Modells erhalten werden. Aufgrund der hohen Anschaffungs- und Betriebskosten von Kommunikationssystemen werden keine Messungen am realen System durchgeführt.

Die Basis der Untersuchung sind Fehlerfälle, welche beim Test von neuentwickelten Kommunikationssystemen in der Industrie aufgetreten sind. Es wird eine Fehlerklasse betrachtet, welche nur sehr selten oder nach sehr langer Latenzzeit auftritt.

## 1.2 Stand der Technik

Kommunikationssysteme müssen hohen Verfügbarkeitsanforderungen genügen. Diese Verfügbarkeitsanforderungen sind in den Verträgen mit den Netzbetreibern

vereinbart. Bei deren Nichterfüllung ist der Hersteller zur Zahlung hoher Konventionalstrafen verpflichtet. Ein Kommunikationssystem mit einer hohen Verfügbarkeit ist folglich ein großer Wettbewerbsvorteil gegenüber den Mitbewerbern. Effiziente Methoden zur Fehlererkennung und Analyse bei derartigen Systemen sind somit ein äußerst sensibles Thema. Aus diesem Grund gibt es nur vereinzelt Veröffentlichungen zu diesem umfassenden Themenkomplex wie insbesondere von *Whitt et. al.* z.B. in [7], [11], [79].

### 1.3 Ziel dieser Arbeit

Ziel dieser Arbeit ist die Herleitung einer systematischen Vorgehensweise zur Unterstützung industrieller Systementwickler bei der Qualitätssicherung großer und komplexer leitungsvermittelnder Kommunikationssysteme mit hohen Zuverlässigkeitsanforderungen. Es wird in dieser Arbeit untersucht, wie die Auftrittshäufigkeit von Softwarefehlern der Fehlerklasse „Konkurrierende Zugriffe“ an realen Systemen durch Lasttests erhöht werden kann.

Es wird in dieser Arbeit einerseits der Nachweis erbracht, dass sich durch Lasttests die Auftrittshäufigkeit konkurrierender Zugriffe im Vergleich zu funktionalen Tests erhöht, andererseits werden Leitlinien für die Durchführung von Lasttests an realen Systemen definiert. Durch die Erkennung und verbesserte Analysemöglichkeit dieses Softwarefehlers wird eine höhere Verfügbarkeit der Kommunikationssysteme im Betrieb beim Kunden erreicht.

### 1.4 Kapitelübersicht

Im Folgenden wird eine Übersicht für die Kapitel dieser Arbeit gegeben.

#### **Kapitel 1:** Einleitung

Dieses Kapitel gibt eine Einleitung in die zu behandelnde Problemstellung. Es wird die Motivation, der Stand der Technik und das Ziel der Arbeit beschrieben. In einer Übersicht werden die Kapitel dieser Arbeit beschrieben.

#### **Kapitel 2:** Beschreibung des betrachteten Systemaufbaus

Es werden die Technologien leitungsvermittelnder Kommunikationssysteme beschrieben. Neben den physikalischen Eigenschaften dieser Systeme werden die funktionalen Eigenschaften wie z.B. der Verbindungsaufbau und -abbau sowie Eigenschaften der gebräuchlichen Betriebssysteme erläutert. Ebenso wird die Fehlerklasse „Konkurrierende Zugriffe“ beschrieben. Einige beim Test derartiger Systeme aufgetretene Versagen und deren Ursache werden geschildert. Die Ei-

enschaften der Last beim Betrieb und beim Test von Kommunikationssystemen werden dargestellt. Die Betrachtung erfolgt mit einer hohen Abstraktion und konzentriert sich auf die Eigenschaften, welche für den betrachteten Systemaufbau von großer Bedeutung sind.

### **Kapitel 3:** Grundlagen der Modellierung

In diesem Kapitel werden die Grundlagen der Modellierung vorgestellt. Es wird beschrieben, welche Modelle von Kommunikationssystemen als Basis für die weiteren Untersuchungen dienen. Aufgrund der Komplexität leitungsvermittelnder Kommunikationssysteme bilden diese Modelle von Kommunikationssystemen nur die Eigenschaften nach, welche für die weitere Untersuchung von Bedeutung sind. Die Struktur des Kapitels lehnt sich an der Struktur von Kapitel 2 an und umfasst die Abschnitte „Leitungsvermittelnde Kommunikationssysteme“, „Konkurrierende Zugriffe“ sowie „Lastgenerator“.

### **Kapitel 4:** Analytisches Modell

Für das analytische Modell werden einige wenige zusätzliche Annahmen getroffen. Um eine gute Übereinstimmung zwischen dem Analytischen Modell und dem Simulationsmodell zu erlangen, sollten möglichst wenige zusätzliche Annahmen in diesem Kapitel getroffen werden. Diese werden als Modelleinschränkungen bezeichnet. Anschliessend wird die Auftrittshäufigkeit „Konkurrierender Zugriffe“ in Abhängigkeit der Parameter hergeleitet.

### **Kapitel 5:** Simulationsmodell

In diesem Kapitel wird der Aufbau des Simulationsmodells vorgestellt. Das Simulationsmodell modelliert das System, einen Fehler der Fehlerklasse „Konkurrierende Zugriffe“ und den Lastgenerator. Für die Durchführung der Simulation ist es erforderlich, zusätzliche Eigenschaften der Schreib-Lesevorgänge festzulegen. Desweiteren werden zwei verschiedene Arten konkurrierender Zugriffe definiert.

### **Kapitel 6:** Modellauswertungen

In diesem Kapitel werden verschiedene Modelle von Kommunikationssystemen und deren zugehörige Last während der Untersuchung vorgestellt. Bei der Auswertung mit dem analytischen Modell wird die Auftrittshäufigkeit bei unterschiedlicher Anzahl der Knoten im Netz und bei unterschiedlicher Anzahl der Verbindungen berechnet. Ebenso wird der Einfluss der Dauer der Schreib-Lesevorgänge betrachtet und insbesondere ein Augenmerk auf den Einfluss von Systemeigenschaften gelegt. Auch wird die Auftrittshäufigkeit mit Hilfe des Simulationsmodells bestimmt. Anschliessend werden die Auswertungsmethoden „Analy-

se“ und „Simulation“ und deren Ergebnisse für die verschiedenen Werte der Last jeweils mit den berechneten Auftrittshäufigkeiten und den durch Simulationsläufe erhaltenen Auftrittshäufigkeiten verglichen. Die Konfidenzintervalle werden bestimmt. Weitere Erkenntnisse in Abhängigkeit der Art des Kommunikationssystems und der Threadingmechanismen werden gewonnen. Der große Einfluss des Betriebssystems im Rahmen der Untersuchung wird deutlich.

### **Kapitel 7:** Leitlinien zur Durchführung von Lasttests

Es werden die Eigenschaften der Modellierungen mit den Eigenschaften des leitungsvermittelnden Systems verglichen. Eine systematische Vorgehensweise bei der Durchführung von Lasttests wird dargestellt. Eine Zusammenfassung und ein Ausblick werden gegeben.





# Kapitel 2

## Beschreibung des betrachteten Systemaufbaus

Albert Einstein *"The mere formulation of a problem is far more essential than its solution, which may be merely a matter of mathematical or experimental skills."*

In diesem Kapitel werden leitungsvermittelnde Kommunikationssysteme in einer allgemeinen, z.T. auf die für die folgende Untersuchung abstrahierten Weise beschrieben. Zugleich werden aber die Eigenschaften, welche einen Einfluss auf die Untersuchung haben, in einer umfassenden Art und Weise dargestellt. Durch die Konzentration auf die wesentlichen Eigenschaften leitungsvermittelnder Kommunikationssysteme ist es möglich, allgemeingültige Schlussfolgerungen für die Durchführung von Lasttests zu ziehen. Würde keine hinreichende Abstraktion durchgeführt werden, so wären Schlussfolgerungen nur für eine spezielle Art leitungsvermittelnder Kommunikationssystemen möglich.

Nach der Beschreibung der Eigenschaften leitungsvermittelnder Kommunikationssysteme werden aufgetretene Versagensfälle vorgestellt. Das Auftreten dieser Versagensfälle ist aufgrund verschiedener Randbedingungen beim Test der Systeme mit üblichen Verfahren schwer zu erreichen und folglich sind die Fehler auch schwer zu finden.

In dieser Arbeit wird der Lasttest von Kommunikationssystemen untersucht. Deshalb wird anschließend ein in der Industrie verwendeter Lastgenerator beschrieben. Als Lastgenerator wird ein Gerät bezeichnet, welches ein Kommunikationssystem durch verschiedene Anforderungen belastet.

### 2.1 Leitungsvermittelnde Kommunikationssysteme

Ein Kommunikationssystem kann z.B.

- ein Vermittlungssystem

- ein optisches Übertragungssystem oder
- ein Mobilfunksystem (z.B. GSM-System)

sein. Ein Kommunikationssystem besteht aus mehreren Netzelementen. Ein Netzelement ist

- bei einem Vermittlungssystem eine telefonische Vermittlungsanlage
- bei einem optischen Übertragungssystem eine optische Übertragungsanlage
- bei einem Mobilfunksystem eine Mobilvermittlungsanlage.

In Kommunikationsnetzen werden zwei Arten von Informationen übertragen:

- Nutzinformationen
- Vermittlungstechnische Informationen

In der vorliegenden Arbeit werden die vermittlungstechnischen Informationen betrachtet, d.h. die Informationen, welche für den Verbindungsaufbau und -abbau, Verbindungskontrolle sowie Managementfunktionen nötig sind. Die Nutzinformationen, welche zwischen den Verbindungsteilnehmern ausgetauscht werden, sind für diese Untersuchung nicht von Bedeutung.

Ein Kommunikationssystem ist ein verteiltes System. Dies bedeutet einerseits, dass die Zustandsinformationen in den verschiedenen Knoten des Netzes verteilt sind. Andererseits sind die Zustandsinformationen durch die Signallaufzeiten zwischen den einzelnen Knoten auch oft veraltet. Da sich der Netzzustand z.B. durch den Zusammenbruch einzelner Knoten jederzeit ändern kann, ist eine komplette und globale Sicht des Netzes im allgemeinen nicht möglich.

### 2.1.1 Netzgröße von Kommunikationssystemen

Kommunikationssysteme sind hierarchische oder nicht-hierarchische Netze. Es ist von der Art des Systems abhängig, ob ein Netzelement in einem hierarchischen Netz verwendet wird oder nicht. Übertragungssysteme mit einer hohen Übertragungsrate werden in Netzwerken wie z.B. dem WAN (Wide Area Network) in einem nichthierarchischen Netz verwendet. In dieser Arbeit wird ein nichthierarchisches, leitungsvermittelndes Kommunikationssystem betrachtet. Es besteht aus einer Netzebene mit mehreren Netzelementen, welche in Verbindung zu einem oder mehreren benachbarten Netzelementen stehen. Es handelt sich um ein offenes Netz, d.h. die Anzahl der Nachrichten im Netz variiert.

Im realen Betrieb eingesetzte nichthierarchische Kommunikationssysteme sind i.A. Kombinationen von Netzelementen in den Grundformen Maschennetz, Sternnetz und Ringnetz und können aus mehreren Tausend Netzelementen bestehen.



Abbildung 2.1: Optisches Übertragungssystem des Netzbetreibers Global Crossing (vgl. [22])

Abbildung 2.1 zeigt einen Ausschnitt des Übertragungssystems des amerikanischen Netzbetreibers Global Crossing. Bei der Durchführung von Tests beim Systementwickler derartiger Netzelemente werden Kommunikationssysteme aus Kostengründen nur in deutlich kleineren Netzkonfigurationen getestet. Abbildung 2.2 stellt eine mögliche Konfiguration dar. Sind alle Systeme dieser Konfiguration vollbestückt und kostet ein Netzelement ca. 500 000 €, so kostet ein derartiger Testaufbau aus fünf Netzelementen ca. 2,5 Millionen €.

### 2.1.2 Übertragungsmedium zwischen Netzelementen

Das Übertragungsmedium zwischen den Netzelementen ist von der Art des Kommunikationssystems abhängig. Bei Vermittlungssystemen werden als Übertragungsmedium Koaxialkabel oder Kabel mit Kupferdoppeladern verwendet. Bei optischen Übertragungssystemen werden Lichtwellenleiter verwendet. In leitungsvermittelnden Kommunikationssystemen können diese Kabel eine Länge von mehreren Tausend Kilometern haben.

Obwohl Mobilfunksysteme über die Funkschnittstelle kommunizieren, gehören auch sie zu den leitungsvermittelnden Kommunikationssystemen. Beim Test von

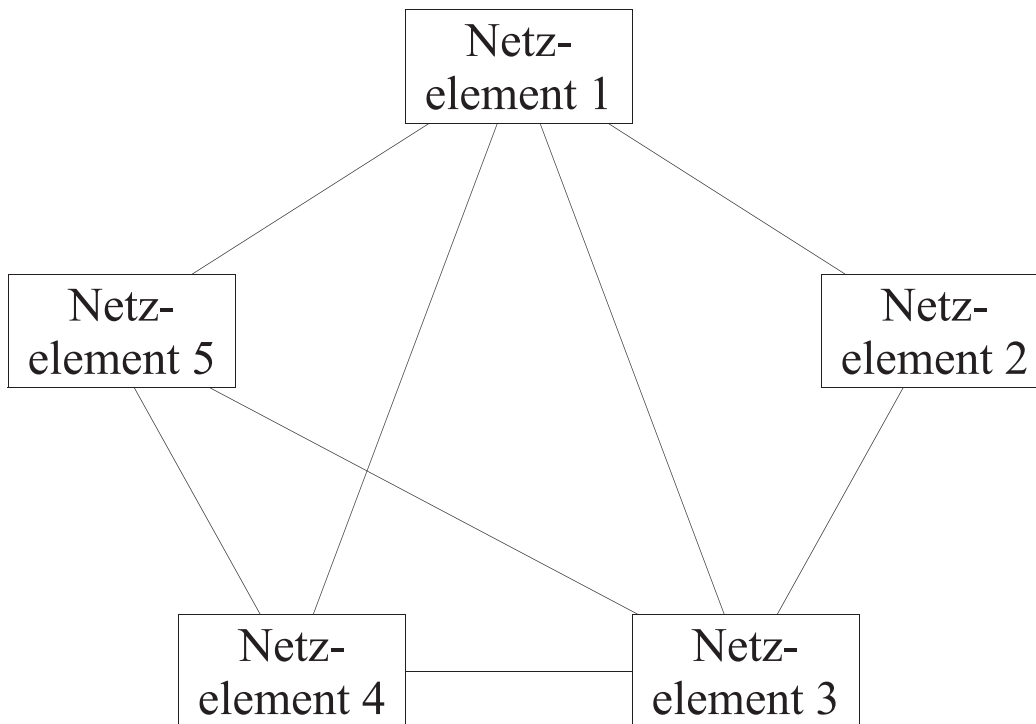


Abbildung 2.2: Topologie eines Kommunikationssystems für die Durchführung von Tests beim Systementwickler

Mobilfunksystemen sind diese sog. „Basisstationen“ meist nur wenige Meter voneinander entfernt. Es werden jedoch ebenso Tests durchgeführt, bei welchen die Entfernungen zwischen den Basisstationen einen Abstand von mehreren Kilometern haben und die daraus resultierenden Funkzellen den realen Zellengrößen entsprechen.

Zur Beschreibung eines Kommunikationssystems ist es jedoch nicht nur von Bedeutung, ob eine physikalische Verbindung zwischen zwei Systemen existiert. Aufgrund physikalischer Restriktionen ist ebenso die Leitungskapazität von großer Bedeutung. Die Leitungskapazität einer Verbindung gibt an, wieviele dynamische Verbindungen gleichzeitig über diese physikalische Verbindung geführt werden können. Bei der Vermittlungstechnik wird durch die Leitungskapazität angegeben, wieviele 64 kbit-Kanäle zwischen zwei Knoten für ISDN-Verbindungen zur Verfügung stehen.

### 2.1.3 Dynamische Verbindungen zwischen Netzelementen

Die Leitungsvermittlung ist eine Vermittlungstechnik, bei der zwischen Sender und Empfänger eine physikalische Leitung über ein Netzelement oder mehrere Netzelemente geschaltet wird. Eine Quelle fordert vom Netz die Herstellung einer durchgängigen Verbindung zu einer gewünschten Senke an. Anschliessend kön-

nen hierüber Nachrichten bis zur Freigabe der Verbindung ausgetauscht werden. Die Leitungsvermittlung wird auch als Durchschaltevermittlung (engl.: circuit switching) bezeichnet. Es können sowohl unidirektionale als auch bidirektionale Verbindungen geschaltet werden. Für die Datenaustauschphase ist bei der Leitungsvermittlung eine feste Punkt-zu-Punkt Verbindung zwischen dem Sender und Empfänger geschaltet. Sie dient der Echtzeitübertragung z.B. von Sprache, Daten oder Video. Die Verbindungen haben eine garantierte Dienstgüte bezüglich Datenrate und Verzögerung. Diese Eigenschaften hängen von der Leitung selbst und nicht von der Netzbelastung ab.

Beim Verbindungsaufbau wird ein fester Leitungsweg gesucht. Dieser wird für die gesamte Dauer der Verbindung reserviert und ausschließlich für die Kommunikation zwischen den beteiligten Partnern genutzt. Die Verbindung steht nach dem Verbindungsaufbau bis zum Verbindungsabbau mit ihrer vollen Übertragungsbandbreite zur Verfügung. Das bedeutet, dass bei der Leitungsvermittlung die nicht benötigte Bandbreite ungenutzt verloren geht. Bei der Leitungsvermittlung fordert ein Sender die Herstellung einer durchgängigen Verbindung zu einem bestimmten Empfänger an und kann nach erfolgreichem Verbindungsaufbau hierüber Daten bis zum Abbau der Verbindung austauschen. Bereits beim Verbindungsaufbau werden Reservierungen vorgenommen. Eine Verbindung über mehrere Netzelemente wird sukzessive, ausgehend vom anfordernden Netzelement, aufgebaut. Der Verbindungsabbau nach Ablauf der Datenaustauschphase kann sowohl vom Sender als auch vom Empfänger initiiert werden.

Der Startknoten einer Verbindung wird auch als A-Seite, der Zielknoten einer Verbindung wird bei Vermittlungssystemen i.A. als B-Seite, bei Übertragungssystemen i.A. als Z-Seite bezeichnet. Im Folgenden wird der Zielknoten einer Verbindung, unabhängig von der Art des Kommunikationssystems, als Z-Seite bezeichnet.

Beim Verbindungsaufbau werden Teilstrecken auch dann reserviert, wenn eine Verbindung letztendlich nicht zustande kommt, weil für eine spätere Teilstrecke keine Leitungskapazität zur Verfügung steht. Ist dies der Fall, so wird die Verbindungsanfrage negativ beantwortet, und die bisher aufgebaute Teilstrecke wieder abgebaut. Es wird kein weiterer Versuch eines Verbindungsaufbaus automatisch initiiert. Der Verbindungsabbau kann sowohl von der A-Seite als auch von der Z-Seite initiiert werden. Die Verbindungen zwischen den einzelnen Knoten werden der Reihe nach abgebaut, die Ressourcen werden freigegeben.

#### **2.1.4 Kommunikation zwischen Netzelementen**

Die Kommunikation zwischen Netzelementen erfolgt über Nachrichten. Sie ist unabhängig von der Leitungskapazität. Existiert eine beliebige Route zwischen zwei Netzelementen eines Kommunikationssystems aufgrund physikalischer Verbindungen, so ist es möglich, zwischen diesen Netzelementen Nachrichten zu versenden. Die Bearbeitung von Nachrichten erfolgt in der Reihenfolge des Eintref-

fens der Nachrichten unter Beachtung der Priorität der Nachrichten. Dies bedeutet, dass Nachrichten niedriger Priorität erst dann bearbeitet werden, wenn keine Nachricht einer höheren Priorität im Netzelement unbearbeitet ist. Nachrichten mit der Information über Signalisierungsfehler haben eine höhere Priorität als z.B. Nachrichten des Bedienrechners. Diese Nachrichten können z.B. Umschaltungen der geschalteten Verbindungen nach sich ziehen, d.h. die Verbindungsdaten werden geändert, die Verbindung wird umgeschaltet.

## 2.1.5 Arten von Netzelementen

Vermittlungssysteme und optische Übertragungssysteme werden im Folgenden als typische Repräsentanten leitungsvermittelnder Kommunikationssysteme näher untersucht.

### 2.1.5.1 Vermittlungssysteme

Vermittlungssysteme sind z.B. das analoge Vermittlungsnetz oder das ISDN-Netz. Im Folgenden werden nur Systeme mit Outbound-Signalisierung betrachtet. Diese Systeme haben eigene Übertragungswege, welche von den Nutzkanälen unabhängig sind. Diese werden als Zeichenkanäle bezeichnet und sind von der ITU-T als Zeichengabesystem Nr. 7 (engl.: Signalling System No. 7, SS7) in internationalen Empfehlungen beschrieben. Bei diesen Kommunikationssystemen werden i.A. Verbindungen mit kurzer Datenaustauschphase aufgebaut, an welchen meist nur zwei Teilnehmer beteiligt sind. Bei der Übermittlung von Sprache entspricht die Datenaustauschphase der Gesprächsdauer. Die durchschnittliche Dauer einer solchen Datenaustauschphase liegt bei 120 s (vgl. *Siegmund* in [61]). Weitere Aussagen zu Eigenschaften der Gesprächsdauer im realen Netz wurden auch von *Duffy et al.* in [16] gemacht. Durch die Art der Nutzung kommt es in jedem Vermittlungssystem zu sehr vielen Schaltvorgängen pro Tag. Der Schaltvorgang wird vom Gesprächsteilnehmer durch die Wahl der Telefonnummer initiiert. Der Aufbau der Verbindung und die Wahl der Route ist bei Vermittlungssystemen abhängig von der Rufnummer des gewünschten Anschlusses. Abbildung 2.3 beschreibt einen erfolgreichen Verbindungsaufbau einer ISDN-Verbindung.

Vermittlungssysteme haben die folgenden Eigenschaften:

- Die Verbindungen haben kurze Datenaustauschphasen.
- Es finden im Durchschnitt viele Verbindungsaufbauten und -abbauten pro Zeiteinheit statt.
- Zu Spitzenzeiten ist eine extrem hohe Anzahl von Verbindungsaufbauten und -abbauten möglich.
- Beim Ausfall einer Verbindung sind meist nur zwei Nutzer betroffen.

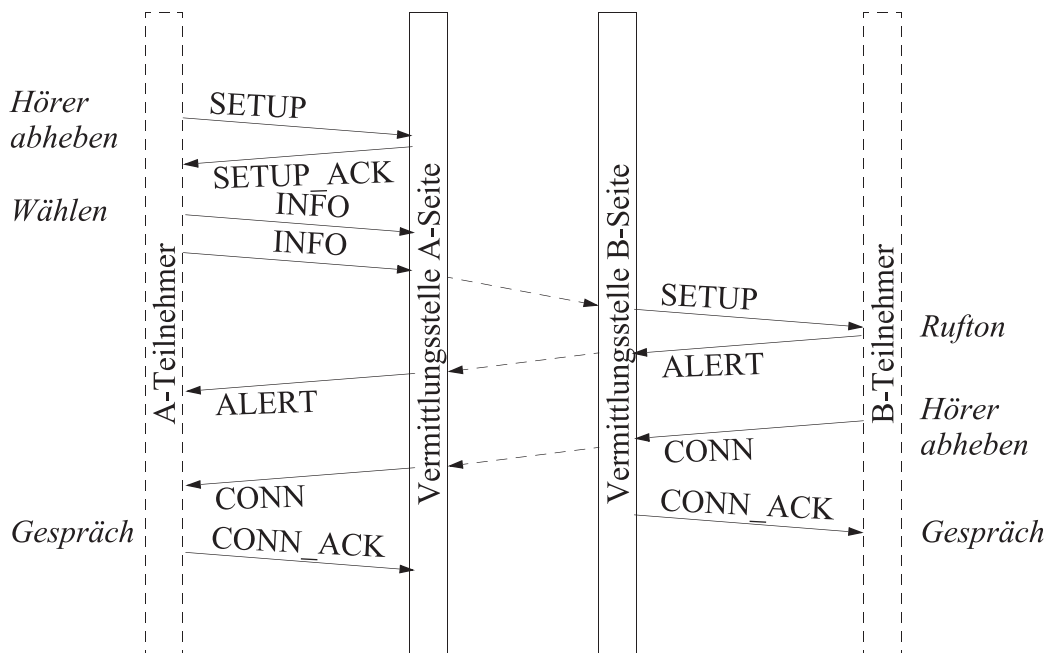


Abbildung 2.3: Aufbau einer ISDN-Verbindung

### 2.1.5.2 Optische Übertragungssysteme

Optische Übertragungssysteme stellen das Übertragungsmedium z.B. für das Vermittlungssystem zur Verfügung. So wird z.B. bei SDH/SONET-Systemen (Synchrone Digitale Hierarchie/Synchronous Optical Network) oder DWDM-Systemen (Dense Wavelength Division Multiplex) über Glasfaserstrecken die Übertragung von Signalisierungsinformationen ermöglicht, die Signalisierungsinformationen der Teilnehmer jedoch nicht ausgewertet. Dabei handelt es sich bei SDH/SONET um ein Zeitmultiplexverfahren, während es sich bei DWDM um ein Wellenlängenmultiplexverfahren handelt. Bei DWDM-Systemen ist die Übertragungsrates bis zu 40 GBit/Sekunde. So können bei 40 GBit Systemen gleichzeitig 500 000 Telefongespräche übertragen werden. Die Einweglaufzeiten von Lichtwellenleitern, wie sie heutzutage bei Übertragungssystemen üblich sind, betragen  $5\mu\text{s}/\text{km}$  und sind physikalisch bedingt durch die Lichtgeschwindigkeit in Glas von ca. 200 000 km/s.

Im Gegensatz zu Vermittlungssystemen werden in Übertragungssystemen nur sehr wenige Schaltvorgänge pro Monat oder pro Jahr durchgeführt. Kommt es zu einem Ausfall einer Verbindung in einem optischen Übertragungssystem, sind alle Nutzer dieser geschalteten Verbindung betroffen. Dies bedeutet, dass der Ausfall einer einzigen Verbindung sehr viele Nutzer des Übertragungsmediums betreffen kann.

Übertragungssysteme haben somit die folgenden Eigenschaften:

- Die Verbindungen haben sehr lange Datenaustauschphasen.



- Es werden vom Personal des Netzbetreibers nur sehr wenige Verbindungsaufbauten und -abbauten pro Zeiteinheit durchgeführt.
- Eine Verbindung hat eine sehr hohe Übertragungsrate.
- Beim Ausfall einer Verbindung ist die Anzahl der betroffenen Nutzer des Übertragungsmediums sehr hoch.

Handelt es sich bei dem Übertragungssystem um ein sog. „Automatic Switched Optical Network“, so werden bei Übertragungsfehlern automatisch Ersatzschaltungen durchgeführt.

### 2.1.6 Aufbau eines Netzelements

Ein Netzelement besteht aus mehreren Baugruppen unterschiedlicher Anzahl:

- Einem Systemcontroller.
- Einem Koppelfeld.
- Mehreren Schnittstellenbaugruppen verschiedener Typen.

Eine typische Systemgröße für Netzelemente sind 32 Schnittstellenbaugruppen. Gemeinsam mit dem Systemcontroller und dem Koppelfeld ergibt sich somit ein System mit 34 Prozessoren.

Der Aufbau eines Netzelements ist exemplarisch in Abbildung 2.4 dargestellt.

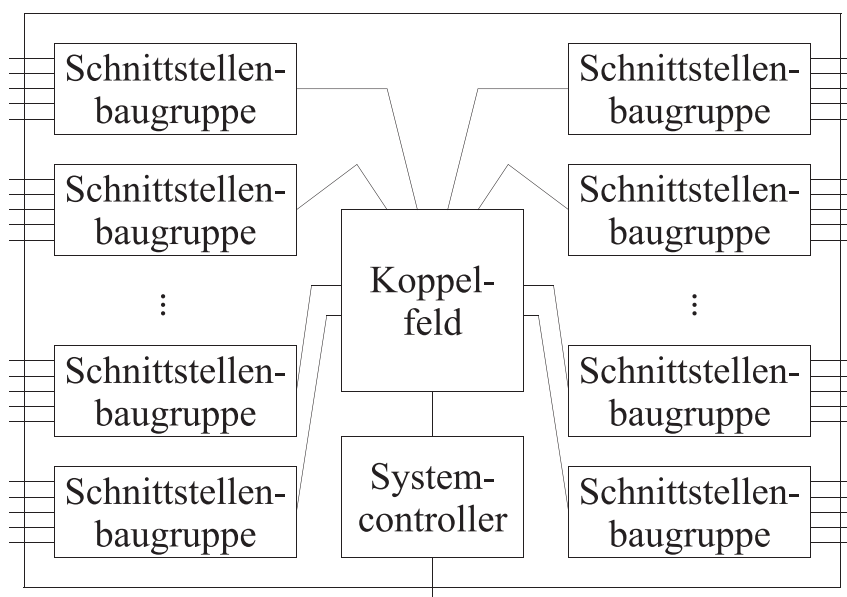


Abbildung 2.4: Aufbau eines Netzelements

Ein Netzelement oder auch mehrere Netzelemente werden durch einen Bedienrechner gesteuert. Durch diesen ist es möglich, Protokollnachrichten an das Netzelement zu senden.

Kommunikationssysteme müssen eine sehr hohe Verfügbarkeit haben. Wartungsarbeiten oder Erweiterungen des Systems durch das Hinzufügen einer Baugruppe dürfen keinerlei Störungen des Betriebs des Kommunikationssystems verursachen und zu keinem Neustart (engl.: reboot) des Systems führen.

Zur Gewährung dieser Eigenschaft besitzt jede Baugruppe einen eigenen Prozessor (engl.: CPU). Die Interprozesskommunikation aller Baugruppen erfolgt über Nachrichten und serielle Verbindungen. Die Nachrichten werden gemäß einem Protokoll gesendet. Beim Verlust von Nachrichten werden diese durch sog. „Retries“ wiederholt. Die Reihenfolge der Bearbeitung eintreffender Nachrichten ist bei manchen Anwendungen prioritätsabhängig. So werden eintreffende Nachrichten vom Bedienrechner des Kommunikationssystems erst dann bearbeitet, wenn keine Nachrichten aufgrund von Signalisierungsfehlern mehr zur Bearbeitung anstehen. Da jede Baugruppe eine eigene CPU besitzt, ist die Rechenleistung des Systems von der Bestückung unabhängig und somit konstant. Beim Ausfall einer Baugruppe ist nur diese Baugruppe betroffen. Alle anderen Baugruppen des Systems können weiter ihre Funktionalität erfüllen. Somit ist eine hohe Ausfallsicherheit gegeben.

Ebenso besitzt jedes Netzelement eine Datenbank, um u.a. jeweils die Daten der Verbindungen zwischen einem Eingang der einen Schnittstellenbaugruppe durch das Koppelfeld zum Ausgang einer anderen Schnittstellenbaugruppe in diesem Netzelement zu speichern.

### 2.1.7 Systemfunktionen

In Kommunikationssystemen sind abhängig von der Anwendung verschiedene Systemfunktionen realisiert. Der Verbindungsaufbau und -abbau wurde bereits in den Abschnitten 2.1.3 und 2.1.5 erläutert.

Nun werden exemplarisch noch drei weitere Systemfunktionen näher betrachtet.

- Überwachung der Signalgüte: Es wird das Signal einer geschalteten Verbindung überwacht. Unterschreitet die Signalgüte eine bestimmte Schranke, so wird dies in der Datenbank gespeichert und dem Betreiber ein Alarm gemeldet.
- Durchführung von Ersatzschaltungen: Bei Auftreten von Signalfehlern für eine Verbindung wird das Signal auf eine neue, bereits eingerichtete Verbindung geschaltet und von der ursprünglichen Verbindung weggeschaltet. Die Datenbankeinträge der Verbindungsdaten ändern sich entsprechend.
- Übertragung von Gebührenimpulsen: In regelmäßigen Abständen werden Gebührenimpulse übertragen und in die Datenbank der Verbindungsdaten eingetragen.

### 2.1.8 Betriebssystemeigenschaften

Bedingt durch die vorgestellten Eigenschaften von Kommunikationssystemen werden als Betriebssystem sog. „Echtzeit-Betriebssysteme für Multithreading-Systeme“ verwendet. Derartige Betriebssysteme sind meist kommerzielle Betriebssysteme. Typische Repräsentanten derartiger Betriebssysteme sind *Chorus*, *VxWorks* und *psos*. Echtzeit-Betriebssysteme unterscheiden sich bzgl. der Threadingmechanismen, bzgl. des Adressraums und des Zeitverhaltens von Betriebssystemen wie Windows und Unix. Sie erfüllen die folgenden Threading-Eigenschaften:

- Jeder Verbindungsaufbau wird durch einen eigenen Thread ausgeführt.
- Der Thread mit höchster Priorität, welcher auf die Ausführung wartet, wird als erstes ausgeführt.
- Besitzen mehrere Threads die höchste Priorität und warten auf die Ausführung, so werden diese im Scheduling-Verfahren „Round Robin“ ausgeführt.
- Der Scheduling Mechanismus ist statisch, d.h. wenn ein Thread kreiert wird, wird ihm eine bestimmte Priorität zugewiesen und diese wird niemals geändert.

Bei Echtzeit-Betriebssystemen befinden sich alle Threads und alle Variablen in einem einzigen großen Adressraum. Dies ist aus den folgenden Gründen sehr effizient:

- Kommt es zu einem Prozessor-Switch, so muss der Adressraum nicht gewechselt werden.
- Die Designmethoden sind durch die sog. „Shared Data“ sehr effizient.

Das Design von Echtzeit-Betriebssystemen garantiert, dass die Verarbeitungsergebnisse innerhalb einer bestimmten Zeitspanne zur Verfügung stehen. D.h. wichtige Betriebssystemfunktionen haben garantierte maximale Laufzeiten. Diese maximalen Laufzeiten sind von der Anzahl der Objekte (wie z.B. Threads) unabhängig.

Für verteilte Systeme werden Synchronisationsmechanismen verwendet, um sicherzustellen, dass kooperierende Prozesse eine konsistente Sicht des Systemzustands haben. Durch diese Synchronisationsmechanismen können Zugriffe koordiniert und Ressourcen gemeinsam genutzt werden. Ebenso ist es möglich, Ereignisse in einer gewissen spezifischen Reihenfolge zu erzwingen. Um die Sequentialität zu erzwingen, werden bei vielen Kommunikationssystemen atomare (unteilbare) Operationen durchgeführt. Auf diese Weise können Aktionen, welche sich nicht

stören, gleichzeitig vorkommen, während sich Aktionen, welche im Konflikt miteinander sind, nur sequentiell ereignen können. Für die Realisierung der Sequentialität sind verschiedene Mechanismen wie z.B. Konkurrenzkontrolle erforderlich. Neben der Verteiltheit und Echtzeitabhängigkeit der Software von Kommunikationssystemen ergibt sich ebenso durch die Nebenläufigkeit die fehlende Wiederholbarkeit (Nichtreproduzierbarkeit) von Programmabläufen.

Die fehlende Wiederholbarkeit von Programmabläufen wird durch die folgende Abbildung 2.5 (vgl. *Herrtwich, Hommel* in [27]) ersichtlich. Bestehen nebenläufige Anweisungen aus mehreren atomaren Operationen und werden diese Anweisungen ausgeführt, dann kann jede atomare Operation der einen Ausführung sequentiell zu jeder atomaren Operation der anderen Anweisung erfolgen. Die Sequentialitätsbeziehungen der atomaren Operationen bei der Ausführung einer Anweisung bleiben dabei gewahrt.

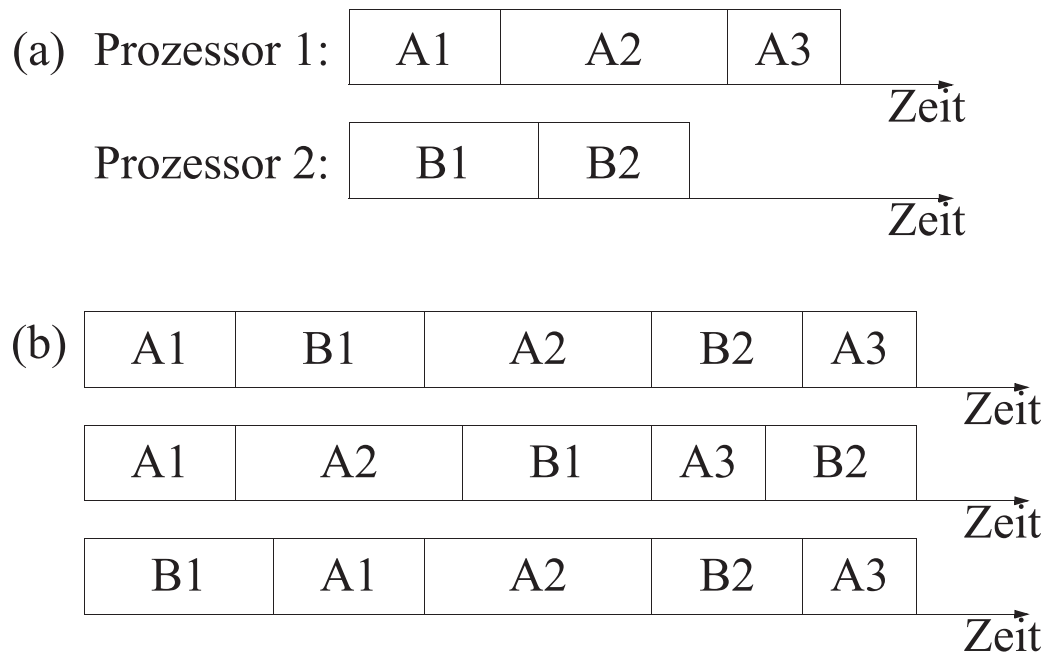


Abbildung 2.5: Ausführungsreihenfolgen zweier nebenläufiger Anweisungen, (a) parallel, (b) verzahnt

In Abbildung 2.5 sind verschiedene Ausführungsarten zweier nebenläufiger Anweisungen dargestellt. Im Teil (a) dieser Abbildung ist die parallele Ausführung der atomaren Operationen A1, A2 und A3 bzw. B1 und B2 gezeigt. Diese werden auf zwei Prozessoren durchgeführt. Teil (b) zeigt drei verschiedene Ausführungsreihenfolgen bei einer verzahnten Ausführung der atomaren Operationen A1, A2 und A3 sowie B1 und B2. Diese dargestellten Ausführungsreihenfolgen sind nur

ein kleiner Teil der potentiell möglichen Ausführungsreihenfolgen dieser Anweisungen und werden - im Gegensatz zu Teil (a) - auf nur einem Prozessor durchgeführt.

