

Institut für Nachrichtenvermittlung und Datenverarbeitung  
der Universität Stuttgart  
Prof. Dr.-Ing. P. Kühn

## **40. Bericht über verkehrstheoretische Arbeiten**

Über die Leistungsfähigkeit  
virtueller Datenverbindungen  
entsprechend der CCITT-Empfehlung X.25

von  
Wolfgang Dieterle

Institute of Communications Switching and Data Technics  
University of Stuttgart  
Prof. Dr.-Ing. P. Kühn

## **40th Report on Studies in Congestion Theory**

On the Performance  
of Virtual Circuits  
According to CCITT Recommendation X.25

by  
Wolfgang Dieterle

© 1987 Institut für Nachrichtenvermittlung und Datenverarbeitung Universität Stuttgart

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

ISBN 3-922403-50-6

## Abstract

During the last 10 years, in many countries packet switched public data networks have been introduced to serve for the growing amount of data and computer communications. With the introduction of the Integrated Services Digital Network (ISDN) the infrastructure for digital communications will be greatly enhanced such that packet switched services will be accessible literally from any place.

The main issue of all these networks is the well defined transport of any data in a save and fast way between any communication entities.

For those purposes a plurality of rules, so called protocols, were committed and negotiated worldwide.

This report deals with the performance evaluation of one of those protocols, especially with the performance of so called virtual circuits within the data network access section, controlled by the protocol CCITT recommendation X.25.

X.25 is the CCITT standard for the interface between data terminal equipment and data circuit termination equipment for terminals operating in the packet mode on public data networks.

The X.25 standard is not only important for the terminal network interface of special packet switched public networks like the Datex-P within the Federal Republic of Germany; the X.25 standard forms also the base of packetized data communication within ISDN where non-ISDN terminal equipment accesses the ISDN-internal packet network through Terminal Adaptors (TA), B- or D-channels of the ISDN-subscriber loop and a Packet Handler within the ISDN-exchange.

The main issues of X.25 are:

- to provide an errorfree interface by error detection and correction
- to multiplex several logical channels onto the one physical transmission line
- to take care of the packet sequence and
- to control the flow of data individually for each of the virtual circuits established at the interface.

The main issue of this report is the performance evaluation of virtual circuits with respect to parameters such as:

- number of active virtual circuits
- amount of data on these circuits
- propagation delay and transmission rate of the physical line as well as of protocol specific parameters such as:
  - window size of the transmission and receiving window flow control mechanism.

## Chapter 1: Introduction

After a short introduction to communication systems and data communication protocols a brief survey of this report is given.

## Chapter 2: Theoretical Base

This chapter gives an overview of switching principles of general data networks as well as of packet switching networks.

After a short introduction and explanation of the architecture and the protocols of the so called open system interconnection reference model the chapter will be closed with principles of queuing and traffic theory.

### Chapter 3: The CCITT Recommendation X.25

In this chapter the structure, functions, special mechanisms as well as performance influencing parameters of X.25 will be explained and outlined in detail.

### Chapter 4: Implementation of a X.25 Interface

This chapter deals with the physical implementation of a X.25 interface and shows aspects of the realization such as process concepts, adjacent layer protocols, special peer-to-peer protocol implementations as well as the chosen structure of the hardware implementation.

### Chapter 5: Performance Evaluation

In the first part of this chapter a detailed queuing model of the X.25 interface will be outlined and described which functions as base for the performance analysis.

The second part deals with different methods of performance evaluation such as simulation and mathematical/queuing analysis.

In the mathematical part the chosen hierarchical decomposition method will be explained as well as the performance analysis of the resulting components.

The chapter ends with the explanation of the whole complex queuing model.

### Chapter 6: Results

The main issue of this chapter is to show performance influencing parameters as well as to give a feeling of the grade of service

of virtual circuits expressed in maximum throughput of data as well as transfer time.

The chapter yields recommendations on how to choose parameters to reach optimal and best performance results.

The chapter ends with a comparison between the results introduced in this chapter and real measurements obtained in the DATEX-P network of the Deutsche Bundespost.

### Chapter 7: Conclusion and Outlook

Concluding remarks are given and a critical outlook of the proposed procedure will close up this report.

INHALTSVERZEICHNIS

ABKÜRZUNGSVERZEICHNIS	5
1. EINLEITUNG	10
1.1 Allgemeines über Kommunikationssysteme und Datenübertragungsprotokolle	10
1.2 Problemstellung und Überblick über die Arbeit	11
2. THEORETISCHE GRUNDLAGEN	13
2.1 Vermittlungsverfahren in Datennetzen	13
2.1.1 Durchschaltevermittlung	14
2.1.2 Teilstreckenvermittlung	15
2.2 Verbindungskonzepte in Paketvermittlungsnetzen	16
2.2.1 Virtuelle Verbindung	17
2.2.2 Datagram-Verbindung	19
2.3 Datenflußsteuerungskonzepte in Paketvermittlungsnetzen	21
2.3.1 Handshaking-Verfahren	21
2.3.2 Fenstermechanismen	22
2.4 Architektur und Protokolle offener Kommunikationssysteme	28
2.4.1 Allgemeine Architekturprinzipien	28
2.4.2 Kommunikationsprotokolle	29
2.4.3 ISO-Referenzmodell	30
2.5 Verkehrsanalyse	32
2.5.1 Modellbildung	32
2.5.1.1 Strukturen	33
2.5.1.2 Parameter	37
2.5.2 Simulative Analyse	39
2.5.3 Mathematische Analyse	43
2.5.3.1 Produktformlösung	44
2.5.3.2 Mittelwertanalyse	47
2.5.3.3 Dekomposition	55

3. DIE CCITT-NETZZUGANGSSCHNITTSTELLE X.25	60
3.1 Allgemeines	61
3.2 Schichtenstrukturierung und Funktionen	63
3.2.1 Logische Struktur und Protokollschichtung	63
3.2.2 Bitübertragungsschicht	66
3.2.3 Sicherungsschicht	67
3.2.4 Paketschicht	72
3.2.4.1 Leistungsumfang	72
3.2.4.2 Virtuelle Verbindung und logischer Kanal	73
3.2.4.3 Paketformate und -funktionen	74
3.2.4.4 Datentransfer	79
3.3 Parameter	82
3.3.1 Struktur- und Verkehrsparameter	82
3.3.2 Optionale Benutzer- und Leistungsmerkmale	82
4. IMPLEMENTIERUNG DER CCITT-NETZZUGANGSSCHNITTSTELLE X.25	84
4.1 Prozeßkonzept	85
4.2 Dienstprotokoll	87
4.3 Implementierungsstruktur	89
4.3.1 Hardware	89
4.3.2 Software	95
5. LEISTUNGSANALYSE DER PAKETSCHICHT VON X.25	100
5.1 Modellbildung der Schnittstelle	101
5.2 Methoden zur Leistungsuntersuchung	105
5.2.1 Simulationsmethode	105
5.2.1.1 Vereinfachende Annahmen	106
5.2.1.2 Hauptmechanismen an der Schnittstelle	107
5.2.1.3 Schnittstelle zur Umwelt	110
5.2.2 Analytische Methode	110
5.2.2.1 Dekomposition und Aggregation	112
5.2.2.2 Analyse des Flußkontrollmechanismus	116
5.2.2.3 Analyse des Netzwerkeinflusses	122
5.2.2.4 Zusammensetzung	129

