

Um ein Bürodokument zwischen zwei Systemen zu übertragen, muß der Baum, der es beschreibt z. B. nach dem Verfahren 'preorder traversal' sequenzialisiert werden. Der resultierende Datenstrom besteht dann aus Deskriptoren über das Dokument und Stücken seines Inhalts.

## 7 Ausblick

Die gesamte nationale und internationale Szene läßt die ziemlich sichere Vermutung zu, daß am Ende dieses Jahrzehnts eine breite Palette unterschiedlicher Produkte auf dem Markt sein wird, die ISO-Normen entspricht.

### Literatur

- [1] ISO, Information Processing Systems – Open Systems Interconnection Basic Reference Model. IS 7498.
- [2] ISO, Information Processing Systems – Open Systems Interconnection Presentation Service Definition and Protocol Specification. DP 8822, DP 8823.
- [3] ISO, Information Processing Systems – Open Systems Interconnection. Basic Connection Oriented Session Service Definition and Protocol Specification. IS 8326, IS 8327.
- [4] ISO, Information Processing Systems – Open Systems Interconnection File Transfer, Access and Management. DP 8571/1, DP 8571/2, DP 8571/3, DP 8571/4.
- [5] ISO, Job Transfer and Manipulation Concepts, Service Definition and Protocol Specification. DP 8831, DP 8832.

- [6] ECMA/TC23/84/162, Distributed Application for Message Interchange (MIDA) Message Transfer Service Access. Fourth Draft, Dec. 84.
- [7] ECMA/TC29/84/62, Office Document Architecture working draft.
- [8] Proceedings of the IEEE, Dec. 83, Vol. 71, No. 12.
- [9] ISO, Information Processing Systems – Open Systems Interconnection Specification of Abstract Syntax Notation One (ASN1) and Basic encoding Rules for Abstract Syntax Notation One DP 8824, DP 8825.
- [10] W. v. Pattay, F. Puhlmann: Wege zur offenen Kommunikation – das ISO-Referenzmodell im Umfeld der Kommunikation. Siemens, Bestell-Nr. U-1474-J-Z72-2.

Dipl.-Ing. *Winfried Blumann* (48). Studium der Nachrichtentechnik an der TH Karlsruhe. Von 1963 bis 1965 Entwicklung von Trägerfrequenzfiltern bei Felten & Guilleaume in Nürnberg. Ab 1966 zunächst Siemens Erlangen, Bereich Automatisierungstechnik, dann Siemens München, Bereich Datentechnik. Von 1966 bis 1974 Entwicklung von Ein-/Ausgabe-Geräten, -Schnittstellen und -Prozessoren für Prozeßrechner und dazugehöriger Systemsoftware. Von 1975 bis 1980 Entwicklung von Transaktionssystemen für die Betriebssysteme BS1000 und BS2000. Von 1981 bis 1984 Entwicklung von portierbaren Filetransfer-Produkten für die Betriebssysteme BS2000, SINIX, BS1000 und IBM/OS/VS2/MVS. Mitarbeit in DIN NI21.5 und ISO TC97/SC21.5 insbesondere auf dem Gebiet Filetransfer. Seit 1985 Entwicklung von Kommunikationsdiensten im Rahmen von Electronic Mail nach CCITT X.400 bzw. X.410. Vertretung von Siemens im Pilotprojekt ESPRIT/ROSE zur Förderung der ISO/OSI-Protokolle. Mitarbeit in DIN NI21.6 und ISO/TC97/SC21.6 insbesondere auf dem Gebiet Common Application Services.

# Approximative Produktformlösungen für Blockiernetze

## Approximate product form solutions for blocking networks

I. F. Akyildiz, Louisiana State University

*Die Blockiernetze haben einen beschränkten Zustandsraum, über den die Zustandswahrscheinlichkeiten in Produktform und daraus die Leistungsgrößen bestimmt werden können. In dieser Arbeit werden verschiedene Algorithmen entwickelt, die sich daher auf einen Teil der ausführbaren Zustände des eigentlichen Zustandsraumes beziehen. Zunächst wird die Produktformlösung für Blockiernetze vorgestellt und danach drei effiziente Algorithmen entwickelt, mit denen dann die Normalisierungskonstante bestimmt werden kann. Neue Formeln werden für die Leistungsgrößen eingeführt, die die ermittelte Normalisierungskonstante benutzen. Zum Schluß werden anhand vieler Beispiele die durch die vorgeschlagenen Algorithmen ermittelten Ergebnisse mit Simulation verglichen.*

*The blocking networks have a smaller state space than the networks with infinite station capacities. Based on this fact we show that the blocking networks may have approximate product form solutions. Three different algorithms are introduced for the computation of the normalization constant which occurs in the product form solution.*

*New formulas are developed for the performance measures of blocking networks. A numerical example is given to show how the suggested algorithms can be applied. Finally by executing several examples the analytical results are compared with simulation.*

### 1 Einführung

Warteschlangennetze mit Produktform haben sich bei der Bildung vieler Rechen- und Kommunikationssystemen als sehr wertvoll erwiesen und sie sind flexibel genug, um hinreichend viele der Besonderheiten, die in solchen Anwendungen entstehen, zu repräsentieren. Die grundlegenden Ergebnisse wurden in [1] gegeben, um Produktformlösungen für Warteschlangennetze mit mehreren Auftragsklassen, allgemeinen Bedienzeitverteilungen und verschiedenen Warteschlangendisziplinen (FCFS, RR-PS, LCFS-PR) zu erhalten. Die Produktformnetze (auch BCMP bzw. zerlegbare Netze genannt) sind jedoch nicht in der Lage, einen großen Einblick in das Phänomen des Blockierens zu geben. Die