Die Vielfalt der Ausführungsreihenfolgen ist bei diesem einfachen Beispiel offensichtlich. Die Applikationssoftware eines Kommunikationssystems besteht aus bis zu 3 Millionen unkommentierter Netto-Codezeilen (NCSL). Diese Software hat folglich aus Effizienzgründen auch eine beachtliche Anzahl atomarer Operationen, was zu einer enormen Vielzahl möglicher verschiedener Sequenzen bei derartigen Systemen führt. Es ist folglich unmöglich, vollständige Tests derartiger Systeme durchzuführen.

## 2.2 Konkurrierende Zugriffe

Im Folgenden werden beispielhaft verschiedene Versagensfälle beschrieben, welche beim Test leitungsvermittelnder Kommunikationssysteme aus dem Bereich der Übertragungstechnik aufgetreten sind. Es kann zu Fehlern aufgrund konkurrierender Zugriffe kommen, wenn mindestens zwei Prozesse gleichzeitig durch mehrere nicht-atomare Operationen die gleichen Daten bearbeiten und diese gleichzeitige Bearbeitung nicht durch einen Semaphor verhindert wird. Diese konkurrierenden Zugriffe werden auch als „race condition“ bezeichnet. Sie können zu fehlerhaften Ergebnissen führen, wenn nebenläufige Prozesse unkoordiniert den Inhalt einer gemeinsamen Variablen modifizieren.

Erfolgen diese nicht-atomaren Operationen hintereinander, d.h. hat der eine Prozess die Operationen auf den Daten vollständig durchgeführt, bevor der andere Prozess diese Daten bearbeitet, so tritt kein Fehler auf. Führt jedoch ein Prozess die Operationen auf den gleichen Daten durch, obwohl der andere Prozess auf diesen Daten die Operationen noch nicht beendet hat und wird dieser Prozess unterbrochen, so kommt es zu einem Fehler. Das Auftreten derartiger Fehler ist somit zufällig, d.h. stochastisch, und tritt umso häufiger auf, je öfter diese Operationen durchgeführt werden. Ein Fehler aufgrund konkurrierender Zugriffe tritt bei Einzeltests (d.h. bei einmaliger Durchführung der Operationen) folglich nicht auf.

Kommunikationssysteme sind Echtzeitsysteme und müssen harte Zeitbedingungen erfüllen, um auf unvorhergesehene oder katastrophale Ereignisse unverzüglich reagieren zu können. Es wird bei diesen Systemen mit harten Zeitbedingungen bei allen Funktionen mit hoher Priorität wenn immer möglich vermieden, nebenläufige Prozesse zu koordinieren. D.h. es gibt keine generelle Konkurrenzkontrolle, wie sie durch bestimmte Werkzeuge möglich wäre. Der Software-Entwickler muss somit bei jedem Datenzugriff entscheiden, ob ein Semaphor notwendig ist. Eine Verwendung von Semaphoren für jeden Datenzugriff ist nicht sinnvoll. Durch die restriktive Verwendung von Semaphoren ist es möglich, dass der Entwickler einen benötigten Semaphor vergisst. Durch konkurrierende Zugriffe kann es

zu Speicherzugriffsfehlern kommen, welche sehr auffällig sind. Aber ebenso kann es zu falschen Einträgen in der Datenbank kommen, welche nicht so leicht zu erkennen sind.

### 2.2.1 Aufgetretene Versagen

Durch die fortlaufende Nummerierung der nun folgenden Beschreibungen der Versagen ist es möglich, diese im weiteren Teil der Arbeit zu referenzieren.

**Versagen 2.1** Es werden gleichzeitig mehrere Baugruppen eines Kommunikationssystems mehrmals hintereinander automatisch eingerichtet und anschließend wieder entfernt. Dadurch werden innerhalb eines kurzen Zeitintervalls sehr viele Nachrichten im System empfangen. Da ein Schutz vor konkurrierenden Zugriffen fehlt, werden bei diesem Ablauf nicht alle für die Nachrichtbearbeitung allokierten Speicherbereiche zurückgegeben. Als Folge steht nach mehrmaliger automatischer Einrichtung der Baugruppen für die Nachrichtbearbeitung kein Speicher mehr zur Verfügung. Bei der Implementierung des betrachteten Kommunikationssystems führte dies zu einem endlosen reboot.

**Versagen 2.2** Die Dauer einer Signalstörung einer Verbindung kann durch die Einrichtung von Ersatzschaltungen minimiert werden. Kommt es bei einer oder mehreren dieser besonders geschützten Verbindungen zu Signalstörungen, so werden alle betroffenen Verbindungen auf störungsfreie Ersatzverbindungen umgeschaltet. Um dies innerhalb von Millisekunden durchzuführen, werden besonders große Nachrichten erstellt, welche durch bis zu 32 Prozesse gleichzeitig beschrieben werden. Da diese Prozesse nicht untereinander vor zeitgleichen Datenzugriffen geschützt sind, kommt es bei der Erstellung einer derartigen großen Nachricht zu falschen Einträgen und daraus resultierend zu einem Speicherzugriffsfehler.

**Versagen 2.3** Es wird der übliche Betrieb eines Kommunikationssystems nachgebildet, d.h. es werden Verbindungen eingerichtet und wieder entfernt, es werden Ersatzschaltungen eingerichtet und durchgeführt. Nach einer gewissen Betriebsdauer kommt es zu korrupten Daten. Aufgrund dieser korrupten Daten kommt es zu einem Speicherzugriffsfehler, welcher zu einer sog. „recovery“ führt.

Die Analyse der Fehlerursache zeigte, dass konkurrierende Zugriffe diese Versagensfälle verursachten. Wie an diesen Beispielen ersichtlich ist, sind die Auswirkungen konkurrierender Zugriffe sehr vielfältig. Aus den Symptomen heraus lässt sich häufig nicht auf die Ursache schließen.

### 2.2.2 Beschreibung der Ursache eines Versagens

Damit es zu einem konkurrierenden Zugriff zweier Prozesse (im folgenden als Prozess A und Prozess B bezeichnet) in einem Netzelement kommt, müssen folgende Bedingungen erfüllt sein:

1. räumliches Zusammentreffen: Die Zugriffe von Prozess A und Prozess B müssen den gleichen Eintrag in der Datenbank desselben Netzelements verändern.
2. zeitliches Zusammentreffen: Die Zugriffe der Prozesse müssen zeitnah zueinander stattfinden. Der Zugriff von Prozess A muss vor der Beendigung der Datenänderung durch einen Prozessorswitch unterbrochen werden. Anschliessend ändert Prozess B dieselben Daten. Frühestens nach dem nächsten Prozessorswitch setzt Prozess A seine geplanten Datenänderungen fort.

Da diese beiden Bedingungen i.A. nur sehr selten gemeinsam erfüllt sind, treten konkurrierende Zugriffe nur sehr selten auf. Es ist somit verständlich, warum die beschriebenen Versagen nur nach speziellen Tests aufgetreten sind.

Im Folgenden wird exemplarisch die Ursache des Versagens 2.3 beschrieben. Der Ausgangspunkt ist eine Liste (siehe Abbildung 2.6). Die Einträge der Liste enthalten z.B. Verbindungsdaten. Diese Verbindungsdaten sind durch Semaphore vor unsynchronisierten Speicherzugriffen geschützt, der Zeiger der Liste ist jedoch nicht vor unsynchronisierten Speicherzugriffen geschützt. Der Prozess A will das Element 27 an den Kopf der Liste anfügen. In Abbildung 2.7 ist zu sehen, wie das neue Element 27, welches von Prozess A erzeugt wurde, einen Zeiger auf die Liste setzt. Nun kommt es zum Prozessorswitch. Prozess B will Element 28 zur Liste hinzufügen.

In Abbildung 2.8 wird gezeigt, wie das neue Element 28 von Prozess B einen Zeiger auf die Liste setzt. Abbildung 2.9 zeigt, wie anschließend der Kopf der Liste durch Prozess B geändert wird, damit er auf das neue Listenelement 28 zeigt und damit die Liste effektiv um den neuen Eintrag erweitert. Nun kommt es zum Prozessorswitch zurück zu Prozess A. Wie in Abbildung 2.10 ersichtlich wird, wird nochmals der Kopf der Liste geändert. In Abbildung 2.11 ist zu sehen, dass das von Prozess B eingetragene Listenelement 28 verloren ist.

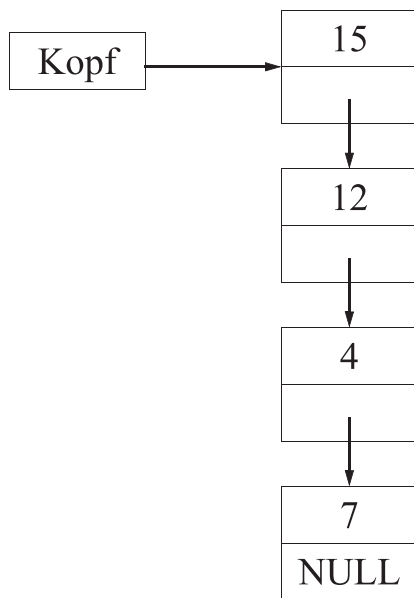


Abbildung 2.6: Verkettete Liste

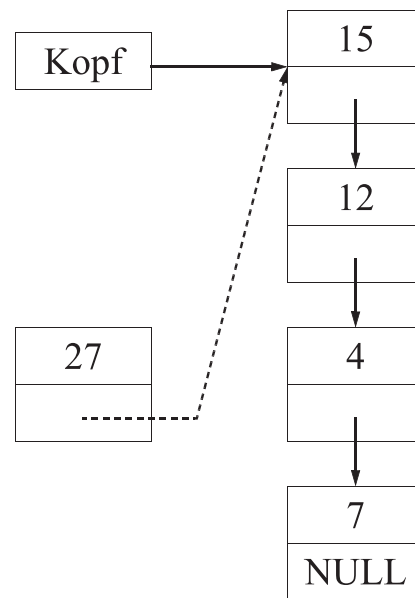


Abbildung 2.7: Prozess A setzt Zeiger von neuem Element 27 auf die Liste

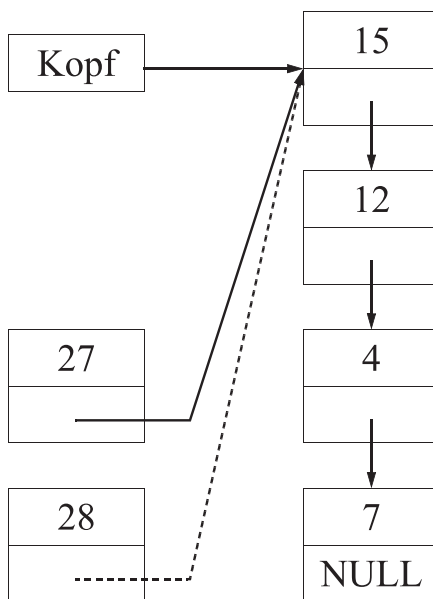


Abbildung 2.8: Prozess B setzt Zeiger von neuem Element 28 auf die Liste

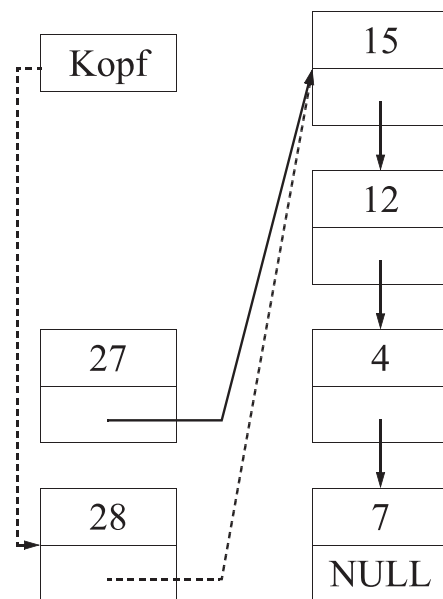


Abbildung 2.9: Kopf der Liste wird durch Prozess B auf Listenelement 28 geändert



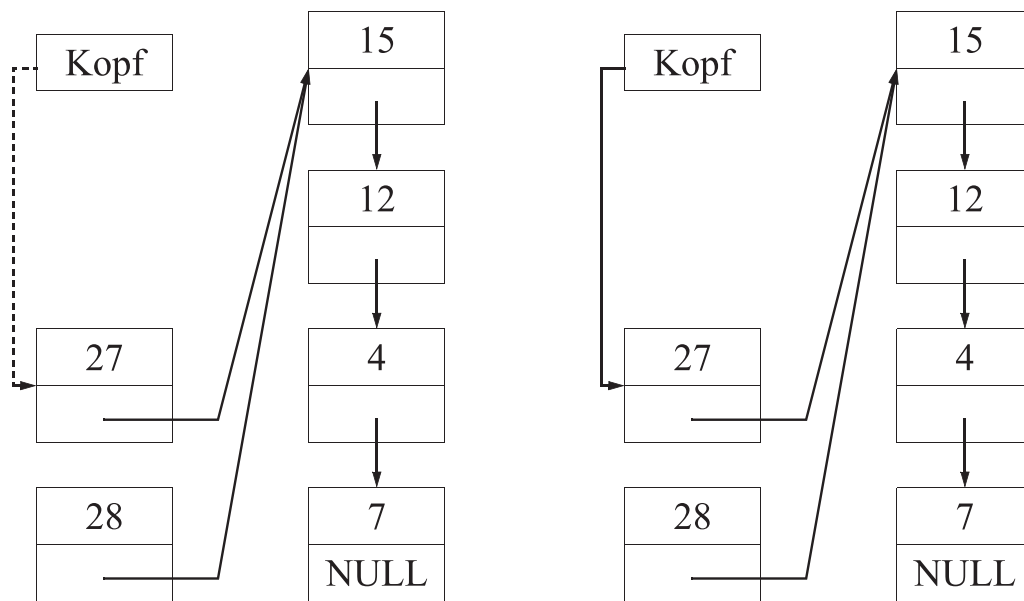


Abbildung 2.10: Kopf der Liste wird durch Prozess A auf Listenelement 27 geändert

Abbildung 2.11: Element 28 von Prozess B ist nicht mehr Teil der Liste

```

1 fuegeElementAnListenkopfAn(zeigerNeuesElement)
2 {
3     zeigerNeuesElement.NaechsterEintrag = Liste.Kopf;
4     Liste.Kopf = zeigerNeuesElement;
5 }

```

Pseudocode 2.1: Fehlerhafter Prozedur zum Eintrag eines neuen Listenelements

```

1 fuegeElementAnListenkopfAn(zeigerNeuesElement)
2 {
3     lock (semaphor)
4     zeigerNeuesElement.NaechsterEintrag = Liste.Kopf;
5     Liste.Kopf = zeigerNeuesElement;
6     unlock (semaphor)
7 }

```

Pseudocode 2.2: Fehlerfreie Prozedur zum Eintrag eines neuen Listenelements

Pseudocode 2.1 beschreibt die fehlerhafte Prozedur zum Eintrag eines neuen Listenelements, während hingegen Pseudocode 2.2 die fehlerfreie Prozedur beschreibt.

## 2.3 Lastgenerator

Wie bei der Beschreibung von aufgetretenen Versagen deutlich wurde, treten diese Versagen häufiger auf, wenn das System einer bestimmten Last ausgesetzt wird. Im Folgenden wird nun beschrieben, welche Arten von Last mit Hilfe eines Lastgenerators erzeugt werden können und welche Eigenschaften dieser Lastgenerator üblicherweise hat.

### 2.3.1 Beschreibung der Last

Beim Lasttest leitungsvermittelnder Kommunikationssysteme wird die Last auf drei verschiedene Arten erzeugt.

Durch den Lastgenerator wird z.B. das Verhalten vieler Nutzer simuliert, d.h. es wird z.B. bei Systemen der Vermittlungstechnik gesteuert, wie viele Teilnehmer ein Telefongespräch aufbauen wollen und wie lange dieses Telefongespräch nach einem erfolgreichen Verbindungsaufbau andauert. Ebenso werden sehr viele Protokollnachrichten generiert und dadurch z.B. Ersatzschaltungen eingerichtet. Als weitere Möglichkeit zur Generierung von Last werden Wartungsaufgaben von

Baugruppen simuliert und dafür diese Baugruppen manuell und ggf. auch in der Datenbank entfernt und neu eingerichtet. Die Entfernung von Baugruppen erzeugt eine sehr hohe Anzahl von Alarmen in den entsprechenden Netzelementen. Sie kann - je nach Konfiguration - zur Durchführung von Ersatzschaltungen führen. Last wird somit durch die gehäufte Anwendung bestehender Anwendungen erzeugt.

Das Aufkommen von Telefongesprächen hängt von Faktoren wie z.B. der Tageszeit und dem Wochentag ab. Zu einem besonders hohen Verkehrsaufkommen kommt es beispielsweise zum Jahreswechsel um Mitternacht. Aus diesem Grund ist das Telefonnetz innerhalb von ein bis zwei Stunden einer enormen Belastung ausgesetzt. Kommunikationssysteme müssen derartige Spitzen (engl.: bursts) fehlerfrei bewältigen.

Lange Zeit wurde das reale Lastaufkommen bei Untersuchungen nicht genügend berücksichtigt. So gab es nach Willinger et al. (vgl. [80]) im Zeitraum von 1966 bis 1987 nur ca. 50 Veröffentlichungen, welche auf Messungen des Verkehrsaufkommens bei Fernsprechverkehr basieren. Im Gegensatz dazu gab es im gleichen Zeitraum jedoch mehrere Tausend Veröffentlichungen über Warteschlangenprobleme. In der jüngeren Vergangenheit wurden jedoch große Mengen von Daten über das Verkehrsaufkommen verschiedener existierender Netzwerke gesammelt. Diese Daten basieren auf Messungen über Stunden, Tage oder Wochen und charakterisieren das Verkehrsaufkommen.

Abbildung 2.12 zeigt das Ergebnis der Messung, welche von Duffy et al. (vgl. [16]) im Jahre 1993 an einem realen Netz durchgeführt wurde. Es zeigt die Verkehrsdaten eines CCSN (Common Channel Signaling Network), durch welches u.a. SS7 (Signaling System Number 7) Protokoll Daten gelenkt werden. Das SS7 Protokoll steuert die Signalisierung für den Auf- und Abbau von 64-kbit/s-Nutzkanalverbindungen und von ISDN-Diensten. Es wird für den Austausch von Signalisierungsnachrichten zwischen digitalen Vermittlungsstellen eingesetzt. In dieser Messung sind die ankommenden Verbindungsanforderungen pro 10 Sekunden dargestellt, welche in einer 4tägigen Messung ab Mittwoch dem 20. Januar 1993 um 22.23 Uhr stattfanden. Wie leicht zu sehen ist, ist das Verkehrsaufkommen bei dieser Messung am Wochenende deutlich geringer als an den Werktagen Donnerstag und Freitag. Ebenso kann die unterschiedliche Last zu den verschiedenen Tageszeiten gut erkannt werden.

Last kann somit z.B. durch vermittlungstechnische Aufgaben wie dem Verbindungsaufbau und -abbau oder durch die Unterbrechung von Leitungen zwischen Netzelementen und die dadurch nötige Alarmbehandlung entstehen.

### 2.3.2 Eigenschaften eines Lastgenerators

Der Lastgenerator ist ein speziell entwickeltes eingebettetes System und wird beim Test von Kommunikationssystemen verwendet. Er kann Verbindungsanforderungen mit einer bestimmten Rate an das zu testende Kommunikationssystem

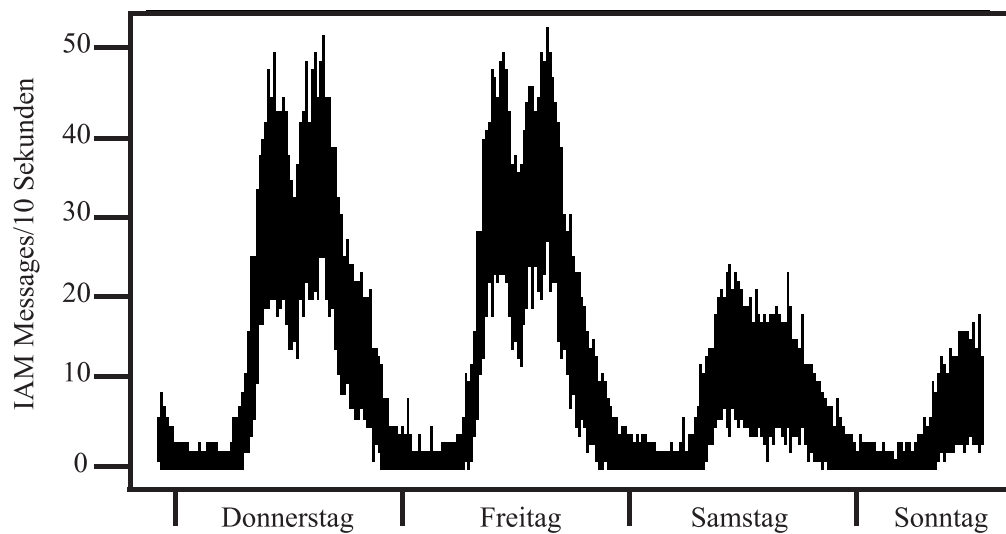


Abbildung 2.12: Verkehrsaufkommen eines realen Netzwerks (nach [16])

senden. Abhängig von den vorhandenen Ressourcen führt eine Verbindungsanforderung entweder zum Aufbau einer Verbindung im Kommunikationssystem durch das Senden von Verbindungsanforderungen zum nächsten Knoten oder die Verbindungsanforderung wird abgewiesen. Die Erzeugung dieser Verbindungsanforderungen und das Senden an den Startknoten der geplanten Verbindung geschieht durch die Software des Lastgenerators. I.A. werden Lastgeneratoren für derartige Anwendungen innerbetrieblich durch den Hersteller der Kommunikationssysteme entwickelt. Die exakten Eigenschaften wie z.B. die Strategien zur Lastgenerierung sind somit hersteller- und anwendungsspezifisch.



# Kapitel 3

## Grundlagen der Modellierung

George E.P. Box *"All models are wrong,  
but some are useful"*

In diesem Kapitel werden die Grundlagen der Modellierung vorgestellt. Die Struktur des Kapitels lehnt sich der Struktur von Kapitel 2 an und umfasst die Modellierung der Kommunikationssysteme, die Modellierung konkurrierender Zugriffe und die Modellierung des Lastgenerators. Es wird vom leitungsvermittelnden Kommunikationssystem zum Modell des Kommunikationssystems stark abstrahiert. Aufgrund der Komplexität realer Kommunikationssysteme werden bei der Modellierung nur die Eigenschaften nachgebildet, welche zur Untersuchung der Auftrittshäufigkeiten konkurrierender Zugriffe nötig sind. Die Modelle sind stochastische Modelle, d.h. die Systemparameter sind stochastisch verteilt.

### 3.1 Leitungsvermittelnde Kommunikationssysteme

In diesem Abschnitt werden Modelle von Kommunikationssystemen vorgestellt. Diese modellieren typische leitungsvermittelnde Kommunikationssysteme und deren wesentlichen Eigenschaften. Es werden für die Modelle von Kommunikationssystemen die Parameter definiert, welche für die verschiedenen Systemfunktionen (wie z.B. den Aufbau von Verbindungen) benötigt werden. Bei der Betrachtung von Verbindungen werden nur Verbindungen zwischen Knoten beschrieben. Es werden nicht Verbindungen zum einzelnen Teilnehmer betrachtet.

#### 3.1.1 Netzgröße

**Eigenschaft 3.1** Es sei  $N$  die Anzahl der Knoten des Modells des Kommunikationssystems,  $N \in \mathbb{N}_+$ . Folgende Eigenschaft ist erfüllt:

Jeder Knoten kann potentiell zu jedem anderen Knoten eine physikalische Verbindung haben.

### 3.1.2 Physikalische Verbindungen zwischen den Knoten

**Eigenschaft 3.2** Eine  $N \times N$ -Matrix beschreibe alle möglichen gerichteten unidirektionalen Verbindungen zwischen den Knoten. Jedes Element  $c_{n_1 n_2}$ ,  $c_{n_1 n_2} \geq 0$ ,  $1 \leq n_1 \leq N$ ,  $1 \leq n_2 \leq N$ , dieser  $N \times N$ -Matrix gebe die Leitungskapazität an den Kanten vom Knoten  $n_1$  zum Knoten  $n_2$  an. Folgende Eigenschaften sind erfüllt:

- (a) Kein Knoten sei durch eine physikalische Verbindung mit sich selbst verbunden.
- (b) Ist die Leitungskapazität gleich null, so existiere keine physikalische Leitung zwischen diesen beiden Knoten.

Damit ist ersichtlich, dass jede Topologie durch diese  $N \times N$ -Matrix beschrieben werden kann.

**Eigenschaft 3.3** Das Wegeermittlungsverfahren des betrachteten Modells eines Kommunikationssystems sei statisch, d.h. nicht adaptiv. Es hat somit die folgenden Eigenschaften:

- (a) Die Berechnung der Leitwege zwischen zwei Knoten erfolgt für alle Knoten vor der Inbetriebnahme des Netzes.
- (b) Für die Bestimmung der nötigen Ressourcen für Verbindungen zwischen zwei Netzelementen werden Mittelwerte verwendet.
- (c) Es ist keine Änderung der Leitwege während des Betriebs des Netzes möglich.
- (d) Der aktuelle Netzzustand wird während des Aufbaus einer Verbindung nicht berücksichtigt.

### 3.1.3 Dynamische Verbindungen zwischen den Knoten

Im Modell des Kommunikationssystems ist die Verkehrslenkung durch feste Routen realisiert. Es sei  $S$  die Anzahl der Routen im Netz,  $S \in \mathbb{N}_+$ . Die  $s$ -te Route,  $s \in \mathbb{N}_+$ ,  $s \leq S$ , wird mit  $r_s$ , der Startknoten der Route  $r_s$  wird als A-Knoten bzw.  $r_{s;A}$ , der Zielknoten der Route  $r_s$  wird als Z-Knoten bzw.  $r_{s;Z}$  bezeichnet. Die Route  $r_s$  von  $r_{s;A}$  zu  $r_{s;Z}$  ist vorgegeben. Mit Hilfe einer Tabelle ist festgelegt, mit welcher Wahrscheinlichkeit jede dieser Routen gewählt wird. Die Wahrscheinlichkeit der Auswahl einer Route  $r_s$  wird mit  $p(r_s)$  bezeichnet. Es folgt:  $\sum_{s=1}^S p(r_s) = 1$ .

Jedes Kommunikationsnetz stellt nur eine beschränkte Anzahl von Ressourcen wie z.B. Leitungen zur Verfügung. Steht für einen Verbindungsaufbau eine benötigte Ressource nicht zur Verfügung, so wird diese Verbindungsanforderung abgewiesen und die bereits bis zu diesem Knoten eingerichtete Verbindung wieder

abgebaut. Ein derartiges Verkehrsmodell wird als Verlustsystem bezeichnet. Die Wahrscheinlichkeit, dass es zu keinem erfolgreichen Verbindungsaufbau aufgrund von fehlenden Ressourcen kommt, wird als Verlustwahrscheinlichkeit bezeichnet. Die mathematische Modellierung derartiger Systeme ist durch *Kelly* in [36] beschrieben.

Der Verbindungsaufbau im Modell des Kommunikationssystem beginnt mit der Erzeugung und dem Start einer Verbindung. Die Route der gewünschten Verbindung ist bekannt. Es wird eine Verbindungsanforderung an den A-Knoten gesendet. In dieser Verbindungsanforderung ist die Route der gewünschten Verbindung enthalten. Durch den A-Knoten wird eine Verbindungsanforderung (CONN\_REQ) an den nächsten Knoten der Route geschickt. Dieser Knoten sendet wiederum eine Verbindungsanforderung an den nächsten Knoten der Route. Dies wiederholt sich so lange, bis der Z-Knoten der Route erreicht ist. Der Z-Knoten sendet eine Bestätigung der Verbindungsanforderung (CONN\_ACK) an den vorherigen Knoten der Verbindung. Dies wiederholt sich, bis der A-Knoten erreicht wurde. Nun ist die Verbindung erfolgreich aufgebaut. Der Verbindungsabbau einer geschalteten Verbindung erfolgt nach dem Ende der Haltephase. Der Verbindungsabbau erfolgt durch das Senden einer Verbindungsabbauanforderung (DISC\_REQ) vom Z-Knoten zum vorherigen Knoten. Das Senden einer Verbindungsabbauanforderung wiederholt sich, bis der A-Knoten erreicht wird. Diese Verbindungsabbauanforderung wird durch den A-Knoten durch eine Verbindungsabbaubestätigung (DISC\_ACK) an den nächsten Knoten der Route bestätigt. Dies wiederholt sich bis zum Z-Knoten. Die allokierten Ressourcen werden freigegeben. Abbildung 3.1 zeigt den erfolgreichen Aufbau und Abbau einer Verbindung des Modells eines Kommunikationssystem.

Die Abbildung 3.2 beschreibt den zeitlichen Ablauf des Auf- und Abbaus einer Verbindung. Die Haltephase einer Verbindung beginnt mit dem Ende der Verbindungsaufbauphase und endet mit dem Start der Verbindungsabbauphase.

**Eigenschaft 3.4** Es sei  $I$  die Anzahl der Verbindungen. Die Dauer der Haltephase einer Verbindung  $i$ ,  $i \in \{1, \dots, I\}$ , wird als  $H_i$  bezeichnet. Folgende Eigenschaften sind erfüllt:

- (a) Die Rate der Haltephase sei  $\gamma$ , der Mittelwert der Dauer der Haltephasen sei folglich  $H_M = \frac{1}{\gamma}$ .
- (b) Es sei  $0 \leq H_i \leq 2H_M$ ,  $i \in \{1, \dots, I\}$ .  $\tilde{H}_i$  beschreibe die gemäß der Verteilung berechnete Dauer der Haltephasen der Verbindung  $i$ ,  $i \in \{1, \dots, I\}$ . Ist  $\tilde{H}_i \leq 0$  oder  $\tilde{H}_i \geq 2H_M$ , so wird  $\tilde{H}_i$  neu generiert. Ist  $0 < \tilde{H}_i < 2H_M$ , so ist  $H_i := \tilde{H}_i$ . Die Verteilung von  $H_i$  zwischen 0 und  $2H_M$  ist folglich identisch.



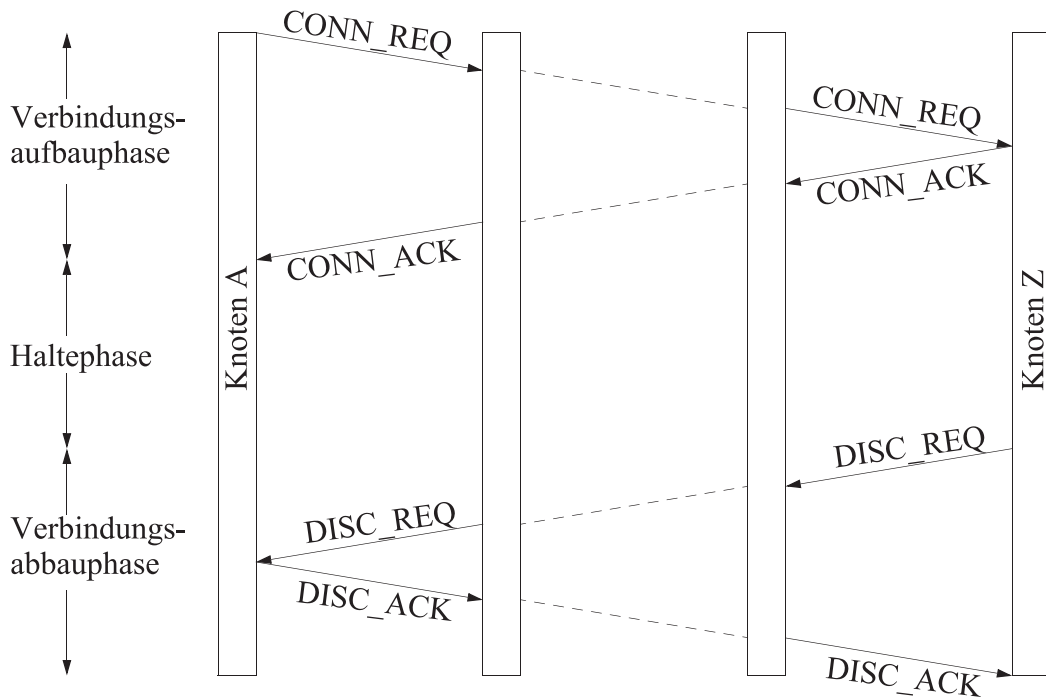


Abbildung 3.1: Aufbau und Abbau einer Verbindung des Modells eines Kommunikationssystems

$V_{i,r_s}$  bezeichnet die geschaltete Verbindung  $i$  der Route  $r_s$  mit  $i \in \{1, \dots, I\}$ ,  $s \in \{1, \dots, S\}$ . Abbildung 3.3 zeigt eine erfolgreich geschaltete, gerichtete Verbindung mit der Nummer 1 von Knoten 2 über die Knoten 3 und 5 zum Knoten 4.

**Eigenschaft 3.5** Das Modell erfüllt die folgenden Eigenschaften:

- (a) Die Leitungskapazität sei stets hinreichend groß, d.h. es kommt zu keinem erfolglosen Verbindungsaufbauversuch aufgrund fehlender Ressourcen.

### 3.1.4 Kommunikation zwischen den Knoten

Die Kommunikation zwischen Knoten des Modells des Kommunikationssystems erfolgt durch das Versenden von Nachrichten. Jeder Knoten ist gemäß der Notation von Kendall ein  $A/B/1/k$ -System. Es beschreibt  $A$  die Verteilung der Zwischenankunftszeiten und  $B$  die Verteilung der Bedienzeiten. Jeder Knoten besitzt eine Bedienstation, und verfügt über  $k - 1$  Plätze in der Warteschlange,  $k \in \mathbb{N}_+$ . Eine neu eintreffende Nachricht wird in die Warteschlange des Knotens eingefügt, wenn weniger als  $k - 1$  Nachrichten im Knoten zur Bearbeitung in der Warteschlange anstehen. Sind bereits  $k - 1$  Nachrichten in der Warteschlange des Systems, so geht die Nachricht verloren.

Der Informationsdatenfluss im Modell des Kommunikationssystems ist zwischen

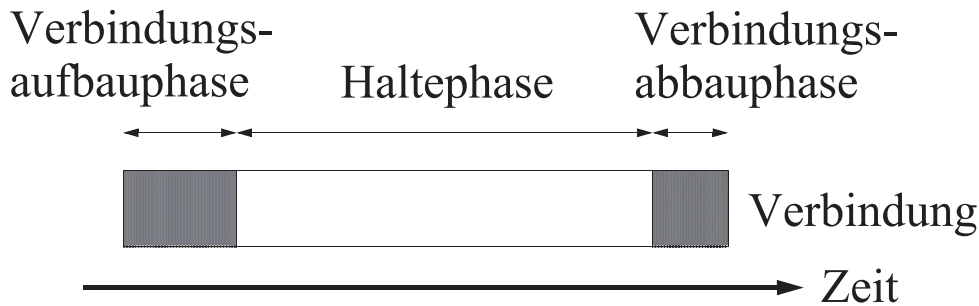


Abbildung 3.2: Zeitlicher Ablauf des Verbindungsaufbaus und -abbaus

miteinander verbundenen Knoten stets möglich, d.h. er ist von der Leitungskapazität unabhängig.

Alle Nachrichten zwischen Knoten und alle Nachrichten von ausserhalb des Netzes besitzen die gleiche Priorität und werden gemäß ihrem Eintreffen in der Warteschlange nach dem First-In-First-Out-Prinzip (FIFO-Prinzip) bearbeitet. Jeder Knoten hat genau eine Warteschlange endlicher Länge, um eintreffende Nachrichten bis zur Bearbeitung zu speichern. Hat ein Knoten mehrere Nachbarknoten und somit mehrere Eingangsports für eintreffende Nachrichten, so werden alle Nachrichten dieser Eingangsports in dieselbe Warteschlange geschrieben.

Es wird ein  $M/M/1/k$ -Warteschlangensystem betrachtet. Es beschreibt  $M$  die Exponentialverteilung. Es ist  $\lambda$  die Ankunftsrate,  $\nu$  die Rate der Bedienzeiten und  $k$  die Warteschlangenlänge (inkl. Bedienstation). In allen weiteren Betrachtungen sei die Bedienzeit nahe null.

**Eigenschaft 3.6** Das Modell erfüllt die folgende Eigenschaft:

- (a) Die Warteschlangen der Knoten seien hinreichend dimensioniert, d.h. es kommt zu keinem Verlust von Nachrichten zwischen den Knoten.

### 3.1.5 Arten von Knoten

Die beiden Arten der betrachteten Kommunikationssysteme werden mit Hilfe bestimmter Eigenschaften der Knoten unterschiedlich modelliert.

#### 3.1.5.1 Vermittlungssysteme

In einem Netz mit Knoten der Art „Vermittlungssystem“ werden ständig Verbindungen aufgebaut, gehalten und abgebaut. Es werden parallel Verbindungsaufbau- und Verbindungsabbauvorgänge durchgeführt. Nach einer „Anlaufphase“ entspricht die Anzahl der in einem bestimmten Intervall aufgebauten Verbindungen in etwa der Anzahl der abgebauten Verbindungen in diesem Intervall. Ein möglicher zeitlicher Ablauf des Verbindungsaufbaus und -abbaus verschiedener Verbindungen

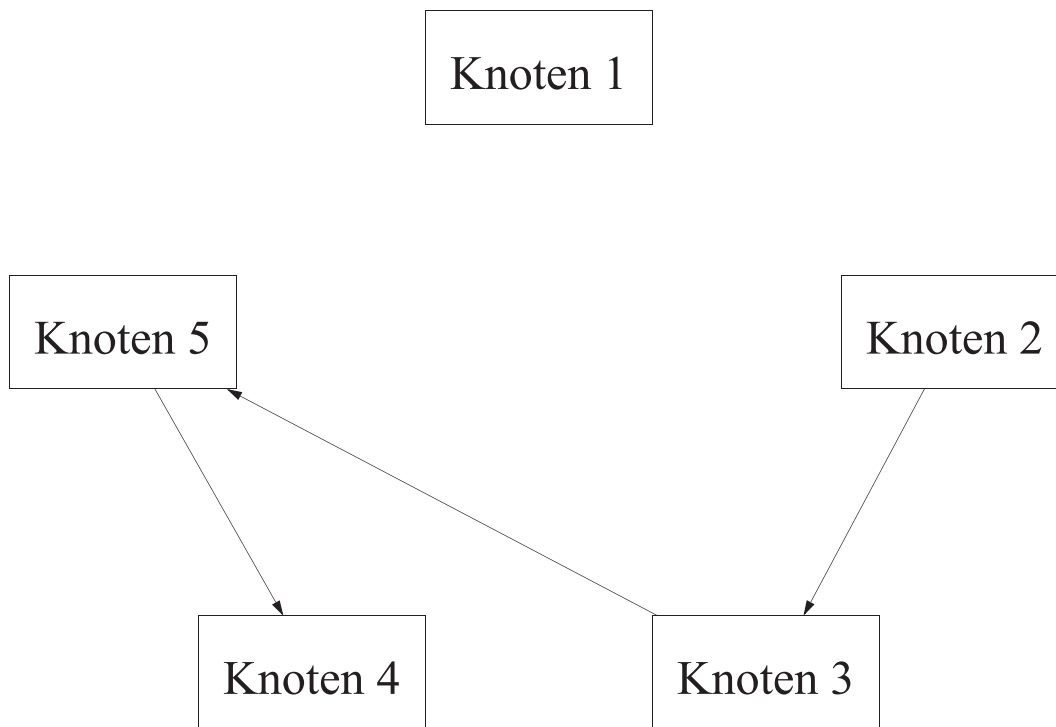


Abbildung 3.3: Gerichtete Verbindung  $V_{1_s}$  mit  $r_s = 2 - 3 - 5 - 4$

eines Vermittlungssystems ist in Abbildung 3.4 dargestellt. Die Darstellung der verschiedenen Phasen entspricht der Darstellung der Abbildung 3.2.