5.3 Geltungsbereich der Analysemethoden	134
5.3.1 Simulative Analyse	135
5.3.2 Mathematische Analyse	135
6. ERGEBNISSE DER LEISTUNGSFAHIGKEIT	137
6.1 Leistungsbeeinflussende Parameter	137
6.2 Durchsatz-Untersuchungen virtueller Verbindungen	138
6.3 Transferzeit-Untersuchungen virtueller Verbindungen	147
6.4 Einstellung von Parametern	171
6.5 Messungen im DATEX-P Netz der Deutschen Bundespost	176
7. ZUSAMMENFASSUNG, AUSBLICK UND KRITIK	178
LITERATURVERZEICHNIS	181
LEBENS LAUF	194

ABKÜRZUNGSVERZEICHNIS

A	Angebot
$a_{VC}$	Mittlerer Nachrichtenblockankunftsabstand pro virtueller Verbindung
$A(t)$	VF des zufallsabhängigen Ankunftsabstandes $T_A$
ABM	Asynchronous Balanced Mode
ACIA	Asynchronous Communications Interface Adapter
ADLC	Advanced Data Link Control
ARM	Asynchronous Response Mode
BE	Bedieneinheit
c	Variationskoeffizient
$C_{iter}$	Komplexitätsgrad, iteratives Verfahren
$C_{rek}$	Komplexitätsgrad, rekursives Verfahren
CCITT	Comité Consultatif International Téléphonique et Télégraphique
CPU	Central Processing Unit
D	Durchsatz
$D_n$	Normierter Durchsatz
$D_{n,gr}$	Normierter Grenzdurchsatz
DCE	Data Circuit Terminating Equipment
DEE	Datenendeinrichtung
DK	Durchsatzklasse
DTE	Data Terminal Equipment
DOE	Datenübertragungseinrichtung
E	Empfangsfolgenummer
$E^*$	Zuletzt empfangene Empfangsfolgenummer
$E_0$	Höchste Folgenummer, die vom Empfänger nicht mehr akzeptiert wird
$E_u$	Niedrigste Folgenummer, die vom Empfänger akzeptiert wird
$\mathcal{E}_i^r$	Verteilwahrscheinlichkeit
$E[NBL]$	Mittelwert der (zufallsabhängigen) Nachrichtenblocklänge
$E[T]$	Mittelwert der allgemeinen Zufallsvariablen T
$E[T_A]$	Mittelwert des (zufallsabhängigen) Ankunftsabstandes $T_A$
$E[T_F]$	Mittelwert der (zufallsabhängigen) Transferzeit $T_F$

$E[T_H]$	Mittelwert der (zufallsabhängigen) Bediendauer $T_H$
$E[T_T]$	Mittelwert der (zufallsabhängigen) Transitzeit $T_T$ im VCPS-Komplement
$E[T_W]$	Mittelwert der (zufallsabhängigen) Wartezeit $T_W$ im VCSSp
$E[X]$	Mittelwert der allgemeinen Zufallsvariablen X
ESp	Empfangsspeicher
F	Flag
FCS	Frame Check Sequence
FIFO	First-in, first-out
FlSt	Flußsteuerung
G	Allgemeiner Bedienprozeß
$\delta$	Durchsatzrate
$\delta^r$	Durchsatz (Enderate) der Kette r
$\delta_i$	Durchsatz (Enderate) der Station i
$\delta_i^*, \delta_i^*$	Durchsatz (Enderate) der markierten Station $i^*$
$\delta(k)$	Zustandsabhängiger Durchsatz (Enderate)
$\delta(k,y)$	Zustandsabhängiger Durchsatz (Enderate)
$\delta(k,y,z)$	Zustandsabhängiger Durchsatz (Enderate)
$g(y)$	Gewichtungsfaktor
GI	Allgemeiner Ankunftsprozeß
$h_i$	Mittlere Bediendauer in der BE von Station i
$h_i^r$	Mittlere Bediendauer in der BE von Station i für Nachrichten der Kette r
$\hat{h}_i^r$	Korrigierte mittlere Bediendauer in der BE von Station i für Nachrichten der Kette r
$H(t)$	VF der zufallsabhängigen Bediendauer $T_H$
HDLC	High Level Data Link Control
i	Tiefindex für Größen, die sich auf Station i beziehen
$i^*$	Willkürlich markierte Station
ISO	International Organization for Standardization
j	Anzahl Quittungen im QSp
$\bar{j}$	Mittlere Anzahl Quittungen im QSp
k	Anzahl Pakete im VCPS-Komplement
$\bar{k}$	Mittlere Anzahl Pakete im VCPS-Komplement

K	Populationsgröße
$K^r$	Populationsgröße in Kette r
IK	Populationsvektor
$IK-e^r$	Populationsvektor mit einer aus Kette r entnommenen Nachricht
KN	Koppelnetz
$\lambda$	Ankunftsrate
$\lambda_{VC}$	Ankunftsrate pro virtueller Verbindung
$\lambda(w)$	Zustandsabhängige Ankunftsrate
LAPB	Link Access Procedure, Balanced Mode of Operation
LAPB-ESp	LAPB-Empfangsspeicher (Schicht-2-Empfangsspeicher)
LAPB-SSp	LAPB-Sendespeicher (Schicht-2-Sendespeicher)
LIFO	Last-in, first-out
M	MODULUS-Wert
$\mu_i$	Bedienrate der Station i
$\mu_i^r$	Bedienrate der Station i für Nachrichten der Kette r
$\hat{\mu}_i^r$	Korrigierte Bedienrate der Station i für Nachrichten der Kette r
$\mu^*(k)$	Zustandsabhängige Bedienrate der Komplementärstation
MDM	Multiplexer/Demultiplexer
modM	Modulo-M-Rechnung
N	Anzahl Stationen
n	Anzahl virtuelle Verbindungen
$n_i^r$	Mittlere Anzahl Nachrichten der Kette r in Station i
$n_i(k)$	Mittlere (zustandsabhängige) WS-Länge in Station i
N(R)	Rahmen-Empfangsfolgennummer (Frame Receive Sequence Number)
N(S)	Rahmen-Sendefolgennummer (Frame Send Sequence Number)
NBL	Nachrichtenblocklänge
NK	Netzknoten
NRM	Normal Response Mode
OSI	Open System Interconnection
OVHD	Overhead (Steuerinformation)
P	Nichtunterbrechende Priorität

P(R)	Paket-Empfangsfolgennummer (Packet Receive Sequence Number)
P(S)	Paket-Sendefolgennummer (Packet Send Sequence Number)
p(x)	Zustandswahrscheinlichkeit
$P\{T \leq t\}$	Mathematische Beschreibung der VF der Zufallsvariablen T
$P\{T_A \leq t\}$	VF des zufallsabhängigen Ankunftsabstandes $T_A$
$P\{T_H \leq t\}$	VF der zufallsabhängigen Bediendauer $T_H$
PIA	Parallel Interface Adapter
PR	Preemptive-Resume
PS	Processor Sharing
Q	Verzweigungsmatrix
$q_{ij}$	Verzweigungswahrscheinlichkeit
$Q(r)$	Menge derjenigen Stationen, welche der Kette r angehören
$ Q(r) $	Mächtigkeit von $Q(r)$
QSp	Quittungsspeicher
R	Anzahl Ketten
r	Hochindex für Größen, die sich auf Kette r beziehen
$g_i$	Verkehrsintensität der Station i
$g_i^r$	Verkehrsintensität der Kette r in Station i
$R(i)$	Menge derjenigen Ketten, welche über Station i verlaufen
REJ	Reject
RNR	Receive Not Ready
RR	Receive Ready
S	Sendefolgennummer
$S_o$	Oberer Fensterrand
$S_u$	Unterer Fensterrand
$S(IK-e^r)$	Systemzustand mit Populationsvektor $IK-e^r$
SI	Steuerinformation
Sp	Speicher
SREJ	Selective Reject
SSp	Sendespeicher
$T_A$	Ankunftsabstand
$T_E$	Empfangsdauer
$T_F$	Transferzeit
$T_H$	Bediendauer