Produktformnetze gehen immer davon aus, daß die Kapazitäten der Warteschlangen keinen Beschränkungen unterliegen. Sind die Kapazitäten der Warteschlangen endlich, können nach gewisser Zeit Blockierungen vorkommen, d. h., wenn die Kapazität einer Warteschlange erschöpft ist, kann sie keinen neuen Auftrag mehr annehmen. Der Auftrag, der in diese Station kommen sollte, muß in der augenblicklichen Station verweilen und diese blockieren, bis in der vollen Station ein Platz frei wird. Es ist ihm dabei nicht möglich, die Blockade der Station dadurch zu verhindern, daß er in eine andere Station, deren Kapazität nicht erschöpft ist, übergeht. Der Fall, daß ein Auftrag wegen einer vollen Station verlorengeht, wird in diesem Zusammenhang nicht berücksichtigt.

Obwohl bisher viele Arbeiten veröffentlicht wurden, gibt es keine allgemeingültigen einfachen Algorithmen für geschlossene Netze mit endlichen Stationskapazitäten. Im folgenden geben wir eine kurze Übersicht über die wichtigsten bisherigen Arbeiten.

Konheim und Reiser [2; 3] untersuchen ein offenes Warteschlangennetz mit einem unendlichen Puffer vor einer zyklischen Anordnung von zwei Bedienstationen. Die Analyse wird unter der Annahme exponentieller Verteilungen durchgeführt. Wenn die Anzahl der Aufträge in den zwei Bedienstationen einen Maximalwert  $M$  erreicht, werden neu ankommende Aufträge im Puffer zurückgehalten, d. h. blockiert. Ein Algorithmus zur Berechnung der stationären Zustandswahrscheinlichkeiten wird angegeben und einige Spezialfälle erarbeitet. Die Vorgehensweise liefert aber nur für die betrachteten Spezialfälle exakte Ergebnisse. Außerdem ist der Algorithmus komplex und sehr aufwendig.

Pittel [4] betrachtet zwei geschlossene Markoffsche Warteschlangennetze mit Blockierung. Im ersten Modell geht ein Auftrag in die gleiche Station wieder zurück, wenn die Station, in die er eintreten möchte, voll besetzt ist. Im zweiten Modell geht ein Auftrag zu einer beliebigen anderen Station, wenn die Station, zu der er übertreten wollte, schon besetzt ist. Die Zeit, die ein Auftrag für das Suchen einer leeren Station verbringt, wird in diesem Modell vernachlässigt. Durch Anwendung der nichtlinearen mathematischen Programmiermethoden erhält [4] eine asymptotische Lösung für die Blockierwahrscheinlichkeiten.

Boxma und Konheim [5] führen eine andere Vorgehensweise für die Behandlung der offenen Netze mit Blockierung an. Wenn bei einer Station  $j$  die Kapazität erreicht ist, werden alle Stationen blockiert, die Aufträge bedienen, die zur Station  $j$  übergehen werden. Außerdem werden die externen Ankünfte zu einer vollen Station ignoriert. Es werden zwei Phasen für eine Station eingeführt, in denen sich das Gesamtsystem abwechselnd befindet. Zunächst werden die Wahrscheinlichkeiten geschätzt, daß sich das System in Phase 2 befindet und dann die Wahrscheinlichkeit approximiert, mit der sich höchstens  $j$  Aufträge im System befinden. Die Methode funktioniert aber nur unter der Bedingung, daß ein Auftrag nicht mehr wieder in die gleiche Station zurückkommt. Die Genauigkeit der Ergebnisse

liegt für 2 bzw. 3 Stationen ungefähr bei 10%. Für größere Netze gibt es große Fehlerabweichungen.

Die Isolationsmethode von Labetoulle und Pujolle [6] und ihre Erweiterung von Georges [7] ist geeignet, Paketvermittlungsnetze mit Blockierung für Kommunikationssysteme zu analysieren. Die Gesamtvorgehensweise basiert darauf, daß das System in einzelne Stationen zerlegt wird. Die Parameter der isolierten Station werden so angepaßt, daß der Einfluß der übrigen Stationen des Netzes mitberücksichtigt wird. Die so entstandene G/G/1-Station wird mit Hilfe der Diffusionsapproximation [8] analysiert. Ein Auftrag, der von Station  $i$  fertig bedient worden ist und nach Station  $j$  übergehen möchte, dieses aber nicht kann, da Station  $j$  bereits voll besetzt ist, wird von Station  $i$  ein weiteres Mal bedient. Die Wahrscheinlichkeiten der Rückweisung eines Auftrags ist beim ersten Versuch gleich der Wahrscheinlichkeit, daß die Warteschlange der Station  $j$  voll besetzt ist. Bei den weiteren Versuchen wird die Information, daß die Station  $j$  eine Bedienzeit zuvor noch voll war, mit in die Rückweisungswahrscheinlichkeit einbezogen. Die Ergebnisse weichen von den Simulationsergebnissen um 10 bis 25% ab.

Hordijk und Van Dijk [9] beschäftigen sich mit geschlossenen Netzwerken aus Stationen mit exponentieller Bedienzeitverteilung und mit Blockierung. Die Blockierung der Aufträge an einer Station soll nur von der Zahl der Aufträge in dieser Station abhängen. Für diese Netze stellen sie das globale Gleichgewichtsgleichungssystem auf und lösen es nach Produktformeln, wobei die Gesamtvorgehensweise auf den möglichen ausführbaren Zuständen basiert. In der Arbeit werden nur Produktformlösungen gegeben und keine explizite Angaben, wie die in der Produktformlösung vorkommende Normalisierungskonstante und die Leistungsgrößen bestimmt werden sollen.