### 3.1.5.2 Übertragungssysteme

Bei einem Netz mit Knoten der Art Übertragungssystem werden zuerst alle Verbindungen aufgebaut. Anschließend erfolgt eine Phase, in welcher weder Verbindungen aufgebaut werden noch abgebaut werden. Diese Phase wird als Haltephase bezeichnet. Nach Abschluss dieser Phase werden die Verbindungen wieder abgebaut. Ein möglicher zeitlicher Ablauf des Verbindungsaufbaus und -abbaus verschiedener Verbindungen eines Übertragungssystems ist in Abbildung 3.5 dargestellt.

### 3.1.6 Eigenschaften der Knoten und Verbindungen

Als Funktionen des Modells des Kommunikationssystems sind Schreibzugriffe und Lesezugriffe realisiert. Schreibzugriffe modellieren Datenänderungen, Lesezugriffe modellieren Datenabfragen. Die Schreibzugriffe und Lesezugriffe werden auf den Speicherfeldern der Knoten durchgeführt. Die Anzahl der Speicherfelder des Knotens  $n$  wird mit  $F_n$  bezeichnet,  $F_n \in \mathbb{N}_+$ ,  $n \in \{1, \dots, N\}$ .

**Eigenschaft 3.7** Schreibzugriffe und Lesezugriffe erfüllen die folgenden Eigenschaften:

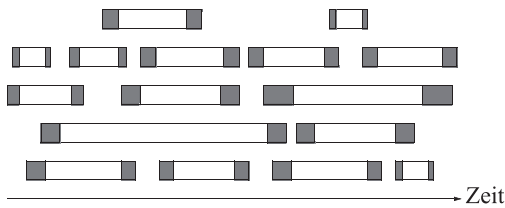


Abbildung 3.4: Zeitlicher Ablauf des Verbindungsaufbaus und Verbindungsabbaus in einem Vermittlungssystem



Abbildung 3.5: Zeitlicher Ablauf des Verbindungsaufbaus und Verbindungsabbaus in einem Übertragungssystem

- (a) Schreibzugriffe und Lesezugriffe einer Verbindung können jederzeit während der Haltephase der Verbindung stattfinden.
- (b) Schreibzugriffe und Lesezugriffe werden durch die Verbindung initiiert.
- (c) Schreibzugriffe und Lesezugriffe werden durch den Knoten ausgeführt.
- (d) Jeder Schreibzugriff und jeder Lesezugriff erfolgt auf einem zufällig ausgewählten Speicherfeld des Knotens.
- (e) Die Dauer von Schreibzugriffen und Lesezugriffen ist sehr kurz und kann vernachlässigt werden.
- (f) Die Auswahl des Speicherfelds eines Knotens und der Zeitpunkt des Schreib-Lesevorgangs sind voneinander unabhängig.

Aus den Eigenschaften des Modells des Kommunikationssystems folgt ein stochastisches Verhalten.

## 3.2 Konkurrierende Zugriffe

Zur Modellierung von konkurrierenden Zugriffen werden Schreib-Lesevorgänge auf Verbindungen durchgeführt. Der Semaphor, welcher die Daten vor zeitgleichen Schreibzugriffen anderer Verbindungen schützen soll, fehlt jedoch bei der Modellierung, und so kann es zu einem konkurrierenden Zugriff kommen. Der modellierte Schreib-Lesevorgang besteht aus einem Schreibzugriff im A-Knoten der Verbindung, einem Schreibzugriff im Z-Knoten der Verbindung, einer Ruhephase, einem Lesezugriff im A-Knoten der Verbindung und einem Lesezugriff im Z-Knoten der Verbindung. Der konkurrierende Zugriff tritt genau dann auf, wenn der Schreib- und Lesezugriff an einem Knoten von einem anderen Prozess unterbrochen wird und dieser Prozess genau das selbe Speicherfeld des Knotens beschreibt. Schreib-Lesevorgänge werden durch die entsprechenden Verbindung

initiiert. Der Schreibzugriff, die Ruhephase und der Lesezugriff modellieren einen langdauernden Schreibzugriff mit sehr großen Dateneinträgen. Das Speicherfeld auf welchem der Schreibzugriff durchgeführt wird, wird mit einem Zufallsgenerator nach einer gegebenen Verteilung ausgewählt.

**Eigenschaft 3.8** Ein Schreib-Lesevorgang (SLV) besteht aus drei Teilen und hat den folgenden Ablauf:

- (a) Es wird nach einer gegebenen Verteilung ein Speicherfeld des A-Knotens und ein Speicherfeld des Z-Knotens ausgewählt. Durch die Verbindung werden die Schreibvorgänge in dem A-Knoten und Z-Knoten durchgeführt. In die ausgewählten Speicherfelder wird jeweils die Zeichenkette „ABC + <connectionID>“ geschrieben. Dabei ist „<connectionID>“ eine Zeichenkette, welche pro Verbindung eindeutig ist.
- (b) Anschließend ruht die Verbindung für die Dauer  $z'$ .
- (c) Das Speicherfeld des A-Knotens und das Speicherfeld des Z-Knotens werden durch den A-Knoten bzw. den Z-Knoten gelesen. Es wird überprüft, ob die Einträge noch den gesetzten Einträgen „ABC + <connectionID>“ entsprechen.

Der modellierte Fehler „Konkurrierende Zugriffe“ äußert sich durch einen inkonsistenten Dateneintrag in einem Speicherfeld. Der konkurrierende Zugriff tritt genau dann auf, wenn der Schreib-Lesevorgang von einem anderen Prozess unterbrochen wird und dieser genau das selbe Speicherfeld beschreibt. Das Auftreten wird dadurch erkannt, dass die gelesene Zeichenkette nicht der erwarteten Zeichenkette entspricht.

In Abbildung 3.6 ist der Prozessor des A-Knotens zu sehen. Während des Beobachtungszeitraums werden vier Schreibzugriffe und vier Lesezugriffe von diesem Prozessor durchgeführt. Die zufällige Auswahl des Speicherfelds ergab das Speicherfeld 1 für die Verbindungen 1 und 2, das Speicherfeld 2 für die Verbindung 3 sowie das Speicherfeld 3 für die Verbindung 4. Die Schreibzugriffe und Lesezugriffe von den Verbindungen 1 und 2 greifen somit beide auf das Speicherfeld 1 zu. Da sie sich überlappen, kommt es zu einem konkurrierenden Zugriff. Die Schreib-Lesevorgänge der Verbindungen 3 und 4 überlappen sich zwar, es greift jedoch Verbindung 3 auf Speicherfeld 2 zu und Verbindung 4 greift auf das Speicherfeld 3 zu. Aus diesem Grund kommt es bei den Schreib-Lesevorgängen von Verbindung 3 und Verbindung 4 zu keinem konkurrierenden Zugriff. Es ist einfach zu verstehen, dass die Anzahl der Speicherfelder einen bedeutenden Einfluss auf die Auftrittshäufigkeit konkurrierender Zugriffe hat.

Der modellierte Fehler „Konkurrierende Zugriffe“ kann folgendermaßen definiert werden:

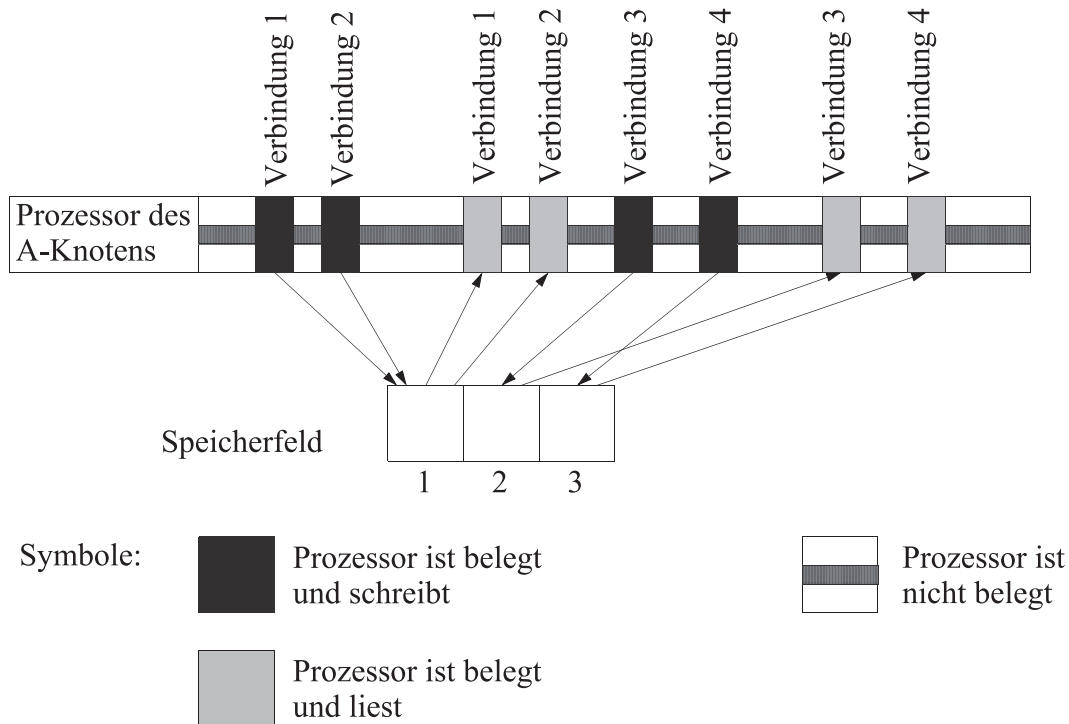


Abbildung 3.6: Prozessorbelegung des A-Knotens während mehrerer Schreib-Lesevorgänge

**Definition 3.9 Konkurrierender Zugriff**

Es wird ein bestimmtes Speicherfeld betrachtet. Gibt es einen Zeitpunkt zu dem dieses Speicherfeld an mindestens zwei Transaktionen beteiligt ist, wovon mindestens eine Transaktion schreibt, so wird dies als konkurrierender Zugriff bezeichnet.

**Bemerkung 3.10** Es wird auch der Fall als konkurrierender Zugriff bezeichnet, wenn nach dem konkurrierenden Zugriff das Speicherfeld an dem die beiden Transaktionen zu einem Zeitpunkt beteiligt sind, unverändert ist, und es somit zu keinem inkorrekten Wert einer Variablen und folglich auch zu keinem Versagen kommt.

Abbildung 3.7 zeigt den Aufbau eines Schreib-Lesevorgangs.

**Bezeichnung 3.11** Zur Beschreibung von Schreib-Lesevorgängen und Schreibzugriffen werden folgende Bezeichnungen verwendet:

- (a) Es sei  $p_{f_{n,m}}$  die Wahrscheinlichkeit, dass bei einem Schreibzugriff im Knoten  $n$  das Speicherfeld  $f_{n,m}$  ausgewählt wird,  $n \in \{1, \dots, N\}$ ,  $m \in \{1, \dots, F_n\}$ .
- (b) Es sei  $p$  die Wahrscheinlichkeit, dass zu einer beliebigen aber festen Verbindung ein Schreib-Lesevorgang innerhalb einer Zeiteinheit stattfindet.

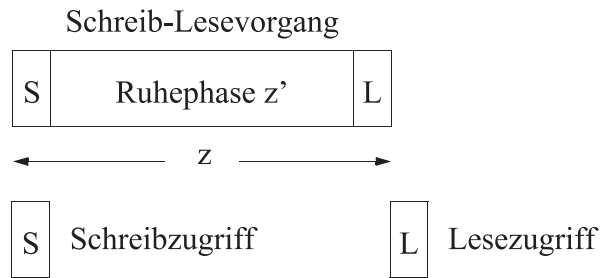


Abbildung 3.7: Aufbau eines Schreib-Lesevorgangs

- (c) Es sei  $z'$  die Dauer der Ruhephase eines Schreib-Lesevorgangs (vgl. Abbildung 3.7).
- (d) Es sei  $z$  die Dauer eines Schreib-Lesevorgangs (vgl. Abbildung 3.7).

**Eigenschaft 3.12** Schreib-Lesevorgänge erfüllen die folgenden Eigenschaften:

- (a) Die durchschnittliche Zugriffsdauer aller Schreib-Lesevorgänge auf ein Speicherfeld ist für alle Knoten und alle Speicherfelder nahezu gleich.
- (b) Es ist  $z - z' = \varepsilon$ .  $\varepsilon$  ist sehr klein und vernachlässigbar.
- (c) Die Dauer eines Schreib-Lesevorgangs entspricht der Dauer eines Timeslots.
- (d) Der Inhalt von Schreib-Lesevorgängen ist bei Schreib-Lesevorgängen verschiedener Verbindungen verschieden.

**Annahme 3.13** Folgende Annahmen gelten für Schreib-Lesevorgänge:

- (a) Der Zeitpunkt der Schreib-Lesevorgänge und die Auswahl des Speicherfelds in einem Knoten seien voneinander unabhängig.
- (b) Der Zeitpunkt der Schreib-Lesevorgänge und die Auswahl des Knotens des Schreib-Lesevorgangs seien voneinander unabhängig.
- (c) Die Zeitpunkte der Schreib-Lesevorgänge verschiedener Verbindungen seien voneinander unabhängig.
- (d) Der Zeitpunkt eines Schreib-Lesevorgangs sei unabhängig vom Zeitpunkt in der Messphase.

Aufgrund der vorliegenden Modellierung ist der Dateneintrag bei Schreib-Lesevorgängen verschiedener Verbindungen stets verschieden. Somit ist das Auftreten eines konkurrierenden Zugriffs stets feststellbar.

## 3.3 Lastgenerator

### 3.3.1 Beschreibung der Last

Die Last wird durch verschiedene Faktoren bestimmt:

- Die Anzahl der existierenden Verbindungen während der Untersuchung mit Last.
- Die durchschnittliche Dauer der Haltephasen der Verbindungen.
- Die Wahrscheinlichkeit einer Route während der Untersuchung mit Last.
- Die Rate der Schreib-Lesevorgänge pro Verbindung während der Untersuchung mit Last.

Es werden durch den Lastgenerator nur Verbindungen initiiert, für welche eine Route definiert ist.

Im Folgenden wird die Last jeweils pro Minute betrachtet. Diese Einheit ist sinnvoll, da Timeslots stets eine Dauer in der Größenordnung von Millisekunden haben. Prinzipiell ist aber auch jede andere Einheit als Zeitbasis möglich.

**Bezeichnung 3.14** Es sei  $I$  die Anzahl der existierenden Verbindungen im Netz in der Messphase,  $I > 0$ . Es sei  $H_M$  die durchschnittliche Dauer der Haltephasen der Verbindungen,  $H_M > 0$ . Es gilt:  $\gamma = \frac{1}{H_M}$ . Es sei  $p(r_s)$  die Wahrscheinlichkeit einer Route  $r_s$  und  $S$  die Anzahl der Routen des Modells des Kommunikationssystems mit  $S \in \mathbb{N}_+$ . Die Rate der Anforderungen für Schreib-Lesevorgänge an eine Verbindung wird mit  $\mu$  bezeichnet. Es sei  $\mu > 0$ . Die Last  $L$  wird beschrieben durch den folgenden Vektor:

$$L = \begin{pmatrix} I \\ \gamma \\ p(r_1) \\ \dots \\ p(r_S) \\ \mu \end{pmatrix}.$$

**Annahme 3.15** Folgende Annahmen gelten bei einer Untersuchung mit Last:

- (a) Die Verbindungsanforderungen des Lastgenerators seien voneinander unabhängig.
- (b) Die Dauer der Haltephasen der Verbindungen seien voneinander unabhängig.

**Bemerkung 3.16** Durch den vollständigen Abbau der Verbindungen nach jeder Messung sind die Messungen voneinander unabhängig.



**Eigenschaft 3.17** Folgende Eigenschaften sind bei der Messaufbauphase eines Modells erfüllt:

- (a) Vermittlungssystem: Es werden während der Messaufbauphase  $i + j$  Verbindungen aufgebaut und  $j$  Verbindungen abgebaut. D.h. am Ende der Messaufbauphase sind  $i$  Verbindungen aufgebaut. Während der Messphase werden genauso viele Verbindungen aufgebaut wie abgebaut. Während der Messabbauphase werden  $i$  Verbindungen abgebaut.
- (b) Übertragungssystem: Am Ende der Messaufbauphase sind alle Verbindungen erfolgreich aufgebaut.

### 3.3.2 Eigenschaften des Lastgenerators

Der Lastgenerator umfasst die folgende Funktionalität:

- Initiierung des Aufbaus einer Verbindung: Der Lastgenerator initiiert eine Verbindung vom Start- zum Zielknoten. Der Lastgenerator kreiert nur dann eine Verbindung, wenn für den Start- und Zielknoten eine Route existiert.
- Initiierung der Schreib-Lesevorgänge für eine Verbindung: Der Lastgenerator bestimmt die Zeitpunkte der Schreib-Lesevorgänge auf einer Verbindung.

**Eigenschaft 3.18** Folgende Eigenschaften sind bei der Initiierung des Aufbaus von Verbindungen erfüllt:

- (a) Der Lastgenerator bestimmt die Anzahl der zu initiiierenden Verbindungen in Abhängigkeit der Art des betrachteten Kommunikationssystems.
- (b) Der Lastgenerator bestimmt den Zeitpunkt eines jeden Verbindungsaufbaus in Abhängigkeit der Art des betrachteten Kommunikationssystems.
- (c) Der Lastgenerator initiiert den Aufbau von Verbindungen zu den berechneten Zeitpunkten.
- (d) Der Lastgenerator bestimmt die Route der Verbindungen in Abhängigkeit der gegebenen Wahrscheinlichkeiten.
- (e) Der Lastgenerator bestimmt die Dauer der Haltephase jeder Verbindung.

**Eigenschaft 3.19** Folgende Eigenschaften sind bei der Initiierung von Schreib-Lesevorgängen für Verbindungen erfüllt:

- (a) Der Lastgenerator bestimmt die Anzahl der Schreib-Lesevorgänge in der Haltephase einer Verbindung.

- (b) Der Lastgenerator bestimmt den Zeitpunkt der Schreib-Lesevorgänge während der Haltephase.
- (c) Die durch den Zufallsgenerator generierte Anzahl bzw. generierte Zeiten entsprechen den gewählten Verteilungen.

**Bemerkung 3.20** Es sei  $I_M$  die Anzahl der zu einem Zeitpunkt durchschnittlich existierenden Verbindungen in der Messphase. Die Rate der Schreib-Lesevorgänge aller Verbindungen entspricht  $\mu \cdot I_M$ . Die Rate der Schreib-Lesevorgänge jeder einzelnen Verbindung muss jedoch nicht der gegebenen Rate  $\mu$  entsprechen.

Durch die vorgestellten Eigenschaften des Lastgenerators wird offensichtlich, dass dieser stochastisches Verhalten besitzt.

Um die Unabhängigkeit der Einzelerperimente sicherzustellen, müssen die folgenden Voraussetzungen für die Verbindungs- und Schreibanforderungen sowie für die Knoten erfüllt sein.

**Voraussetzung 3.21** Die Verbindungsanforderungen und Anforderungen für Schreib-Lesevorgänge des Lastgenerators müssen die folgenden Eigenschaften erfüllen:

- (a) Die zeitliche Verteilung der Verbindungsanforderung und die zeitliche Verteilung der Anforderungen für Schreib-Lesevorgänge des Lastgenerators innerhalb der Haltephase dieser Verbindung sind voneinander unabhängig.
- (b) Die Dauer der Haltephase einer Verbindung ist unabhängig vom Zeitpunkt der Verbindungsanforderung für diese Verbindung.
- (c) Die zeitlichen Verteilungen der Anforderungen für Schreib-Lesevorgänge zweier beliebiger Verbindungen sind voneinander unabhängig.

Um die Voraussetzungen 3.15 – 3.21 zu gewährleisten, ist es nötig, weitere Annahmen zu treffen:

**Annahme 3.22** Die Rate der Verbindungsanforderungen des Lastgenerators und die Haltephasen müssen die folgenden Eigenschaften erfüllen:

- (a) Die Rate der Verbindungsanforderungen des Lastgenerators sei im Mittel für jedes Teilintervall des betrachteten Intervalls gleich.
- (b) Die Haltephasen der Verbindungen seien normalverteilt.
- (c) Für jede Verbindung seien die Schreib-Lesevorgänge im Mittel gleichverteilt während der Haltephase dieser Verbindung.



# Kapitel 4

## Analytisches Modell

Albert Einstein *"As far as the laws of mathematics refer to reality,  
they are not certain;  
and as far as they are certain,  
they do not refer to reality."*

Um die Auftrittshäufigkeit konkurrierender Zugriffe durch mathematische Methoden herzuleiten und zu bestimmen, müssen weitere Annahmen getroffen werden. Diese Annahmen werden im Folgenden als Modelleinschränkungen bezeichnet. Unter diesen Annahmen wird anschliessend die Auftrittshäufigkeit „Konkurrierender Zugriffe“ mathematisch hergeleitet.

### 4.1 Modelleinschränkungen

Die Modelleinschränkungen werden nur beim analytischen Modell verwendet und werden in diesem Abschnitt vorgestellt. Es ist ein Ziel der Modellierung, möglichst wenige zusätzliche Annahmen für das analytische Modell zu treffen. Das analytische Modell und das Modell des Kommunikationssystems sollen bestmöglich übereinstimmen. Eigenschaften, welche nicht explizit aufgeführt sind, entsprechen exakt der Modellierung in Kapitel 3.

#### 4.1.1 Modelle von Kommunikationssystemen

Bei dem Modell von Kommunikationssystemen sind die folgenden Parameter gemäß Verteilungsfunktionen verteilt:

- (a) Routenwahrscheinlichkeiten von Verbindungen.
- (b) Dauer der Haltephase bei Vermittlungssystemen.
- (c) Zeitpunkte der Verbindungsanforderungen vom Lastgenerator.

Durch diese Eigenschaften ergibt sich eine bestimmte Varianz der durchschnittlichen Anzahl der Verbindungen im Modell des Kommunikationssystems. Bei dem analytischen Modell sind diese Parameter fest gewählt, d.h. die Varianz ist null. Das analytische Modell hat somit die folgenden Eigenschaften:

- (a) Die Routenwahrscheinlichkeiten entsprechen exakt den vorgestellten Werten.
- (b) Bei Vermittlungssystemen ist die Dauer der Haltephase konstant  $H_M$ .
- (c) Die Wahrscheinlichkeit einer Verbindungsanforderung des Lastgenerators ist - abhängig von der Art des Kommunikationssystems - zu jedem Zeitpunkt gleich.

Unter diesen Voraussetzungen ist die durchschnittliche Anzahl von Verbindungen im Kommunikationssystem konstant.

#### 4.1.2 Konkurrierende Zugriffe

Bei dem Modell von konkurrierenden Zugriffen sind die folgenden Parameter gemäß Verteilungsfunktionen verteilt:

- (a) Zeitpunkte der Schreib-Lesevorgänge.
- (b) Wahrscheinlichkeit der Auswahl eines Speicherfelds.
- (c) Zeitpunkte der Anforderungen für Schreib-Lesevorgänge.

Beim analytischen Modell werden folgende Eigenschaften verwendet:

- (a) Die Wahrscheinlichkeit eines Schreib-Lesevorgangs einer Verbindung ist zu jedem Zeitpunkt der Haltephase der Verbindung gleich.
- (b) Die Wahrscheinlichkeit der Auswahl eines Speicherfelds im Knoten  $n$  ist exakt  $p_{f_{n,m}}$ ,  $n \in \{1, \dots, N\}$ ,  $m \in \{1, \dots, F_n\}$ .
- (c) Die Wahrscheinlichkeit einer Anforderung für einen Schreib-Lesevorgang ist zu jedem Zeitpunkt der Haltephase einer Verbindung gleich.

#### 4.1.3 Last

Der Vektor  $L$ , welcher die Last beschreibt und in der Bezeichnung 3.14 vorgestellt wurde, wird für das analytische Modell verwendet.

#### 4.1.4 Eigenschaften des Gesamtmodells

Beim analytischen Modell wird ein Monoprocessor mit nur einem Thread modelliert. Durch diese Eigenschaften ist bei dem analytischen Modell ein stochastisches Verhalten gegeben. Beim Modell von Kommunikationssystemen wird ein Monoprocessorsystem mit mehreren Threads betrachtet. Während beim Simulationsmodell des Kommunikationssystem eine Taktung durch das Betriebssystem vorgegeben ist, ist dies beim analytischen Modell nicht gegeben. D.h. bei dem analytischen Modell sind die Taktung des Prozessors und der Prozessorswitch nicht modelliert.

## 4.2 Herleitung der Auftrittshäufigkeit

In diesem Abschnitt wird die Auftrittshäufigkeit konkurrierender Zugriffe hergeleitet. Zum besseren Verständnis werden nun drei verschiedene Darstellungen betrachtet, durch welche die Nichtlinearität der Auftrittshäufigkeit konkurrierender Zugriffe deutlich wird. In Abbildung 4.1 sind zwei Verbindungen dargestellt. Auf diesen Verbindungen finden jeweils zwei Schreib-Lesevorgänge (SLV) statt. Der Schreib-Lesevorgang 1 der Verbindung A (d.h. SLV A1) kann mit den Schreib-Lesevorgängen 1 und 2 der Verbindung B (d.h. SLV B1 und SLV B2) potentiell einen konkurrierenden Zugriff haben. Es ist jedoch nicht möglich, dass der Schreib-Lesevorgang 1 der Verbindung A (d.h. SLV A1) mit dem Schreib-Lesevorgang 2 der Verbindung A (d.h. SLV A2) einen konkurrierenden Zugriff hat. Folglich kann jeder dieser Schreib-Lesevorgänge in dieser Abbildung mit jeweils zwei anderen Schreib-Lesevorgängen potentiell einen konkurrierenden Zugriff haben.

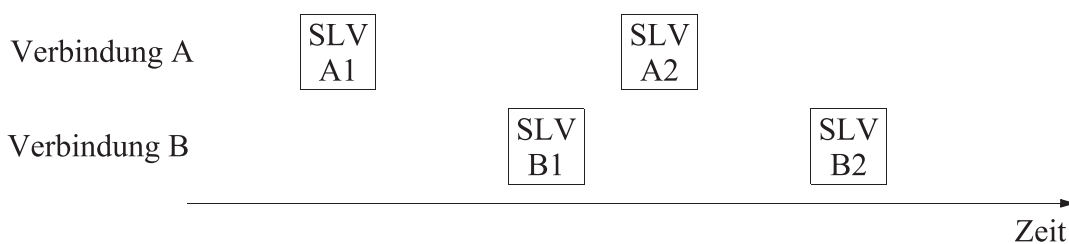


Abbildung 4.1: Zwei Verbindungen mit je zwei Schreib-Lesevorgängen

In Abbildung 4.2 sind hingegen zwei Verbindungen mit jeweils vier Schreib-Lesevorgängen dargestellt, d.h. die Rate der Schreib-Lesevorgänge ist doppelt so hoch. Analog zu Abbildung 4.1 kann in Abbildung 4.2 der Schreib-Lesevorgang 1 der Verbindung A (d.h. SLV A1) mit den Schreib-Lesevorgängen 1, 2, 3 und 4 der Verbindung B (d.h. SLV B1, SLV B2, SLV B3 und SLV B4) einen konkurrierenden Zugriff haben. Folglich kann jeder dieser acht Schreib-Lesevorgänge in

dieser Abbildung mit jeweils vier anderen Schreib-Lesevorgängen potentiell einen konkurrierenden Zugriff haben.

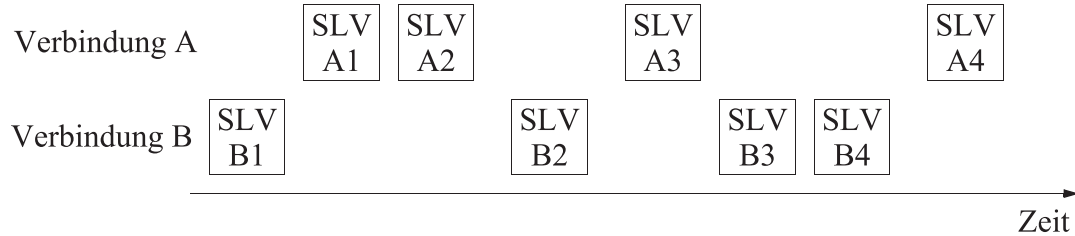


Abbildung 4.2: Zwei Verbindungen mit je vier Schreib-Lesevorgängen

In Abbildung 4.3 sind vier verschiedene Verbindungen mit jeweils zwei Schreib-Lesevorgängen dargestellt. D.h. die Anzahl der Schreib-Lesevorgänge ist identisch zur Anzahl der Schreib-Lesevorgänge in Abbildung 4.2. In Abbildung 4.3 kann der Schreib-Lesevorgang 1 der Verbindung A (d.h. SLV A1) mit den Schreib-Lesevorgängen 1 und 2 der Verbindung B, der Verbindung C und der Verbindung D (d.h. SLV B1, SLV B2, SLV C1, SLV C2, SLV D1 und SLV D2) einen konkurrierenden Zugriff haben. Folglich kann jeder dieser acht Schreib-Lesevorgänge in dieser Abbildung mit jeweils sechs anderen Schreib-Lesevorgängen potentiell einen konkurrierenden Zugriff haben.

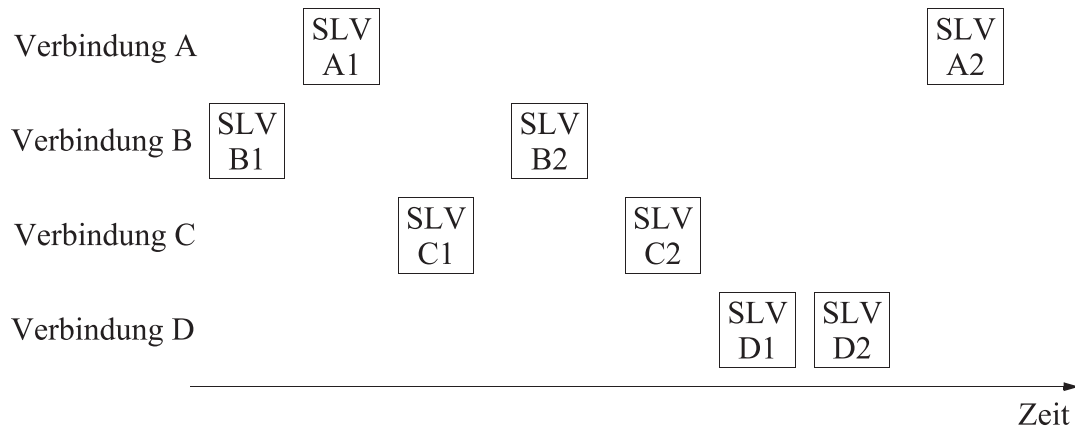


Abbildung 4.3: Vier Verbindungen mit je zwei Schreib-Lesevorgängen

Anhand dieser Darstellungen ist ersichtlich, dass bei fester Anzahl Schreib-Lesevorgänge die Chance konkurrierender Zugriffe mit der Anzahl der Verbindungen steigt.

Im Folgenden wird die Auftrittshäufigkeit selten auftretender konkurrierender Datenzugriffe mathematisch hergeleitet.

In jedem Knoten  $n$  sei  $f_{n,m}$  ein Speicherfeld,  $n \in \{1, \dots, N\}$ ,  $m \in \{1, \dots, F_n\}$ .

Für jedes beliebige Speicherfeld  $f_{n,m}$  sei für jeden Schreib-Lesevorgang dieses Knotens:

$$D_{f_{n,m}} = \begin{cases} 1 & \text{ein Schreib-Lesevorgang im Knoten } n \text{ betrifft das Speicherfeld} \\ & f_{n,m}, \\ 0 & \text{sonst.} \end{cases}$$

Die Wahrscheinlichkeit des Ereignisses  $D_{f_{n,m}} = 1$  entspricht  $p_{f_{n,m}}$  der Bezeichnung 3.11 (a). Nach Bezeichnung 3.11 (b) bezeichne  $p$  die Wahrscheinlichkeit, dass zu einer beliebigen aber festen Verbindung ein Schreib-Lesevorgang innerhalb einer Zeiteinheit stattfindet.

Es wird eine beliebige Verbindung und ein Knoten  $n$ ,  $n \in \{1, \dots, N\}$ , betrachtet. Die Zufallsgröße  $X_{r_s;A;Z=n}$  beschreibt, ob diese Verbindung die Route  $r_s$ ,  $s \in \{1, \dots, S\}$ , und den Knoten  $n$  als A-Knoten oder als Z-Knoten hat. D.h.

$$X_{r_s;A;Z=n} = \begin{cases} 1 & \text{Verbindung hat die Route } r_s \text{ und Knoten } n \text{ als} \\ & \text{A-Knoten oder Z-Knoten,} \\ 0 & \text{sonst.} \end{cases}$$

Die Wahrscheinlichkeit, dass eine Verbindung den Knoten  $n$  als A-Knoten oder Z-Knoten hat, wird mit  $p_n$  bezeichnet. Da der A-Knoten und der Z-Knoten einer Verbindung stets verschieden sind, gilt:

$$p_n = \sum_{s=1}^S p(X_{r_s;A;Z=n} = 1)p(r_s).$$

Die Wahrscheinlichkeit, dass innerhalb einer Zeiteinheit ein Schreib-Lesevorgang einer beliebigen Verbindung erfolgt, welche den Knoten  $n$  als Start- oder Zielknoten hat und das Speicherfeld  $f_{n,m}$  beschreibt, ist durch  $q_{f_{n,m}}$  beschrieben. Es gilt:

$$q_{f_{n,m}} = p \cdot p_n \cdot p_{f_{n,m}}.$$

Es ist  $I$  die Anzahl von Verbindungen in einer Minute in der Messphase. Die Zufallsgröße  $Y_{i,f_{n,m}}$  beschreibt einen Schreib-Lesevorgang der Verbindung  $i$  auf das Speicherfeld  $f_{n,m}$ .

$$Y_{i,f_{n,m}} = \begin{cases} 1 & \text{Schreib-Lesevorgang der Verbindung } i \text{ greift auf das} \\ & \text{Speicherfeld } f_{n,m} \text{ innerhalb einer Zeiteinheit zu,} \\ 0 & \text{sonst.} \end{cases}$$

$i \in \{1, \dots, I\}$ ,  $n \in \{1, \dots, N\}$ ,  $m \in \{1, \dots, F_n\}$ .

Wie in Eigenschaft 3.7(f) beschrieben ist, sind die Auswahl des Speicherfelds und der Zeitpunkt des Schreib-Lesevorgangs voneinander unabhängig. Wie in Abschnitt 3.2 beschrieben, kommt es bei der gewählten Modellierung genau dann zu einem konkurrierenden Zugriff, wenn mindestens zwei Schreib-Lesevorgänge



unterschiedlicher Verbindungen gleichzeitig auf das selbe Speicherfeld zugreifen. Die Zufallsgröße  $K_{f_{n,m}}$  beschreibe den konkurrierenden Zugriff unterschiedlicher Verbindungen innerhalb einer Zeiteinheit auf das Speicherfeld  $f_{n,m}$ .

$$K_{f_{n,m}} = \begin{cases} 1 & \text{zwei oder mehr Schreib-Lesevorgänge unterschiedlicher} \\ & \text{Verbindungen greifen auf das Speicherfeld } f_{n,m} \\ & \text{innerhalb einer Zeiteinheit zu,} \\ 0 & \text{sonst.} \end{cases}$$

$n \in \{1, \dots, N\}$ ,  $m \in \{1, \dots, F_n\}$ .

Es gilt:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= p\left(\sum_{i=1}^I Y_{i,f_{n,m}} \geq 2\right) \\ &= \left(1 - p\left(\sum_{i=1}^I Y_{i,f_{n,m}} = 1\right) - p\left(\sum_{i=1}^I Y_{i,f_{n,m}} = 0\right)\right). \end{aligned}$$

Aufgrund von Bernoulli gilt:

$$\begin{aligned} p\left(\sum_{i=1}^I Y_{i,f_{n,m}} = 1\right) &= \binom{I}{1} \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} \\ &= I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} \end{aligned}$$

und

$$p\left(\sum_{i=1}^I Y_{i,f_{n,m}} = 0\right) = (1 - q_{f_{n,m}})^I.$$

Es folgt:

$$p(K_{f_{n,m}} = 1) = 1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I.$$

$E[K_{f_{n,m}}]$  ist die erwartete Anzahl konkurrierender Zugriffe auf das Speicherfeld  $f_{n,m}$  innerhalb von  $t$  Zeiteinheiten. Aufgrund der Annahme der Unabhängigkeit der Ereignisse gilt:

$$\begin{aligned} E[K_{f_{n,m}}] &= t \cdot p(K_{f_{n,m}} = 1) \\ &= t \cdot \left(1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I\right). \end{aligned}$$

$E[K_n]$  ist die erwartete Anzahl konkurrierender Zugriffe im Knoten  $n$  innerhalb von  $t$  Zeiteinheiten. Aufgrund der Annahme der Unabhängigkeit der Ereignisse gilt:

$$\begin{aligned} E[K_n] &= \sum_{m=1}^{F_n} E[K_{f_{n,m}}] \\ &= \sum_{m=1}^{F_n} t \cdot \left(1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I\right). \end{aligned}$$

$E[K_N]$  ist die erwartete Anzahl konkurrierender Zugriffe im Modell des Kommunikationssystems innerhalb von  $t$  Zeiteinheiten. Aufgrund der Annahme der Unabhängigkeit der Ereignisse folgt wiederum:

$$\begin{aligned} E[K_N] &= \sum_{n=1}^N E[K_n] \\ &= \sum_{n=1}^N \sum_{m=1}^{F_n} t \cdot (1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I). \end{aligned}$$



# Kapitel 5

## Simulationsmodell

Albert Einstein *"No amount of experimentation  
can ever prove me right;  
a single experiment can prove me wrong."*

In diesem Kapitel werden die Eigenschaften des Simulationsmodells vorgestellt. Es handelt sich dabei um ein Modell der Diskreten-Ereignis-Simulation. Es ist ein stochastisches Simulationsmodell.