$t_i(k)$	Zustandsabhängige Aufenthaltsdauer in Station $i$
$t_{i^*,i}^r$	Mittlere Aufenthaltsdauer einer Nachricht der Kette $r$ in Station $i$ zwischen zwei aufeinanderfolgenden Besuchen bei der willkürlich gewählten Station $i^*$ dieser Kette $r$
$t_{i^*,i}$	Mittlere Aufenthaltsdauer einer Nachricht in Station $i$ zwischen zwei aufeinanderfolgenden Besuchen bei einer willkürlich gewählten Station $i^*$
$T_L$	Laufzeit (inklusive Bearbeitungszeiten in Schicht 1 und 2)
$T_S$	Sendedauer
$T_T$	Transitzeit im VCPS-Komplement
$T_0$	Übertragungszeit
$T_{ODP}$	Datenpaketübertragungszeit
$T_{OSP}$	Steuerpaketübertragungszeit
$T_W$	Wartezeit im VCSSp
$\theta$	Vektor
$\theta_i$	Mittlere Anzahl Besuche einer Nachricht bei Station $i$
$\theta_i^r$	Mittlere Anzahl Besuche einer Nachricht der Kette $r$ bei Station $i$ zwischen zwei aufeinanderfolgenden Besuchen bei einer willkürlich gewählten Station dieser Kette
$v$	Übertragungsgeschwindigkeit
$V(E)$	Empfangsfolgezähler
$V(S)$	Sendefolgezähler
$VC$	Virtual Circuit (virtuelle Verbindung)
$VCESp$	VC-Empfangsspeicher
$VCSSp$	VC-Sendespeicher
$VCPS$	VC-Protokollsteuerung
$VDX$	Vollduplex
$VF$	Verteilungsfunktion
$W$	Anzahl Warteplätze im VCSSp
$w$	Anzahl wartender Nachrichtenblöcke im VCSSp
$\bar{w}$	Mittlere Anzahl wartender Nachrichtenblöcke im VCSSp
$w_E$	Empfangsfenstergröße
$w_S$	Sendefenstergröße
$WS$	Warteschlange
$X$	Zustandsvariable

## 1. EINLEITUNG

### 1.1 Allgemeines über Kommunikationssysteme und Datenübertragungsprotokolle

Die Fähigkeit der modernen Telekommunikation, auf schnelle Art und Weise Daten zwischen wechselnden Kommunikationspartnern auszutauschen, hat sich in den vergangenen Jahren in der Rationalisierung und Automatisierung von betrieblichen Arbeitsvorgängen ausgewirkt.

Die rasche Entwicklung und die Produktion von mikroelektronischen Komponenten in großen Stückzahlen, und damit der preisgünstige Einsatz auf dem Gebiet der Datenverarbeitung, führen zu einem weiter wachsenden Bedürfnis an Datenkommunikation. Diesem zunehmenden Nachrichtenaustausch wird dadurch Rechnung getragen, daß weltweit neue Datennetze mit unterschiedlichen Kommunikationsformen geplant, aufgebaut und betrieben werden. Derartige Kommunikationsformen können sein, (vgl. /1/):

- Individualkommunikation, Breitenkommunikation
- Sammeln, Verteilen von Information
- wechselseitige, einseitige Kommunikation.

Es leuchtet ein, daß derart verschiedene Kommunikationsformen auch unterschiedliche Aufgaben an Kommunikationssysteme stellen. Hauptaufgabe aller Kommunikationssysteme ist jedoch der gesicherte und möglichst schnelle Transport von Nachrichten zwischen beliebigen Kommunikationspartnern.

Um sowohl schwankende Verkehrslasten auszugleichen als auch zufällig auftretende Fehler zu erkennen und zu beheben, bedarf es innerhalb des Kommunikationssystems einer Vielzahl von Regeln, um den geforderten zuverlässigen und schnellen Informationsaustausch zu gewährleisten. Die Gesamtheit solcher Regeln wird als Protokoll bezeichnet. Um die vielfältigen Aufgaben der Kommunikationspartner, die unterschiedlichen Kommunikationsformen, die Eigenschaften der Übertragungswege sowie die verschiedensten Realisierungsformen

der Kommunikationspartner zu beherrschen, wird die Menge der Regeln insofern hierarchisch strukturiert, daß sog. Protokollschichten definiert werden. Dieses von den internationalen Standardisierungsgremien ISO und CCITT entwickelte Architekturmodell legt fest, welche Funktionen in welchem Zusammenhang in einer Kommunikation auftreten und stellt damit den Rahmen für die Erarbeitung neuer bzw. Einordnung bestehender Normen auf, die nötig sind, um das Ziel - offene Kommunikationssysteme - zu erreichen. Unter offenen Kommunikationssystemen werden Systeme verstanden, deren Schnittstellen und Prozeduren so gegliedert und standardisiert sind, daß beliebige Kommunikationspartner in Verbindung treten können.

## 1.2 Problemstellung und Überblick über die Arbeit

Es ist offensichtlich, daß sowohl beim Entwurf von Datenübertragungsprotokollen als auch bei bereits bestehenden Normen eine Fülle theoretischer und praktischer Probleme zu lösen ist.

Die vorliegende Arbeit beschäftigt sich mit Fragestellungen der Leistungsbewertung von Datenübertragungsprotokollen, im speziellen mit der Bestimmung der Leistungsfähigkeit virtueller Datenverbindungen auf einem entsprechend der CCITT-Empfehlung X.25, /2/, gesteuerten Übertragungsabschnitt.

Ziel der Arbeit ist, ein realistisches Verkehrsmodell, welches X.25 detailliert nachbildet, zu entwickeln und zu analysieren, um ein Hilfsmittel für die praktische und schnelle Bemessung und zur Identifizierung leistungsbeeinflussender Parameter eines bestehenden Datenübertragungsprotokolls bereitzustellen. Insbesondere soll der Einfluß dieser Parameter auf Gütekriterien einer Datenverbindung, wie Durchsatz von übertragenen Datenpaketen und Transferzeit, quantitativ ermittelt werden.

Die Arbeit soll weiterhin Anhaltspunkte liefern, zukünftige oder auch bestehende, ähnlich gelagerte Normen (z.B. CCITT-Empfehlung X.75 /3/, Pacing Mechanismus /4/) mittels der aufgezeigten Methode zu untersuchen.

Kapitel 2 "Theoretische Grundlagen" gibt einen Überblick über Vermittlungsverfahren in Datennetzen und Verbindungsprinzipien in paketvermittelnden Netzen. Der Vorstellung und Erläuterung von Architektur und Protokollen offener Kommunikationssysteme folgen abschließend Grundprinzipien der Verkehrsanalyse.

In Kapitel 3 "Die CCITT-Netzzugangsschnittstelle X.25" werden Strukturierung, Funktionen, spezielle Mechanismen und leistungsbeeinflussende Parameter von X.25 erläutert.

Kapitel 4 "Implementierung der CCITT-Netzzugangsschnittstelle X.25" befaßt sich mit Realisierungsaspekten, wie Prozeßkonzept, Dienstprotokoll und Implementierungsstrukturen.

In Kapitel 5 "Leistungsanalyse der Paketschicht von X.25" wird ein detailliertes Warteschlangenmodell der Schnittstelle vorgestellt, auf dessen Basis sowohl die simulative als auch die mathematische Analyse von X.25 eingehend erläutert werden.