In den letzten Jahren wurden noch viel mehr Algorithmen zur Analyse von Blockiernetzen veröffentlicht. Hiervon sind z. B. noch [10; 11] zu nennen. Eine ausführliche Liste von den bisherigen Untersuchungen über die Blockiernetze wurde von H. G. Perros [12] angefertigt.

## 2 Produktformlösung für Netze mit Blockierung

Wir betrachten geschlossene Warteschlangennetze, die die folgenden Annahmen erfüllen:

- i) Es gibt  $N$  Bedienstationen und  $K$  Aufträge im System.
- ii) Jede Station hat eine Bedieneinheit ( $m_i = 1$ ) und exponentielle Bedienzeitverteilung mit Mittelwert  $\frac{1}{\mu_i}$  für  $i = 1, 2, \dots, N$ .
- iii) Jede Station hat die endliche, konstante Kapazität  $M_i$  für  $i = 1, \dots, N$ . ( $M_i = \text{Warteschlangenkapazität} + 1$ ).
- iv) Ein in der Station  $i$  fertig bedienter Auftrag geht mit der Wahrscheinlichkeit  $p_{ij}$  zur Station  $j$  über, falls in der  $j$ -ten Station die Anzahl der Aufträge die Kapazität  $M_j$  nicht erreicht ist. Sonst blockiert sich der

Auftrag in der  $i$ -ten Station so lange, bis in der  $j$ -ten Station ein Auftrag fertig bedient und ein Platz für diesen Auftrag verfügbar wird. Falls sich mehrere Aufträge an ein und derselben Station blockieren, so werden sie nach FCFS deblockiert.

- v) Die Gesamtanzahl der Aufträge  $K$  im System muß kleiner sein als die Summe der Stationskapazitäten, d.h.:

$$K < \sum_{i=1}^N M_i$$

- vi) Die Aufträge werden in den Stationen nach FCFS abgearbeitet.

Sei  $k_i$  die Anzahl der Aufträge in der  $i$ -ten Station (wartend und in Bedienung) und der Zustand des Netzwerkes ist definiert durch den folgenden Vektor:

$$\underline{k} = (k_1, \dots, k_N)$$

Sei  $p(k_1, \dots, k_N)$  die Wahrscheinlichkeit, daß das Netz sich im Zustand  $\underline{k}$  befindet. Diese Wahrscheinlichkeit aller Systemzustände im statistischen Gleichgewicht zu bestimmen, ist eines der Hauptprobleme der analytischen Modellbildung. Denn aus diesen Zustandswahrscheinlichkeiten lassen sich alle Leistungsgrößen leicht ableiten.

Hordijk und Van Dijk [9] haben gezeigt, daß sich die Wahrscheinlichkeit für den Gleichgewichtszustand  $\underline{k}$  in einem geschlossenen Netzwerk mit endlichen Stationskapazitäten durch das Produkt der Randwahrscheinlichkeiten der einzelnen Stationen wie folgt ergibt:

$$p(k_1, k_2, \dots, k_N) = \frac{1}{G(K)} \prod_{i=1}^N x_i^{k_i} \delta_i(k_i), \quad (1)$$

wobei  $G(K)$  die Normalisierungskonstante ist, so daß die Summe aller Wahrscheinlichkeiten auf 1 normiert wird.  $x_i$  ist die relative Auslastung der  $i$ -ten Station und gegeben durch:

$$x_i = \frac{e_i}{\mu_i} \quad (2)$$

$e_i$  ist die mittlere Anzahl der Besuche eines Auftrags in die  $i$ -te Station:

$$e_i = \sum_{j=1}^N e_j p_{ji} \text{ für } i = 1, 2, \dots, N,$$

wobei  $e_i$  die mittlere Besuchshäufigkeit eines Auftrags in die  $i$ -te Station ist. Da es im geschlossenen Netz  $(N-1)$  unabhängige Gleichungen gibt, nimmt man  $e_i = 1$  für beliebige  $i = 1, 2, \dots, N$  an. Die binäre Funktion  $\delta_i(k_i)$  ergibt sich aus

$$\delta_i(k_i) = \begin{cases} 0 & \text{if } k_i > M_i \\ 1 & \text{if } k_i \leq M_i \end{cases}$$

Durch  $\delta_i(k_i)$  werden alle Flußwahrscheinlichkeiten in oder aus nicht ausführbaren Zuständen (aufgrund der Kapazitäten) gleich 0 gesetzt. M.a.W. durch diese Funktion werden die Zustände eliminiert, die die Kapazitäten der einzelnen Stationen überschreiten.

Zur Verdeutlichung dieser Definitionen geben wir an dieser Stelle ein Beispiel.

Gegeben sei ein geschlossenes Netz mit  $N = 3$  Stationen und  $K = 5$  Aufträgen. Die Kapazität der ersten

Station ist  $M_1 = 3$ , von der zweiten Station  $M_2 = 2$  und von der dritten Station  $M_3 = 4$ . Wenn wir die Stationskapazitäten des Netzwerkes nicht berücksichtigen, hat der Zustandsraum folgende Struktur:

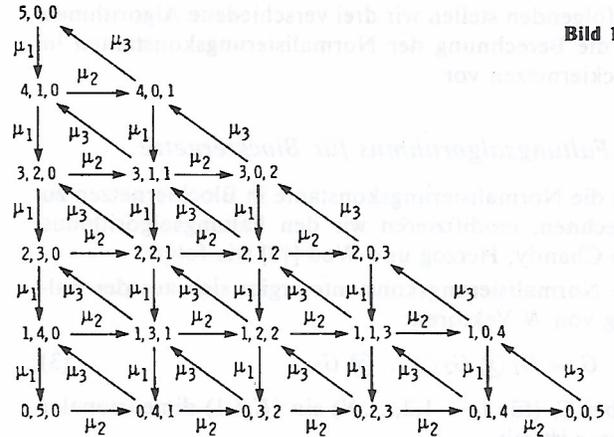


Bild 1.

wobei sich die Übergangsraten von einem Zustand in den anderen aus den Bedienraten der Stationen  $\mu_i$  multipliziert mit den Übergangswahrscheinlichkeiten  $p_{ij}$  (für  $i, j = 1, 2, \dots, N$ ) ergeben.