In dieser Arbeit wird die Auftrittswahrscheinlichkeit konkurrierender Zugriffe durch ein Bernoulli-Experiment approximiert. Basierend auf dieser Auftrittswahrscheinlichkeit kann anschließend die absolute Auftrittshäufigkeit geschätzt werden. Ein Bernoulli-Experiment beschreibt die  $m$ -malige Durchführung unabhängiger Einzelexperimente. Die Anzahl der Bernoulli-Experimente entspricht der Anzahl der Messungen  $u$ .

Bevor sich jedoch dem Simulationsmodell zugewandt werden kann, müssen zuerst die Eigenschaften der Schreib-Lesevorgänge näher betrachtet werden.

### 5.1 Eigenschaften der Schreib-Lesevorgänge

Bei der erstmaligen Durchführung von Simulationen zeigten sich einige — durch den Prozessor bedingte Eigenschaften — welche weitere Voraussetzungen nötig machen. Diese werden nun vorgestellt.

**Eigenschaft 5.1** Das Simulationsmodell erfüllt die folgende Eigenschaft: Die Schreibzugriffe erfolgen auf dem RAM, da es — abhängig vom Betriebssystem — bei jedem Zugriff auf ein Speichermedium (wie z.B. eine Datei) durch Einfluss des Betriebssystems zu einer Verfälschung kommen kann und dies durch eine Untersuchung auf dem RAM verhindert wird.

In der folgenden Abbildung 5.1 sind drei Threads dargestellt, welche Schreib-Lesevorgänge auf demselben Speicherfeld eines Knotens durchführen. Der Schreib-Lesevorgang von Thread B beginnt, nachdem der Schreib-Lesevorgang von Thread A vollständig durchgeführt wurde. Die gelesene Zeichenkette von Thread A entspricht der erwarteten Zeichenkette. Der Schreib-Lesevorgang von Thread C beginnt, bevor der Schreib-Lesevorgang von Thread B vollständig durchgeführt wurde. Die gelesene Zeichenkette von Thread B entspricht nicht der erwarteten Zeichenkette. Es ist ein konkurrierender Zugriff aufgetreten. Die gelesene Zeichenkette von Thread C entspricht hingegen der erwarteten Zeichenkette.

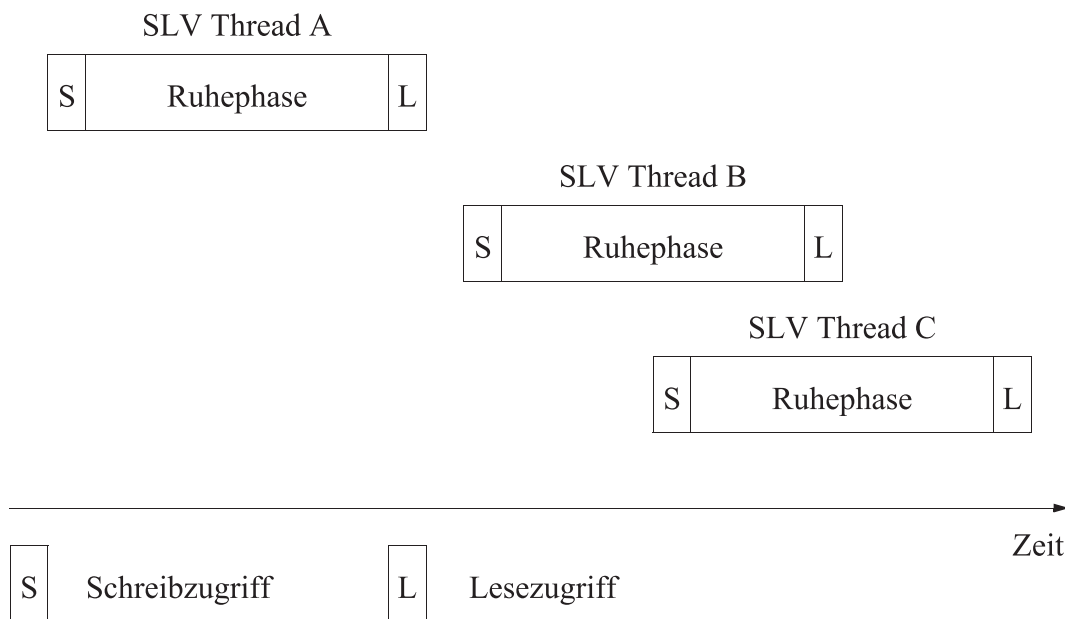


Abbildung 5.1: Schreib-Lesevorgänge von Threads

Der Startzeitpunkt von Schreib-Lesevorgängen ist bei einem Mehrprozessorsystem zu Beginn eines Timeslots - vorausgesetzt es stehen keine weiteren Tasks für diesen Thread an. In dem Simulationsmodell wird der Startzeitpunkt von Schreib-Lesevorgängen durch einen Zufallsgenerator erzeugt. Da durch die Generierung von Zufallszahlen keine ganzzahligen Werte erhalten werden, wurde der Startzeitpunkt jeweils auf den Beginn von Takten (rechnerabhängig) gerundet und jeder Schreib-Lesevorgang soll somit zum Zeitpunkt eines Taktbeginns beginnen.

Zum besseren Verständnis wird die folgende Abbildung 5.2 betrachtet. Sie beschreibt, wie Schreib-Lesevorgänge zweier verschiedener Threads in zwei benachbarten Takten durchgeführt werden, wenn diese exakt zu dem geplanten Zeitpunkt stattfinden.

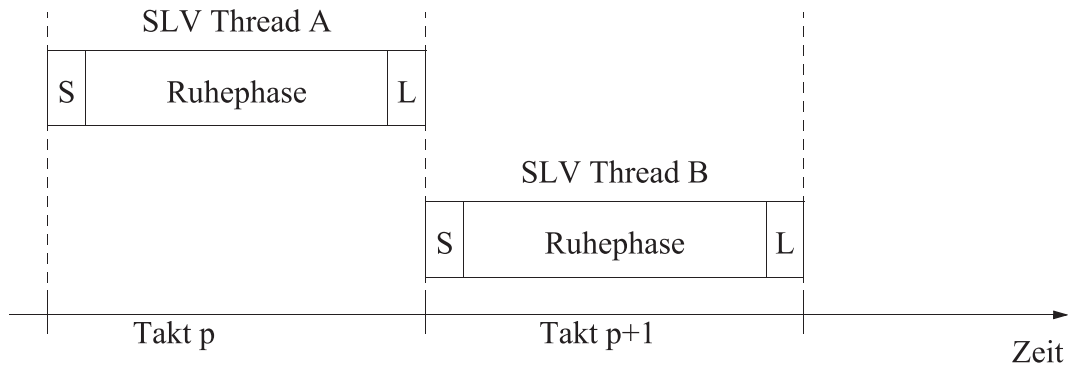


Abbildung 5.2: Theoretische Betrachtung von Schreib-Lesevorgängen zweier Threads

**Bemerkung 5.2** Die Taktlänge ist durch den Simulationsrechner vorgegeben. Sie kann bestimmt werden durch die Durchführung zeitaufwändiger Tasks und die sehr häufige Ausgabe der Uhrzeit in Millisekunden. Die Ausgabe erfolgt nicht im 1 Millisekunden-Raster, sondern i.A. im Raster der Taktlänge des Simulationsrechner.

Bei der Durchführung erster Untersuchungen hat sich gezeigt, dass die Schreib-Lesevorgänge nicht stets genau in der geplanten Millisekunde gestartet werden. In Abbildung 5.3 wurden diese Untersuchungsergebnisse bzgl. der Startzeitpunkte der Schreib-Lesevorgänge dargestellt. Da der Start dieser Schreib-Lesevorgänge nicht stets genau in der geplanten Millisekunde durchgeführt wird, kann es zu einem Konflikt kommen, obwohl theoretisch die beiden Schreib-Lesevorgänge während verschiedener Takte durchgeführt werden sollten.

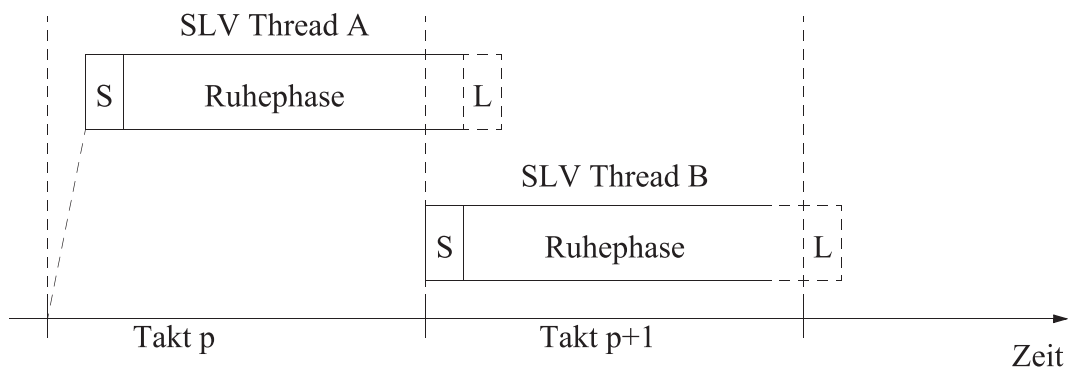


Abbildung 5.3: Messergebnis von Schreib-Lesevorgängen zweier Threads

Um die Effekte im Simulationsmodell statistisch günstig zeigen zu können, empfiehlt es sich, die Transaktionslänge in einer geeigneten Größe zur Taktlänge zu wählen. Bei den ersten Untersuchungen erwies sich ein Wert von 60% der Taktlänge als besonders geeignet. Aufgrund dieser Messergebnisse müssen für das Simulationsmodell die Schreib-Lesevorgänge noch die folgenden Voraussetzungen erfüllen:

**Voraussetzung 5.3** Das Simulationsmodell erfülle bei der Durchführung der Schreib-Lesevorgänge die folgenden Eigenschaften:

- (a) Die Dauer des Schreibzugriffs, der Ruhezeit und des Lesezugriffs soll in etwa 60% der Taktlänge entsprechen.
- (b) Die Zeitachse sei hinreichend klein getaktet und entspricht den Gegebenheiten des Simulationsrechners.
- (c) Ist ein Schreib-Lesevorgang für einen Zeitpunkt innerhalb eines Taktes geplant, so wird der Start dieses Schreib-Lesevorgang bei dem Simulationsmodell bereits zu Beginn des Taktes geplant.

In Abbildung 5.4 wird dargestellt, wie es unter der Voraussetzung 5.3 zu einem konkurrierenden Zugriff kommen kann.

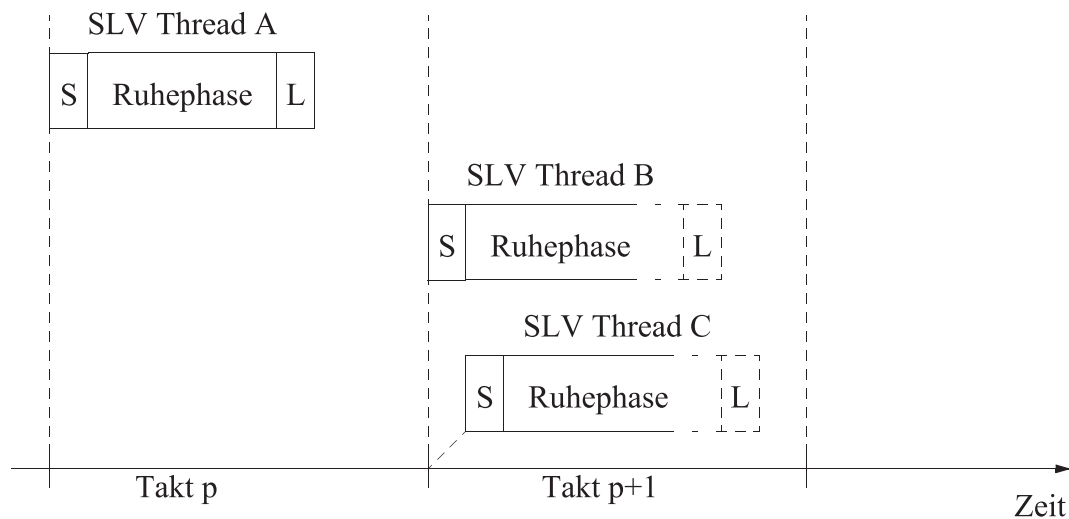


Abbildung 5.4: Schreib-Lesevorgänge dreier Threads

**Bemerkung 5.4** Werden zwei Schreib-Lesevorgänge auf derselben Verbindung durchgeführt, so werden diese stets aufeinander folgend und niemals überlappend durchgeführt, da jede Verbindung durch einen Thread ausgeführt wird. Es kann also in einem Knoten, welcher für nur eine Verbindung den A- oder Z-Knoten darstellt, niemals ein konkurrierender Zugriff auftreten.

## 5.2 Aufbau des Simulationsmodells

Das Simulationsmodell wurde auf einem PC mit dem Betriebssystem Windows XP mit Hilfe der Programmiersprache Java<sup>TM</sup>2 Platform Standard Edition v.1.4.2 entwickelt. Das Modell des Kommunikationssystems und des Lastgenerators werden gemeinsam auf einem Simulationsrechner simuliert.

Es gibt zwei Arten von Ereignissen im Simulationsmodell.

Einerseits wird der Verbindungsaufbau und -abbau durch das Versenden von Nachrichten (CONN\_REQ, CONN\_ACK, DISC\_REQ, DISC\_ACK) gesteuert. Der Zeitpunkt des Versendens der Nachrichten CONN\_REQ und DISC\_REQ wird durch die Anwendung eines Zufallszahlengenerator bestimmt. Die Nachrichten zur Empfangsbestätigung (CONN\_ACK und DISC\_ACK) werden als Antworten vom Empfängerknoten an den Senderknoten geschickt.

Andererseits erfolgt die Durchführung von Schreib-Lesevorgängen in Abhängigkeit der Systemzeit. Die Schreib-Lesevorgänge werden durchgeführt, sobald eine gewisse Zeit vergangen ist. Die Dauer dieser Zeit wird durch die Anwendung eines Zufallszahlengenerators bestimmt. Die Generierung der Zufallszahlen erfolgt durch einen in Java<sup>TM</sup>2 Platform Standard Edition v.1.4.2 vorhandenen Zufallszahlengenerator, welcher normal-verteilte reelle Zahlen mit dem Mittelwert 0 und der Standardabweichung 1.0 erzeugt. Dieser Zufallszahlengenerator verwendet die sog. *polynomial method* von *G.E.P.Box*, *M.E.Muller* und *G. Marsaglia*, die von *Knuth* in [42] vorgestellt wurde. Die Uhr des Simulationsmodells entspricht der Systemzeit des Rechners. Die Bestimmung der Systemzeit erfolgt ebenso durch eine in der Java Bibliothek enthaltene Routine. Deren Granularität ist vom verwendeten Betriebssystem abhängig. Die Uhr des Simulationsrechners wird nur in festen Schrittgrößen weitergeschaltet (sog. *fixed-increment time advance*), welche dieser Granularität entspricht.

Der Zeitpunkt eines Schreib-Lesevorgangs einer Verbindung, welcher ursprünglich für einen Zeitpunkt in einem bestimmten Takt geplant wurde, wird jeweils zu Beginn dieses Taktes geplant. Es sei im Zeitpunkt  $t_A$  der Schreib-Lesevorgang der Verbindung A, im Zeitpunkt  $t_B$  der Schreib-Lesevorgang der Verbindung B geplant. Es sei  $t_A < t_B$ . Durch die Taktung des Simulationsrechners werden beide Schreib-Lesevorgänge zu Beginn eines Taktes  $k$  geplant. Jede Verbindung hat eine eigene Liste, in welcher die geplanten Takte für die Schreib-Lesevorgänge aufgeführt sind. Es kann folglich sein, dass der Schreib-Lesevorgang der Verbindung B vor dem Schreib-Lesevorgang der Verbindung A stattfindet.

Es gibt sowohl für die Knoten als auch für die Verbindungen Ereignisse. Die Knoten werden durch die Nachrichten CONN\_REQ, CONN\_ACK, DISC\_REQ und DISC\_ACK getrieben, die Verbindungen werden durch die Systemzeit zum Verbindungsaufbau bzw. zum Schreib-Lesevorgang getrieben.



Sobald alle Verbindungen auf- und abgebaut worden sind, wird die Simulation beendet.

**Eigenschaft 5.5** Folgende Eigenschaft seien bei einer Untersuchung mit unterschiedlicher Last erfüllt:

- (a) Eine Untersuchung mit unterschiedlicher Last wird als eine Sequenz von Messblöcken mit verschiedenen Raten durchgeführt (vgl. Abbildung 5.5).
- (b) Ein Messblock besteht aus mehreren Messungen (vgl. Abbildung 5.6). Es beschreibt  $u$  die Anzahl der Messungen in einem Messblock.
- (c) Eine Messung besteht aus einer Messaufbauphase, einer Messphase und einer Messabbauphase (vgl. Abbildung 5.7). Es beschreibt  $T$  die Dauer der Messphase,  $T > 0$ .

**Bemerkung 5.6** Durch den vollständigen Abbau der Verbindungen nach jeder Messung sind die Messungen voneinander unabhängig.

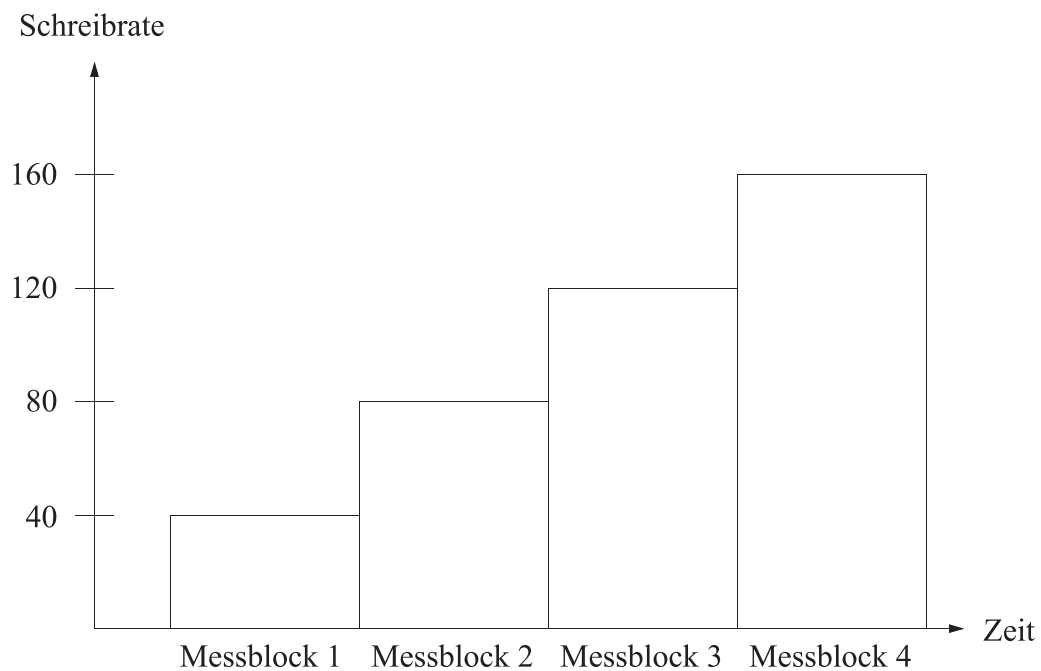


Abbildung 5.5: Sequenz von vier Messblöcken

**Eigenschaft 5.7** Folgende Eigenschaften erfüllt der Lastgenerator bei der Durchführung einer Untersuchung mit unterschiedlicher Last:

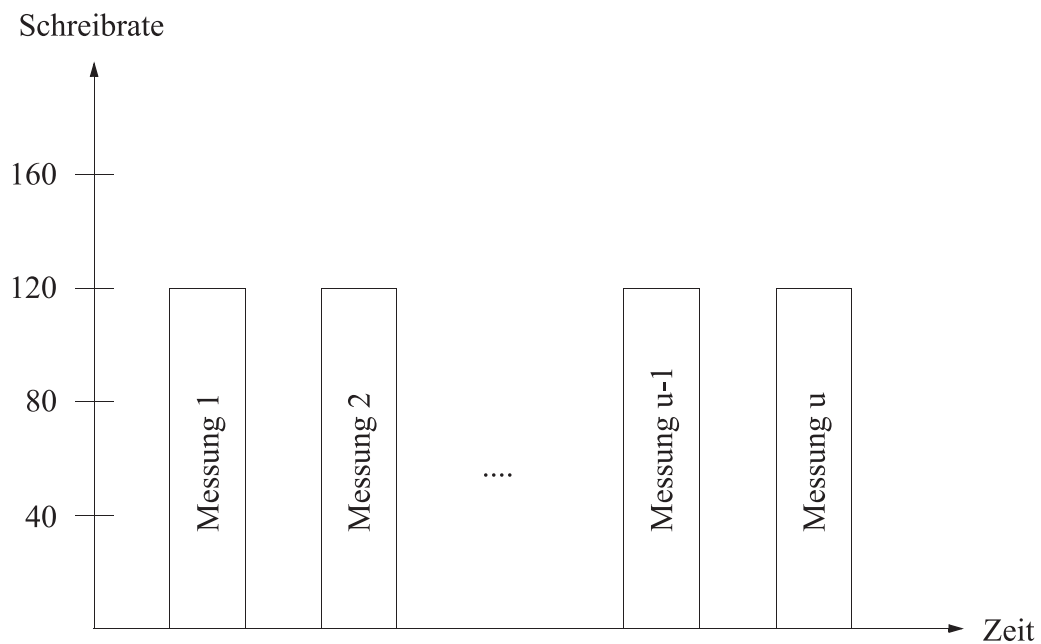


Abbildung 5.6: Aufbau eines Messblockes

- (a) Der Lastgenerator initiiert den Start und das Ende der Messblöcke, die Messungen sowie die Messaufbauphase, Messphase und Messabbauphase.
- (b) Der Lastgenerator ermittelt die Anzahl der aufgetretenen Fehlerereignisse.

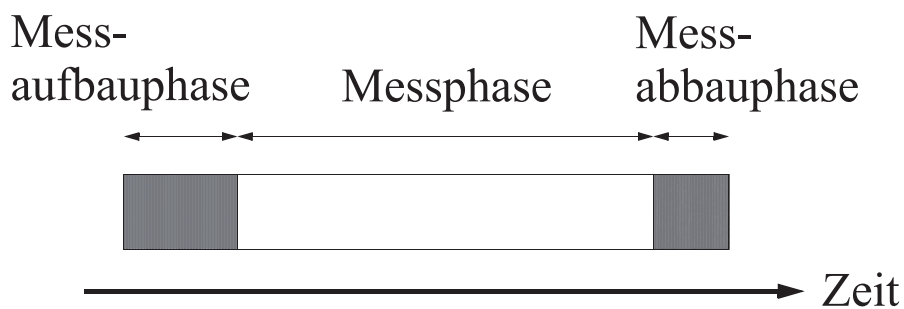


Abbildung 5.7: Aufbau einer Messung

**Eigenschaft 5.8** Folgende Eigenschaften seien beim Simulationsmodell erfüllt:

- (a) Jede Verbindung und jeder Knoten werden durch jeweils einen eigenen Thread ausgeführt.
- (b) Der Lastgenerator wird durch einen eigenen Thread ausgeführt.

- (c) Alle Threads der Knoten und der Verbindungen haben die gleiche Priorität. Diese Priorität ist statisch.
- (d) Die Threads werden durch einen Monoprozessor des Simulationsrechners nachgebildet.
- (e) Für das Modell ist die „Round Robin“ Strategie realisiert.
- (f) Alle Threads eines Knotens arbeiten auf dem gleichen Adressraum.

Die Prozessorbelegung der verschiedenen Threads während zweier Schreib-Lesevorgänge ist in Abbildung 5.8 dargestellt. Die Schreib-Lesevorgänge verschiedener Verbindungen sind voneinander unabhängig.

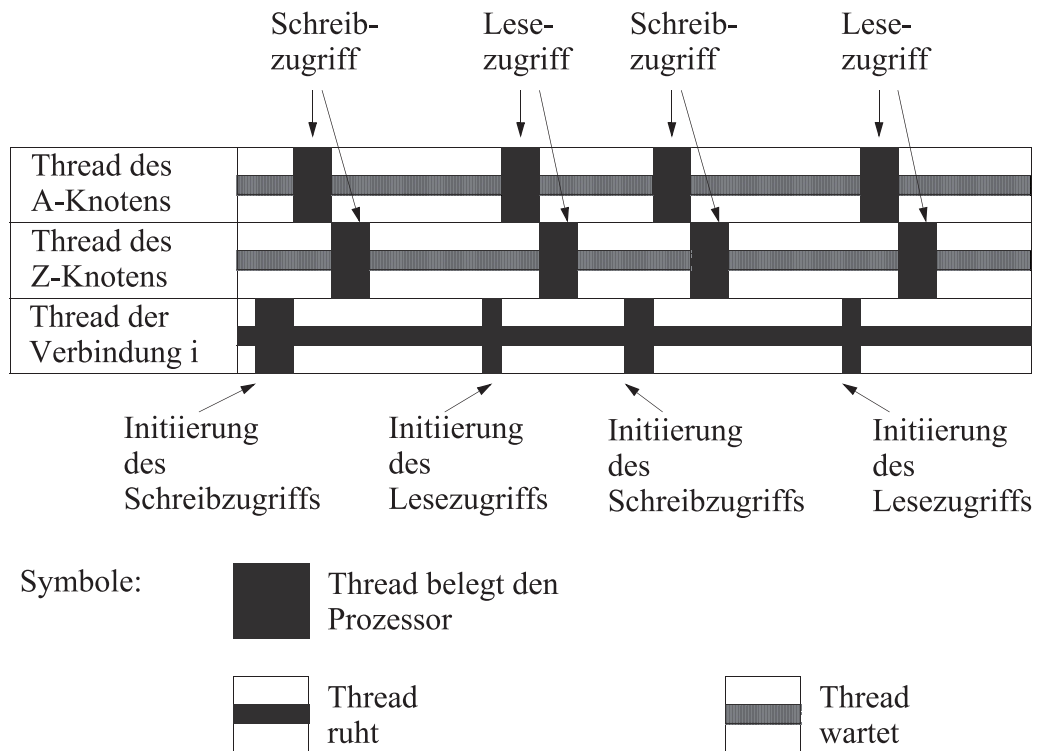


Abbildung 5.8: Prozessorbelegung der Threads während zweier Schreib-Lesevorgänge

In Abbildung 5.9 ist die Prozessorbelegung aller beteiligten Threads während des Verbindungsaufbaus und -abbaus dargestellt.

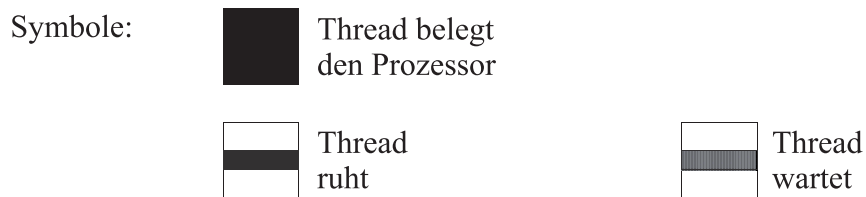
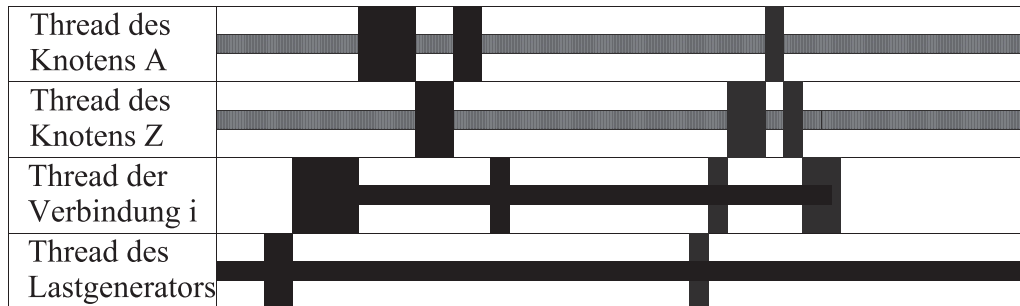


Abbildung 5.9: Prozessorbelegung aller beteiligten Threads während des Verbindungsaufbaus und -abbaus

Zum besseren Verständnis des Simulationsmodells wird nun das Ablaufdiagramm des Lastgenerators pro Messung in Pseudocode 5.1 vorgestellt.

```

1 run(Lastgenerator)
2 {
3   definiereAundZKnotenDerRouten;
4   I = berechneAnzahlDerVerbindungen;
5   AnzahlSLVGesamt = berechneAnzahlSLVGesamt;
6   for Knoten = (1,...,N)
7     erzeugeKnoten(Knoten);
8   for connectionID = (1,...,I)
9     erzeugeVerbindungen(connectionID);
10    StartVerbindung(connectionID) =
11      bestimmeRelZeitpunkteFuerVerbindungsaufbau
12      (Zufallszahlengenerator);
13    dauerZStrich(connectionID) =
14      bestimmeHaltedauer(Zufallszahlengenerator);
15    route(connectionID) =
16      bestimmeRouteVerbindung(Zufallszahlengenerator);
17    sortiereStartVerbindungsaufbauAufsteigend;
18    richteRelZeitpunkteVerbindungsaufbauAmZeitrasterSimRechnersAus;
19    for connectionID = (1,...,I)
20      speichereRelZeitpunkteVerbindungsaufbau(connectionID);
21    for writeID = (1,...,AnzahlSLVGesamt)
22      SLVZeit(writeID) =
23        bestimmeZeitpunktSLV(Zufallszahlengenerator);
24      sortiereZeitpunktSLVAufsteigend;
25      richteZeitpunkteSLVAmZeitrasterSimRechnersAus;
26      legeBetroffeneVerbindungDesSLVFest;
27      for connectionID = (1,...,I)
28        speichereRelZeitpunkteSLV(writeID,connectionID);
29    for connectionID = (1,...,I)
30      speichereVerbindungsdaten(connectionID);
31  Do
32    for connectionID = (1,...,I)
33      weckeVerbindung(connectionID);
34    for Knoten = (1,...,N)
35      weckeKnoten(Knoten);
36  until alleVerbindungenAbgebaut;
37  GebeErgebnisseAus;
38 }

```

Pseudocode 5.1: Ablaufdiagramm des Lastgenerators pro Messung

Das Ablaufdiagramm einer Verbindung wird in Pseudocode 5.2 beschrieben.

```
1 run (ConnectionID)
2 {
3   erhoeheAnzahlAktiverThreads ;
4   sleep (RelZeitpunktVerbindungsaufbau ) ;
5   sendeNachricht (CONN_REQ,A);
6   if (systemType = Uebertragungssystem)
7     ueberpruefeObAlleVerbindungenAufgebaut ;
8   if (systemType = Vermittlungssystem)
9     warteAufStartSignal ;
10  for (1 ,... , writeID)
11    warteBisNaechstenSchreibLesevorgangDerVerbindung ;
12    schreibLeseVorgang (A,Z, connectionID , dauerZStrich ) ;
13  warteBisHaltezeitVorbei ;
14  sendeNachricht (DISC_REQ,Z);
15  warteAufDiscAck ;
16  weckeAlleKnoten ;
17 }
```

Pseudocode 5.2: Ablaufdiagramm einer Verbindung

Das Ablaufdiagramm eines Knoten wird in Pseudocode 5.3 beschrieben.

```
1 run(NodeID)
2 {
3     warteBisNachrichtErhalten;
4     leseNachricht;
5     if (Nachricht = CONN_REQ)
6         allokiereLeitung;
7         sendeNachricht(CONN_ACK,Z);
8     if (Nachricht = CONN_ACK)
9         setzeStartSignalDerVerbindung;
10        weckeAlleVerbindungen;
11        weckeAlleKnoten;
12    if (Nachricht = DISC_REQ)
13        gebeLeitungFrei;
14        sendeNachricht(DISC_ACK,A);
15    if (Nachricht = DISC_ACK)
16        verringereAnzahlAktiverThreads;
17 }
```

Pseudocode 5.3: Ablaufdiagramm eines Knotens

Die Prozedur `schreibLeseVorgang` wird von dem Thread der Verbindung ausgeführt. Sie wählt Speicherfelder, beschreibt sie und überprüft den Inhalt der Speicherfelder.

```

1 schreibLeseVorgang (A, Z, connectionID , dauerZStrich)
2 {
3     SpeicherfeldA = waehleSpeicherfeldVonKnoten(A);
4     SpeicherfeldZ = waehleSpeicherfeldVonKnoten(Z);
5     A.schreibeSpeicherfeldVonKnoten
6         (SpeicherfeldA , connectionID);
7     Z.schreibeSpeicherfeldVonKnoten
8         (SpeicherfeldZ , connectionID);
9     sleep(dauerZStrich);
10    A.pruefeSpeicherfeldVonKnoten
11        (SpeicherfeldA , connectionID);
12    Z.pruefeSpeicherfeldVonKnoten
13        (SpeicherfeldZ , connectionID);
14 }

```

Pseudocode 5.4: Ablauf eines Schreib-Lesevorgangs

Die Prozedur `schreibeSpeicherfeldVonKnoten` wird von dem Thread des entsprechenden Knotens ausgeführt und wird in Pseudocode 5.5 beschrieben.

```

1 schreibeSpeicherfeldVonKnoten (Speicherfeld , connectionID)
2 {
3     global [Speicherfeld] = "ABC"+connectionID;
4     globalInt [Speicherfeld] = connectionID;
5     print ("Schreibzugriff der Verbindung "+connectionID+":_ "
6         + global [Speicherfeld]);
7 }

```

Pseudocode 5.5: Ablauf eines Schreibzugriffs auf ein Speicherfeld eines Knotens



Die Prozedur `pruefeSpeicherfeldVonKnoten` wird durch den Knoten ausgeführt, der Knoten überprüft, ob der Dateneintrag unverändert ist. Sie wird in Pseudocode 5.6 beschrieben.

```

1 pruefeSpeicherfeldVonKnoten ( Speicherfeld , connectionID )
2 {
3     print ( "Pruefen_des_Speicherfelds_" + Speicherfeld +
4             "_der_Verbindung_" + connectionID + " :_"
5             + global [ Speicherfeld ] );
6     if ( connectionID == globalInt [ Speicherfeld ] )
7     {
8         print ( "Korrektער_Eintrag_des_Speicherfelds_der_Verbindung_"
9                 + connectionID );
10    }
11    else
12    {
13        print ( connectionID + " !=_" + globalInt [ Speicherfeld ] );
14        print ( "Falscher_Eintrag_des_Speicherfelds_der_Verbindung_"
15                + connectionID );
16    }
17 }

```

Pseudocode 5.6: Ablauf der Überprüfung des Inhalts des Speicherfelds

### 5.3 Arten konkurrierender Zugriffe

In Abbildung 5.10 wird dargestellt, welche Arten konkurrierender Zugriffe auftreten können. Es können konkurrierende Zugriffe mit gleichem geplanten Timeslot und konkurrierende Zugriffe mit verschiedenen geplanten Timeslots auftreten. D.h. im Fall (b) ist der Schreib-Lesevorgang von Thread A zwar für den Takt  $p$  geplant, aufgrund der Auslastung des Prozessors und der Threadingmechanismen wird dieser Schreib-Lesevorgang jedoch erst im Takt  $p+1$  durchgeführt.

Bei der Durchführung der Simulationen wurde nicht nur die Gesamtanzahl aufgetretener konkurrierender Zugriffe erfasst, sondern zugleich, ob es sich um konkurrierende Zugriffe mit gleichen geplanten Timeslots (gTS) oder mit verschiedenen geplanten Timeslots (ngTS) handelt.

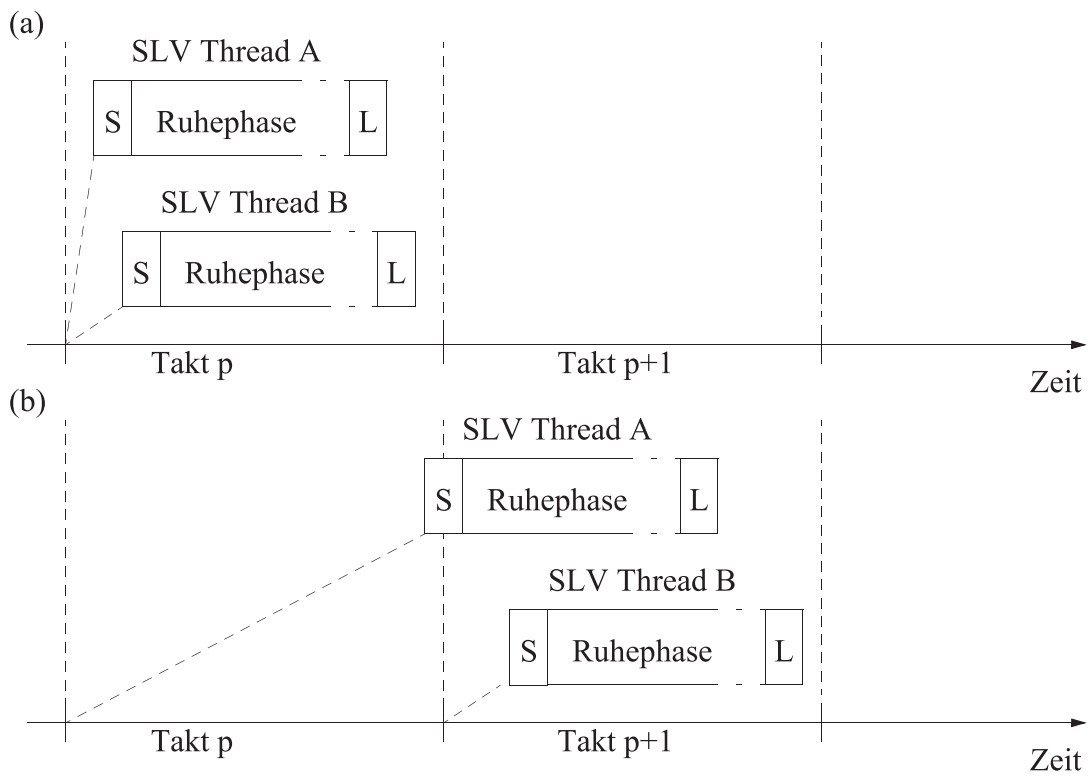


Abbildung 5.10: Arten konkurrierender Zugriffe



# Kapitel 6

## Modellauswertungen

Bertrand Russell *"Although this may seem a paradox,  
all exact science is dominated  
by the idea of approximation"*

Im Folgenden werden fünf verschiedene Modelle vorgestellt und anschliessend die Auswertungen des analytischen Modells und des Simulationsmodells aufgeführt und verglichen. Während vier dieser Modelle mit verschiedenen Ausprägungen unter moderater Last untersucht werden, wird ein Modell mit verschiedenen Ausprägungen und mit jeweils einer verschieden hohen Anzahl von Prozessen für eine Untersuchung unter hoher Last vorgestellt.