Kapitel 6 "Ergebnisse der Leistungsfähigkeit" zeigt leistungsbeeinflussende Parameter auf und gibt Aufschluß über die Dienstgüte virtueller Verbindungen bezüglich Durchsatz von Datenpaketen und Transferzeit. Das Kapitel schließt mit einem Vergleich zwischen Ergebnissen der vorliegenden Arbeit und Messungen im DATEX-P Netz der Deutschen Bundespost.

Kapitel 7 enthält sowohl eine Zusammenfassung als auch einen kritischen Ausblick der Arbeit.

## 2. THEORETISCHE GRUNDLAGEN

Dieses Kapitel soll einen Abriß über theoretische Grundlagen liefern, soweit sie für die Untersuchungen dieser Arbeit von Bedeutung sind, bezüglich Vermittlungsverfahren in Datennetzen, Verbindungsprinzipien in paketvermittelnden Netzen, Architektur und Protokollen offener Kommunikationssysteme sowie bezüglich simulativer und mathematischer Verkehrsanalyse.

### 2.1 Vermittlungsverfahren in Datennetzen

Voraussetzung für den Austausch von Daten zwischen beliebigen Stationen eines Datennetzes ist die dauernd oder nur für eine bestimmte Zeit bestehende Verbindung zwischen diesen beiden Kommunikationspartnern. Im einfachsten Fall steht den Stationen ständig ein Übertragungsweg zur Verfügung. Derartige Verbindungen werden als Standverbindung bezeichnet. Demgegenüber finden Vermittlungsvorgänge statt, wenn Verbindungen auf- und abgebaut werden müssen. Heutzutage realisierte Verfahren zur Vermittlung von Daten sind in Bild 2.1 aufgeführt und werden im folgenden kurz charakterisiert, /5,6/.

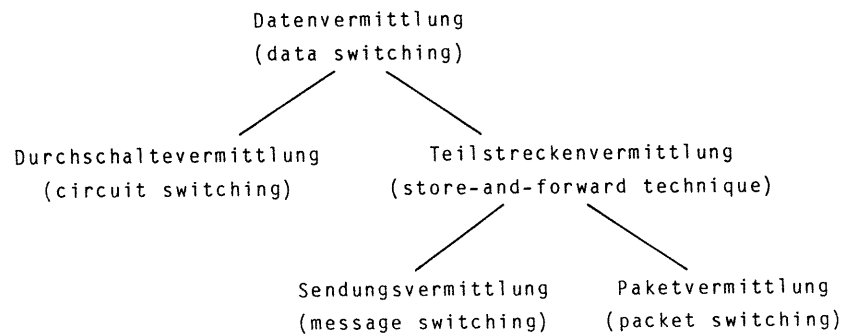


Bild 2.1: Vermittlungsverfahren in Datennetzen

### 2.1.1 Durchschaltevermittlung

Bei diesem Vermittlungsverfahren steht ein durchgehender, während einer Verbindungsaufbauphase aufgebaute Verbindungsweg (Kanal) mit konstanter Bandbreite ausschließlich den am Dialog beteiligten Kommunikationspartnern zur Verfügung (siehe Bild 2.2). Durchgehend bedeutet, daß außer einer kurzzeitigen Zwischenspeicherung zwecks eventuell erforderlichem Zeitkanalumstieg keinerlei Nachrichtenspeicher im Netz zwischengeschaltet werden. Während der Übertragungsphase werden die Daten im Prinzip ohne Verzögerung und ohne weitere Bearbeitung weitergeleitet. Die Durchschaltung wird abschließend in einer Verbindungsabbauphase aufgrund eines speziellen Auslösesignals abgebaut.

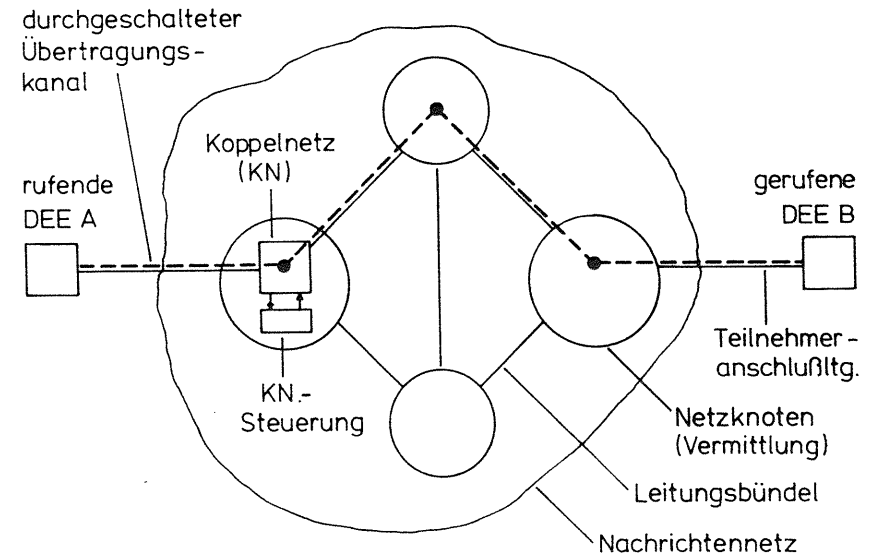


Bild 2.2: Durchschaltevermittelte Verbindung

### 2.1.2 Teilstreckenvermittlung

Beim Betrieb mit Teilstreckenvermittlung steht den Kommunikationspartnern kein durchgehender, exklusiver Verbindungsweg zur Verfügung. Bei diesem Vermittlungsverfahren werden die Nachrichten nur über Teilstrecken übertragen: von der sendenden Station zu einer Vermittlungsstelle, von dort evtl. zu einer anderen und schließlich zur empfangenden Station. Da die Daten nur dann weitergeleitet werden, wenn der nächste Teilstreckenabschnitt zur Verfügung steht, ist eine Zwischenspeicherung in den Vermittlungsknoten unumgänglich (Speichervermittlung, store-and-forward switching), siehe Bild 2.3.

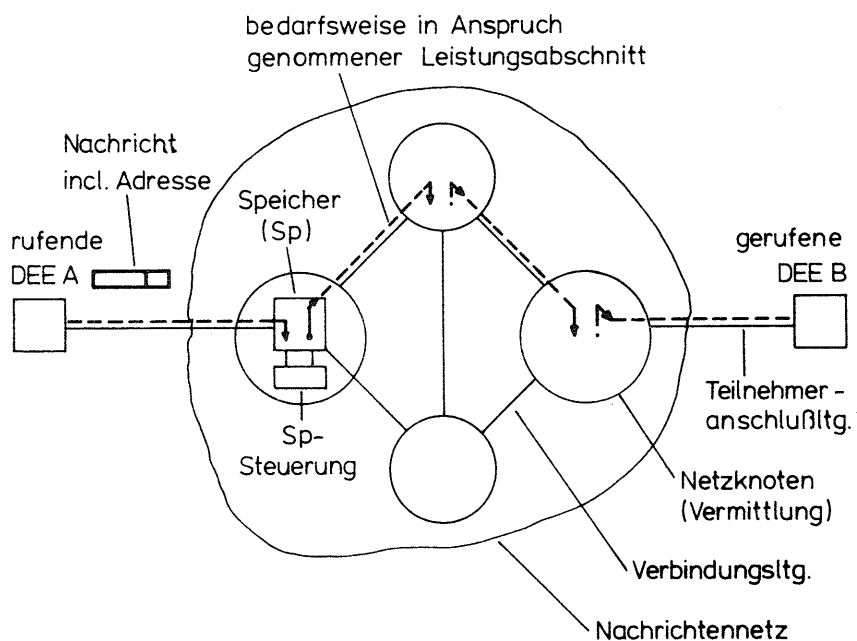


Bild 2.3: Teilstreckenvermittelte Verbindung

Bei Teilstreckenvermittlung existiert eine Verbindung in Form einer Adreßzuordnung zwischen rufender und gerufener Datenendrichtung, welche adressierte Nachrichten über das Netz austauschen. Abhängig davon, ob die Daten der Kommunikationspartner im Ganzen durch das Netz transportiert werden oder aufgeteilt in Datenblöcke bestimmter (maximaler) Größe, spricht man von Sendungsvermittlung (message switching) oder Paketvermittlung (packet switching).