Da wir endliche Stationskapazitäten haben, können wir nicht annehmen, daß alle Zustände in Bild 1 ausführbar sind. Die ausführbaren Zustände für das Blockiernetz sind die Zustände, bei denen die Anzahl  $k_i$  der Aufträge die Kapazität der Station  $i$  nicht überschreitet. Zum Beispiel der Zustand  $(4, 1, 0)$  ist nicht ausführbar, da sich 4 Aufträge in der ersten Station wegen der Kapazitätsüberschreitung nicht aufhalten dürfen.

So erhalten wir den folgenden Zustandsraum für das Blockiernetz:

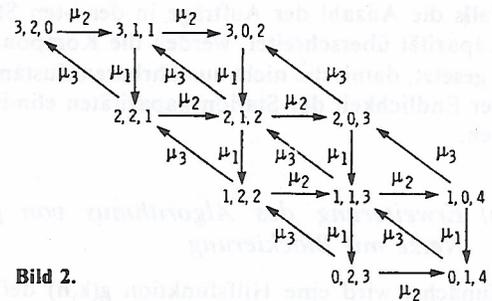


Bild 2.

Wie in der Gleichung 1 ersichtlich, muß die Normalisierungskonstante berechnet werden, um die Zustandswahrscheinlichkeiten zu erhalten.

### 3 Algorithmen zur Berechnung der Normalisierungskonstanten

Geschlossene Produktformnetze haben einen großen Einfluß auf die Rechensystemanalyse. Sie werden benutzt um z.B. Mehrprogramm- und „Time Sharing“-Systeme sowie „Window-Flow-Control“ in Kommunikationsnetzen als Modell abzubilden. Es ist daher nicht verwunderlich, daß erhebliche Anstrengungen unternommen werden, um ein Verfahren zur Berechnung der in geschlossenen Netzen vorkommenden Normalisie-

rungskonstanten zu finden. Die Normalisierungskonstante ist das einzige Element bei der Produktformlösung, das einen größeren rechnerischen Aufwand erfordert.

Im folgenden stellen wir drei verschiedene Algorithmen für die Berechnung der Normalisierungskonstanten in Blockiernetzen vor.

a) *Faltungsalgorithmus für Blockiernetze*

Um die Normalisierungskonstante in Blockiernetzen zu berechnen, modifizieren wir den Faltungsalgorithmus von Chandy, Herzog und Woo [13] wie folgt:

Die Normalisierungskonstante ergibt sich aus der Faltung von  $N$  Vektoren:

$$G = G_1 \otimes G_2 \otimes \dots \otimes G_N \quad (3)$$

wobei  $G_i$  (für  $i = 1, 2, \dots, N$ ) ein  $(K + 1)$  dimensionaler Vektor ist mit

$$G_i = \begin{bmatrix} y_i(0) \\ y_i(1) \\ y_i(2) \\ \dots \\ y_i(K) \end{bmatrix}$$

Die Komponenten  $y_i(\cdot)$  lassen sich wie folgt bestimmen:

$$y_i(k) = \begin{cases} x_i^k & \text{if } k = 0, 1, 2, \dots, M_i \\ 0 & \text{if } k = M_i + 1, \dots, K \end{cases} \quad (4)$$

wobei sich der  $x_i$  Wert aus der Gleichung (2) ergibt.

Falls die Anzahl der Aufträge in der  $i$ -ten Station die Kapazität überschreitet, werden die Komponenten auf 0 gesetzt, damit die nicht ausführbaren Zustände wegen der Endlichkeit der Stationskapazitäten eliminiert werden.

b) *Erweiterung des Algorithmus von [14] auf Netze mit Blockierung*

Zunächst wird eine Hilfsfunktion  $g(k, n)$  definiert mit der Eigenschaft, daß der erste Parameter die Anzahl der Aufträge und der zweite die Stationen im System zeigt. Der Algorithmus wird wie folgt durchgeführt:

Die Initialwerte

$$g(0, n) = 1 \text{ für } n = 1, 2, \dots, N$$

$$\text{und } g(k, 1) = \begin{cases} x_1^k & \text{if } k \leq M_1 \\ 0 & \text{if } k > M_1 \end{cases} \quad (5)$$

Die Hilfsfunktion  $g(k, n)$  ergibt sich aus:

$$g(k, n) = g(k, n-1) + x_n g(k-1, n) \quad (6)$$

für  $n = 1, 2, \dots, N$  und  $k = 1, 2, \dots, M_n$ .

Wenn die Kapazität der Station  $M_n$  überschritten wird, d.h. für  $k > M_n$ , läßt sich  $g(k, n)$  wie folgt ermitteln:

$$g(k, n) = g(k, n-1) + x_n g(k-1, n) - x_n^k h(l, n) \quad (7)$$

Die Funktion  $h(l, n)$  eliminiert die nicht ausführbaren Zustände. Sie ergibt sich aus:

$$h(l, n) = \frac{1}{l} \sum_{i=1}^{n-1} \sum_{j=1}^l x_i^j h(l-j, n) \quad (8)$$

für  $l = 0, 1, \dots, K$ .