### 6.1 Parametrisierung der Modelle

Es werden in diesem Abschnitt sog. „Einfache Modelle“ von sog. „Komplexen Modellen“ unterschieden. Einfache Modelle bestehen aus zwei Knoten, haben genau eine Route und pro Knoten ein Speicherfeld. Komplexe Modelle bestehen dagegen aus fünf Knoten, mehreren Routen und pro Knoten aus bis zu zwei Speicherfeldern. Eine derartige Dimensionierung der Modelle zur Durchführung der Untersuchung ist sinnvoll, da auch bei der Durchführung von Tests an leitungsvermittelnden Kommunikationssystemen i.A. keine größeren Netze verwendet werden. Bei einem Netz aus fünf Knoten können alle üblichen Netzstrukturen wie z.B. Maschennetz, Sternnetz, Ringnetz oder auch ein hierarchisches Netz dargestellt werden.

#### 6.1.1 Modelle von Kommunikationssystemen

Kommunikationssysteme werden aufgrund der Eigenschaften der Gedächtnislosigkeit häufig durch Warteschlangen mit einer Poissonverteilung der Zwischenankunftszeiten modelliert (vgl. *Bolch* in [9]). So werden auch in dieser Arbeit die Warteschlangensysteme aller Modelle durch M/M/1/k-Systeme modelliert. D.h.

die Warteschlangensysteme haben die Länge  $k-1$ , die Verteilung der Zwischenankunftszeiten und die Verteilung der Bedienzeiten sind exponentialverteilt. Dies bedeutet, dass die Zwischenankunftszeiten gemäß  $1 - e^{-\lambda t}$  verteilt sind, sowie die Bedienzeiten gemäß  $1 - e^{-\nu t}$ . Die Raten der Verbindungsanforderungen sind durch die Modellparameter gegeben. Alle Routen im Netz haben die Länge 1. In den Tabellen ist jeweils die Art des Systems durch „V“ bzw. „Ü“ angegeben, d.h. ob die Parameter für ein Vermittlungssystem bzw. ein Übertragungssystem definiert werden. Beim Modell des Vermittlungssystems ist die Dauer der Haltephasen normalverteilt. Bei Modellen von Übertragungssystemen beginnt die Haltephase nachdem alle Verbindungen erfolgreich aufgebaut wurden und endet bevor die erste Verbindung wieder abgebaut wird. Deshalb ist bei diesen Modellen die mittlere Ankunftsrate  $\lambda$  der Verbindungsanforderungen nicht relevant und in den Tabellen nicht aufgeführt.

### 6.1.1.1 Einfache Modelle

In diesem Abschnitt wird ein einfaches Modell eines Übertragungssystems und ein einfaches Modell eines Vermittlungssystems vorgestellt. Beide Modelle besitzen eine minimale Dimensionierung. Bereits bei dieser Dimensionierung ist es möglich, einige prinzipielle Eigenschaften zur Erkennung konkurrierender Zugriffe nachweisen zu können. In Tabelle 6.1 werden die Parameter für diese Modelle vorgestellt.

Art des Systems	Anzahl der Knoten	Parameter der Leitungskapazität	Parameter der Haltephasen	Parameter der Routen	Parameter des Warteschlangensystems
V	$N = 2$	$c_{11} = 0,$ $c_{12} = 500,$ $c_{21} = 0,$ $c_{22} = 0.$	$H_i,$ $i \in \mathbb{N},$ normalverteilt, $\gamma = 2,$ $\sigma = 1.$	$S = 1,$ $r_{1;A} = 1,$ $r_{1;Z} = 2,$ $r_{1;1} = 1,$ $p(r_1) = 1.$	M/M/1/k, $\lambda$ variabel, $\nu$ nahe 0, $k = 500,$ FCFS.
Ü	$N = 2$	$c_{11} = 0,$ $c_{12} = 500,$ $c_{21} = 0,$ $c_{22} = 0.$	$\gamma = 1$	$S = 1,$ $r_{1;A} = 1,$ $r_{1;Z} = 2,$ $r_{1;1} = 1,$ $p(r_1) = 1.$	M/M/1/k, $\nu$ nahe 0, $k = 500,$ FCFS.

Tabelle 6.1: Einfaches Modell eines Kommunikationssystems

Es handelt es sich um ein Netz aus zwei Knoten, in welchem nur eine Route von Knoten 1 zu Knoten 2 existiert.

### 6.1.1.2 Komplexe Modelle

Beide komplexen Modelle von Kommunikationssystemen bestehen aus einem nahezu vollständig vermaschten Netz aus fünf Knoten. Die Topologie dieses Modells ist in Abbildung 6.1 dargestellt. In Tabelle 6.2 werden die Parameter für das Modell eines Vermittlungssystems und eines Übertragungssystems vorgestellt.

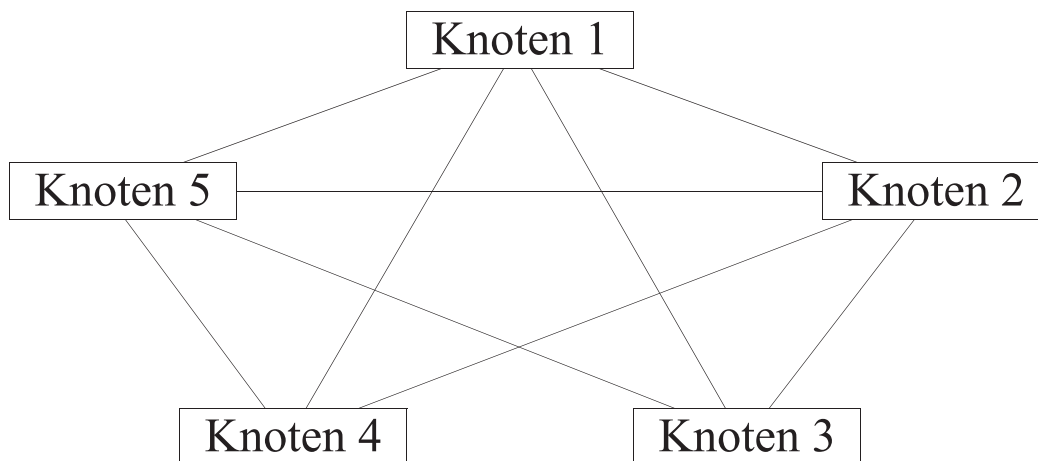


Abbildung 6.1: Topologie des komplexen Modells

Die Wahrscheinlichkeiten  $p(r_s)$ ,  $s \in \{1, \dots, S\}$ , dass eine Verbindung die Route  $r_s$  hat und vom A-Knoten zum Z-Knoten führt, sind in Tabelle 6.3 aufgeführt. Durch die unterschiedlichen Routenwahrscheinlichkeiten ergibt sich folglich auch pro Knoten eine unterschiedliche mittlere Anzahl von Verbindungen, welche im Knoten starten oder im Knoten enden.

### 6.1.2 Konkurrierende Zugriffe

In den Tabellen 6.4 und 6.5 werden für die vorgestellten Modelle von Kommunikationssystemen unterschiedlicher Komplexität die Parameter der Schreib-Lesevorgänge für die Untersuchung von konkurrierenden Zugriffen vorgestellt. Die Parameter der Schreib-Lesevorgänge sind von der Art des Systems unabhängig. Die Verteilungsfunktion der Schreib-Lesevorgänge wird durch  $G$  beschrieben.

#### 6.1.2.1 Einfache Modelle von Kommunikationssystemen

Die Parameter der konkurrierenden Zugriffe für die einfachen Modelle von Kommunikationssystemen werden in der Tabelle 6.4 aufgezeigt. Jeder der Knoten besitzt nur ein Speicherfeld.

Art des Systems	Anzahl der Knoten	Parameter der Leitungskapazität	Parameter der Haltephasen	Parameter der Routen	Parameter des Warteschlangensystems
V	$N = 5$	$c_{ii} = 0$ $\forall i =$ $\{1, \dots, 5\},$ $c_{34} = 0,$ sonst $c_{ij} = 2500$ $\forall i, j =$ $\{1, \dots, 5\}.$	$H_i,$ $i \in \{1, \dots, I\},$ normalverteilt, $\gamma = 4,$ $\sigma = 1.$	$S = 18,$ weitere Parameter siehe Tabelle 6.3.	M/M/1/k, $\lambda$ variabel, $\nu$ nahe 0, $k = 500,$ FCFS.
Ü	$N = 5$	$c_{ii} = 0$ $\forall i =$ $\{1, \dots, 5\},$ $c_{34} = 0,$ sonst $c_{ij} = 2500$ $\forall i, j =$ $\{1, \dots, 5\}.$	$\gamma = 1.$	$S = 18,$ weitere Parameter siehe Tabelle 6.3.	M/M/1/k, $\nu$ nahe 0, $k = 500,$ FCFS.

Tabelle 6.2: Komplexes Modell eines Kommunikationssystems

### 6.1.2.2 Komplexe Modelle von Kommunikationssystemen

Die komplexen Modelle besitzen pro Knoten eine unterschiedliche Anzahl von Speicherfeldern. Die Parameter der konkurrierenden Zugriffe für die komplexen Modelle von Kommunikationssystemen werden in der Tabelle 6.5 aufgeführt.

$s$	$p(r_s)$	$r_{s;A}$	$r_{s;Z}$	$r_s$
1	0.10	1	2	1-2
2	0.02	1	3	1-3
3	0.10	1	4	1-4
4	0.04	1	5	1-5
5	0.04	2	1	2-1
6	0.07	2	3	2-3
7	0.02	2	4	2-4
8	0.03	2	5	2-5
9	0.06	3	1	3-1
10	0.13	3	2	3-2
11	0.01	3	5	3-5
12	0.06	4	1	4-1
13	0.04	4	2	4-2
14	0.12	4	5	4-5
15	0.03	5	1	5-1
16	0.01	5	2	5-2
17	0.02	5	3	5-3
18	0.10	5	4	5-4

Tabelle 6.3: Definition der Routen und der Wahrscheinlichkeit je Route für die komplexen Modelle von Kommunikationssystemen

Verteilungsfunktion	Dauer der Ruhephase	Anzahl der Speicherfelder	Wahrscheinlichkeit der Auswahl des Speicherfeldes
$G = 1 - e^{-\mu t}$	$z = 32ms$	$F_1 = F_2 = 1$	$p_{1,1} = p_{2,1} = 1.$

Tabelle 6.4: Parameter der Schreib-Lesevorgänge für einfache Modelle von Kommunikationssystemen

Verteilungsfunktion	Dauer der Ruhephase	Anzahl der Speicherfelder	Wahrscheinlichkeit der Auswahl des Speicherfeldes
$G = 1 - e^{-\mu t}$	$z = 32ms$	$F_1 = F_3 = F_5 = 1,$ $F_2 = F_4 = 2$	$p_{1,1} = p_{3,1} = p_{5,1} = 1,$ $p_{2,1} = p_{2,2} = 0.5,$ $p_{4,1} = p_{4,2} = 0.5.$

Tabelle 6.5: Parameter der Schreib-Lesevorgänge für komplexe Modelle von Kommunikationssystemen



### 6.1.3 Last zur Untersuchung von konkurrierenden Zugriffen

In den Tabellen 6.6 und 6.7 werden für die vorgestellten Modelle von Kommunikationssystemen unterschiedlicher Komplexität verschiedene Lastbeschreibungen für die Untersuchung von konkurrierenden Zugriffen vorgestellt. Die Lastbeschreibungen werden in Bezug zum betrachteten Modell des Kommunikationssystems gesetzt. Die Anzahl der Knoten gibt an, ob ein einfaches oder ein komplexes Modell eines Kommunikationssystems betrachtet wird. Die verschiedenen Lasten in Kombination mit der Eigenschaft der Knoten des Modells werden als Lastvarianten der Untersuchungen bezeichnet. Die Lastvarianten A1,..., A5, C1, C2, E1, E2 und E3 beschreiben Varianten für Modelle von Vermittlungssystemen, die Lastvarianten B1,..., B5, D1 und D2 beschreiben Varianten für Modelle von Übertragungssystemen. Es ist  $H_M$  die durchschnittliche Dauer der Haltephase eines Vermittlungssystems. Die durchschnittliche Anzahl der zu einem Zeitpunkt in der Messphase bestehenden Verbindungen  $I_M$  wird bei einem Vermittlungssystem durch das Produkt der Ankunftsrate  $\lambda$  und der durchschnittlichen Dauer  $H_M$  der Haltephase berechnet. D.h. es gilt:

$$I_M = \lambda \cdot H_M.$$

Es ist  $I$  die Anzahl der in einer Minute existierenden Verbindungen im Netz in der Messphase. Bei Übertragungssystemen gilt  $I_M = I$ . Die Rate von Schreib-Lesevorgängen im Netz wird durch das Produkt der Anzahl der zu einem Zeitpunkt in der Messphase bestehenden Verbindungen  $I_M$  und der Schreibrate  $\mu$  auf eine Verbindung bestimmt. Die Rate von Schreib-Lesevorgängen steigt linear mit  $I_M$  bzw.  $\mu$ . Die Rate der Schreibzugriffe im Netz entspricht in der Messphase bei Übertragungssystemen dem Produkt

$$2\mu I_M.$$

Die Rate der Schreibzugriffe im Netz entspricht in der Messphase bei Vermittlungssystemen dem Produkt

$$2\mu\lambda H_M.$$

Die maximale Rate der Schreibzugriffe im Netz ist für die Lastvarianten A1,...,A5 und B1,...,B5 320, für die Lastvarianten C1, C2, D1 und D2 ist sie 1080, für die Lastvarianten E1, E2 und E3 ist sie 36000. Die verschiedenen Lasten wurden so definiert, dass die Anzahl der Schreib-Lesezugriffe im Netz für das System der Art Vermittlungssystem jeweils mit der Anzahl des Systems der Art Übertragungssystem übereinstimmt. Pro Lastvariante werden vier Untersuchungen durchgeführt. Es wird untersucht, ob die Auftrittshäufigkeit konkurrierender Zugriffe linear mit der Rate der Schreib-Lesevorgänge im Netz ansteigt. Um die Unterschiede bzgl. der Auftrittshäufigkeit in Abhängigkeit von der Anzahl der Verbindungen und von der Schreibrate zu erkennen, werden diese Lasten betrachtet und in den folgenden Abschnitten analysiert und ausgewertet.

### 6.1.3.1 Einfache Modelle von Kommunikationssystemen

Die verschiedenen Lastbeschreibungen für die einfachen Modelle von Kommunikationssystemen werden in der Tabelle 6.6 aufgezeigt.

Art des Systems	Anzahl der Knoten	Lastvariante	Untersuchungen für		
			$\lambda$	$I_M$	$\mu$
V	2	A1	2	$\lambda H_M$	40, 80, 120, 160
		A2	4	$\lambda H_M$	20, 40, 60, 80
		A3	20	$\lambda H_M$	4, 8, 12, 16
		A4	80	$\lambda H_M$	1, 2, 3, 4
		A5	80, 160, 240, 320	$\lambda H_M$	1
Ü	2	B1	N.A.	1	40, 80, 120, 160
		B2	N.A.	2	20, 40, 60, 80
		B3	N.A.	10	4, 8, 12, 16
		B4	N.A.	40	1, 2, 3, 4
		B5	N.A.	40, 80, 120, 160	1

Tabelle 6.6: Lastparameter für einfache Modelle von Kommunikationssystemen

### 6.1.3.2 Komplexe Modelle von Kommunikationssystemen

Die verschiedenen Lastbeschreibungen für die komplexen Modelle von Kommunikationssystemen werden in der Tabelle 6.7 aufgezeigt.

Art des Systems	Anzahl der Knoten	Lastvariante	Untersuchungen für		
			$\lambda$	$I_M$	$\mu$
V	5	C1	72	$\lambda H_M$	5, 10, 15, 20, 25, 30
		C2	144	$\lambda H_M$	2.5, 5, 7.5, 10, 12.5, 15
Ü	5	D1	N.A.	18	5, 10, 15, 20, 25, 30
		D2	N.A.	36	2.5, 5, 7.5, 10, 12.5, 15
V	5	E1	72	$\lambda H_M$	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
		E2	144	$\lambda H_M$	50, 100, 150, 200, 250, 300, 350, 400, 450, 500
		E3	288	$\lambda H_M$	25, 50, 75, 100, 125, 150, 175, 200, 225, 250

Tabelle 6.7: Lastparameter für komplexe Modelle von Kommunikationssystemen

## 6.2 Auswertung durch das analytische Modell

### 6.2.1 Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbindungen

Im Folgenden werden die Auftrittshäufigkeiten konkurrierender Zugriffe der einfachen Modelle von Kommunikationssystemen in Abhängigkeit der Rate der Schreib-Lesevorgänge pro Verbindung und der Anzahl der in einer Minute existierenden Verbindungen bestimmt. Es wird eine Formel für das Vermittlungssystem und eine Formel für das Übertragungssystem bestimmt. Nach deren Bestimmung erfolgt die Instanzierung für die Lastvarianten A1, B1 und B2. Durch die Anwendung der jeweiligen Formel kann für das Vermittlungssystem bzw. für das Übertragungssystem jeweils die Auftrittshäufigkeit für alle Ankunftsdaten der Anforderungen für Schreib-Lesevorgänge  $\mu$  berechnet werden. Es werden die Raten der Verbindungsanforderungen und der Anforderungen für einen Schreib-Lesevorgang pro Minute beschrieben. Eine Zeiteinheit ist ein sog. Timeslot und hat eine Dauer von 32 Millisekunden. Die Anzahl der Timeslots pro Minute ist folglich  $t = \frac{60000}{32} = 1875$ . Es werden die Auftrittshäufigkeiten konkurrierender Zugriffe jeweils pro Minute bestimmt.

**Vermittlungssystem:** Betrachtung eines Vermittlungssystems in Abhängigkeit von  $\mu$  Schreib-Lesevorgängen pro Verbindung und der Anzahl  $I$  der in einer Minute existierenden Verbindungen in der Messphase (Vermittlungssystem mit zwei Knoten ( $N = 2$ ) und einer Route ( $S = 1$ ), jeder Knoten hat ein Speicherfeld ( $F_1 = 1, F_2 = 1$ ), die Ankunftsrate der Verbindungsanforderungen pro Minute ist  $\lambda = 2, \gamma = 2$  (vgl. Tabellen 6.1 und 6.4). Die Betrachtung erfolgt in der Messphase, d.h. es werden genauso viele Verbindungen aufgebaut wie abgebaut (vgl. Eigenschaft 3.17(a)).

Es ist  $I_M = \gamma H_M$ . Es ist  $I \simeq \lambda$ .

Da nur ein Speicherfeld pro Knoten existiert, gilt:

$$p_{f_{1,1}} = p_{f_{2,1}} = 1.$$

Da nur eine Route existiert, gilt für  $n \in \{1, 2\}$ :

$$p_n = p(r_1) = 1.$$

Daraus folgt für  $n \in \{1, 2\}$  und  $m = 1$ :

$$q_{f_{n,m}} = p \cdot p_n \cdot p_{f_{n,m}} = p$$

In einer Minute gibt es  $t = 1875$  Zeiteinheiten. Die Rate der Schreib-Lesevorgänge auf eine beliebige aber feste Verbindung kann durch die Rate der Schreib-Lesevor-

gänge pro Minute im Netz geteilt durch die Anzahl der in einer Minute existierenden Verbindungen approximiert werden. Die Wahrscheinlichkeit  $p$  eines Schreib-Lesevorgangs auf eine beliebige aber feste Verbindung innerhalb einer Zeiteinheit wird durch die Rate der Schreib-Lesevorgänge auf eine beliebige aber feste Verbindung geteilt durch die durchschnittliche Dauer der Haltephasen approximiert. Die Wahrscheinlichkeit eines Schreibzugriffs in einem Knoten entspricht der Wahrscheinlichkeit eines Schreib-Lesevorgangs in dem Knoten.

Es folgt somit:

$$p \simeq \frac{\mu}{t} = \frac{\mu}{1875}$$

Es ist:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I \\ &= 1 - I \cdot p \cdot (1 - p)^{I-1} - (1 - p)^I \\ &\simeq 1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I \end{aligned}$$

Es folgt:

$$\begin{aligned} E[K_{1,1}] &= E[K_{2,1}] = tp(K_{f_{n,m}} = 1) = 1875 \cdot p(K_{f_{n,m}} = 1) \\ &\simeq 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

Da  $F_1 = F_2 = 1$  ist, folgt:

$$\begin{aligned} E[K_1] &= E[K_2] = E[K_{1,1}] = E[K_{2,1}] \\ &\simeq 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

und

$$\begin{aligned} E[K_N] &= E[K_1] + E[K_2] \\ &\simeq 2 \cdot 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

**Beispiel 6.1** Lastvariante A1 mit  $\mu = 80$  Schreib-Lesevorgänge pro Verbindung und  $I \simeq \lambda = 2$  in einer Minute existierenden Verbindungen:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - 2 \cdot \frac{80}{1875} \cdot \left(1 - \frac{80}{1875}\right)^1 - \left(1 - \frac{80}{1875}\right)^2 \\ &\simeq 1 - 0.081692444 - 0.916487111 \\ &= 0.001820445 \end{aligned}$$

$$\begin{aligned} E[K_{1,1}] = E[K_{2,1}] &= tp(K_{f_{n,m}} = 1) \simeq 1875 \cdot 0.001820445 = 3.4133, \\ E[K_1] = E[K_2] &= E[K_{1,1}] = E[K_{2,1}] \simeq 3.4133, \end{aligned}$$

$$E[K_N] = E[K_1] + E[K_2] \simeq 6.8266.$$

Für ein Vermittlungssystem aus zwei Knoten werden unter den beschriebenen Parametern bei einer Last von  $\mu = 80$  pro Verbindung ca. 7 konkurrierende Zugriffe pro Minute erwartet.

**Beispiel 6.2** Lastvariante A2 mit  $\mu = 40$  Schreib-Lesevorgänge pro Verbindung und  $I \simeq \lambda = 4$  in einer Minute existierenden Verbindungen:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - 4 \cdot \frac{40}{1875} \cdot \left(1 - \frac{40}{1875}\right)^3 - \left(1 - \frac{40}{1875}\right)^4 \\ &\simeq 1 - 0.079987679 - 0.917358704 \\ &= 0.002653616 \end{aligned}$$

$$E[K_{1,1}] = E[K_{2,1}] = tp(K_{f_{n,m}} = 1) \simeq 1875 \cdot 0.002653616 = 4.9756,$$

$$E[K_1] = E[K_2] = E[K_{1,1}] = E[K_{2,1}] \simeq 4.9756,$$

$$E[K_N] = E[K_1] + E[K_2] \simeq 9.9511.$$

Für ein Vermittlungssystem aus zwei Knoten werden unter den beschriebenen Parametern bei einer Last von  $\mu = 40$  pro Verbindung ca. 10 konkurrierende Zugriffe pro Minute erwartet.

Die Werte für die anderen Schreibraten und für die anderen Lastvarianten eines Vermittlungssystems können analog berechnet werden. Die Ergebnisse sind in Abbildung 6.2 dargestellt.

**Übertragungssystem:** Betrachtung eines Übertragungssystems in Abhängigkeit der Rate der Schreib-Lesevorgänge  $\mu$  pro Verbindung und der Anzahl  $I$  der existierenden Verbindungen in der Messphase

(Übertragungssystem mit zwei Knoten ( $N = 2$ ) und einer Route ( $S = 1$ ), jeder Knoten hat ein Speicherfeld ( $F_1 = 1, F_2 = 1$ ) (vgl. Tabellen 6.1 und 6.4). Die Betrachtung erfolgt in der Messphase, d.h. es wurden in dieser Phase bereits alle Verbindungen aufgebaut, keine Verbindung wird in dieser Phase abgebaut (vgl. Eigenschaft 3.17(b)).

Da nur ein Speicherfeld pro Knoten existiert, gilt:

$$p_{f_{1,1}} = p_{f_{2,1}} = 1.$$

Da nur eine Route existiert, gilt für  $n \in \{1, 2\}$ :

$$p_n = p(r_1) = 1.$$

Daraus folgt für  $n \in \{1, 2\}$  und  $m = 1$ :

$$q_{f_{n,m}} = p \cdot p_n \cdot p_{f_{n,m}} = p$$

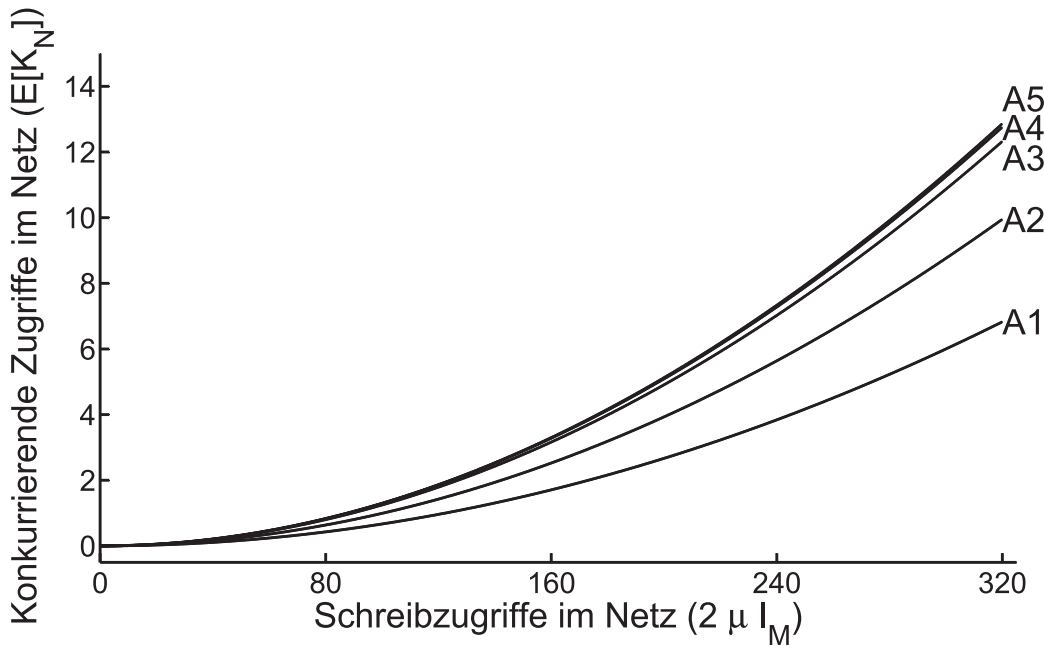


Abbildung 6.2: Ergebnisse des einfachen Modells eines Vermittlungssystems — Lastvarianten A1 – A5

In einer Minute gibt es  $t = 1875$  Zeiteinheiten. Die Rate der Schreib-Lesevorgänge pro Verbindung auf eine beliebige aber feste Verbindung kann durch die Rate der Schreib-Lesevorgänge pro Minute im Netz geteilt durch die Anzahl der existierenden Verbindungen pro Minute approximiert werden. Die Wahrscheinlichkeit  $p$  eines Schreib-Lesevorgangs auf eine beliebige aber feste Verbindung in einer Zeiteinheit, wird durch die Rate der Schreib-Lesevorgänge auf eine beliebige aber feste Verbindung geteilt durch die Anzahl der Zeiteinheiten pro Minute approximiert. Die Wahrscheinlichkeit eines Schreibzugriffs in einem Knoten entspricht der Wahrscheinlichkeit eines Schreib-Lesevorgangs.

Es folgt somit:

$$p \simeq \frac{\mu}{t} = \frac{\mu}{1875}.$$

Es ist:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I \\ &= 1 - I \cdot p \cdot (1 - p)^{I-1} - (1 - p)^I \\ &\simeq 1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I \end{aligned}$$

Es folgt:

$$\begin{aligned} E[K_{1,1}] &= E[K_{2,1}] = tp(K_{f_{n,m}} = 1) = 1875 \cdot p(K_{f_{n,m}} = 1) \\ &\simeq 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

Da  $F_1 = F_2 = 1$  ist, folgt:

$$\begin{aligned} E[K_1] &= E[K_2] = E[K_{1,1}] = E[K_{2,1}] \\ &\simeq 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

und

$$\begin{aligned} E[K_N] &= E[K_1] + E[K_2] \\ &\simeq 2 \cdot 1875 \cdot \left(1 - I \cdot \frac{\mu}{1875} \cdot \left(1 - \frac{\mu}{1875}\right)^{I-1} - \left(1 - \frac{\mu}{1875}\right)^I\right) \end{aligned}$$

**Beispiel 6.3** Lastvariante B1 mit  $\mu = 160$  Schreib-Lesevorgänge pro Verbindung und  $I = 1$  Verbindung

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - I \cdot q_{f_{n,m}} \cdot (1 - q_{f_{n,m}})^{I-1} - (1 - q_{f_{n,m}})^I \\ &= 1 - I \cdot p \cdot (1 - p)^{I-1} - (1 - p)^I \\ &= 1 - 1 \cdot p \cdot (1 - p)^0 - (1 - p) = 0. \end{aligned}$$

Unabhängig von der Wahl von  $t$  und  $\mu$  folgt:

$$E[K_{1,1}] = E[K_{2,1}] = 0$$

sowie

$$E[K_1] = E[K_2] = 0$$

und

$$E[K_N] = 0.$$

Insbesondere gilt für  $\mu = 160$ :

$$E[K_N] = 0.$$

Unabhängig von der Rate der Schreib-Lesevorgänge tritt bei einem Modell der Art Übertragungssystem mit nur einer Verbindung kein konkurrierender Zugriff auf.

**Beispiel 6.4** Lastvariante B2 mit  $\mu = 80$  Schreib-Lesevorgänge pro Verbindung und  $I = 2$  Verbindungen

$$p \simeq \frac{\mu}{t} = \frac{80}{1875} = 0.042\bar{6}$$

Es folgt:

$$\begin{aligned} p(K_{f_{n,m}} = 1) &= 1 - I \cdot p \cdot (1 - p)^{I-1} - (1 - p)^I \\ &= 1 - 2 \cdot p \cdot (1 - p)^1 - (1 - p)^2 \\ &= 1 - 2 \cdot p + 2 \cdot p^2 - 1 + 2 \cdot p - p^2 \\ &= p^2 \\ &\simeq 0.001820 \end{aligned}$$

$$E[K_{1,1}] = E[K_{2,1}] = t \cdot p^2 \simeq 1875 \cdot 0.001820 = 3.41\bar{3}.$$

$$E[K_1] = E[K_2] = E[K_{1,1}] = E[K_{2,1}] \simeq 3.41\bar{3}$$

$$E[K_N] = E[K_1] + E[K_2] \simeq 2 \cdot 3.41\bar{3} = 6.82\bar{6}.$$

Für ein Modell eines Übertragungssystems aus zwei Knoten werden unter den beschriebenen Parametern bei einer Last von  $\mu = 80$  ca. 7 konkurrierende Zugriffe pro Minute erwartet.

Die Werte für andere Raten von Schreib-Lesevorgängen und für die Lastvarianten B3 – B5 können analog berechnet werden. Die Ergebnisse sind in Abbildung 6.3 dargestellt.

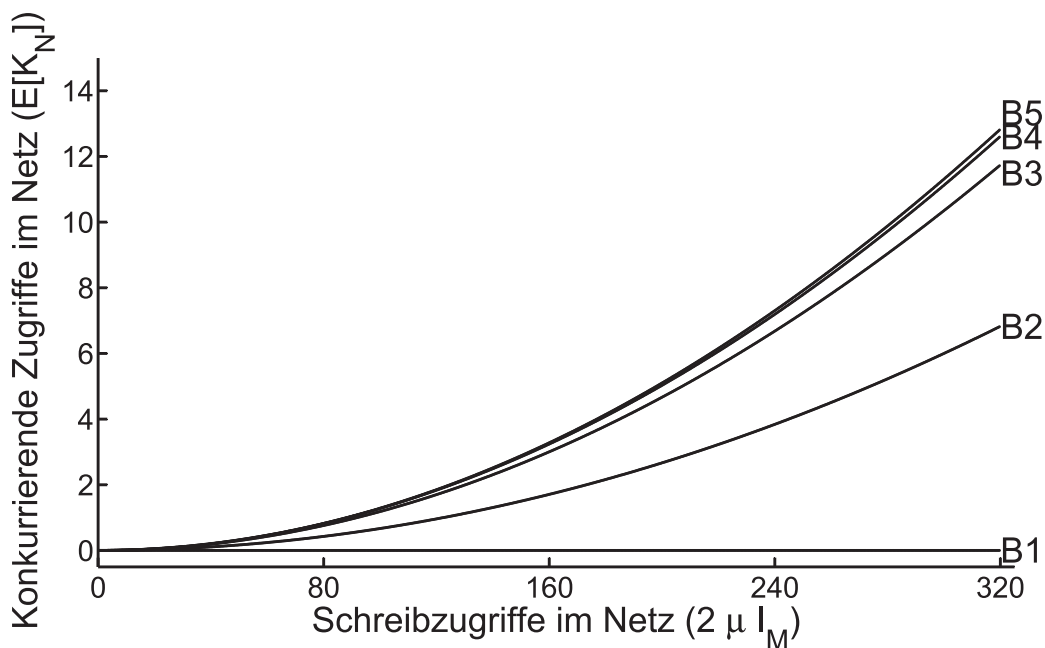


Abbildung 6.3: Ergebnisse des einfachen Modells eines Übertragungssystems — Lastvarianten B1 – B5

Zum Vergleich zwischen Modellen von Vermittlungssystemen und Modellen von Übertragungssystemen sind die Ergebnisse für das Netz mit der Lastvariante A2 für das Modell eines Vermittlungssystems und mit der Lastvariante B2 für das Modell eines Übertragungssystems in der Abbildung 6.4 dargestellt.

Ist die Anzahl von Verbindungen  $I$  in einer Minute in der Messphase gering, so kann bei ansonsten gleichen Parametern eine deutlich höhere Anzahl von konkurrierenden Zugriffen bei einem Vermittlungssystem in Vergleich zu einem Übertragungssystem erreicht werden.



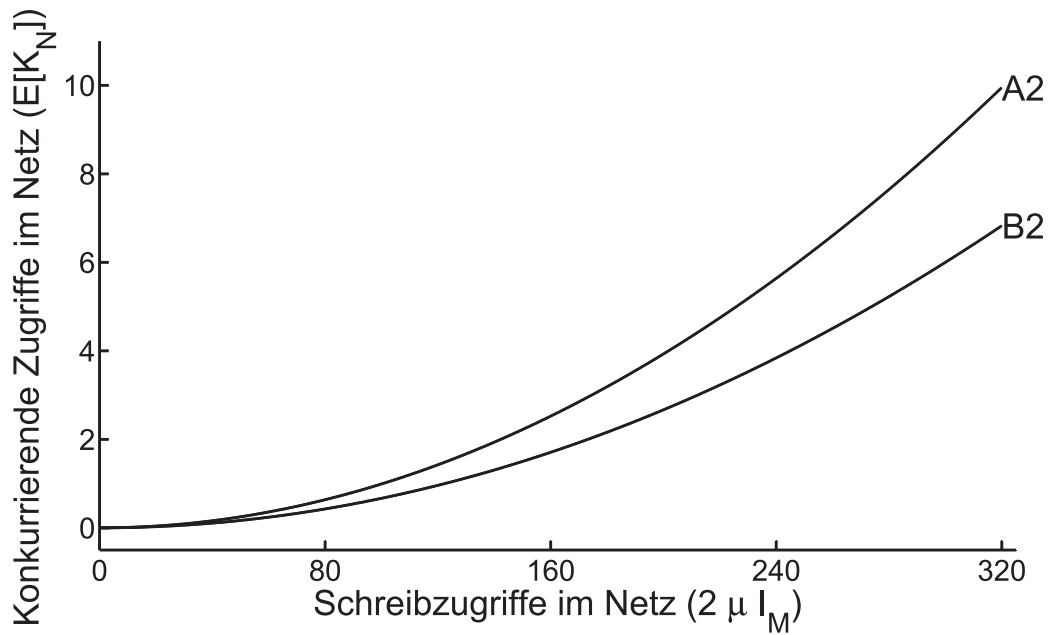


Abbildung 6.4: Ergebnisse des einfachen Modells eines Vermittlungssystems — Lastvariante A2 — und eines Übertragungssystems — Lastvariante B2

In den Abbildungen 6.5 und 6.6 sind die Ergebnisse für das komplexe Modell eines Vermittlungssystem dargestellt. Die Auftrittshäufigkeit für das komplexe Modell unterscheidet sich bei doppelter Anzahl von Verbindungen bei gleicher Rate von Schreibzugriffen im Netz nur marginal.