Sendungsvermittlung: Charakteristisches Merkmal von Sendungsvermittlungsnetzen ist die Verwendung relativ langsamer Massenspeicher zur Aufnahme der gesamten Nachrichten. Hierin ist auch der Grund zu suchen, daß derartige Netze für Dialogverkehr weniger geeignet sind.

Paketvermittlung: Paketvermittelnde Netze können dagegen durch geeignete Wahl der Paketgrößen dahingehend ausgelegt werden, daß mit ihnen Transferzeiten entstehen, die weit unter einer Sekunde liegen und sich somit für einen Dialogbetrieb eignen. Ein weiteres wichtiges Merkmal von Paketvermittlungsnetzen ist schließlich, daß auf einfache Weise ein Datenaustausch zwischen Stationen ermöglicht werden kann, die nicht in Code und/oder Übertragungsgeschwindigkeit übereinstimmen.

### 2.2 Verbindungskonzepte in Paketvermittlungsnetzen

Charakteristisches Merkmal paketvermittelnder Nachrichtennetze ist, daß Leitungsabschnitte im Netz nur während der reinen Übertragung des Paketes belegt werden. Dabei kommen in der Regel zwei Verbindungskonzepte zur Anwendung: virtuelle Verbindungen oder Datagrammverbindungen, //.

2.2.1 Virtuelle Verbindung

Unter dem Begriff der virtuellen Verbindung versteht man, daß den am Dialog beteiligten Partnern eine Verbindung zur Verfügung steht, welche bis zu einem gewissen Grad, d.h. "dem Wesen nach" (virtuell), die Eigenschaften real durchgeschalteter Verbindungen aufweist. Im einzelnen besitzt eine virtuelle Verbindung folgende Eigenschaften: die fehlergesicherte Übertragung der Daten und die Einhaltung der Paketreihenfolge beim Zielteilnehmer, /8/. Bei einer virtuellen Verbindung unterscheidet das Datennetz - wie bei der Durchschaltvermittlung - drei Phasen: Verbindungsaufbau-, Datenübertragungs- und Verbindungsabbauphase, /5,7/.

Verbindungsaufbauphase: In dieser Phase wird mittels Steuerpaketen die virtuelle Verbindung zwischen rufendem und gerufenem Teilnehmer aufgebaut und somit der Weg durch das Netz für die Dauer der Verbindung festgelegt. Die Wahl des (zum Aufbauzeitpunkt) günstigsten Weges erfolgt durch Verkehrslenkungsalgorithmen (routing) ebenfalls innerhalb der Verbindungsaufbauphase.

Durch die in der Regel übliche Anwendung eines sog. Adreßmultiplexverfahrens, d.h. eines dynamischen (asynchronen) Multiplexens der zu unterschiedlichen Verbindungen gehörenden Nachrichten, werden die physikalischen Verbindungsleitungen wesentlich besser ausgenutzt und damit letztendlich auch wirtschaftlicher. Dabei wird jeder Verbindung auf jedem Übertragungsabschnitt eine (freie) logische Kanalnummer zugeordnet.

Die gesamte virtuelle Verbindung ist dann festgelegt durch die Verknüpfung der einzelnen logischen Kanalnummern, welche in Steuerdatenbereichen in den an dieser Verbindung beteiligten Netzknoten abgelegt ist (siehe Bild 2.4).

Während der Verbindungsaufbauphase legen die Ursprungs- und Zielnetz-knoten je virtueller Verbindung zusätzlich Datenspeicherbereiche an zur Pufferung von Paketen während der Übertragungsphase (Bild 2.4).

von	nach	log. Kanal	Verknüpfung. in	von log. Kanal	nach log. Kanal
DEE A	DOE A	3	DOE A	3	7
DOE A	NK 2	7	NK 2	7	11
NK 2	DOE B	11	DOE B	11	9
DOE B	DEE B	9			

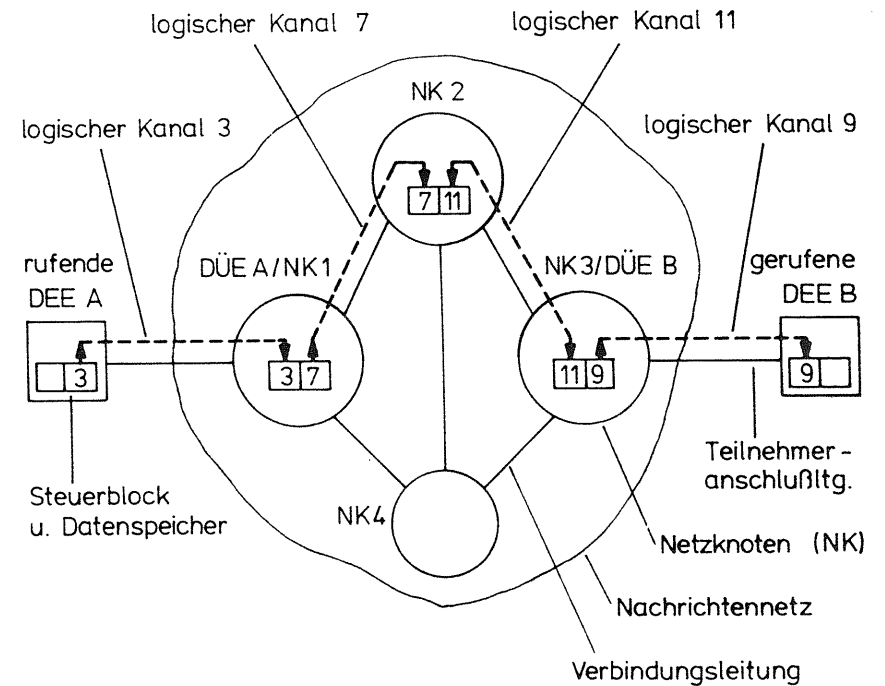


Bild 2.4: Virtuelle Verbindung

Innerhalb des reinen Vermittlungsnetzes kann eine virtuelle Verbindung durch Reservierung von Speicherplätzen weiter unterstützt werden.

Datenübertragungsphase: In dieser Phase werden die Pakete der einzelnen Verbindungen entlang des virtuellen Verbindungsweges übertragen, wobei die Vermittlung in den beteiligten Netzknoten durch Zugriff auf die in den Steuerblöcken abgelegten Verknüpfungsbeziehungen erfolgt.

Weitere Aufgaben beim virtuellen Verbindungskonzept innerhalb der Übertragungsphase sind:

- Einhaltung der Paketreihenfolge
- Fehlererkennung und -korrektur
- Datenflußsteuerung.

Verbindungsabbauphase: Nach Beendigung der Datenübertragungsphase wird die virtuelle Verbindung durch Aussenden eines speziellen Steuerpaketes abgebaut. Dies wird durch Löschen der Verknüpfungsbeziehung in den Steuerblöcken realisiert sowie dadurch, daß vorgenommene Reservierungen von Datenspeicherbereichen wieder rückgängig gemacht werden.