Mit dem Initialwert  $h(0, n) = 1$  für  $n = 1, 2, \dots, N$ .

Das Ziel des Algorithmus ist die Bestimmung des Wertes der Funktion  $g(K, N)$ , weil dieser Wert die Normalisierungskonstante  $G(K)$  ist.

c) *LBANCBLO (Local Balance Algorithm for Normalizing Constant in Networks with Blocking)*

Ausgehend von der Mittelwertanalyse entwickelten Sauer und Chandy [15] einen Algorithmus mit der Bezeichnung LBANC, der aber nur auf geschlossene Produktformnetze anwendbar ist. Für die geschlossenen Netzwerke mit endlichen Stationskapazitäten kann der Algorithmus in der gegebenen Form nicht angewandt werden. Daher wird im folgenden eine modifizierte Version von LBANC vorgestellt.

LBANCBLO benutzt die folgende Rekursionsformel, um die *unnormierte mittlere Warteschlangenlänge* zu bestimmen:

$$q_i(k) = x_i [G(k-1) + q_i(k-1) - x_i^{M_i} (M_i + 1) b_i(l)] \quad (9)$$

mit den Initialwerten

$$G(0) = 1 \text{ und } q_i(0) = 0$$

für  $i = 1, 2, \dots, N$ .

Die Funktion  $b_i(l)$  eliminiert die nicht ausführbaren Zustände und ist gegeben durch:

$$b_i(l) = \begin{cases} \frac{1}{l} \sum_{j=1}^N s_j(l) & \text{if } l > M_i \\ 0 & \text{if } l \leq M_i \end{cases}$$

wobei

$$s_j(l) = x_j [b_j(l-1) + s_j(l-1)] \text{ for } j = 1, 2, \dots, N$$

für  $l = 0, 1, 2, \dots, K$  und mit den Initialwerten  $s_j(0) = 0$  und  $b_i(0) = 1$ .

Die Normalisierungskonstante wird dann wie folgt ermittelt:

$$G(K) = \frac{1}{K} \sum_{i=1}^N q_i(K) \quad (10)$$

4 *Leistungsgrößen für Blockiernetze*

Da wir Netze mit Blockierungen behandeln, können wir die klassischen Formeln [8; 15] für die Leistungsgrößen nicht verwenden. Im folgenden führen wir neue Formeln für die Leistungsgrößen ein, die über den ausführbaren Zustandsraum der Blockiernetze gelten.

*Die Randwahrscheinlichkeiten*, d.h. die Wahrscheinlichkeiten dafür, daß in der  $i$ -ten Station genau  $l$  Aufträge sind, ergeben sich aus

$$p_i(l) = \frac{x_i^l G_i - (K-l)}{G(K)} \quad (11)$$

wobei  $G_i$ — die Normalisierungskonstante ist, die ohne Bedienstation  $i$  berechnet wird.

Die mittlere Anzahl von Aufträgen in der  $i$ -ten Station ergibt sich aus:

$$\bar{k}_i(K) = \sum_{l=1}^{M_i} l p_i(l) \quad (12)$$

Durch Einsetzen der Gleichung (11) erhält man

$$\bar{k}_i(K) = \frac{1}{G(K)} \sum_{l=1}^{M_i} l x_i^l G_i(K-l) \quad (13)$$

Auf ähnlichem Wege erhält man die mittleren Warteschlangenlängen:

$$\bar{Q}_i(K) = \frac{1}{G(K)} \sum_{l=1}^{\min\{K, M_i\}} (l-1) x_i^{l+1} G(K-l-1) \quad (14)$$

Den Durchsatz des Gesamtsystems errechnet man mit

$$\lambda(K) = \frac{K}{\frac{1}{G(K)} \cdot S + \sum_{i=1}^N x_i} \quad (15)$$

wobei

$$S = \sum_{i=1}^N \sum_{l=1}^{\min\{K, M_i\}} l x_i^{l+1} G(K-l-1).$$

Aus dem Durchsatz  $\lambda(K)$  ermitteln wir dann die Durchsätze der einzelnen Stationen mit

$$\lambda_i(K) = e_i \lambda(K) \quad \text{für } i = 1, 2, \dots, N. \quad (16)$$

Die mittlere Verweilzeit eines Auftrags in der  $i$ -ten Station ermittelt man durch Anwendung von Littles Gesetz:

$$\bar{t}_i(K) = \frac{\bar{k}_i(K)}{\lambda_i(K)} \quad \text{für } i = 1, 2, \dots, N. \quad (17)$$

Die mittlere Wartezeit eines Auftrags in der  $i$ -ten Station erhält man auf demselben Weg:

$$\bar{w}_i(K) = \frac{\bar{Q}_i(K)}{\lambda_i(K)} \quad \text{für } i = 1, 2, \dots, N. \quad (18)$$

Die Auslastungen der einzelnen Stationen erhält man mit der bekannten Formel:

$$\rho_i(K) = \frac{\lambda_i(K)}{\mu_i} \quad \text{für } i = 1, 2, \dots, N. \quad (19)$$

Im nächsten Abschnitt zeigen wir anhand eines numerischen Beispiels die Anwendung der eingeführten Algorithmen und Formeln.

## 5 Beispiel

Gegeben sei ein geschlossenes Netz mit  $N=2$  Bedienstationen (z. B. ein CPU und ein E/A-Gerät) und  $K=6$  Aufträgen. Die Bedienstationen haben eine Bedieneinheit ( $m_i = 1$ ) und exponentielle Bedienzeitverteilungen mit den Mittelwerten  $\frac{1}{\mu_1} = 3$  sec bzw.  $\frac{1}{\mu_2} = 2$  und FCFS Strategien. Die erste Station hat die konstante Kapazität  $M_1 = 4$  und die zweite Station  $M_2 = 3$ .