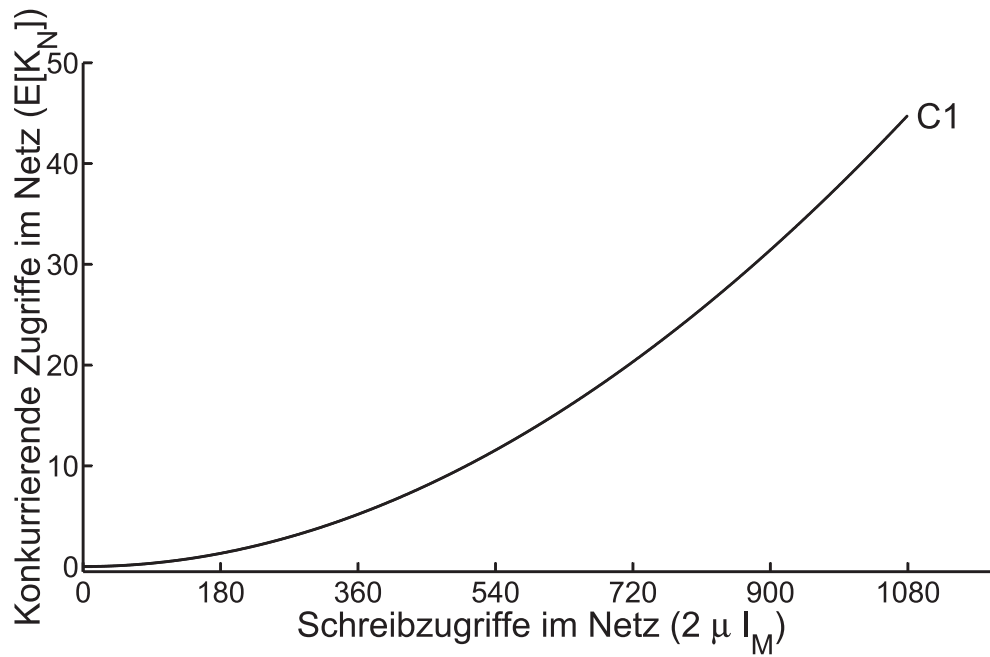


Abbildung 6.5: Ergebnisse des komplexen Modells eines Vermittlungssystems — Lastvariante C1

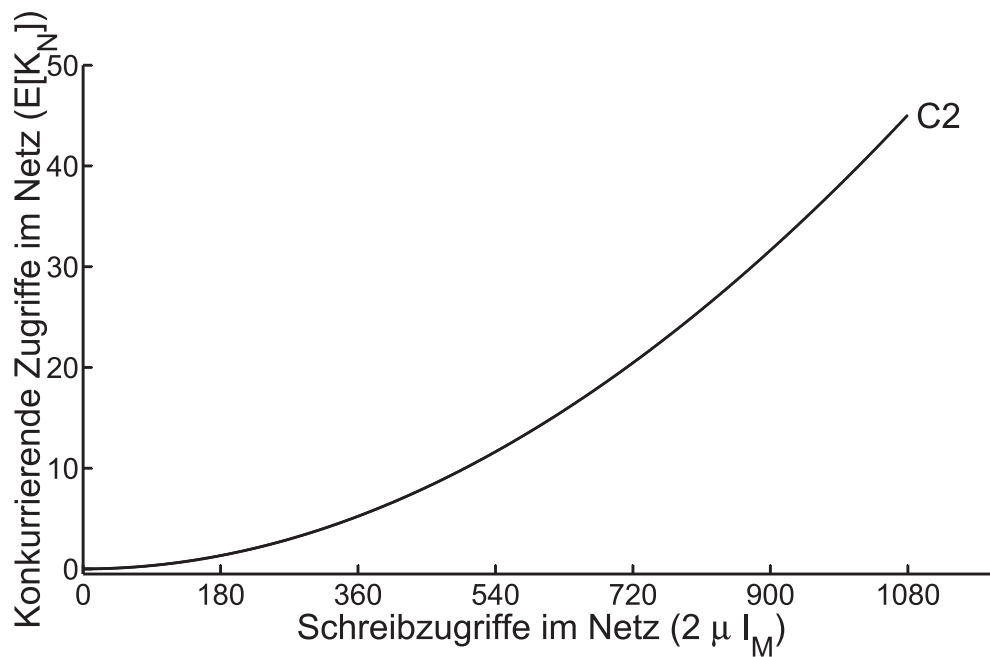


Abbildung 6.6: Ergebnisse des komplexen Modells eines Vermittlungssystems — Lastvariante C2

Die unterschiedlichen Erwartungswerte konkurrierender Zugriffe bei dem komplexen Modell eines Übertragungssystems in den verschiedenen Speicherfeldern der verschiedenen Knoten sind in Abbildung 6.7 dargestellt. Finden in den verschiedenen Speicherfeldern eines Knotens jeweils unterschiedlich viele Schreib-Lesevorgänge statt, so unterscheidet sich auch die Erwartungswerte der konkurrierenden Zugriffe in den jeweiligen Speicherfeldern.

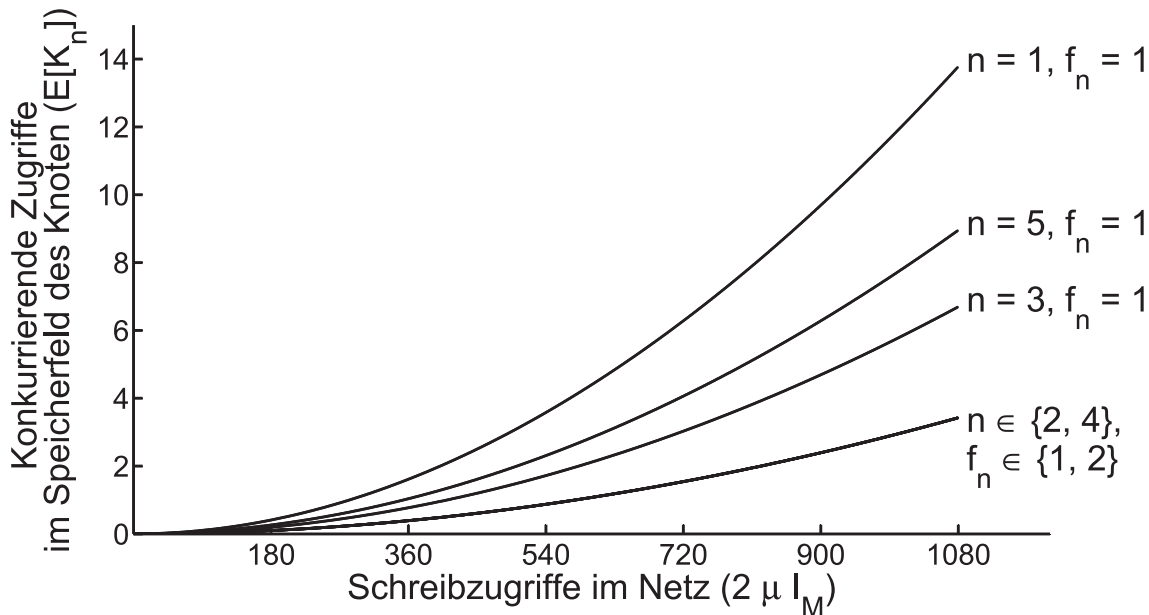


Abbildung 6.7: Ergebnisse des komplexen Modells eines Übertragungssystems — Lastvariante D1 — für die verschiedenen Speicherfelder der Knoten

Die vorherigen Berechnungen haben gezeigt, dass mit steigender Rate von Schreibzugriffen die Auftrittshäufigkeit konkurrierender Zugriffe überproportional steigt. Ebenso ist eine geschickte Wahl der Parameter „Art der Knoten“ und „Anzahl der Verbindungen“ sinnvoll, da es bei gleicher Rate von Schreibzugriffen im Netz zu einer doppelt so hohen Auftrittshäufigkeit konkurrierender Zugriffe kommen kann. Wenige Speicherfelder pro Knoten bedingen auch eine hohe Auftrittshäufigkeit konkurrierender Zugriffe.

Die Auftrittshäufigkeit konkurrierender Zugriffe im Netz ist die Summe der Auftrittshäufigkeit konkurrierender Zugriffe in den verschiedenen Knoten im Netz. Bei steigender Rate von Schreibzugriffen steigt die Auftrittshäufigkeit konkurrierender Zugriffe überproportional. Aus diesem Grund ist es offensichtlich, dass bei fester Anzahl von Schreibzugriffen im Netz und Gleichverteilung der Schreibzugriffe über alle Knoten die Auftrittshäufigkeit konkurrierender Zugriffe bei kleinen Netzen größer ist als bei großen Netzen.

Die maximale Anzahl konkurrierender Zugriffe pro Minute im Netz ist für alle Konfigurationen durch die zweifache Rate der Schreib-Lesevorgänge begrenzt und

entspricht der Anzahl von Schreibzugriffen pro Minute im Netz. Diese maximale Anzahl kann nur bei einem Netz aus zwei Knoten mit jeweils nur einem Speicherfeld erhalten werden. D.h. es muss gelten:  $q_{f_{n,m}} = 1$  für  $n = 1, 2$ . Ebenso muss die Rate der Schreib-Lesevorgänge pro Minute mindestens der Anzahl der Timeslots pro Minute entsprechen.

In Abbildung 6.8 sind die Ergebnisse für das komplexe Modell eines Vermittlungssystem für eine hohe Anzahl von Verbindungen und eine sehr hohe Anzahl von Schreibzugriffen im Netz dargestellt. Bei dieser sehr hohen Anzahl von Schreibzugriffen zeigt sich, dass die Anzahl der konkurrierenden Zugriffe im Netz nur noch nahezu linear steigen. Dies ist der Fall, da bei so vielen Schreibzugriffen häufig mehr als zwei Schreib-Lesezugriffe in einem Takt kollidieren und dies jedoch nur als ein konkurrierender Zugriff erfasst wird. Die Ergebnisse für die Lastvarianten E2 und E3 sind analog und werden nicht explizit dargestellt.

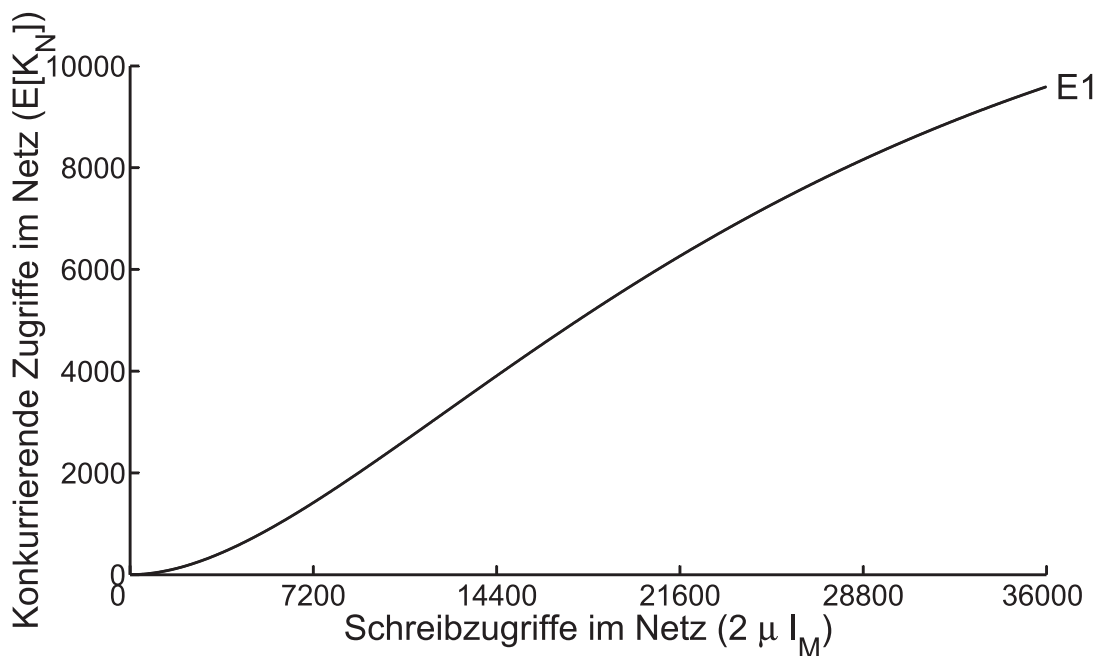


Abbildung 6.8: Ergebnisse des komplexen Modells eines Vermittlungssystem — Lastvariante E1

### 6.2.2 Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Dauer der Schreib-Lesevorgänge

Im Folgenden wird die Auftrittshäufigkeit des komplexen Modells eines Übertragungssystems bei unterschiedlicher Dauer der Schreib-Lesevorgänge für die Lastvariante D1 bestimmt. Für die Dauer der Schreib-Lesevorgänge  $z$  wurden verschiedene Werte ( $z = 0.5 \delta, z = \delta, z = 1.5 \delta, z = 2 \delta$ ) mit  $\delta = 32 \text{ ms}$  gewählt. In Abbildung 6.9 sind die berechneten Werte eingetragen.

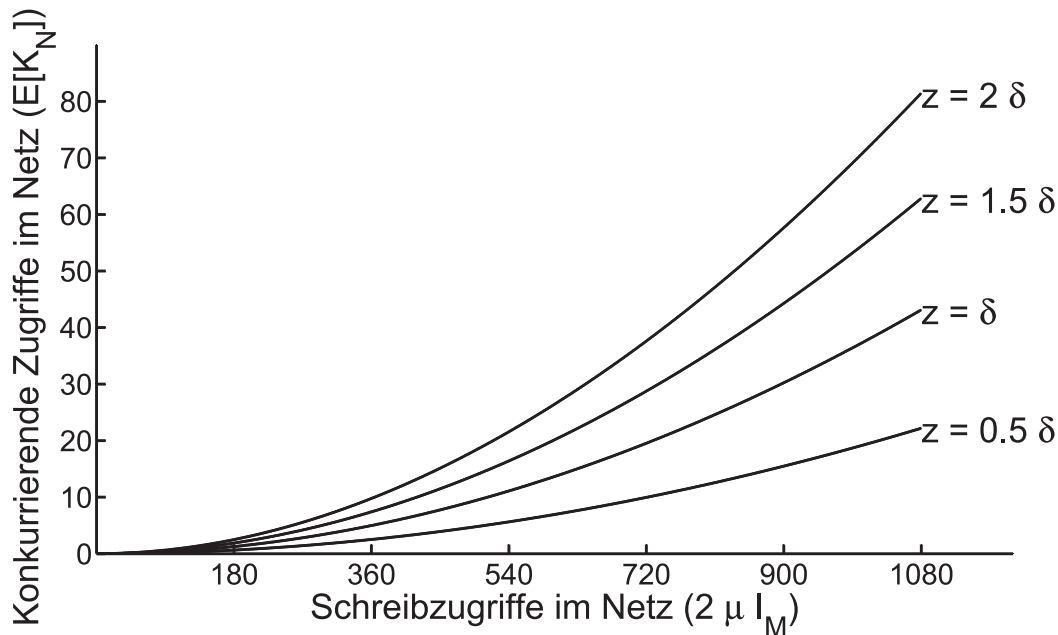


Abbildung 6.9: Ergebnisse des komplexen Modells eines Übertragungssystems — Lastvariante D1 — bei unterschiedlicher Dauer der Schreib-Lesevorgänge mit  $\delta = 32 \text{ ms}$

Bei Verdopplung der Dauer der Schreib-Lesevorgänge verdoppelt sich nahezu auch die Auftrittshäufigkeit konkurrierender Zugriffe.

### 6.2.3 Bestimmung der Auftrittshäufigkeit in Abhängigkeit von Betriebssystemeigenschaften

Jedes Betriebssystem versucht, abhängig von der Priorität der Prozesse, die Wartezeit der Prozesse möglichst gering zu halten und die Prozesse schnellstmöglich abzuarbeiten. In den meisten Betriebssystemen wird dies durch das Timeslicing-Verfahren „Round Robin“ mit fester Dauer der Timeslots realisiert. Es wird versucht, Restriktionen bzgl. der Wartezeit einzuhalten. Kommt es zu Spitzen mit besonders viel Last (den sog. „Bursts“), so werden Restriktionen bzgl. der Verzögerung der Abarbeitung verletzt.

Sobald ein Prozess seine Prozessorzeit aufgebraucht hat, wird er - wenn ein anderer Prozess mit gleicher oder höherer Priorität auf die Abarbeitung wartet - von diesem Prozess verdrängt. Diese Verdrängung wird als Context Switch bezeichnet. Dieser Context Switch benötigt Zeit, die so genannte Context Switch Time. Übersteigt die Context Switch Time eine kritische Grenze, so ist der verwendete Scheduling-Algorithmus ineffizient, da zu viel Zeit beim Context Switch verloren geht (vgl. *Probst in [58]*). Aus diesem Grund haben manche speziellen Betriebssysteme für Systeme, bei welchen derartige Spitzen vorkommen können, spezielle Scheduling-Mechanismen.

*Galli* führt in [19] verschiedene Arten von Scheduling-Algorithmen auf. So existieren einige sogenannte Adaptive Scheduler, welche zusätzliche Informationen verwenden, um ggf. die ursprüngliche Entscheidung bzgl. des Scheduling zu ändern. Beim „Short Term Quality of Service Deficit Round Robin“ wird die Dauer eines Timeslots (auch als Quantum bezeichnet) periodisch an den gemessenen Inputraten und dem Rückstand angepasst. *Kim* untersuchte in [38], wie sich ein derartiger Algorithmus bei sehr hoher Systemlast verhält.

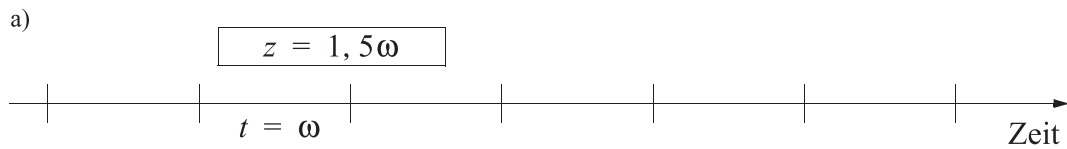
Aus diesen Gründen wird nun die Auftrittshäufigkeit konkurrierender Zugriffe unter der Annahme untersucht, dass sich die Dauer der Timeslots bei einer gewissen Systemlast verändert.

Ein konkurrierender Zugriff kann nur auftreten, wenn mindestens ein Prozessorswitch während eines Schreib-Lesevorgangs durchgeführt wird. In Abbildung 6.10 ist dargestellt, wie sich die Wahrscheinlichkeit mindestens eines Prozessorswitches  $p_{switch}$  in Abhängigkeit der Dauer der Timeslots bei konstanter Dauer des Schreib-Lesevorgangs verändert. Es beschreibt  $t$  die Dauer eines Timeslots und  $z$  die Dauer eines Schreib-Lesevorgangs. Beide Parameter werden in Abhängigkeit des Parameters  $\omega$  beschrieben. Die Tabelle 6.8 beschreibt die Wahrscheinlichkeit für verschiedene  $t$  und  $z$ . Es zeigt sich, dass es stets zu einem Prozessorswitch kommt, wenn die Dauer des Schreib-Lesevorgangs  $z$  die Dauer des Timeslots  $t$  übersteigt. Das mathematische Modell zur Berechnung der Auftrittshäufigkeit enthält als Parameter die Dauer des Timeslots  $t$ . Ändert das Betriebssystem ab einer gewissen Systemlast die Dauer der Timeslots, so ändert sich der Wert  $t$ . Wird  $t$  vergrößert, so sinkt die Wahrscheinlichkeit eines Prozessorswitches. Steigt die Systemlast weiter, so ist wiederum ein Anstieg der Auftrittshäufigkeit möglich.

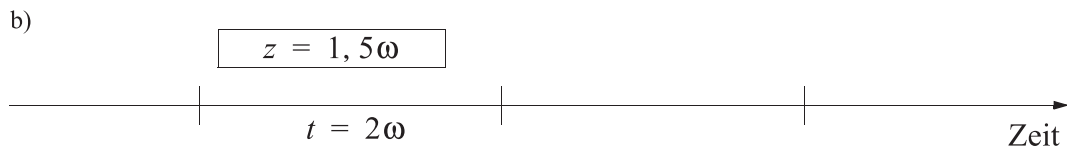
	$z = 0.75 \omega$	$z = 1.5 \omega$	$z = 3.0 \omega$	$z = 6.0 \omega$
$t = \omega$	$p_{switch} = 0.75$	$p_{switch} = 1.0$	$p_{switch} = 1.0$	$p_{switch} = 1.0$
$t = 2 \omega$	$p_{switch} = 0.375$	$p_{switch} = 0.75$	$p_{switch} = 1.0$	$p_{switch} = 1.0$
$t = 4 \omega$	$p_{switch} = 0.1875$	$p_{switch} = 0.375$	$p_{switch} = 0.75$	$p_{switch} = 1.0$

Tabelle 6.8: Wahrscheinlichkeit mindestens eines Prozessorswitches während eines Schreib-Lesevorgangs

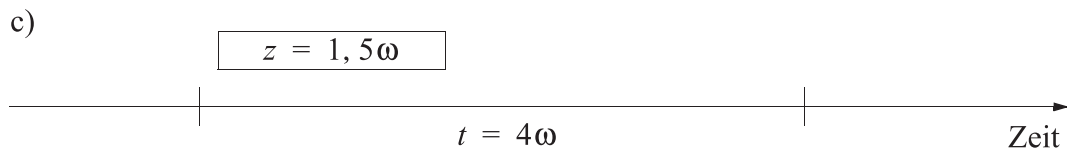
Abbildung 6.11 stellt die Auftrittshäufigkeit des Modells B2 dar unter der Annah-



Unabhängig vom Startzeitpunkt des Schreib-Lesevorgangs findet ein Prozessorswitch während des Schreib-Lesevorgangs statt.



Bei einer doppelten Taktlänge findet mit einer Wahrscheinlichkeit  $p_{\text{switch}} = 0,75$  ein Prozessorswitch während des Schreib-Lesevorgangs statt.



Bei einer vierfachen Taktlänge findet mit einer Wahrscheinlichkeit  $p_{\text{switch}} = 0,375$  ein Prozessorswitch während des Schreib-Lesevorgangs statt.

Abbildung 6.10: Prozessorswitch während eines Schreib-Lesevorgangs

me, dass bei einer Rate von 100 Schreib-Lesevorgängen im Knoten 1 die Dauer des Timeslots  $t$  verdoppelt wird.

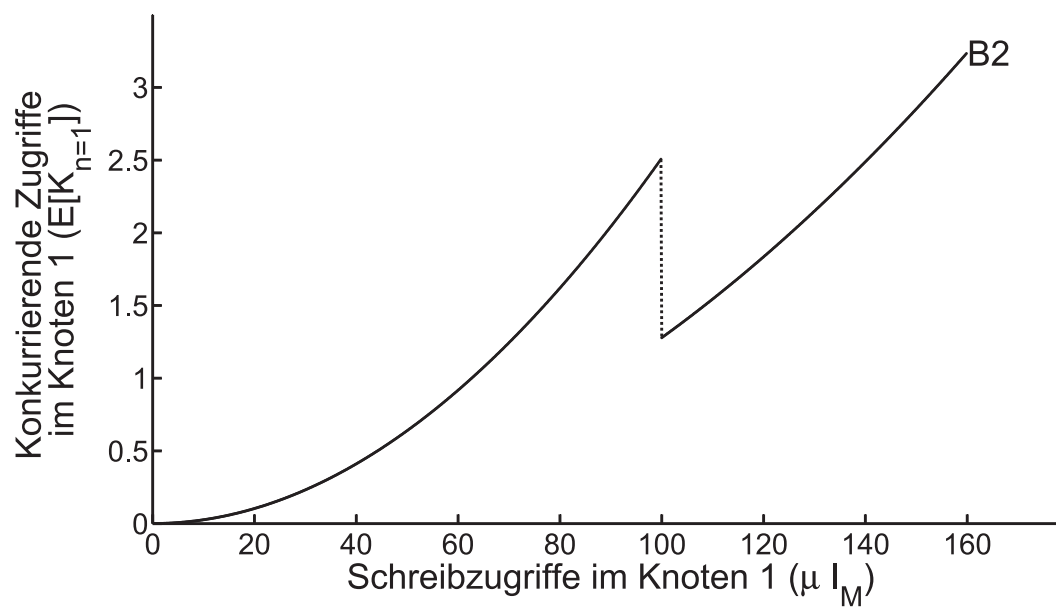


Abbildung 6.11: Ergebnisse des einfachen Modells eines Übertragungssystems B2 mit Verdopplung der Dauer des Timeslots  $t$  bei einer Rate von 100 Schreib-Lesevorgängen im Knoten 1



## 6.3 Auswertung durch das Simulationsmodell

In diesem Abschnitt werden die Ergebnisse des Simulationsmodells präsentiert.

### 6.3.1 Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbindungen

**Ergebnis 6.5** In der Tabelle 6.9 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvarianten A1 bis A5 dargestellt. Sie wurden durch  $u = 30$  Messungen erhalten. Die Konfidenzintervalle, deren untere Grenze jeweils mit  $K_-$  und deren obere Grenze mit  $K_+$  bezeichnet sind, sind in Tabelle 6.10 aufgeführt.

Lastvariante		$2\mu I_M = 80$	$2\mu I_M = 160$	$2\mu I_M = 240$	$2\mu I_M = 320$
A1	gesamt	0.73	3.27	4.80	8.27
	gTS	0.73	2.13	2.93	5.80
	ngTS	0	1.13	1.87	2.47
A2	gesamt	0.53	4.13	7.00	12.00
	gTS	0.40	3.00	5.13	9.07
	ngTS	0.13	1.13	1.87	2.93
A3	gesamt	1.00	4.13	8.93	14.73
	gTS	0.87	3.13	6.67	11.20
	ngTS	0.13	1.0	2.27	3.53
A4	gesamt	1.27	3.80	10.27	20.20
	gTS	0.80	2.67	7.47	15.07
	ngTS	0.47	1.13	2.80	5.13
A5	gesamt	1.20	5.27	11.00	18.30
	gTS	0.73	3.73	8.20	13.13
	ngTS	0.47	1.53	2.80	5.17

Tabelle 6.9: Auftrittshäufigkeit konkurrierender Zugriffe für das einfache Modell eines Vermittlungssystems (Lastvarianten A1 bis A5)

Lastvariante		$2\mu I_M = 80$	$2\mu I_M = 160$	$2\mu I_M = 240$	$2\mu I_M = 320$
A1	gesamt	0.73	3.27	4.80	8.27
	$K_-$	0.24	1.77	2.98	5.00
	$K_+$	1.23	4.80	6.62	11.50
A2	gesamt	0.53	4.13	7.00	12.00
	$K_-$	0.20	2.67	4.20	9.04
	$K_+$	0.87	5.60	9.80	15.00
A3	gesamt	1.00	4.13	8.93	14.73
	$K_-$	0.57	3.14	6.75	12.40
	$K_+$	1.43	5.13	11.10	17.10
A4	gesamt	1.27	3.80	10.27	20.20
	$K_-$	0.73	2.56	8.56	17.80
	$K_+$	1.80	5.04	12.00	22.60
A5	gesamt	1.20	5.27	11.00	18.30
	$K_-$	0.45	3.96	9.08	16.20
	$K_+$	1.95	6.57	12.90	20.40

Tabelle 6.10: 95%-Konfidenzintervalle für das einfache Modell eines Vermittlungssystems (Lastvarianten A1 bis A5)

**Ergebnis 6.6** In der Tabelle 6.11 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvarianten B1 bis B5 dargestellt. Sie wurden durch  $u = 30$  Messungen erhalten. Die Konfidenzintervalle sind in Tabelle 6.12 aufgeführt.

Lastvariante		$2\mu I_M = 80$	$2\mu I_M = 160$	$2\mu I_M = 240$	$2\mu I_M = 320$
B1	gesamt	0	0	0	0
	gTS	0	0	0	0
	ngTS	0	0	0	0
B2	gesamt	0.67	0.93	3.67	7.57
	gTS	0.67	0.87	3.47	7.33
	ngTS	0	0.07	0.20	0.23
B3	gesamt	0.53	3.60	7.43	11.00
	gTS	0.40	3.33	7.27	10.87
	ngTS	0.13	0.27	0.17	0.13
B4	gesamt	1.07	3.23	7.93	12.33
	gTS	0.93	2.90	7.47	11.53
	ngTS	0.13	0.33	0.47	0.80
B5	gesamt	1.13	2.87	8.40	12.33
	gTS	1.13	2.40	7.20	10.13
	ngTS	0	0.47	1.20	2.20

Tabelle 6.11: Auftrittshäufigkeit konkurrierender Zugriffe für das einfache Modell eines Übertragungssystems (Lastvarianten B1 bis B5)

Lastvariante		$2\mu I_M = 80$	$2\mu I_M = 160$	$2\mu I_M = 240$	$2\mu I_M = 320$
B2	gesamt	0.67	0.93	3.67	7.57
	$K_-$	0.21	0.43	2.81	6.40
	$K_+$	1.11	1.44	4.52	8.73
B3	gesamt	0.53	3.60	7.43	11.00
	$K_-$	0.15	2.54	6.14	8.90
	$K_+$	0.92	4.66	8.72	13.10
B4	gesamt	1.07	3.23	7.93	12.33
	$K_-$	0.56	2.47	6.45	10.80
	$K_+$	1.57	4.00	9.42	13.80
B5	gesamt	1.13	2.87	8.40	12.33
	$K_-$	0.67	2.02	8.56	10.50
	$K_+$	1.60	3.71	12.00	14.10

Tabelle 6.12: 95%-Konfidenzintervalle für das einfache Modell eines Übertragungssystems (Lastvarianten B2 bis B5)

**Ergebnis 6.7** In der Tabelle 6.13 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvarianten C1 und C2 dargestellt. Sie wurden durch  $u = 20$  Messungen erhalten. Die Konfidenzintervalle sind in Tabelle 6.14 aufgeführt.

Last-variante		$2\mu I_M$ = 180	$2\mu I_M$ = 360	$2\mu I_M$ = 540	$2\mu I_M$ = 720	$2\mu I_M$ = 900	$2\mu I_M$ = 1080
C1	gesamt	1.55	6.95	16.95	29.45	42.75	61.35
	gTS	1.20	5.60	12.55	22.60	30.00	47.35
	ngTS	0.35	1.35	4.40	6.85	12.75	14.00
C2	gesamt	1.55	7.90	17.30	28.80	38.35	59.85
	gTS	1.15	6.10	12.40	20.35	29.15	43.00
	ngTS	0.40	1.80	4.90	8.45	9.20	16.85

Tabelle 6.13: Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems (Lastvarianten C1 und C2)

Last-variante		$2\mu I_M$ = 180	$2\mu I_M$ = 360	$2\mu I_M$ = 540	$2\mu I_M$ = 720	$2\mu I_M$ = 900	$2\mu I_M$ = 1080
C1	gesamt	1.55	6.95	16.95	29.45	42.75	61.35
	$K_-$	0.74	5.57	14.25	26.30	37.80	55.88
	$K_+$	2.36	8.33	19.70	32.60	47.70	66.80
C2	gesamt	1.55	7.90	17.30	28.80	38.35	59.85
	$K_-$	1.08	6.34	15.81	25.50	33.90	53.60
	$K_+$	2.02	9.46	18.80	32.10	42.80	66.10

Tabelle 6.14: 95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems (Lastvarianten C1 und C2)

**Ergebnis 6.8** In der Tabelle 6.15 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvarianten D1 und D2 dargestellt. Sie wurden durch  $u = 20$  Messungen erhalten. Die Konfidenzintervalle sind in Tabelle 6.16 aufgeführt.

**Ergebnis 6.9** In der Tabelle 6.17 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvarianten E1, E2 und E3 dargestellt. Sie wurden durch  $u = 10$  Messungen erhalten. Die Konfidenzintervalle sind in Tabelle 6.18 aufgeführt.

Wie an den Werten in der Tabelle 6.17 gesehen werden kann, kommt es bei der höchsten Rate von Schreib-Lesevorgänge bei den Lastvarianten E1, E2 und E3

Last- variante		$2\mu I_M$ = 180	$2\mu I_M$ = 360	$2\mu I_M$ = 540	$2\mu I_M$ = 720	$2\mu I_M$ = 900	$2\mu I_M$ = 1080
D1	gesamt	1.20	4.85	12.50	20.10	29.75	43.80
	gTS	1.20	4.70	12.15	19.25	28.00	40.35
	ngTS	0	0.15	0.35	0.85	1.75	3.45
D2	gesamt	1.55	5.40	12.45	23.10	30.65	46.30
	gTS	1.45	5.30	12.25	22.00	28.90	44.15
	ngTS	0.10	0.10	0.20	1.10	1.75	2.15

Tabelle 6.15: Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Übertragungssystems (Lastvarianten D1 und D2)

Last- variante		$2\mu I_M$ = 180	$2\mu I_M$ = 360	$2\mu I_M$ = 540	$2\mu I_M$ = 720	$2\mu I_M$ = 900	$2\mu I_M$ = 1080
D1	gesamt	1.20	4.85	12.50	20.10	29.75	43.80
	$K_-$	0.64	3.86	10.62	17.20	26.58	40.14
	$K_+$	1.75	5.84	14.40	22.98	32.92	47.46
D2	gesamt	1.55	5.40	12.45	23.10	30.65	46.30
	$K_-$	0.99	4.31	10.21	20.94	28.32	43.30
	$K_+$	2.11	6.49	14.69	25.26	32.98	49.30

Tabelle 6.16: 95%-Konfidenzintervalle für das komplexe Modell eines Übertragungssystems (Lastvarianten D1 und D2)

bei ca. 80 % der Schreib-Lesevorgängen zu konkurrierenden Zugriffen. Es ist zu beachten, dass die Rate der Schreib-Lesevorgänge nicht beliebig erhöht werden kann. Der Prozessor kann nur eine begrenzte Rate der Schreib-Lesevorgänge zeitnah abarbeiten.

### 6.3.2 Bestimmung der Auftrittshäufigkeit bei unterschiedlicher Dauer der Schreib-Lesevorgänge

Wie in Abschnitt 5.1 beschrieben wird, müssen für die Durchführung der Simulation gewisse Voraussetzungen erfüllt sein. So soll die Dauer des Schreib-Lesevorgangs etwa 60% der Taktlänge entsprechen (vgl. Voraussetzung 5.3). Wird die Untersuchung mit doppelt langen Schreiblesevorgängen bei gleicher Taktlänge durchgeführt, so ist diese Voraussetzung nicht mehr erfüllt. Ebenso kommt es - wie es in Tabelle 6.8 dargestellt wird, zu einer Erhöhung der Wahrscheinlichkeit mindestens eines Prozessorswitches während eines Schreib-Lesevorgangs.

Aus diesen Gründen wurde auf eine Simulation dieser Problemstellung verzichtet, da wesentliche Voraussetzungen nicht erfüllt sind und somit die Aussagekraft sehr gering wäre.

Last-variante		$2\mu I_M$ = 3600	$2\mu I_M$ = 7200	$2\mu I_M$ = 10800	$2\mu I_M$ = 14400	$2\mu I_M$ = 18000
E1	gesamt	595	2344	4701	7040	10296
	gTS	419	1539	2809	4180	5699
	ngTS	176	805	1892	2860	4597
E2	gesamt	620	2219	4852	7363	11123
	gTS	446	1512	3131	4536	6639
	ngTS	174	707	1721	2827	4484
E3	gesamt	666	2420	5208	7835	11297
	gTS	470	1618	3160	4597	6739
	ngTS	196	802	2048	3238	4558

Last-variante		$2\mu I_M$ = 21600	$2\mu I_M$ = 25200	$2\mu I_M$ = 28800	$2\mu I_M$ = 32400	$2\mu I_M$ = 36000
E1	gesamt	14197	17842	22762	25345	29805
	gTS	6780	8575	9837	10616	10164
	ngTS	7417	9267	12925	14729	19641
E2	gesamt	14715	18362	21898	25829	28372
	gTS	8141	8469	10992	14568	15290
	ngTS	6574	9893	10906	11262	13082
E3	gesamt	14608	18716	23437	26407	30193
	gTS	8404	10370	12163	13089	14293
	ngTS	6204	8346	11274	13318	15900

Tabelle 6.17: Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems (Lastvarianten E1, E2 und E3)

### 6.3.3 Bestimmung der Auftrittshäufigkeit in Abhängigkeit von Betriebssystemeigenschaften

Wie in Abbildung 6.13 ersichtlich ist, fällt ab einer bestimmten Anzahl von Schreib-Lesevorgängen im Netz die Anzahl der konkurrierenden Zugriffe, welche durch mindestens zwei im geplanten Timeslot durchgeführten Schreib-Lesevorgänge verursacht wurden, ab oder stagniert. Die Anzahl der konkurrierenden Zugriffe, welche durch mindestens zwei durchgeführte Schreib-Lesevorgänge verursacht wurden, wobei mindestens einer dieser Schreib-Lesevorgänge nicht im geplanten Timeslot durchgeführt wurde, steigt zeitgleich an.

Um herauszufinden, ob dies auf die Prozessorauslastung zurückzuführen ist, wird nun ein zeitlicher Streckfaktor  $q$  mit  $q > 1$  eingeführt. Bei der folgenden Untersuchung wurde bezogen auf das Zeitintervall des Simulationsrechners

- die Schreibrate um den zeitlichen Streckfaktor  $1/q$  verringert,

Last- variante		$2\mu I_M$ = 3600	$2\mu I_M$ = 7200	$2\mu I_M$ = 10800	$2\mu I_M$ = 14400	$2\mu I_M$ = 18000
E1	gesamt	595	2344	4701	7040	10296
	$K_-$	518	2062	3939	5991	9360
	$K_+$	672	2625	5462	8088	11230
E2	gesamt	620	2219	4852	7363	11123
	$K_-$	553	2068	4287	6742	10533
	$K_+$	685	2369	5415	7984	11712
E3	gesamt	666	2420	5208	7835	11297
	$K_-$	594	2230	4891	7342	10868
	$K_+$	738	2609	5524	8326	11725
Last- variante		$2\mu I_M$ = 21600	$2\mu I_M$ = 25200	$2\mu I_M$ = 28800	$2\mu I_M$ = 32400	$2\mu I_M$ = 36000
E1	gesamt	14197	17842	22762	25345	29805
	$K_-$	12972	16267	20754	23517	27658
	$K_+$	15422	19419	24768	27172	31950
E2	gesamt	14715	18362	21898	25829	28372
	$K_-$	14281	17225	20344	24370	26857
	$K_+$	15148	19497	23453	27288	29885
E3	gesamt	14608	18716	23437	26407	30193
	$K_-$	13894	17823	22183	25740	28979
	$K_+$	15321	19609	24690	27073	31405

Tabelle 6.18: 95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems (Lastvarianten E1, E2 und E3)

- die Rate der Verbindungsanforderungen um den zeitlichen Streckfaktor  $1/q$  verringert,
- die Dauer der Verbindungsaufbauphase um den zeitlichen Streckfaktor  $q$  erhöht,
- die Dauer der Schreib-Lesevorgänge um den zeitlichen Streckfaktor  $q$  erhöht,
- die Dauer der Haltephase um den zeitlichen Streckfaktor  $q$  erhöht,
- die Dauer der Untersuchung um den zeitlichen Streckfaktor  $q$  erhöht,
- das Intervall vom Start der Haltephase bis zum Start der Schreib-Lesevorgänge um den zeitlichen Streckfaktor  $q$  erhöht,
- das Intervall vom Start der Messung bis zum Start des Verbindungsaufbaus um den zeitlichen Streckfaktor  $q$  erhöht.