### 2.2.2 Datagram-Verbindung

Hauptunterschied zwischen dem Datagram-Verbindungskonzept und dem der virtuellen Verbindung ist, daß es keine Verbindungsaufbau- und -abbauphasen gibt. Pakete werden bei diesem Verfahren völlig autonom vom Netz behandelt. Das paketvermittelnde Netz überträgt jedes von den Kommunikationspartnern abgesandte Paket individuell und unmittelbar aufgrund der im Paketkopf enthaltenen Steuerinformation (siehe Bild 2.5), /5,7/.

Die autonome Übermittlung jedes Paketes durch das Netz erfordert beim Datagram-Verbindungskonzept folgende Hauptfunktionen, /7/:

- Verkehrslenkung je Paket nach momentanen Belastungssituationen
- vollständige Auswertung der Paketkopfinformation in jedem Knoten

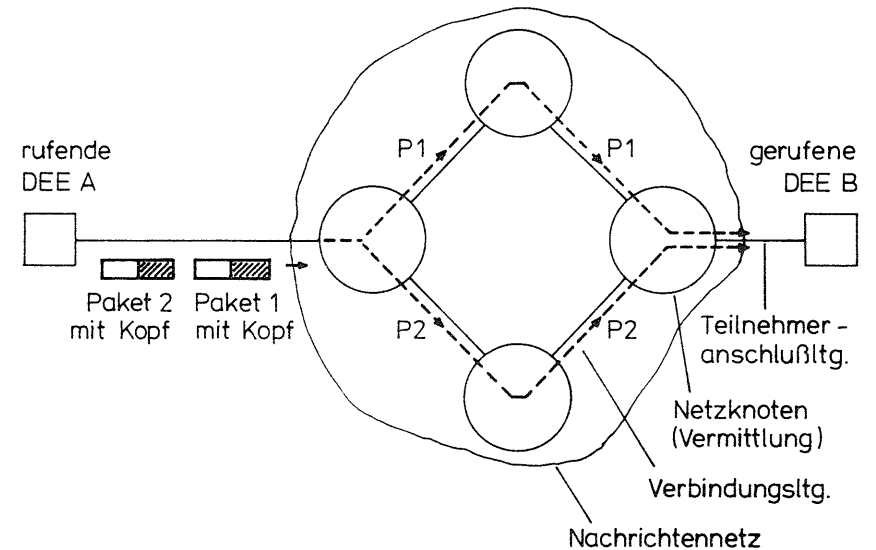


Bild 2.5: Datagram-Verbindung

- Anforderung fehlender Pakete durch Zielteilnehmer
- Anforderung fehlerhafter Pakete durch Zielteilnehmer.

Durch die paketindividuelle Verkehrslenkung sind unterschiedliche Wege der einzelnen Pakete durch das Netz durchaus möglich, so daß die Zielstation zusätzlich auch noch für die richtige Reihenfolge sorgen muß. Desweiteren ist ersichtlich, daß bei einer Datagram-Verbindung eine Datenflußsteuerung innerhalb des Netzes nicht vorgesehen ist; diese Funktion obliegt den beteiligten Datenendrichtungen.

### 2.3 Datenflußsteuerungskonzepte in Paketvermittlungsnetzen

Verkehrslenkungsalgorithmen dienen dazu, stark ausgelastete oder gar bereits überlastete Verbindungsleitungsabschnitte zu umgehen. Sie sind jedoch nicht imstande, durch momentan starkes Überangebot eingetretene Überlastsituationen zu verhindern. Aus diesem Grund sind in Paketvermittlungsnetzen Mechanismen vorgesehen, die sowohl den Zustrom an Daten als auch den Datenfluß innerhalb des Netzes regulieren können. Derartige Mechanismen werden auch als Datenflußsteuerung (flow control) bezeichnet, /7,8,9,10/.

Die Datenflußsteuerung hat zur Aufgabe, für einen geordneten Datenfluß innerhalb der einzelnen Verbindungen zu sorgen. Zusammen mit dieser Aufgabe werden auch Funktionen wie Reihenfolgesteuerung, Fehlererkennung und -behebung erfüllt, da hierzu dieselben Flußsteuergrößen und -parameter verwendet werden können.

Häufig angewandte Verfahren sind:

- Handshaking-Verfahren
- Fenstermechanismen.

#### 2.3.1 Handshaking-Verfahren

Beim Handshaking-Verfahren (stop-and-wait), dem einfachsten Mechanismus zur Datenflußregelung, synchronisieren sich Sender und Empfänger nach jedem Übermittlungsvorgang eines Paketes dadurch neu, daß unmittelbar nach Empfang eines Paketes ein Quittiersignal in Gegenrichtung gesendet wird.

Dieses Verfahren hat eine sehr einfache Steuerung und die automatische Einhaltung der Paketreihenfolge als Vorteile aufzuweisen. Demgegenüber steht jedoch als Nachteil eine sehr geringe Auslastung von Verbindungsleitungsabschnitten mit großen Ausbreitungsverzögerungen.

Zur Vermeidung von Verklemmungssituationen (deadlock), z.B. Warten auf Quittiersignal, wobei jedoch dieses Signal während der Übertragung gestört wurde und deswegen vom Empfänger nicht mehr als solches erkannt wird, bedarf es weiterer Mechanismen, die die zunächst einfache Steuerung zwangsläufig aufwendiger machen. Aus diesem Grunde wird dieses Verfahren i.a. nur auf kürzesten Übertragungsabschnitten angewandt wie z.B. innerhalb von Rechnerbaugruppen.

#### 2.3.2 Fenstermechanismen

Ein weiteres, häufig angewandtes Verfahren zur Datenflußsteuerung ist der Fenstermechanismus (sliding window protocol), bei dem von der Quittierung von Einzelpaketen abgegangen wird. Bei diesem Verfahren kann die sendende Station in ununterbrochener Reihenfolge eine fest vorgegebene Anzahl, die sog. Fenstergröße (window size), an Paketen absenden, um danach erst auf eine Quittung der Gegenstation zu warten, die ihr die ausdrückliche Erlaubnis zum Weitersenden erteilt. Der Sender überträgt also bereits nachfolgende Pakete, obwohl Quittungen für frühere Pakete noch ausstehen. Mit diesem Verfahren kann der Durchsatz auf längeren Übertragungsstrecken erhöht werden, der Verkehrsfluß läßt sich im Netz durch Änderung der Fenstergröße individuell regulieren und es kann damit sog. Quittierungsbelastung eingespart werden, indem mehrere richtig empfangene Pakete mit nur einer Quittung bestätigt werden.

Zur Erzielung einer Datenflußsteuerung mittels eines Fenstermechanismus kommen zwei Grundprinzipien in Betracht:

- Folgenummernsteuerung
- Pacing Verfahren, /7/.

Folgenummernsteuerung: Datenpakete werden fortlaufend durch eine Sendefolgenummer S (send sequence number) numeriert, wobei sich der Wertebereich von 0 bis MODULUS-1 erstreckt, mit MODULUS-Werten von 8 bzw. 128 bei erweiterter Zählweise. Somit kann die Sendefolgenummer S zyklisch die Werte 0...7 (bzw. 0...127) annehmen.

Die von einer Station gesendeten Datenpakete werden von der Gegenstation durch Übermitteln einer Empfangsfolgenummer E (receive sequence number) quittiert und somit der ordnungsgemäße und fehlerfreie Empfang bestätigt. Durch die Empfangsfolgenummer E werden alle Pakete bis  $S = E-1$  quittiert und gleichzeitig das nächste erwartete Datenpaket mit Sendefolgenummer  $S = E$  angegeben.