Zunächst ermitteln wir aus Gleichung (3) die Normalisierungskonstante wie folgt:

$$G = G_1 \otimes G_2,$$

wobei

$$G = \begin{bmatrix} 1 \\ 3 \\ 9 \\ 27 \\ 81 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 19 \\ 65 \\ 195 \\ 342 \\ 540 \end{bmatrix}$$

Jetzt sind wir in der Lage, durch Verwendung der gegebenen Formeln die Leistungsgrößen für dieses Netz zu ermitteln:

Die mittlere Anzahl von Aufträgen (Gleichung 13):

$$\bar{k}_1 = \frac{1}{G(K)} \sum_{l=1}^4 l x_1^l G_2(K-l) = 3,6$$

$$\bar{k}_2 = \frac{1}{G(K)} \sum_{l=1}^3 l x_2^l G_1(K-l) = 2,4$$

Der Durchsatz (Gleichungen 15 und 16):

$$\lambda_1 = 0,328 \quad \lambda_2 = 0,328$$

Die mittlere Verweilzeit (Gleichung 17):

$$\bar{t}_1 = 10,959 \quad \bar{t}_2 = 7,306$$

Die anderen Leistungsgrößen lassen sich analog aus den entsprechenden Formeln berechnen. In der folgenden Tabelle stellen wir diese Werte den exakten Werten [16] gegenüber. Man kann leicht feststellen, daß die Abweichungen gering sind. OKAP bedeutet, daß das Netz ohne Berücksichtigung der Kapazitäten analysiert worden ist.

Tabelle 1.

	Analytisch	Numerische Analyse	Abweichung (%)	OKAP
$\bar{k}_1$	3,6	3,69	2,5	1,564
$\bar{k}_2$	2,4	2,30	4	4,435
$\bar{t}_1$	10,959	12,63	13,2	4,843
$\bar{t}_2$	7,306	7,89	7,4	13,734
$\lambda$	0,328	0,29	12,4	0,323

## 6 Bewertung

Zur Bewertung wurden die im Abschnitt 3 und 4 vorgestellten Algorithmen bzw. Formeln auf der Rechenanlage CDC Cyber 845 implementiert [17]. Es wurden insgesamt mehr als 75 Netzwerke zur Bewertung getestet und die Ergebnisse mit der Simulation [18] sowie mit exakten Resultaten, die durch die numerische Analyse berechnet wurden, verglichen.

Die Zustandswahrscheinlichkeit eines Netzwerks im Gleichgewicht mit der Produktform zu berechnen, heißt u. a. der Reihenfolge, in der die Stationen gekoppelt sind, keine Bedeutung zu geben. Die Übergangswahr-

scheinlichkeiten gehen in Gleichung (1) nur indirekt durch die mittleren Besuchshäufigkeiten  $e_i$  ein, in denen keinerlei Information mehr darüber enthalten ist, welche Stationen mit welchen Übergangswahrscheinlichkeiten miteinander gekoppelt sind. Da aber in Blockiernetzen eine Station durch Blockierung unmittelbaren Einfluß auf ihre Vorgängerinnen nehmen kann, ist diese Information bei Blockiernetzen, wie sie in dieser Arbeit behandelt werden, von großer Wichtigkeit.

Die Güte der Ergebnisse von unserer Vorgehensweise hängt demzufolge in starkem Maße davon ab, wie die Stationen miteinander gekoppelt sind und wie das Blockierverhalten des Netzes ist.

Netze, die keine ausgeprägte Engpaßstation besitzen und solche, in denen aufgrund von „low traffic“ nur relativ wenige Blockierungen vorkommen, werden von der Produktformvorgehensweise mit gutem Ergebnis analysiert. Hier liegen die Abweichungen für die mittleren Auftragszahlen bei 10-12% und für den Durchsatz bei 0-10%. Höhere Abweichungen ergeben sich nur bei numerisch kleinen Werten. Doch bei Netzen, in denen ausgeprägte Engpässe und demzufolge zahlreiche Blockierungen vorkommen, können sehr große Abweichungen auftreten. In solchen Fällen wird z. B. einer schnellen Station, auf die eine Engpaßstation folgt und die dementsprechend häufig blockiert wird, viel zu wenig Aufträge zugeteilt.

Im folgenden zeigen wir Ergebnisse von fünf Beispielen, die sich auf 2-Stationen Netze beziehen.

Die Eingabegrößen lauten wie folgt:

Tabelle 2.

Nr.	K	M <sub>1</sub>	M <sub>2</sub>	1/μ <sub>1</sub>	1/μ <sub>2</sub>
1	20	18	10	2	3,33
2	25	18	13	0,8	0,5
3	15	10	10	1,111	1,5
4	8	4	7	10	2,5
5	50	31	24	4	2

Die analytischen Ergebnisse werden in der folgenden Tabelle gezeigt:

Tabelle 3.

$\bar{k}_1$	$\bar{k}_2$	$\bar{n}_1$	$\bar{n}_2$	λ
11,41	8,59	31,23	23,52	0,365
16,60	8,39	12,12	6,13	1,369
6,67	8,37	9,48	11,84	0,703
3,68	4,32	26,48	31,05	0,139
30,01	19,9	98,78	65,33	0,304

Die Simulationsergebnisse werden in der folgenden Tabelle aufgelistet:

Tabelle 4.