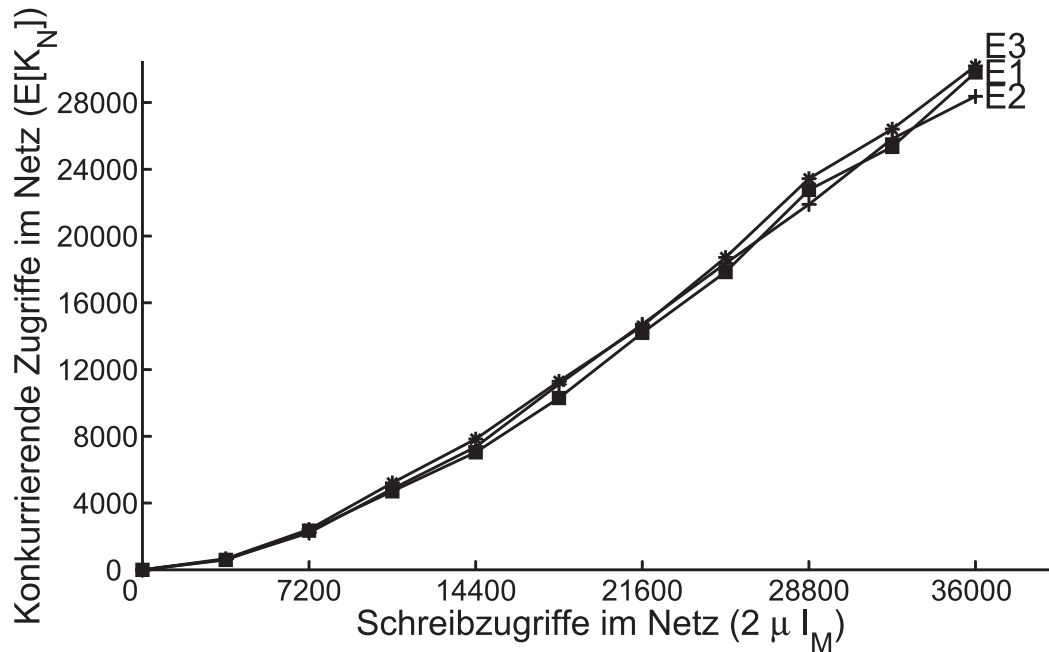


Abbildung 6.12: Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante E1 versus Lastvariante E2 versus Lastvariante E3

Abbildung 6.14 stellt eine Haltephase mit einem Schreib-Lesevorgang ohne zeitliche Streckung und mit zeitlicher Streckung (mit  $q = 2$ ) dar.

Durch die zeitliche Streckung wird

- eine geringere Prozessorauslastung und folglich
- eine geringere Spitzenlast

erhalten.

**Ergebnis 6.10** In der Tabelle 6.19 sind die durch das Simulationsmodell erhaltenen Mittelwerte für die Lastvariante E1 unter Verwendung des zeitlichen Streckfaktors  $q = 1.2$  dargestellt. Sie wurden durch  $u = 10$  Messungen erhalten. Die Konfidenzintervalle sind in Tabelle 6.20 aufgeführt.

Die nachfolgende Abbildung 6.15 zeigt die Ergebnisse unter Verwendung des zeitlichen Streckfaktors  $q = 1.2$ . Wie zu sehen ist, steigt die Auftrittshäufigkeit konkurrierender Zugriffe im gleichen geplanten Timeslot leicht weiter.



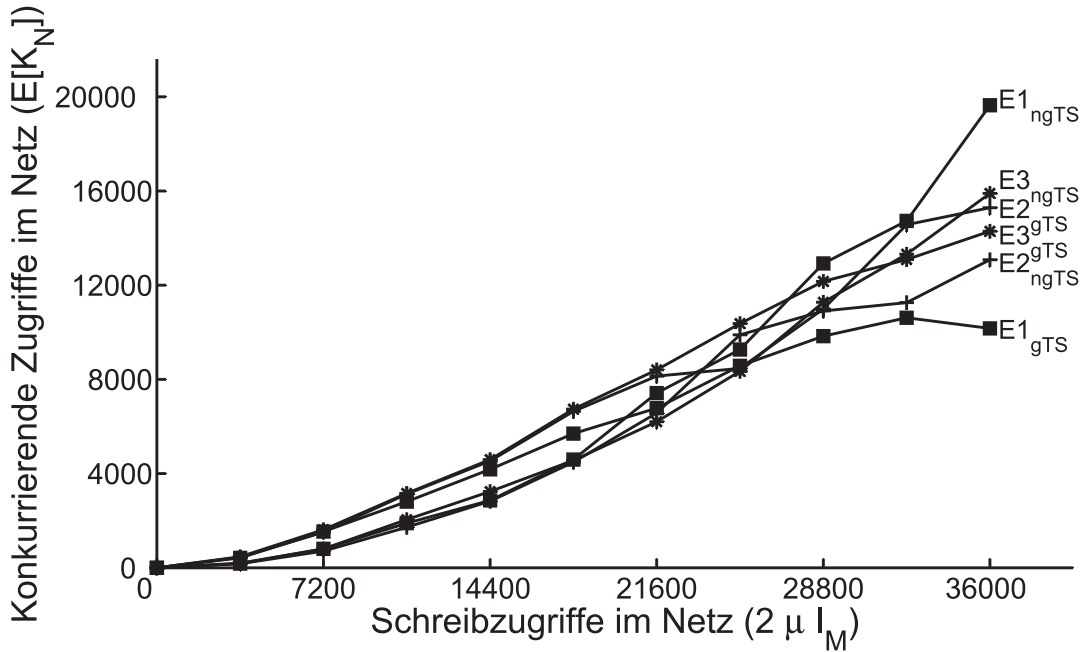


Abbildung 6.13: Ergebnisse bzgl. geplanter und nicht geplanter Timeslots für das komplexe Modell eines Vermittlungssystems — Lastvariante E1 versus Lastvariante E2 versus Lastvariante E3

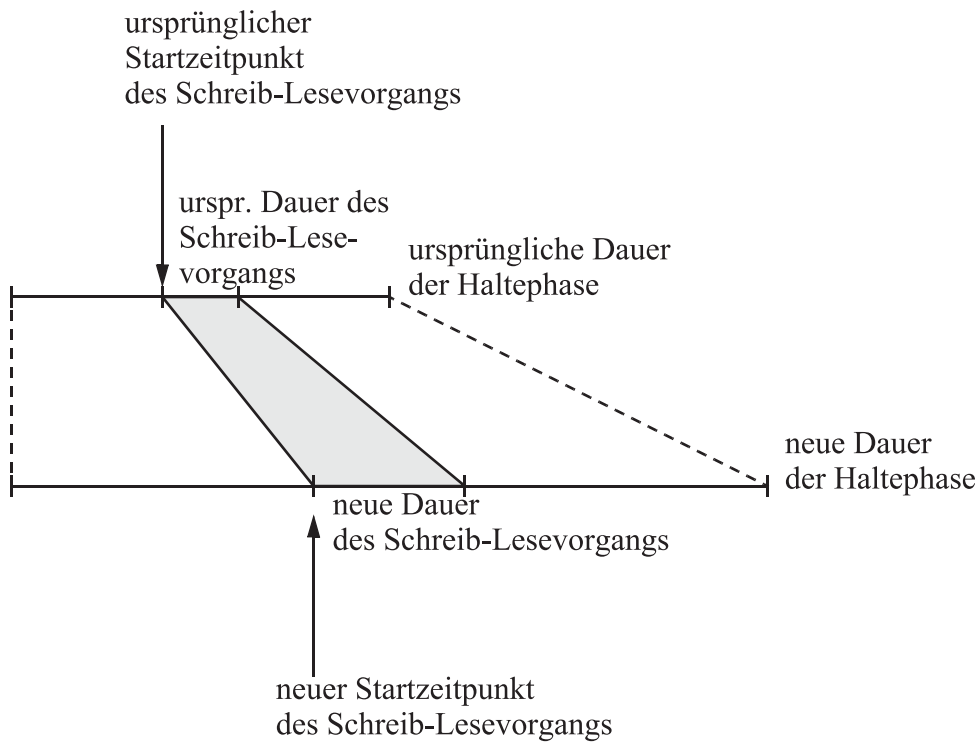


Abbildung 6.14: Zeitliche Streckung einer Haltephase mit einem Schreib-Lesevorgang

Last-variante		$2\mu I_M$ = 3600	$2\mu I_M$ = 7200	$2\mu I_M$ = 10800	$2\mu I_M$ = 14400	$2\mu I_M$ = 18000
E1 <sub>q=1.2</sub>	gesamt	687	2296	4633	7409	11648
E1 <sub>q=1.2</sub>	gTS	499	1569	3022	4621	6809
E1 <sub>q=1.2</sub>	ngTS	188	727	1612	2788	4839
Last-variante		$2\mu I_M$ = 21600	$2\mu I_M$ = 25200	$2\mu I_M$ = 28800	$2\mu I_m$ = 32400	$2\mu I_M$ = 36000
E1 <sub>q=1.2</sub>	gesamt	14568	17927	22412	26563	28244
E1 <sub>q=1.2</sub>	gTS	8111	9619	11375	12572	12877
E1 <sub>q=1.2</sub>	ngTS	6457	8308	11037	13991	15367

Tabelle 6.19: Auftrittshäufigkeit konkurrierender Zugriffe für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor  $q = 1.2$  (Lastvariante E1)

Last-variante		$2\mu I_M$ = 3600	$2\mu I_M$ = 7200	$2\mu I_M$ = 10800	$2\mu I_M$ = 14400	$2\mu I_M$ = 18000
E1 <sub>q=1.2</sub>	gesamt	687	2296	4633	7409	11648
E1 <sub>q=1.2</sub>	$K_-$	614	2043	4353	6509	10491
E1 <sub>q=1.2</sub>	$K_+$	759	2548	4912	8309	12805
Last-variante		$2\mu I_M$ = 21600	$2\mu I_M$ = 25200	$2\mu I_M$ = 28800	$2\mu I_m$ = 32400	$2\mu I_M$ = 36000
E1 <sub>q=1.2</sub>	gesamt	14568	17927	22412	26563	28244
E1 <sub>q=1.2</sub>	$K_-$	13525	15901	20504	25458	25932
E1 <sub>q=1.2</sub>	$K_+$	15610	19954	24319	27667	30555

Tabelle 6.20: 95%-Konfidenzintervalle für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor  $q = 1.2$  (Lastvariante E1)

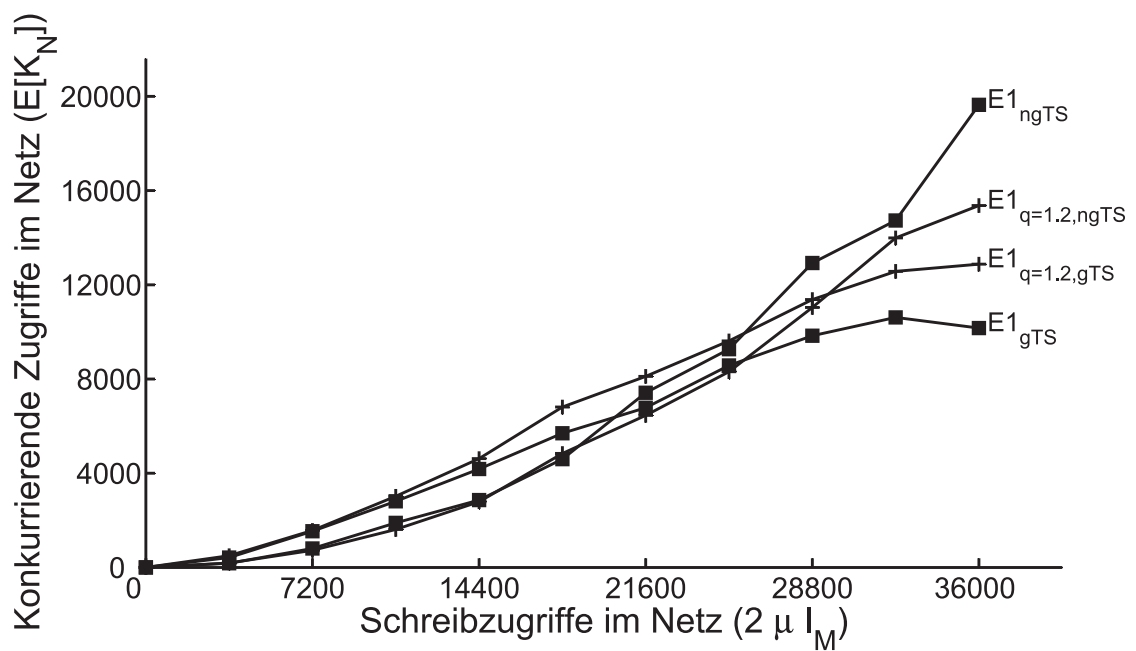


Abbildung 6.15: Ergebnisse für das komplexe Modell eines Vermittlungssystems mit zeitlichem Streckfaktor — Lastvariante E1 versus Lastvariante E1 mit Streckfaktor  $q = 1.2$

## 6.4 Vergleich der Ergebnisse

In Abschnitt 6.2 wurden Auswertungen durch das analytische Modell und in Abschnitt 6.3 durch das Simulationsmodell durchgeführt. In diesem Abschnitt werden die Ergebnisse der Auswertungen verglichen, und es wird untersucht, unter welchen zusätzlichen Voraussetzungen die Ergebnisse des analytischen Modells und die Ergebnisse des Simulationsmodells einander entsprechen. Dieser Vergleich wird hinsichtlich der Art der Knoten und der Anzahl der Verbindungen durchgeführt. Am Ende dieses Abschnitts wird der Einfluss des Betriebssystems betrachtet.

### 6.4.1 Vergleich der Ergebnisse bei unterschiedlicher Art der Knoten und unterschiedlicher Anzahl der Verbindungen

Für jede Lastvariante eines einfachen Modells eines Übertragungssystems wurden pro Schreibrate jeweils 20 Simulationen durchgeführt und in die folgende Abbildung 6.16 gemeinsam mit den Ergebnissen des analytischen Modells eingetragen. Wie an der Abbildung ersichtlich ist, wurde durch diese relativ kleine Anzahl von Simulationen ein Ergebnis erlangt, welches eine gute Übereinstimmung zum analytischen Modell bietet. Der nichtlineare Verlauf und der deutliche Unterschied in Abhängigkeit der Anzahl der Verbindungen (Lastvariante B2 versus B5) wird durch die Ergebnisse des Simulationsmodells bestätigt.

Abbildung 6.17 stellt die Ergebnisse des analytischen Modells und des Simulationsmodells für die Lastvariante D1 eines Übertragungssystems dar. Durch die höhere Anzahl der Schreib-Lesevorgänge und die höhere Anzahl konkurrierender Zugriffe gibt es mehr Stichproben, die Ergebnisse sind weniger gestreut und führen bei gleicher Anzahl Simulationen zu einer besseren Übereinstimmung als das einfache Modell.

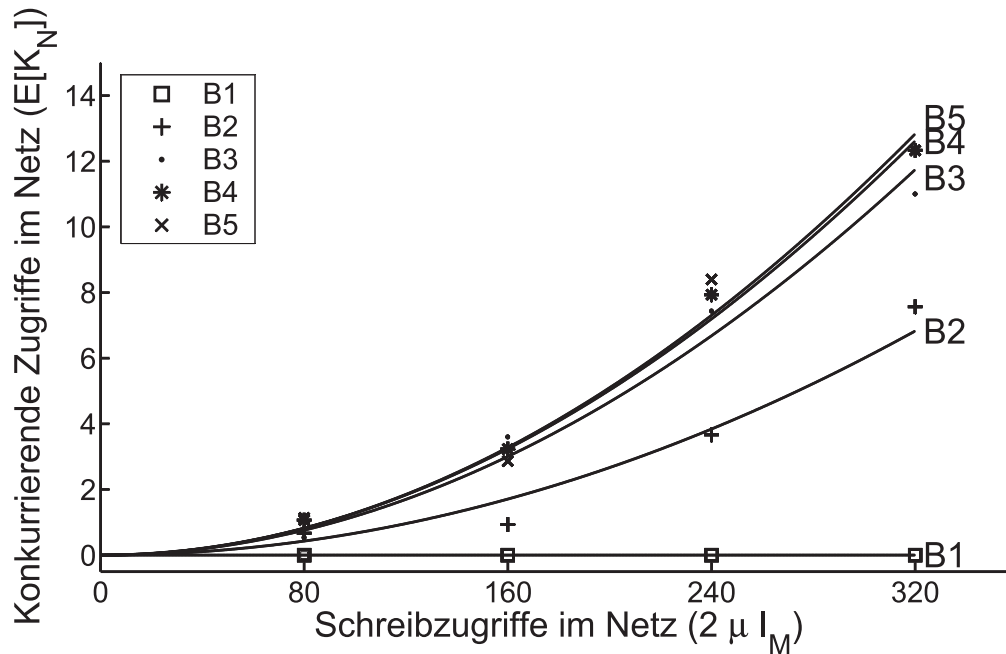


Abbildung 6.16: Vergleich der Ergebnisse für das einfache Modell eines Übertragungssystems — Lastvarianten B1 – B5

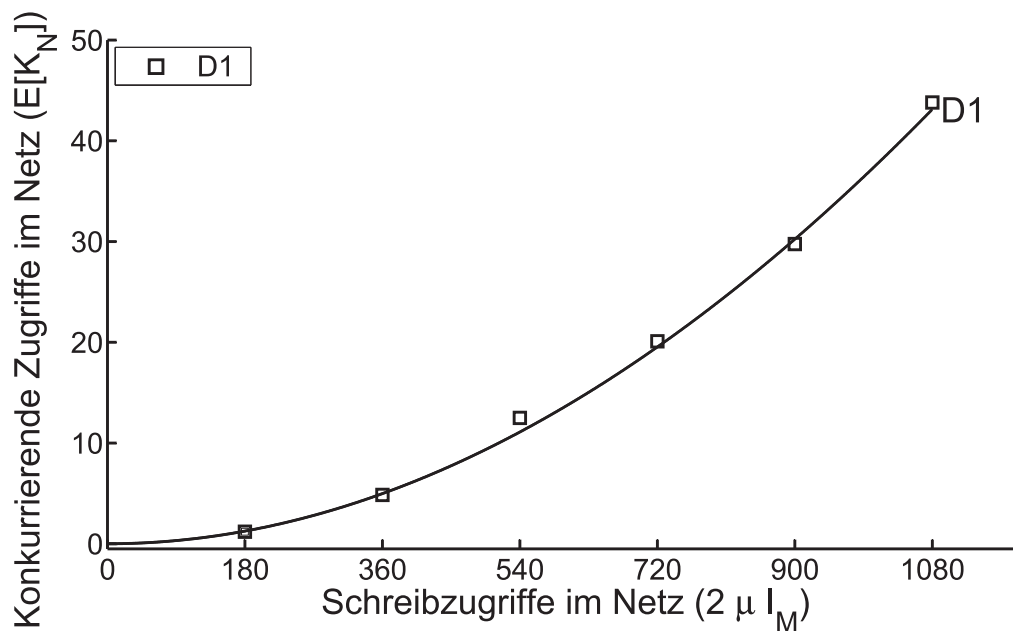


Abbildung 6.17: Vergleich der Ergebnisse für das komplexe Modell eines Übertragungssystems — Lastvariante D1

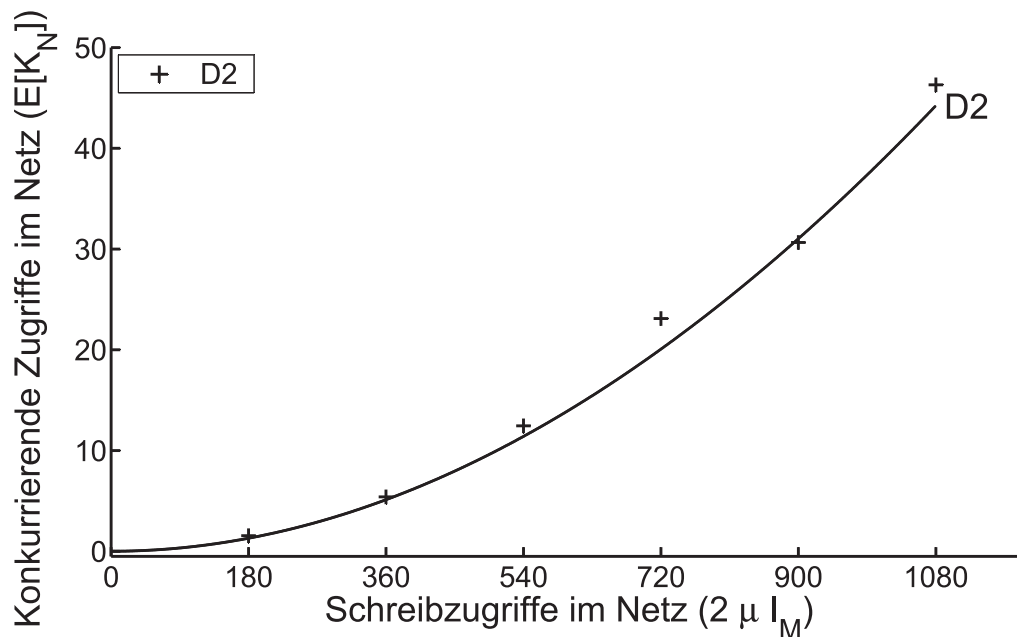


Abbildung 6.18: Vergleich der Ergebnisse für das komplexe Modell eines Übertragungssystems — Lastvariante D2

Ein analoges Ergebnis wurde für die Lastvariante D2 eines komplexen Übertragungssystems erhalten (vgl. Abbildung 6.18).

Werden die Ergebnisse für die Lastvarianten A1 bis A5 eines einfachen Vermittlungssystems (vgl. Abbildung 6.19) bzw. C1 und E1 eines komplexen Vermittlungssystems (vgl. Abbildung 6.20 und 6.21) betrachtet, so ist eine deutliche Abweichung zwischen den Ergebnissen des analytischen Modells und des Simulationsmodells sichtbar.

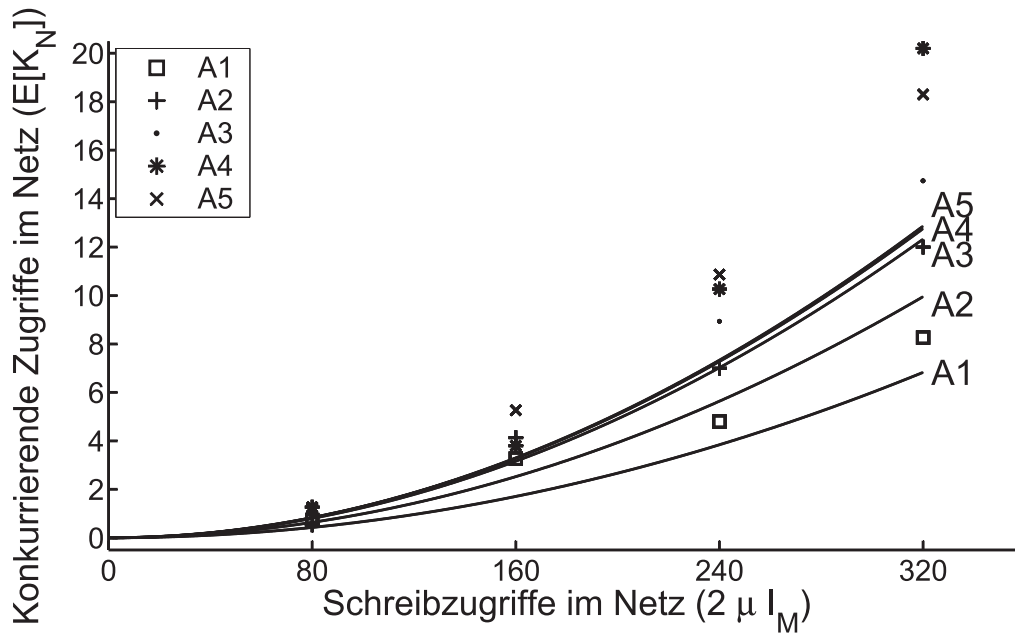


Abbildung 6.19: Vergleich der Ergebnisse für das einfache Modell eines Vermittlungssystems — Lastvarianten A1 bis A5

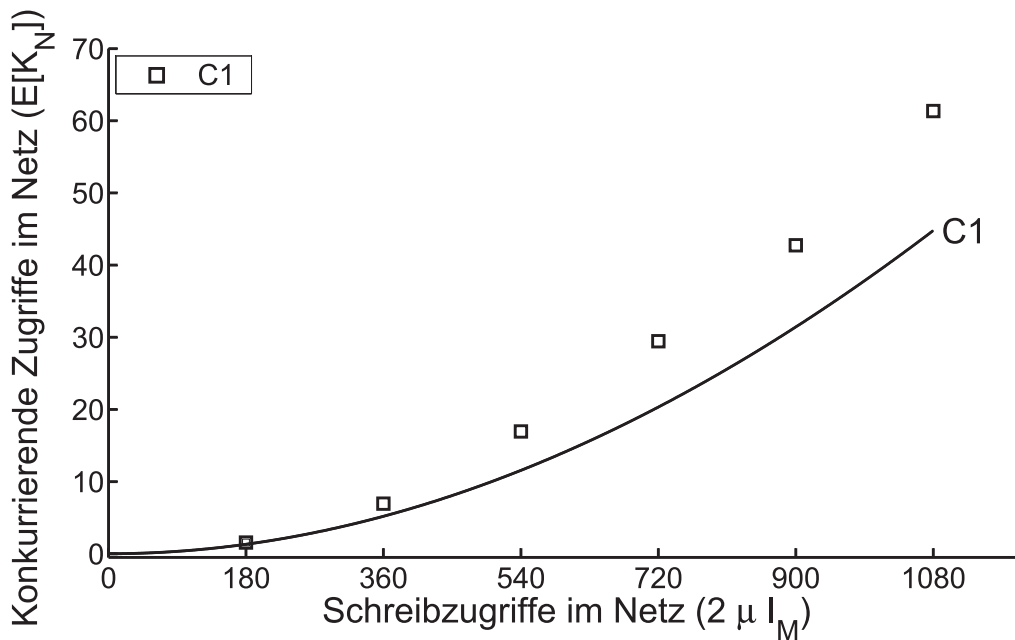


Abbildung 6.20: Vergleich der Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante C1

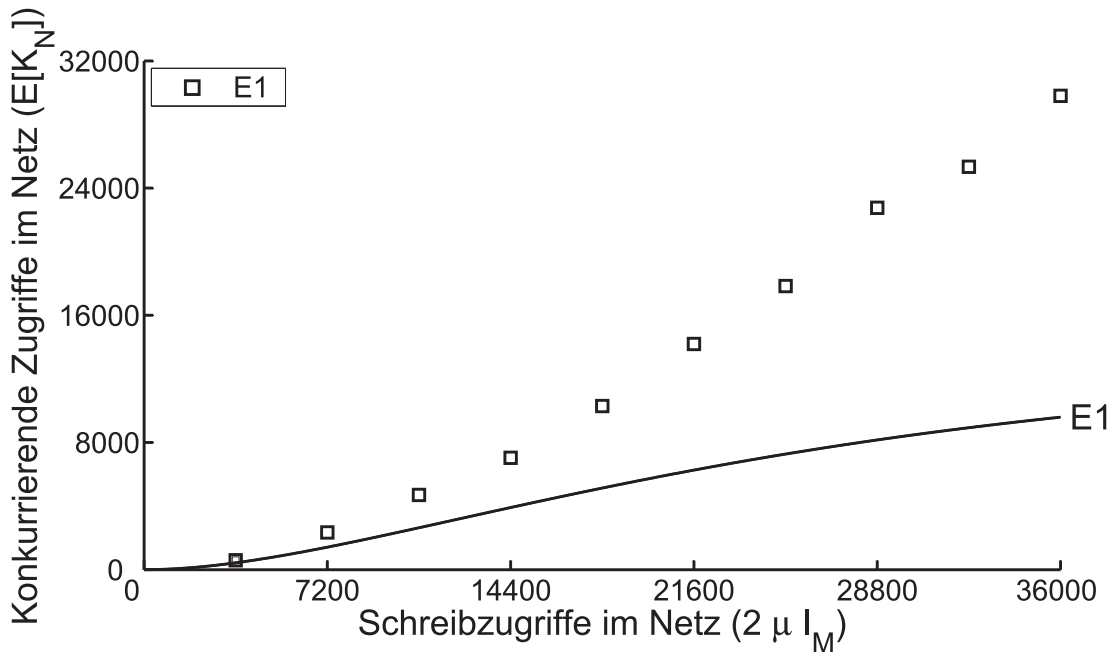


Abbildung 6.21: Vergleich der Ergebnisse für das komplexe Modell eines Vermittlungssystems — Lastvariante E1

Während die Ergebnisse für die Lastvariante A1 und A2 noch eine gewisse Übereinstimmung zeigen, so ist die Abweichung umso größer, je mehr Verbindungen im Systemaufbau existieren und je mehr Schreib-Lesevorgänge durchgeführt werden. Die Verbindungen werden bei Systemen der Art Vermittlungssystem während der Untersuchung aufgebaut und abgebaut. Der Aufbau erfolgt vom A-Knoten zum Z-Knoten, der Abbau erfolgt vom Z-Knoten zum A-Knoten. Wie in Abbildung 5.9 ersichtlich ist, ist der Prozessor während des Verbindungsaufbaus und -abbaus durch verschiedene Threads belegt. Ist in diesem Timeslot ein Schreib-Lesevorgang geplant, so kann dieser nicht in diesem Timeslot stattfinden, falls der Prozessor durch den Thread des Lastgenerators, durch den Thread einer anderen Verbindung oder durch den Thread eines anderen Knoten belegt ist. Dies bedeutet, dass die Anzahl der „freien Timeslots“ geringer ist als bei einem System der Art Übertragungstechnik. Da diese Verbindungsaufbauvorgänge und -abbauvorgänge jedoch nur einen kleinen Teil des Timeslots benötigen, kann es sein, dass mehrere Nachrichten verschiedener Verbindungsaufbauvorgänge bzw. -abbauvorgänge in einem Timeslot bearbeitet werden.

In Abbildung 6.22 wurde nun für das analytische Modell der Lastvarianten A1 bis A3 die folgende modifizierte Formel für die Berechnung von  $q_{f_{n,m}}$  verwendet.

$$q_{f_{n,m}} = \frac{p \cdot p_n \cdot p_{f_{n,m}}}{t - c \cdot I}$$

Es ist  $c$  eine Konstante. Es wurde davon ausgegangen, dass jede Belegung eines Threads des Verbindungsaufbaus und des Verbindungsabbaus einen separaten Timeslot belegt und folglich weniger freie Timeslots zur Verfügung stehen. Wie



in der Abbildung 5.9 zu sehen ist, kommt es bei der betrachteten Modellierung pro Verbindungsaufbau und Verbindungsabbau zu zwölf verschiedenen Prozessorbelegungen (zweimalige Belegung durch den Lastgenerator, viermalige Belegung durch den Thread der Verbindung, je dreimalige Belegung durch den Knoten A und durch den Knoten Z der Verbindung). Folglich wird  $c = 12$  gewählt. Die Ergebnisse des analytischen Modells und des Simulationsmodells stimmen bei der Verwendung dieser Formel besser überein.

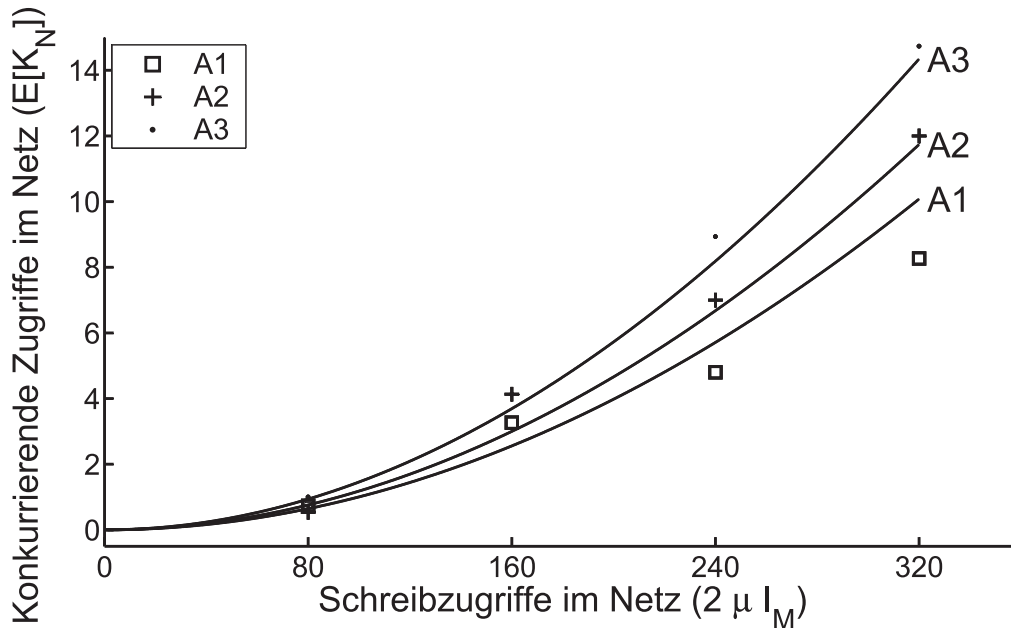


Abbildung 6.22: Ergebnisse des einfachen Modells eines Vermittlungssystems für die Lastvarianten A1 bis A3 mit Einfluss der Threadverdrängung durch den Verbindungsaufbau und -abbau

Das Ergebnis des analytischen Modells mit Einfluss der Threadverdrängung für die Lastvariante C1 eines komplexen Vermittlungssystems ist vergleichbar mit den Ergebnissen des Simulationsmodells und in Abbildung 6.23 dargestellt. Für die Lastvariante C2 ist das Ergebnis analog.

Bei einem Vermittlungssystem mit einer sehr hohen Rate von Schreib-Lesevorgängen kommt es aufgrund der hohen Rate von Schreib-Lesevorgängen zu einer starken Threadverdrängung. Werden sehr viele Verbindungen auf- und abgebaut, so ist die Berechnung durch das modifizierte Modell nicht so genau, da mehrere Nachrichten verschiedener Verbindungsaufbauvorgänge bzw. -abbauvorgänge in einem Timeslot bearbeitet werden können und dies nicht so gut approximiert werden kann. Auch mit der modifizierten Formel ist es bei einer sehr hohen Rate von Schreib-Lesevorgängen nicht möglich, die Ergebnisse des Simulationsmodells zu approximieren.

Aussagen zur Übertragbarkeit der Ergebnisse des Simulationsmodells bei einer

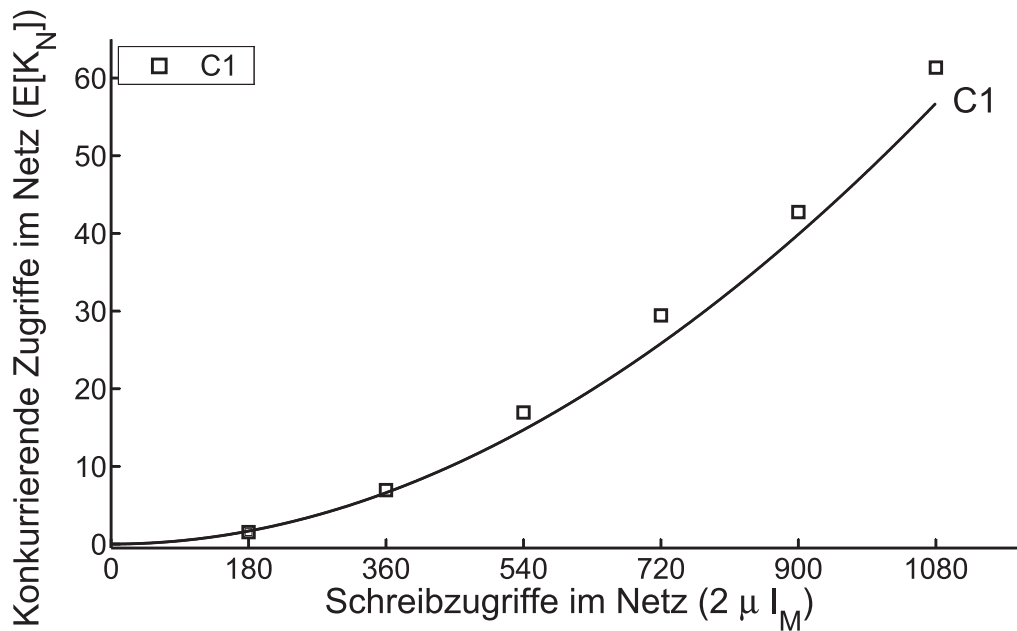


Abbildung 6.23: Ergebnisse des komplexen Modells eines Vermittlungssystems für die Lastvariante C1 mit Einfluss der Threadverdrängung durch den Verbindungsaufbau und -abbau

sehr hohen Rate von Schreib-Lesevorgängen oder bei einem System mit sehr vielen Verbindungsaufbauten auf das reale System sind folglich schwierig. Bei vielen Verbindungen ist es bei der vorliegenden Modellierung möglich, dass in einem Timeslot die Verbindungsaufbauten und Verbindungsabbaute verschiedener Verbindungen erfolgt. Aus diesem Grund wird die modifizierte Formel nicht noch genauer untersucht.

Es ist auch zu beachten, dass es sich bei realen Kommunikationssystemen um Systeme mit mindestens einem Prozessor je Knoten handelt, während die Modellierung auf einen Monoprozessor basiert. Es hat sich jedoch gezeigt, dass die Auftrittshäufigkeit in diesen Fällen beim Simulationsmodell stets höher als beim analytischen Modell ist.

### 6.4.2 Betrachtung des Einflusses der Betriebssystemeigenschaften

In Abschnitt 6.2.3 wurde die Auftrittshäufigkeit konkurrierender Zugriffe unter der Annahme untersucht, dass sich die Dauer der Timeslots bei einer gewissen Systemlast verändert. Unter dieser Annahme ist es möglich, dass sich die Auftrittshäufigkeit verringert, obwohl die Rate der Schreib-Lesevorgängen erhöht wird. Derartige Untersuchungen sind jedoch nicht ohne einen Eingriff in das Betriebssystem möglich und wurden in dieser Untersuchung nicht durchgeführt.

Basierend auf den Erkenntnissen von Abschnitt 6.2.3 wird deutlich, dass der Scheduler des Betriebssystems großen Einfluss auf die Ergebnisse hat. Wird durch den Scheduler aufgrund einer Erhöhung der Schreibrate die Dauer des Quantums geändert, so kann dies zu einer niedrigeren Auftrittshäufigkeit konkurrierender Zugriffe als bei niedrigerer Schreibrate führen. Dies kann zur Folge haben, dass manche Softwarefehler bei einer bestimmten Last gehäuft auftreten, aufgrund einer Änderung der Dauer des Quantums bei einer höheren Last jedoch nicht mehr so häufig auftreten.

Im Rahmen von Abschnitt 6.2 wurde untersucht, wie sich das Verhältnis zwischen konkurrierenden Zugriffen im geplanten Timeslot versus nicht geplanten Timeslot ändert. Bei Anwendung eines zeitlichen Streckfaktors erhöht sich die mögliche Last durch Schreibzugriffe bis zu einer gleichhohen Anzahl konkurrierender Zugriffe im Vergleich zur Untersuchung ohne zeitlichen Streckfaktor leicht. Dies lässt die Annahme zu, dass durch die verringerte Last bei der Untersuchung mit zeitlichem Streckfaktor es zu weniger Verdrängungsvorgängen kommt.