Zur Realisierung der Folgenummernsteuerung sind in jeder Station sowohl Sendefolgezähler  $V(S)$  (send state variable) und Empfangsfolgezähler  $V(E)$  (receive state variable) als auch Sende- und Empfangsfenstermechanismen realisiert.

Die Funktionsweise der beiden Mechanismen soll im folgenden kurz erläutert werden, //:

Sendefenstermechanismus: Hauptfunktion des Sendefenstermechanismus ist das geordnete und kontrollierte Aussenden von Datenpaketen.

Vereinbarungen:

- S Sendefolgenummer des zu sendenden Paketes
- $S_u$  niedrigste Folgenummer der gesendeten, nichtquittierten Pakete (= unterer Fensterrand)
- $E^*$  zuletzt empfangene Empfangsfolgenummer E
- $V(S)$  Folgenummer S des nächsten zu sendenden Paketes
- $S_o$  höchste Folgenummer, welche nicht mehr gesendet werden darf (= oberer Fensterrand)
- $w_s$  Sender-Fenstergröße
- M MODULUS-Wert

Mit o.a. Vereinbarungen lassen sich folgende Beziehungen angeben:

- $S_u = E^*$ , unterer Fensterrand,
- $S_o = (S_u + w_s)_{\text{mod}M}$ , oberer Fensterrand,
- $S_u \leq S < S_o$ , momentaner Gültigkeitsbereich für S
- $(S_o - V(S))_{\text{mod}M}$ , momentane Fenstergröße.

Erreicht  $V(S)$  den Wert des oberen Fensterrandes, d.h. ist die momentane Fenstergröße gleich Null, so stoppt der Sender automatisch den Sendevorgang. Der Empfang einer neuen (und gültigen) Empfangsfolgenummer E bewirkt das neue Festsetzen von  $S_u$ , mit  $S_u := E$ , d.h. das Weiterdrehen des Fensters und somit das Aussenden weiterer Pakete. Jeder Sendevorgang inkrementiert den Sendefolgezähler  $V(S)$ , so daß gilt

$$V(S) := (V(S) + 1)_{\text{mod}M}$$

Der Zusammenhang der einzelnen Größen ist beispielhaft in Bild 2.6 dargestellt.

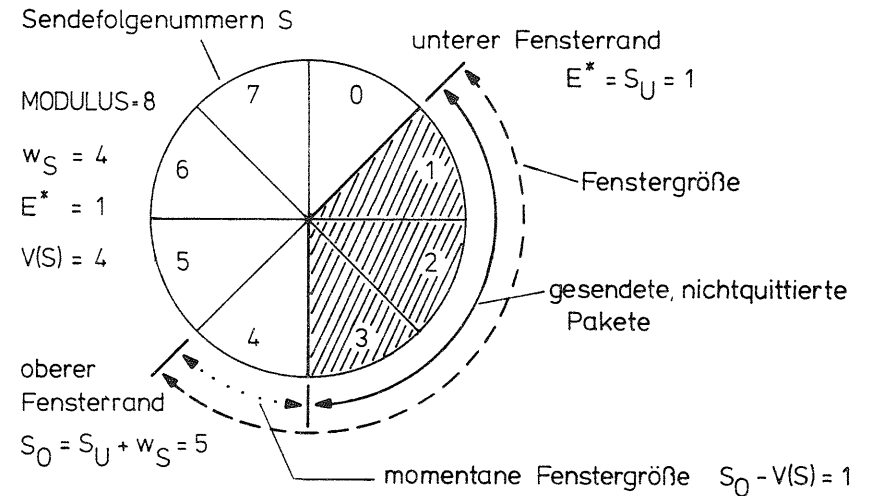


Bild 2.6: Sendefenstermechanismus



Empfangsfenstermechanismus: Dieser Mechanismus dient der Erkennung von Sequenzfehlern von Sendefolgennummern empfangener Pakete.

Vereinbarungen:

- S Sendefolgennummer des empfangenen Paketes
- E Empfangsfolgennummer des zu sendenden Paketes
- $E_U$  niedrigste Folgennummer, die vom Empfänger akzeptiert wird
- $E_O$  höchste Folgennummer, die vom Empfänger nicht mehr akzeptiert wird
- $w_E$  Empfangs-Fenstergröße
- $V(E)$  Folgennummer S des nächsten erwarteten Paketes
- M MODULUS-Wert.

Mit diesen Vereinbarungen gilt:

$$E_U = V(E), \quad \text{unterer Empfangsfensterrand,}$$

$$E_O = (E_U + w_E)_{\text{mod} M}, \quad \text{oberer Empfangsfensterrand.}$$

Pakete mit Sendefolgennummern S, die innerhalb des Empfangsfensters liegen, werden vom Empfänger akzeptiert, d.h.

$$E_U \leq S < E_O.$$

Dagegen werden Pakete verworfen, deren Sendefolgennummern außerhalb des Empfangsfensters liegen, d.h. wenn gilt:

$$S < E_U \quad \text{oder} \quad S \geq E_O.$$

Nach Empfang des Paketes mit der Sendefolgennummer  $S = E_U$  gilt:

$$E_U := (E_U + 1)_{\text{mod} M}$$

$$E_O := (E_O + 1)_{\text{mod} M}.$$

Eine graphische Darstellung des o.a. Sachverhaltes ist in Bild 2.7 ersichtlich.

Folgender in der Regel zur Anwendung kommender Spezialfall läßt sich angeben:

Spezialfall:  $w_E = 1$ ;

in diesem Fall werden vom Empfänger nur Pakete mit Sendefolgennummern in zyklischer Reihenfolge akzeptiert.

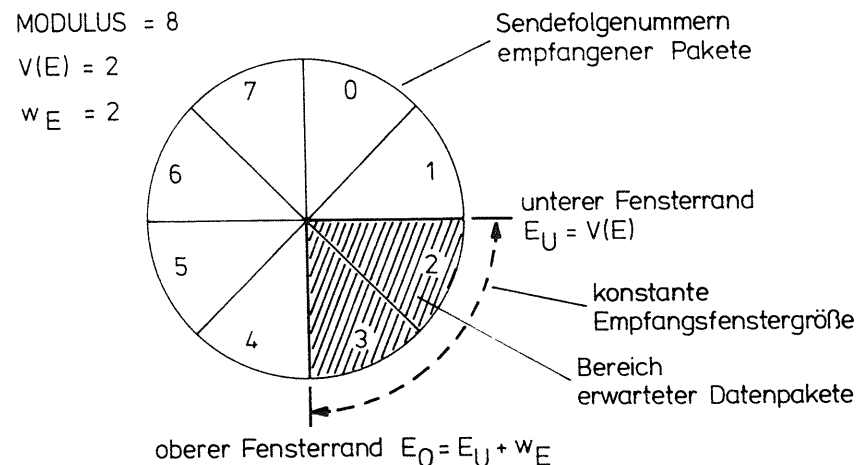


Bild 2.7: Empfangsfenstermechanismus

Zeitliche Zusammenhänge zwischen den Stationen einerseits sowie deren individueller Zählerstände und Folgennummern andererseits sind beispielhaft aus Bild 2.8 zu ersehen.