$\bar{k}_1$	$\bar{k}_2$	$\bar{n}_1$	$\bar{n}_2$	λ
11,02	8,987	36,58	29,84	0,3013
17,12	7,884	13,85	6,383	1,235
6,369	8,631	9,803	13,27	0,6504
3,925	4,075	39,82	41,34	0,0986
30,57	19,43	124,4	79,16	0,2457

Die nachstehende Tabelle zeigt die Abweichungen (in %) der berechneten Lösungen von den simulierten Werten:

$$\delta = \frac{|\text{Simulationswert} - \text{berechneter Wert}|}{\text{Simulationswert}} * 100$$

Tabelle 5.

Beispiel	$\bar{k}_1$	$\bar{k}_2$	$\bar{n}_1$	$\bar{n}_2$	λ
1	3,418	4,62	17,1	16,8	17,5
2	3,1	6,09	14,2	4,1	17,1
3	4,5	3,11	3,38	12,0	7,5
4	6,657	5,67	50,3	33,1	29,0
5	1,56	2,38	25,9	21,1	24,0

Wie man leicht feststellen kann, sind die Abweichungen relativ gering. Größere Abweichungen kommen dort vor, wo in den Netzen große Abhängigkeit zwischen den Stationen vorliegt.

Im folgenden listen wir Ergebnisse von verschiedenen numerischen Beispielen mit mehr als 2-Stationen auf, die mit den in dieser Arbeit vorgeschlagenen Algorithmen und Formeln erzielt worden sind. Alle analytischen Ergebnisse werden mit denen der Simulation verglichen.

Beispiel 1

N = 3 Stationen, K = 5 Aufträge, M<sub>i</sub> = 2, for i = 1,2,3  
1/μ<sub>1</sub> = 1, 1/μ<sub>2</sub> = 0.9, 1/μ<sub>3</sub> = 2 (seriell geschaltet)

Tabelle 6.

	PROBLO	Simulation	Abweichung (%)	OKAP
$\bar{n}_1$	3,058	2,921	4,5	1,752
$\bar{n}_2$	2,977	4,129	38,6	1,478
$\bar{n}_3$	3,420	4,217	23,3	7,212
$\bar{k}_1$	1,617	1,297	19,7	0,839
$\bar{k}_2$	1,575	1,832	16,2	0,708
$\bar{k}_3$	1,809	1,871	3,3	3,454
λ	0,529	0,444	19,1	0,479

Beispiel 2

N = 3 Stationen, K = 8 Aufträge, M<sub>1</sub> = 6, M<sub>2</sub> = 5, M<sub>3</sub> = 8, 1/μ<sub>1</sub> = 0.6, 1/μ<sub>2</sub> = 0.8, 1/μ<sub>3</sub> = 1 (seriell geschaltet)

Tabelle 7.

	PROBLO	Simulation	Abweichung (%)	OKAP
$\bar{n}_1$	1,319	1,49	13	1,272
$\bar{n}_2$	1,955	2,214	13,2	2,465
$\bar{n}_3$	5,304	4,794	9,6	4,798
$\bar{k}_1$	1,23	1,403	14,1	1,922
$\bar{k}_2$	1,823	2,086	14,4	2,311
$\bar{k}_3$	4,947	4,512	8,8	4,497
λ	0,933	0,942	i	0,937

## Beispiel 3

$N = 4$  Stationen,  $K = 8$  Aufträge,  $M_i = 3$  for  $i = 1, 2, 3, 4$   
 $1/\mu_1 = 0.6$ ,  $1/\mu_2 = 0.7$ ,  $1/\mu_3 = 0.4$ ,  $1/\mu_4 = 0.8$  (seriell geschaltet)

Tabelle 8.

	PROBLO	Simulation	Abweichung (%)	OKAP
$\bar{t}_1$	1,807	1,647	9,6	1,467
$\bar{t}_2$	1,961	1,865	5,1	2,121
$\bar{t}_3$	1,255	1,767	29	0,678
$\bar{t}_4$	2,071	2,505	17,3	3,064
$\bar{k}_1$	2,038	1,693	20,3	1,601
$\bar{k}_2$	2,211	1,918	15,3	2,315
$\bar{k}_3$	1,416	1,816	22	0,74
$\bar{k}_4$	2,336	2,574	9,3	3,344
$\lambda$	1,128	1,028	9,7	1,091

## Beispiel 4

$N = 5$  Stationen,  $K = 8$  Aufträge,  $M_1 = 2$ ,  $M_2 = 4$ ,  $M_3 = 3$ ,  $M_4 = 4$ ,  $M_5 = 2$ ,  $1/\mu_1 = 1$ ,  $1/\mu_2 = 2$ ,  $1/\mu_3 = 1.5$ ,  $1/\mu_4 = 1.8$ ,  $1/\mu_5 = 1.6$  (seriell geschaltet)

Tabelle 9.

	PROBLO	Simulation	Abweichung (%)	OKAP
$\bar{t}_1$	1,701	2,046	16	1,598
$\bar{t}_2$	6,393	5,937	7,6	6,338
$\bar{t}_3$	3,537	3,667	3,5	3,274
$\bar{t}_4$	5,781	6,084	4,9	4,878
$\bar{t}_5$	2,501	2,873	12	3,743
$\bar{k}_1$	0,683	0,794	14	0,645
$\bar{k}_2$	2,569	2,306	11	2,557
$\bar{k}_3$	1,421	1,425	-	1,321
$\bar{k}_4$	2,323	2,361	1,6	1,968
$\bar{k}_5$	1,005	1,115	9,8	1,511
$\lambda$	0,402	0,388	3,5	0,403

## Beispiel 5

$N = 4$  Stationen,  $K = 15$  Aufträge,  $M_1 = 3$ ,  $M_2 = 6$ ,  $M_3 = 7$ ,  $M_4 = 4$ ,  $1/\mu_1 = 1$ ,  $1/\mu_2 = 0.9$ ,  $1/\mu_3 = 0.667$ ,  $1/\mu_4 = 0.8$ ,  $p_{12} = 0.6$ ,  $p_{13} = 0.4$ ,  $p_{21} = 0.55$ ,  $p_{24} = 0.45$ ,  $p_{32} = 0.5$ ,  $p_{34} = 0.5$ ,  $p_{43} = 1.0$

Tabelle 10.