Basierend auf den Erkenntnissen durch Systeme der Art Vermittlungstechnik wird deutlich, dass sich durch weitere Aufgaben des Schedulers (mit der gleichen Prozesspriorität) aufgrund der Threadverdrängung die Auftrittshäufigkeit konkurrierender Zugriffe erhöht. Auch häufige Prozessorswitches benötigen Ressourcen und haben einen Einfluss auf die Auftrittshäufigkeit konkurrierender Zugriffe. Alle Untersuchungen wurden unter der Bedingung durchgeführt, dass alle Threads die gleiche Prozesspriorität besitzen und folglich mit der gleichen Prozesspriorität abgearbeitet werden. Würden die Schreib-Lesevorgänge eine höhere Prozesspriorität als der Verbindungsaufbau und -abbau haben, dann wäre die Auftrittshäufigkeit konkurrierender Zugriffe geringer.

Würden die Schreib-Lesevorgänge eine niedrigere Prozesspriorität als der Verbindungsaufbau und -abbau haben, dann wäre die Auftrittshäufigkeit konkurrierender Zugriffe höher, da es - aus Sicht der Schreib-Lesevorgänge - zu einer massiven Threadverdrängung aufgrund der Verbindungsaufbau und -abbauvorgänge kommen würde.

### 6.4.3 Zusammenfassung

#### 6.4.3.1 Vergleich der Auswertungsmethoden

In Tabelle 6.21 werden die Auswertungsmethoden durch ein analytisches Modell und ein Simulationsmodell verglichen. Beide Auswertungsmethoden haben Vorteile und auch Grenzen. D.h. es ist nicht möglich, eine generelle Aussage zu machen, welche Methode die Bessere ist. Manche Erkenntnisse können durch das analytische Modell, jedoch nicht - ohne größere Eingriffe in das Betriebssystem - durch das Simulationsmodell erhalten werden (z.B. Änderung der Dauer der Timeslots - siehe Abschnitt 6.2.3). Andere Erkenntnisse hingegen können durch das Simulationsmodell, jedoch nicht durch das analytische Modell erhalten werden (z.B. Unterscheidung zwischen Schreib-Lesevorgängen im geplanten versus nicht geplanten Timeslot - siehe Abschnitt 6.3.3). Wie bereits an diesen beiden Beispielen zu sehen ist, hat das Betriebssystem einen wesentlichen Einfluss auf die Ergebnisse.

	<b>Vorteile</b>	<b>Grenzen</b>
Auswertung durch analytisches Modell	<ul style="list-style-type: none"> <li>• Meist analytische Bestimmbarkeit</li> <li>• Ist Modell vorhanden, so sind Ergebnisse mit neuen Parametern leicht erhältlich</li> <li>• Keine Einschränkung bzgl. Dimensionierung der Problemstellung (z.B. Anzahl Knoten)</li> </ul>	<ul style="list-style-type: none"> <li>• Stochastisch</li> <li>• Zusätzliche Eigenschaften führen zu Modelländerungen, welche hohen Aufwand erfordern</li> <li>• Betriebssystem wird nicht modelliert</li> <li>• Prozessorauslastung wird nicht modelliert</li> </ul>
Auswertung durch Simulationsmodell	<ul style="list-style-type: none"> <li>• Zusätzliche Eigenschaften führen zu Modelländerungen → Erste Ergebnisse sind schnell erhältlich</li> <li>• Abhängigkeit vom Betriebssystem des Simulators</li> <li>• Das Verhalten bei hoher Prozessorauslastung ist sichtbar</li> <li>• Prozesseinfluss ähnlich zu Verhalten an leitungsvermittelndem System</li> </ul>	<ul style="list-style-type: none"> <li>• Zuverlässige Aussagen erst nach hinreichend vielen Simulationsläufen möglich</li> <li>• Abhängigkeit vom Betriebssystem des Simulators</li> <li>• Streuung der Ergebnisse</li> <li>• Hoher Zeitaufwand für Simulationsläufe</li> <li>• Einschränkung bzgl. der Dimensionierung der Problemstellung durch den Simulationsrechner</li> </ul>

Tabelle 6.21: Vorteile und Grenzen der Auswertungsmethoden

In der Tabelle 6.22 werden Vergleiche hinsichtlich verschiedener Aspekte durchgeführt und die Erkenntnisse der vorangegangenen Untersuchungen zusammengefasst.

Es ist zu beachten, dass von einem Betriebssystem mit konstanter Dauer der Timeslots ausgegangen wurde. Alle Aussagen folgen direkt aus der Berechnung der Auftrittshäufigkeit konkurrierender Zugriffe in Kapitel 4.

Aspekt	Einfluss
Art des Systems	<ul style="list-style-type: none"> <li>• Bei Vermittlungssystemen existieren zeitweise mehr Verbindungen als bei Übertragungssystemen. Deshalb ist die Auftrittshäufigkeit konkurrierender Zugriffe bei einem Vermittlungssystem stets höher als bei einem Übertragungssystem mit gleicher Rate von Schreib-Lesevorgängen.</li> <li>• Bei einem Vermittlungssystem kann es bei durchschnittlich einer Verbindung zu konkurrierenden Zugriffen kommen. Bei einem Übertragungssystem mit genau einer Verbindung hingegen nicht.</li> <li>• Durch die Auslastung des Prozessors beim Verbindungsaufbau und -abbau ist die Auftrittshäufigkeit konkurrierender Zugriffe bei einem Vermittlungssystem höher als bei einem Übertragungssystem.</li> </ul>
Anzahl der Knoten	<ul style="list-style-type: none"> <li>• Die Auftrittshäufigkeit konkurrierender Zugriffe wird nicht durch die Anzahl der Knoten bestimmt. Die Rate der Schreib-Lesevorgänge in den verschiedenen Speicherfeldern der jeweiligen Knoten und die Auslastung des Prozessors ist vielmehr entscheidend. Deshalb sind auch Netze mit wenigen Knoten für die Erkennung konkurrierender Zugriffe ausreichend.</li> <li>• Bei fester Rate der Schreib-Lesevorgänge im Netz sind wenige Knoten besser als viele Knoten.</li> </ul>
Wahrscheinlichkeit der Routen	<ul style="list-style-type: none"> <li>• Durch die Wahrscheinlichkeit der Routen wird die Anzahl der Schreib- und Lesezugriffe in den einzelnen Knoten und folglich in den einzelnen Speicherfeldern in Abhängigkeit der Schreibrate bestimmt.</li> <li>• Die Rate von Schreib- und Lesezugriffen in den einzelnen Knoten und in den einzelnen Speicherfeldern kann aus der Wahrscheinlichkeit der Routen und der Schreibrate im Netz berechnet werden.</li> </ul>
Wahrscheinlichkeit des Speicherfelds	<ul style="list-style-type: none"> <li>• Ist die Wahrscheinlichkeit der Auswahl eines Knotens fest, so bestimmt die Wahrscheinlichkeit der Auswahl eines Speicherfelds die Auftrittshäufigkeit konkurrierender Zugriffe.</li> </ul>

Tabelle 6.22: Einfluss verschiedener Aspekte auf die Auftrittshäufigkeit konkurrierender Zugriffe

Aspekt	Vergleich
Anzahl der Speicherfelder	<ul style="list-style-type: none"> <li>• Gibt es nur wenige Speicherfelder und werden diese gleichverteilt ausgewählt, so ist die Auftrittshäufigkeit höher als bei vielen Speicherfeldern, welche auch gleichverteilt ausgewählt werden.</li> <li>• Gibt es eine feste Anzahl von Speicherfeldern und werden diese gleichverteilt ausgewählt, so ist die Auftrittshäufigkeit niedriger als bei der gleichen Anzahl von Speicherfeldern, welche nicht gleichverteilt ausgewählt werden.</li> </ul>
Anzahl der Verbindungen	<ul style="list-style-type: none"> <li>• Existieren bei einer Problemstellung bereits viele Verbindungen, so ist die absolute Steigerung der Auftrittshäufigkeit konkurrierender Zugriffe durch eine weitere Verbindung geringer, als bei einer Problemstellung mit wenigen Verbindungen.</li> </ul>
Anzahl der Schreib-Lesevorgänge	<ul style="list-style-type: none"> <li>• Bei fester Anzahl Schreib-Lesevorgänge ist die Auftrittshäufigkeit konkurrierender Zugriffe bei vielen Verbindungen mit kurzer Haltephase höher als bei wenigen Verbindungen mit langer Haltephase (<math>I_M \cdot H_M</math> konstant). Ebenso ist sie bei vielen Verbindungen mit niedriger Rate von Schreib-Lesevorgängen höher als bei wenigen Verbindungen mit hoher Rate von Schreib-Lesevorgängen (<math>I_M \cdot \mu</math> konstant).</li> <li>• Bei fester Anzahl Schreib-Lesevorgänge <math>w</math> liegt das Maximum der Auftrittshäufigkeit konkurrierender Zugriffe bei <math>w</math> Verbindungen mit jeweils 1 Schreib-Lesevorgang.</li> </ul>
Dauer der Schreib-Lesevorgänge	<ul style="list-style-type: none"> <li>• Die Auftrittshäufigkeit konkurrierender Zugriffe verdoppelt sich nahezu bei Verdopplung der Dauer der Schreib-Lesevorgänge.</li> </ul>
Art des Betriebssystems	<ul style="list-style-type: none"> <li>• Die Threadingmechanismen des Betriebssystems haben einen grossen Einfluss auf die Auftrittshäufigkeit konkurrierender Zugriffe. Je besser das Betriebssystem des Simulationsmodell mit dem Betriebssystem des Kommunikationssystems übereinstimmt, desto besser können aussagekräftige Ergebnisse für das leitungsvermittelnde Kommunikationssystem erhalten werden.</li> </ul>

Tabelle 6.22: Einfluss verschiedener Aspekte auf die Auftrittshäufigkeit konkurrierender Zugriffe

In der Tabelle 6.23 werden Vergleiche bzgl. der Variation der Raten durchgeführt und die Erkenntnisse der vorangegangenen Untersuchungen zusammengefasst.

Aspekt	Vergleich
Steigende Rate von Schreib-Lesevorgängen $\mu$	<ul style="list-style-type: none"> <li>• Bei steigender Rate von Schreib-Lesevorgängen steigt die Auftrittshäufigkeit konkurrierender Zugriffe überproportional. Es ist effektiver, mit variabler Rate von Schreib-Lesevorgängen mit dem Mittelwert <math>\mu'</math> zu testen als mit konstanter Rate von Schreib-Lesevorgängen <math>\mu'</math>. Dies ist effektiver, da die Auftrittshäufigkeit konkurrierender Zugriffe zwar bei niedrigerer Rate sinkt, bei höherer Rate jedoch überproportional steigt.</li> </ul>
Steigende Rate von Verbindungsanforderungen $\lambda$	<ul style="list-style-type: none"> <li>• Bei steigender Rate von Verbindungsanforderungen steigt die Anzahl der Verbindungen und folglich auch die Rate der Schreib-Lesevorgänge proportional. Die Auftrittshäufigkeit konkurrierender Zugriffe steigt somit überproportional.</li> </ul>

Tabelle 6.23: Vergleich der Ergebnisse bzgl. Variation der Raten

# Kapitel 7

## Leitlinien zur Durchführung von Lasttests

Alexander Graham Bell *"Leave the beaten track occasionally and dive into the woods. Every time you do so you will be certain to find something you have never seen before."*

In diesem Kapitel werden Leitlinien gegeben, wie Lasttests in der Praxis durchgeführt werden sollen. Wie in den vorherigen Kapiteln dargestellt wurde, erhöht sich die Auftrittshäufigkeit bei geeigneter Wahl der Parameter und bei passender Wahl des Modells deutlich. Bei der Durchführung von Tests sollen die Parameter variiert werden.

### 7.1 Eigenschaften der Modellierungen und des leitungsvermittelnden Systems

In der Tabelle 7.1 werden die Eigenschaften der Modellierungen mit dem leitungsvermittelnden System verglichen.



	Wahrscheinlichkeitsmodell	Simulationsmodell	Leitungsvermittelndes System	Relevanz
Anzahl der Knoten	Beliebig groß	Begrenzt durch maximal mögliche Anzahl der Threads	Begrenzt durch Kosten der Systeme	Niedrig
Häufigkeit einer Verbindung	Identische Häufigkeit, Häufigkeit fest vorgegeben	Gemäß probabilistischer Verteilung, unterschiedlich pro Simulationslauf	Variabel	Niedrig
Rate der SLVs	Konstant	Variabel	Variabel	Mittel
Dauer der SLVs	Ideal, beliebig groß	Nur in gewissen Grenzen mit bestimmter Genauigkeit realisierbar	Variabel (abhängig von geschriebenen Daten, Hardware und Betriebssystem)	Hoch
Häufigkeit bzgl. Auswahl eines Speicherfelds	Identische Häufigkeit, Häufigkeit fest vorgegeben	Gemäß probabilistischer Verteilung, unterschiedlich pro Simulationslauf	Variabel	Niedrig
Betriebssystem	Keinen Einfluss	Betriebssystem des Simulators	Betriebssystem des leitungsvermittelnden Systems	Hoch
Prozessor	Monoprozessor mit einem Thread	Monoprozessor mit <ul style="list-style-type: none"> <li>• 1 Thread pro Verbindung</li> <li>• 1 Thread pro Knoten</li> <li>• 1 Thread für Lastgenerator</li> </ul>	Multiprozessor	Mittel bzw. hoch bei hoher Last
Prozessorauslastung	Keinen Einfluss	Überlast des Prozessors führt zur Verschiebung der Schreib-Lesevorgänge	Überlast der Prozessoren kann zu unterschiedlichem Verhalten führen	Hoch

Tabelle 7.1: Vergleich der Modellierungen und des leitungsvermittelnden Systems

## 7.2 Systematische Vorgehensweise bei der Durchführung von Lasttests

Bei begrenzter Zeit zur Durchführung von Lasttests am realen System sollen manche Parameter mit einer höheren Priorität variiert werden als andere Parameter. In der Tabelle 7.2 sind die Parameter und empfohlene Varianzmöglichkeiten dieser bei der Durchführung von Lasttests aufgeführt. Es wird durch den Pfeil „→“ angegeben, durch welche Änderung des Parameters sich die Auftrittshäufigkeit erhöht.

Bei der Durchführung von Tests bevorzugen Tester ein kleines Quantum, da sich durch dieses die Auftrittshäufigkeit konkurrierender Zugriffe erhöht. Netzbetreiber hingegen würden im Hinblick auf konkurrierende Zugriffe ein größeres Quantum bevorzugen. Dies würde aber zugleich zu einer längeren Abarbeitungsdauer z.B. bei Ersatzschaltungen führen. D.h. es würden weniger konkurrierende Zugriffe auftreten, aber zugleich wären die Ersatzschaltungszeiten länger, was wiederum nicht im Interesse der Netzbetreiber ist. Bei der Varianz der Priorität der Schreib-Lesevorgänge ist zu beachten, dass dies einen großen Einfluss auf das Systemverhalten haben kann. Deshalb kann es sein, dass die Durchführung dieser Untersuchung zu keinen relevanten Ergebnissen führt.

Parameter	Varianzmöglichkeiten	Priorität der Durchführung
Anzahl der Verbindungen	Geringe Anzahl → hohe Anzahl Variation der Anzahl der Verbindungen z.B. 1, 2, 4, 8, ...	Hoch
Anzahl der Prozesse	Geringe Anzahl → hohe Anzahl z.B. 1, 2, 4, 8, ...	Hoch
Schreibrate	Geringe Schreibrate → hohe Schreibrate z.B. 10, 20, 40, 80, 160, 320, ...	Hoch
Änderung der Hardwarekonfiguration	Hardwarekonfiguration ändern (wenn die störungsfreie Änderung der Hardwarekonfiguration eine Systemeigenschaft ist) Wenig Änderungen → viele Änderungen	Hoch
Zusatzlast	Geringe Zusatzlast → mittlere Zusatzlast	Mittel

Tabelle 7.2: Potentielle Varianzmöglichkeiten bei der Durchführung von Lasttests zur Steigerung der Auftrittshäufigkeit

Parameter	Varianzmöglichkeiten	Priorität der Durchführung
Speichermedium	Verwendung von Medien mit unterschiedlichen Speicherzugriffszeiten, wenn relevant für System Schnelle Speicherzugriffszeit → langsame Speicherzugriffszeit z.B. Hauptspeicher, Magnetplattenspeicher, Magnetbandspeicher	Mittel
Quantum des Betriebssystems	Wenn änderbar, Quantum für einige Testläufe verringern Großes Quantum → kleines Quantum	Niedrig
Priorität der Schreib-Lesevorgänge	Hohe Priorität → geringe Priorität z.B. hohe Priorität, mittlere Priorität, niedrige Priorität	Niedrig

Tabelle 7.2: Potentielle Varianzmöglichkeiten bei der Durchführung von Lasttests zur Steigerung der Auftrittshäufigkeit

Lasttests werden durchgeführt, um statistisch auftretende konkurrierende Zugriffe zu finden. Da konkurrierende Zugriffe i.A. nur sehr selten auftreten, ist eine gewisse Laufzeit der Tests nötig. Deshalb ist es sinnvoll, Lasttests für mehrere Aspekte zeitgleich durchzuführen.

In der Tabelle 7.3 sind die Optima bzgl. der verschiedenen Parameter zur Erreichung einer möglichst hohen Auftrittshäufigkeit konkurrierender Zugriffen aufgeführt.

Parameter	Optimum	Effizienz
Routenwahrscheinlichkeit	Bei Betriebssystem mit konstanter Dauer der Timeslots: maximieren Sonst: Für jede Route unterschiedlich	Hoch
Art der Knoten	Vermittlungssystem	Hoch
Dauer der Haltephase	Dauer der Haltephase $H_i$ gering	Hoch
Zusatzlast	Zusatzlast mit möglichst häufigem Prozessorwechsel durch andere Systemfunktionen erzeugen	Hoch
Rate der Verbindungsanforderungen	I.A. im Bereich der vorhandenen HW-Ressourcen (Grenzen nur bei einigen Tests überschreiten)	Mittel
Anzahl SLVs pro Verbindung	Ein SLV pro Verbindung	Mittel
Prozessorauslastung	Prozessorauslastung etwas kleiner als die Auslastung, welche zu einer Erhöhung des Quantums führt	Mittel
Zugriff auf Daten	Funktionen mit Zugriffen auf gleichen Daten zeitnah durchführen	Mittel
Anzahl Speicherfelder	Minimal	Mittel
Netzgröße	Minimale Netzgröße ausreichend	Niedrig

Tabelle 7.3: Optimale Parameter bei der Durchführung von Lasttests

In der Tabelle 7.4 ist der Ablauf der Planung von Lasttests beschrieben.

Schritt	Beschreibung	Festgelegte Parameter
1.	Art des Systems auswählen	Vermittlungssystem auswählen
2.	Zur Verfügung stehende HW-Ressourcen bestimmen	Anzahl der Netzknoten und Leitungskapazität bestimmen
3.	Anzahl von Netzaufbauten bestimmen	Anzahl paralleler Netzaufbauten bestimmen
4.	Parameter der Routen festlegen	Routen zwischen Netzknoten und deren Routenwahrscheinlichkeit festlegen
5.	Verteilungen der Verbindungsanforderung bestimmen	Dauer der Haltephase, Anzahl der Speicherfelder und Wahrscheinlichkeit der Auswahl der Speicherfelder bestimmen

Tabelle 7.4: Ablauf der Planung von Lasttests

In der Tabelle 7.5 ist der Ablauf der Testdurchführung beschrieben. Steht nur beschränkte Zeit zur Verfügung, so sollten auf jeden Fall die Variationen mit hoher Durchführungspriorität durchgeführt werden. Bei fehlender Zeit oder fehlenden Ressourcen können die Schritte 2 und 3 eingespart werden.

Schritt	Beschreibung	Varierte Parameter
1.	Lasttests mit variierten Parametern mit hoher Durchführungspriorität durchführen	Anzahl der existierenden Verbindungen, Anzahl der Prozesse, Schreibrate, HW-Konfiguration
2.	Lasttests mit variierten Parametern mit mittlerer Durchführungspriorität durchführen	Speichermedium, Zusatzlast
3.	Lasttests mit variierten Parametern mit niedriger Durchführungspriorität durchführen	Quantum des Betriebssystems, Priorität der Schreib-Lesevorgänge

Tabelle 7.5: Ablauf der Durchführung von Lasttests

Wie durch die Analyse deutlich wurde, steigt die Auftrittshäufigkeit konkurrierender Zugriffe überproportional. Verändert das verwendete Betriebssystem die Dauer des Quants nicht, dann kann die Dauer der Tests mit einer bestimmten Last bei einer erhöhten Last um einen gewissen Faktor sinken, um mindestens die gleiche Auftrittshäufigkeit zu erhalten. D.h. wird z.B. bei einer bestimmten

Schreibrate für eine Dauer von  $n$  Minuten getestet, ist es ausreichend, bei der doppelten Schreibrate nur für eine Dauer von  $\frac{n}{2}$  Minuten zu testen, um mindestens die gleiche Auftrittshäufigkeit zu erhalten. Ist die Rate der Verbindungsanforderungen so hoch, dass die vorhandenen Hardware-Ressourcen nicht ausreichen, um alle Verbindungen zu schalten, werden Verbindungsanforderungen abgewiesen. Dies bedeutet, es wird möglicherweise die gleiche Auftrittshäufigkeit konkurrierender Zugriffe erlangt wie bei niedrigerer – im Bereich der vorhandenen Hardwareressourcen – Rate von Verbindungsanforderungen.

Im Modell wurde die Unabhängigkeit der Schreib-Lesevorzugriffe angenommen. Bei leitungsvermittelnden Kommunikationssystemen können Schreib- und Lesezugriffe verschiedener Prozesse aber voneinander abhängig sein. Dies kann sowohl zu einer erhöhten als auch zu einer verringerten Auftrittshäufigkeit konkurrierender Zugriffe führen. D.h. es ist möglich, dass bei einer bestimmten Last aufgrund der zeitlichen Bedingungen keine konkurrierenden Zugriffe auftreten. Aus diesem Grund sollen Tests stets mit gut variiertes Last durchgeführt werden. Dadurch werden unterschiedliche Zeitbedingungen erzeugt, die auch bei gekoppelten Ereignissen zu konkurrierenden Zugriffen führen können.

### 7.3 Zusammenfassung

In dieser Arbeit wurde durch eine modellgestützte Untersuchung gezeigt, wie konkurrierende Zugriffe bei leitungsvermittelnden Kommunikationssystemen hervorgerufen werden können. Obwohl die Untersuchungen an einem Modell eines Kommunikationssystems durchgeführt wurden, ist der Lasttest nicht auf diese Art von Systemen beschränkt. Er kann ebenso auf andere Kommunikationssysteme, wie z.B. paketorientierte Kommunikationssysteme, übertragen werden. Bei genauerer Betrachtung der elementaren Eigenschaften des untersuchten Systems zeigt sich, dass dieses Verfahren ebenso für andere Mehrprozessorsysteme geeignet ist. Sobald zwei verschiedene Prozesse potentiell zeitgleich auf die gleichen Daten zugreifen können, ist dies ein Anwendungsbereich für Lasttests. Durch die hohen Sicherheitsanforderungen z.B. in der Medizintechnik oder in der Automobiltechnik ist dieses Verfahren ebenso für derartige komplexe Softwaresysteme geeignet. Zeitkritische Echtzeitanwendungen mit hohen Verfügbarkeitsanforderungen sind für die Anwendung von Lasttests geeignet.

Bei der Einführung des Lasttests kann häufig auf die verstärkte Anwendung von automatisierten Tests zurückgegriffen werden. Die parallele Ausführung von Datenzugriffen verschiedener Art vergrößert i.A. die Auftrittshäufigkeit. Gibt es zugleich die Möglichkeit, den oder die Multiprozessor des Systems durch Eingriffe in die Hardware-Konfiguration zu belasten, so erhöht sich die Auftrittshäufigkeit derartiger konkurrierender Zugriffe i.A. weiter. Durch die Nutzung vorhandener Methoden und durch die äußere Stimulation des Systems ist der Mehraufwand bei der Einführung von Lasttests begrenzt, die Auftrittshäufigkeit konkurrierender

Zugriffe steigt jedoch deutlich, was sonst nur durch langdauernde Tests möglich ist. Es ist wesentlich, die Last zu variieren. Es sollten nicht nur Tests mit hoher Last durchgeführt werden.

Die Durchführung des Lasttests am leitungsvermittelnden System ist der Durchführung des Lasttests auf einem Simulationsrechner vorzuziehen. Hat das Betriebssystem des Simulationsrechners andere Threadingmechanismen als das Betriebssystem des realen Systems, so kann sich die Auftrittshäufigkeit konkurrierender Zugriffe deutlich unterscheiden. Beachtet man die hohen Kosten derartiger hochkomplexer Systeme, so wird deutlich, dass in möglichst kurzer Zeit eine hohe Qualität erlangt werden soll. Durch die überproportionale Steigerung der Auftrittshäufigkeit konkurrierender Zugriffe wird die Identifikation und auch Analyse von konkurrierenden Zugriffen deutlich erleichtert. Eine rasche Steigerung der Qualität während des Entwicklungszyklus ist somit möglich.

# Literaturverzeichnis

- [1] Artho C., Havelund K., Biere A.:  
*High-Level Data Races*, In VVEIS'03, The First International Workshop on Verification and Validation of Enterprise Information Systems, Angers France, 2003
- [2] Avritzer A., Larson B.:  
*Load Testing Software using Deterministic State Testing*, Proceedings of the 1993 ACM SIGSOFT international symposium on Software testing and analysis, ACM, New York, 1993, pp. 82 – 88
- [3] Avritzer A., Weyuker E.J.:  
*Generating Test Suites for Software Load Testing*, Proceedings of the 1994 ACM SIGSOFT international symposium on Software testing and analysis, ACM, New York, 1994, pp. 44 – 57
- [4] Balzert H.:  
*Lehrbuch der Software-Technik, Bd. 1: Software-Entwicklung*, Spektrum Akademischer Verlag, Heidelberg, 2000
- [5] Balzert H.:  
*Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*, Spektrum Akademischer Verlag, Heidelberg, 1998
- [6] Banks J., Carson J.S., Nelson B.L.:  
*Discrete-Event System Simulation*, Prentice Hall, Upper Saddle River NJ, 1995
- [7] Berger A.W., Whitt W.:  
*Maximum values in queueing processes*, Probability in the Engineering and Informational Sciences, Vol. 9, 1995, pp. 375 – 409
- [8] Biran A., Breiner M.:  
*MATLAB 5 für Ingenieure: Systematische und praktische Einführung*, Addison Wesley, Bonn, 2000



- [9] Bolch G.:  
*Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*, Teubner, Stuttgart, 1989
- [10] Bolch G., Greiner S., de Meer H., Trivedi K.S.:  
*Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*, John Wiley & Sons, New York, 1998
- [11] McCalla C., Whitt W.:  
*A Time-Dependent Queueing-Network Model to Describe the Life-Cycle Dynamics of Private-Line Telecommunication Services*, Telecommunication Systems, Vol. 19, No. 1, 2002, pp. 9 – 38
- [12] Cohen J.W.:  
*The single server queue*, North-Holland Publishing Company, Amsterdam, 1969
- [13] Cox D.R., Miller H.D.:  
*The Theory of stochastic processes*, Chapman and Hall Ltd, London, 1972
- [14] Crovella M.E., Lipsky L.:  
*Long-lasting Transient Conditions In Simulations With Heavy-tailed Workloads*, Simulation Conference, Proceedings of the 1997 Winter, 1997, pp. 1005 – 1012
- [15] Dahl O.-J., Dijkstra E.W., Hoare C.A.R.:  
*Structured Programming*, Academic Press, London, 1974
- [16] Duffy D., McIntosh A., Rosenstein M., Willinger W.:  
*Analyzing telecommunications traffic data from working common channel signaling subnetworks*, Proceedings of the 25th Interface, San Diego California, 1993, pp. 156 – 165
- [17] *Entwicklungsstandard für IT-Systeme des Bundes, Vorgehensmodell*, Teil 1: Regelungsteil, Teil 3: Handbuchsammlung, Allgemeiner Umdruck Nr. 250/1, BWB IT 15, Koblenz, 1997
- [18] Fischer M.J., Harris C.M.:  
*A Method For Analyzing Congestion in Pareto and Related Queues*, The Telecommunication Review, Vol. 10, Mitretek Systems, McLean, VA, 1999, pp. 15 – 28
- [19] Galli D.L.:  
*Distributed operating systems, concepts and practice*, Prentice Hall, Upper Saddle River NJ, 2000

- [20] Glynn P.W., Whitt W.:  
*Heavy-Traffic Extreme-Value Limits for Queues*, Operations Research Letters, Vol. 18, No. 3, 1995, pp. 107 – 111
- [21] Gopal G., Griffeth N.D.:  
*Software Fault Tolerance in Telecommunications Systems*, Proceedings of the 4th workshop on ACM SIGOPS European workshop, ACM, New York, 1990, pp. 1 – 5
- [22] <http://www.globalcrossing.com/html/map.html>
- [23] Greiner M., Jobmann M., Klüppelberg C.:  
*Telecommunication Traffic, Queueing Models, and Subexponential Distributions*, Queueing Systems Vol. 33, No. 1-3, 1999, pp. 125 – 152
- [24] Haaß W.-D.:  
*Handbuch der Kommunikationsnetze: Einführung in die Grundlagen und Methoden der Kommunikationsnetze*, Springer, Berlin, 1997
- [25] Halang W.A.:  
*Load adaptive dynamic scheduling of tasks with hard deadlines useful for industrial applications*, Computing, Vol. 47, Issue 3-4, Springer, New York, 1991, pp. 199 – 213
- [26] Henderson S., Meyn S., Tadić V.:  
*Performance evaluation and policy selection in multiclass networks*, Discrete Event Dynamic Systems, Vol. 13, Issue 1-2, Kluwer Academic Publishers, Hingham MA, 2003, pp. 149 – 189
- [27] Herrtwich R.G., Hommel G.:  
*Kooperation und Konkurrenz: nebenläufige, verteilte und echtzeitabhängige Programmsysteme*, Springer, Berlin, 1989
- [28] Hjalmtýsson G., Whitt W.:  
*Periodic Load Balancing*, Queueing Systems, Vol. 30, 1998, pp. 203 – 250
- [29] Hofmann F.:  
*Betriebssysteme: Grundkonzepte und Modellvorstellungen*, Teubner, Stuttgart, 1984
- [30] Huang J.-H., Kleinrock L.:  
*Performance evaluation of dynamic sharing of processors in two-stage parallel processing systems*, Transactions on Parallel and Distributed Systems, Vol. 4, Issue 3, 1993, pp. 306 – 317
- [31] Institute of Electrical and Electronics Engineers:  
*Software Engineering IEEE Standards Collection*, IEEE, New York, 1993

- [32] International Telecommunication Union:  
*Key Global Telecom Indicators for the World Telecommunication Service Sector*, www.itu.int
- [33] International Telecommunication Union:  
*World telecommunication indicators*, Geneva, 2001
- [34] Jain R.:  
*The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*, John Wiley & Sons, New York, 1991
- [35] Jonsson J., Lönn H., Shin K.G.:  
*Non-Preemptive Scheduling of Real-Time Threads on Multi-Level-Context Architectures*, Lecture Notes In Computer Science, Vol. 1586, Proceedings of the 11 IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing, Springer, London, 1999, pp. 363 – 374
- [36] Kelly F.P.:  
*Mathematical models of multiservice networks*, Complex Stochastics Systems and Engineering, 1993
- [37] Kelly F.P., Williams R.J.:  
*Stochastic Networks*, Springer-Verlag, New York, 1995
- [38] Kim M.H., Park H.S.:  
*Scheduling self-similar traffic in packet-switching systems with high utilisation*, IEE Proceedings-Communications, Vol. 151, No. 5, 2004, pp. 429 – 437
- [39] Kleinrock L.:  
*Communication nets*, Mc Graw-Hill, New York, 1964
- [40] Kleinrock L.:  
*Queueing systems Volume I: Theory*, John Wiley & Sons, New York, 1975
- [41] Kleinrock L.:  
*Queueing systems Volume II: Computer applications*, John Wiley & Sons, New York, 1976
- [42] Knuth D.E.:  
*The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Addison-Wesley, Reading, Massachusetts, 1998

- [43] Kobayashi H.:  
*Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1978
- [44] Lamport L.:  
*The Mutual Exclusion Problem: Part I - The Theory of Interprocess Communication*, Journal of the Association for Computing Machinery, Vol. 33, No. 2, 1986, pp. 313 – 326
- [45] Lamport L.:  
*The Mutual Exclusion Problem: Part II - Statement and Solutions*, Journal of the Association for Computing Machinery, Vol. 33, No. 2, 1986, pp. 327 – 348
- [46] Law A.M., Kelton W.D.:  
*Simulation modeling and analysis*, Mc Graw Hill, Boston, 2000
- [47] Lea D.:  
*Concurrent programming in Java*, Addison-Wesley-Longman, Bonn, 1997
- [48] Liggesmeyer P.:  
*Software-Qualität: Testen, Analysieren und Verifizieren von Software*, Spektrum Akademischer Verlag, Heidelberg, 2002
- [49] Lipsky L.:  
*Queueing Theory: A Linear Algebraic Approach*, Macmillan Publishing Company, New York, 1992
- [50] Liu M., Baras J.S.:  
*Performance analysis using a hierarchical loss network model*, Global Telecommunications Conference 2000, Vol. 3, 2000, pp. 1793 – 1797
- [51] Martorell X., Labarta J., Navarro N., Ayguade E.:  
*Analysis of Several Scheduling Algorithms under the Nano-Threads Programming Model*, Proceedings of 11th IEEE International Parallel Processing Symposium, 1997, pp. 281 – 287
- [52] Mason C., Bushaus D.:  
*Software problem cripples AT&T long-distance network*, Telephony, Vol. 218/4, January 22, 1990, pp. 10 – 11
- [53] Mason C., Karpinsky R.:  
*SS7 errors torpedo networks in Washington, Los Angeles*, Telephony, 1991

- [54] Massalin H., Pu C.:  
*Fine-Grain Adaptive Scheduling using Feedback*, Computing Systems, Vol. 3, No. 1, 1989, pp. 139 – 173
- [55] Netzer R.H.B., Miller B.P.:  
*What are Race Conditions?, Some Issues and Formalizations*, ACM Letters on Programming Languages and Systems, Vol. 1, No. 1, 1992, pp. 74 – 88
- [56] Neumann P.G.:  
*Computer-related risks*, Addison-Wesley Publishing Company, ACM Press, New York, 1995
- [57] Oechsle R.:  
*Parallele Programmierung mit Java Threads*, Carl Hanser Verlag, München, 2001
- [58] Probst S.:  
*Betriebssysteme*, Unterlagen zur Lehrveranstaltung, Band 1, Fachhochschule Hagenberg
- [59] Resnick S.I.:  
*Heavy tail modeling and teletraffic data*, In: S. Resnick: Heavy tail modeling and teletraffic data, Annals of Statistics 25, 1997, pp. 1805 – 1869
- [60] Scratchley, W.C.:  
*Evaluation and Diagnosis of Concurrency Architectures*, Ph.D. thesis, Carleton University, Canada, 2000
- [61] Siegmund G.:  
*Technik der Netze*, Hüthig, Heidelberg, 1999
- [62] Singhal M., Shivaratri N.G.:  
*Advanced concepts in operating systems, distributed, database, and multiprocessor operating systems*, McGraw-Hill, New York, 1994
- [63] Sinha P.K.:  
*Distributed operating systems, concepts and design*, IEEE Press, Piscataway NJ, 1997
- [64] Sloman M., Kramer J.:  
*Distributed systems and computer networks*, Prentice-Hall International, London, 1987
- [65] Smith C.U., Williams L.G.:  
*Performance solutions: a practical guide to creating responsive, scalable software*, Addison-Wesley, Boston, 2001

- [66] Sommerville I.:  
*Software Engineering*, Pearson Studium, München, 2001
- [67] Stallings W.:  
*Betriebssysteme*, Person Studium, München, 2003
- [68] Stuck B.W., Arthurs E.:  
*A computer and communications network performance analysis primer*,  
Prentice-Hall Inc., Englewood Cliffs, 1984
- [69] Takács L.:  
*Introduction to the Theory of Queues*, Oxford University Press, New York,  
1962
- [70] Takagi H.(editor):  
*Stochastic analysis of computer and communication systems*, North-Holland,  
Amsterdam, 1990
- [71] Tanenbaum A.S.:  
*Computer Networks*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981
- [72] Tanenbaum A.S.:  
*Moderne Betriebssysteme*, Pearson Studium, München, 2002
- [73] Tanenbaum A.S., Van Steen M.:  
*Distributed Systems Principles and Paradigms*, Prentice-Hall, Upper Saddle  
River, New Jersey, 2002
- [74] Thaller G.E.:  
*Software-Test: Verifikation und Validation*, Heise, Hannover, 2002
- [75] Travis P.:  
*Why the AT&T network crashed*, Telephony, Vol. 218/4, January 22, 1990,  
pp. 11
- [76] Voas J.M., McGraw G.:  
*Software fault injection*, John Wiley & Sons, New York, 1998
- [77] Wegener J.:  
*Evolutionärer Test des Zeitverhaltens von Realzeit-Systemen*, Berichte aus  
der Softwaretechnik, Aachen, Shaker, 2001
- [78] Whitt W.:  
*Heavy-Traffic Limits for Loss Proportions in Single-Server Queues*, Queu-  
ing Systems, Vol. 46, 2004, pp. 507 – 536

- [79] Whitt W.:  
*Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and their Application to Queues*, Springer Series in Operations Research, Springer, Berlin, 2002
- [80] Willinger W., Taqqu M., Erramilli A.:  
*A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks*, Stochastic Networks: Theory and Applications, F.P. Kelly, S. Zachary, I.Ziedins (ed.), Oxford University Press, Oxford, 1996, pp. 339 – 366
- [81] Yang C.D., Pollock L.L.:  
*The challenges in automated testing of multithreaded programs*, 14th International Conference on Testing Computer Software, 1997, pp 157 – 166
- [82] Yu P.S., Dias D.M., Lavenberg S.S.:  
*On the Analytical Modeling of Database Concurrency Control*, Journal of the Association for Computing Machinery, Vol. 40, No. 4, 1993, pp. 831 – 872
- [83] Zöbel D., Hogenkamp H.:  
*Konzepte der parallelen Programmierung*, Teubner, Stuttgart, 1988
- [84] Zukerman M.:  
*Bandwidth Allocation for Bursty Isochronous Traffic in a Hybrid Switching System*, IEEE Transactions of Communications, Vol. 37, No. 12, 1989, pp. 1367 – 1371
- [85] Zuser W., Grechenig T., Köhle M.:  
*Software Engineering mit UML und dem Unified Process*, Pearson Studium, München, 2002