	PROBLO	Simulation	Abweichung (%)	OKAP
$\bar{t}_1$	2,598	3,141	20,8	2,095
$\bar{t}_2$	4,97	5,331	11,2	4,834
$\bar{t}_3$	4,461	4,364	2,1	3,495
$\bar{t}_4$	2,65	3,335	25,8	4,234
$\bar{k}_1$	1,397	1,539	9,2	1,122
$\bar{k}_2$	4,858	4,923	1,3	4,705
$\bar{k}_3$	5,844	5,21	10,8	4,558
$\bar{k}_4$	2,901	3,328	12,8	4,616
$\lambda$	0,538	0,492	9,2	0,535

## Literatur

- [1] Baskett, F.; Chandy, K. M.; Muntz, R.; Palacios, G.: Open, Closed and Mixed Network of Queues with Different Classes of Customers. Journal of the ACM, Vol. 22, 2, April 1975, pp. 248-260.
- [2] Konheim, A. G.; Reiser, M.: A Queueing Model with Finite Waiting Room and Blocking. Journal of the ACM, Vol. 23, Number 2, April 1976, pp. 328-341.
- [3] Konheim, A. G.; Reiser, M.: Finite Capacity Queueing Systems with Applications in Computer Modelling. SIAM Journal on Computing, Vol. 7, Number 2, May 1977, pp. 210-229.
- [4] Pittel, B.: Closed Exponential Networks of Queues with Saturation: The Jackson Type Stationary Distribution and Its Asymptotic Analysis. Mathematics of Operations Research, Vol. 4, 1979, pp. 367-378.
- [5] Boxma, O. J.; Konheim, A. G.: Approximate Analysis of Exponential Queueing Systems with Blocking. Acta Informatica, Vol. 15, January 1981, pp. 19-66.
- [6] Labetoulle, J.; Pujolle, G.: Isolation Method in a Network of Queues. IEEE-Transactions on Software Engineering, Vol. SE-6, Nr. 4, July 1980.
- [7] Georges, J.: Amendments to an Analysis Method for Queueing Networks with Blocking. Philips, Technical Report, 1981.
- [8] Bolch, G.; Akyildiz, I. F.: Analyse von Rechensystemen. Teubner Verlag, Herbst 1982.
- [9] Hordijk, A.; van Dijk, N.: Networks of Queues with Blocking. Proceedings, 8th International Symposium on Computer Performance Modelling, Measurement, and Evaluation, Amsterdam, November 4-6, 1981.
- [10] Akyildiz, I. F.: Leistungsanalyse von Multiprozessorsystemen mit Prozeßkommunikation. Arbeitsbericht des IMMD der Universität Erlangen-Nürnberg, Bd. 17, Nr. 8, Erlangen, Dez. 1984.
- [11] Suri, R.; Diehl, G. W.: A New Building Block for Performance Evaluation of Queueing Networks with Finite Buffers. ACM Sigmetrics Conference Proceedings, Cambridge/Mass., Aug. 1984, pp. 134-142.
- [12] Perros, H. G.: „Queueing Networks with Blocking“: A Bibliography, ACM Sigmetrics Performance Evaluation Evaluation Review, August 1984.
- [13] Chandy, K. M.; Herzog, U.; Woo, L.: Parametric Analysis of Queueing Network Models. IBM Journal Res. Dev. 19, 1, Jan. 1975, pp. 43-49.
- [14] Buzen, J. P.: Queueing Network Models of Multiprogramming. PhD. Thesis, Div. Eng. and Applied Science, Harvard Univ., Cambridge/Mass., August 1971.
- [15] Sauer, C. H.; Chandy, K. M.: Computer Systems Performance Modelling. Prentice Hall, Englewood Cliffs, N.J., 1981.
- [16] Stewart, W. G.: A Comparison of Numerical Techniques in Markov Modeling, CACM, Vol. 21, Nr. 2, Feb. 1978, pp. 144-152.
- [17] Paterok, M.: QUEBLO - ein Programmpaket zur Analyse von Warteschlangennetzen mit Blockierung. Diplomarbeit am IMMD IV der Univ. Erlangen-Nürnberg, Okt. 1984.
- [18] Schmidt, B.: GPSS-Fortran Version 3, Springer Verlag, Herbst 1984.

Prof. Dr.-Ing. I. F. Akyildiz, geboren 1954 in Istanbul. Von 1976 bis 1981 Studium der Informatik an der Universität Erlangen-Nürnberg. Schwerpunkt Betriebssysteme. Von 1981 bis Febr. 1985 wiss. Mitarbeiter am Lehrstuhl für Betriebssysteme der Universität Erlangen-Nürnberg. Promotion 1984 bei Prof. Herzog auf dem Gebiet der Leistungsanalyse von Multiprozessorsystemen. Prof. Akyildiz ist Co-Autor des Buches „Analyse von Rechensystemen“. Seit Februar 1985 Professor an der Louisiana State University und Gastprofessor an der Universität von Florida.

Anschrift: Louisiana State University, Department of Computer Science, Baton Rouge, Louisiana 70803-4020, U.S.A.