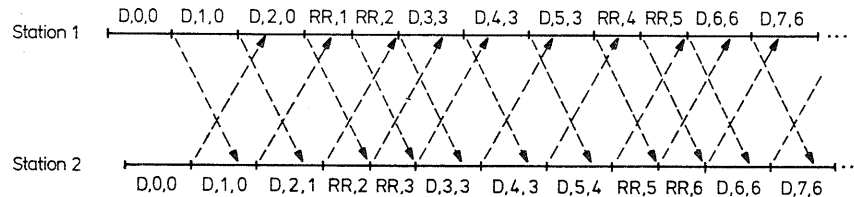
Beim Empfang von Datenpaketen mit ungültiger Empfangsfolgennummer wird in der Regel die Verbindung neu initialisiert, um somit Zählerstände, Folgennummern und Fenstermechanismen auf beiden Seiten der Verbindung auf denselben Ausgangszustand zu bringen. Dagegen bewirkt das Auftreten eines Sequenzfehlers lediglich den Ablauf eines Fehlerkorrekturmechanismus, der im allgemeinen darin besteht, bestimmte Pakete beim Sender anzufordern, der diese dann wiederholt überträgt.

Wie aus den Beschreibungen des Sende- und Empfangsfenstermechanismus als auch aus den Bildern 2.6 und 2.7 unschwer zu erkennen ist, folgt das Handshaking-Verfahren (vgl. Kap.2.3.1) als Spezialfall aus dem allgemeinen Folgennummernsteueralgorithmus durch Festlegung der Sende-Fenstergröße  $w_S$  als auch der Empfangsfenstergröße  $w_E$  auf jeweils den Wert 1.

MODULUS=8,  $w_S=3$ ,  $w_E=1$

$\overbrace{D, S, R}^{\text{Datenpaket mit S- und R-Folgennummer, (Quittung über R-Empfangsfolgennummer)}}$   
 $\overbrace{RR, R}^{\text{Quittier-Steuerpaket mit R-Folgennummer, (soweit kein Datenpaket vorhanden bzw. momentane Fenstergröße gleich Null; hier: Receive Ready)}}$

$S_U$	0	0	0	0	1	2 2	3 3	3	4	5 5	6
$V(S)$	0	1	2	3	3	4 4	5 6	6	6	7 7	0
$V(E)$	0		1	2	3		4	5	6		
$E_O$	1		2	3	4		5	6	7		



$S_U$	0	0	0	0	1	2 2	3 3	3 3	4	5 5	6
$V(S)$	0	1	2	3	3	4 4	5 5	6 6	6	7 7	0
$V(E)$	0		1	2	3		4	5	6		7
$E_O$	1		2	3	4		5	6	7		0

Bild 2.8: Folgennummernsteuerung

Pacing-Verfahren: Ein weiterer, der Folgennummernsteuerung sehr ähnlicher Fenstermechanismus ist das Pacing-Verfahren. Bei diesem Algorithmus hat der Sender zunächst ein "Guthaben" (credits),  $p$  Datenpakete aussenden zu dürfen. Mit jedem übertragenen Paket verringert sich das Guthaben um 1; die Aussendung weiterer Pakete stoppt, wenn das Guthaben aufgebraucht ist. Mit dem Eintreffen jedes ersten der jeweils insgesamt  $p$  aufeinanderfolgenden Pakete beim Empfänger wird gleichzeitig um Vergabe eines neuen Guthabens für weitere  $p$  Pakete ersucht, //.

## 2.4 Architektur und Protokolle offener Kommunikationssysteme

Internationale Normungsarbeiten an offenen Kommunikationssystemen haben zum Ziel, informationsverarbeitende Einrichtungen unterschiedlicher Hersteller und Herkunft in die Lage zu versetzen, miteinander zu kommunizieren und zusammenzuarbeiten.

Diese Fähigkeit erhalten die Kommunikationspartner dadurch, daß ihr nach außen hin erkennbares Verhalten bestimmten Protokollnormen folgt, welche in einer einheitlichen Kommunikationsarchitektur festgelegt sind.

### 2.4.1 Allgemeine Architekturprinzipien

Das grundlegende Architekturprinzip ist das der Schichtung (layering).

Schichtung: Bei diesem Prinzip wird jedes System als aus einem geordneten Satz von Untersystemen zusammengesetzt betrachtet, //1,7, 11-17/. Untersysteme derselben Ordnung bilden schließlich gemeinsam eine Schicht. Untersysteme, und damit auch die Schichten, bestehen wiederum aus einer oder mehreren Instanzen (entity). Das Prinzip der Schichtung bewirkt somit eine Gliederung der Systemfunktionen und dient der Auflösung der Komplexität in überschaubare Einzelkomponenten.

Innerhalb jeder Schicht werden bestimmte Dienstleistungen erbracht und der ihr überlagerten Schicht - mit Ausnahme bei der höchsten Schicht - angeboten.

Dienstleistungen: Die allgemein mit  $N$  bezeichnete Schicht erbringt Schicht- $N$ -spezifische Leistungen, wobei sie auf den Dienstleistungen ihrer darunterliegenden Schicht  $N-1$  aufbaut. Dienstleistungen tieferer Schichten, also  $N-2$ ,  $N-3$ , ..., bleiben dem Zugang aus Schicht  $N$  verborgen.



Schicht 7: Verarbeitungsschicht; in der Verarbeitungsschicht als der höchsten Schicht des Referenzmodells manifestiert sich die Funktion der Informationsverarbeitung in einer Kommunikation. Konkret gesagt, hier erfolgt die Einbettung von Anwendungs- und Verarbeitungsprozessen in die Systemumgebung zum Zugriff auf die OSI-Umgebung (OSI: open system interconnection).

Schicht 6: Darstellungsschicht; hier werden Sprachmittel zur Verfügung gestellt, die es den Instanzen ermöglichen, Begriffe eindeutig zu benennen. Im Darstellungsprotokoll werden Regeln festgelegt, wie die in der gemeinsamen Sprache dargestellte Information auszutauschen ist.

Schicht 5: Kommunikationssteuerungsschicht; diese Schicht stellt Sprachmittel zur Verfügung, die zur Eröffnung einer Kommunikationsbeziehung, auch Sitzung genannt, ihrer geordneten Durchführung und Beendigung notwendig sind.

Schicht 4: Transportschicht; in dieser Schicht erfolgt eine Ende-zu-Ende Steuerung des Nachrichtenaustausches zwischen zwei Datenendeinrichtungen über das gesamte Netz hinweg.

Schicht 3: Vermittlungsschicht; diese Schicht verknüpft gesicherte Verbindungsabschnitte (von den Endsystemen zu den jeweiligen zugehörigen Netzknoten und zwischen den Netzknoten selbst) zu Endsystemverbindungen, d.h. von Endsystem zu Endsystem.

Schicht 2: Sicherungsschicht; hier erfolgt die Übertragungssteuerung auf einem Verbindungsabschnitt für die gesicherte Übertragung von Nachrichten; d.h. diese Schicht setzt ungesicherte Verbindungsabschnitte in gesicherte um.

Schicht 1: Bitübertragungsschicht; in dieser Schicht als der untersten Schicht des Referenzmodells sind die funktionellen, elektrischen und mechanischen Eigenschaften für die Übertragung von Bits festgelegt.

## 2.5 Verkehrsanalyse

Der Sinn dieses Unterkapitels ist die Beschreibung grundlegender Verfahren zur Leistungsbewertung datenflußgesteuerter Verbindungen in paketvermittelnden Netzen. Solche Verfahren können sein:

- die simulative Analyse, ein detailgetreues, aber auch zeitaufwendiges Verfahren,
- die mathematische Analyse, ein meist auf gewissen Vereinfachungen und Approximationen aufbauendes, aber dafür wesentlich schnelleres Berechnungsverfahren.

Der Vorstellung der Verfahren vorangestellt ist die Beschreibung von Verkehrsmodellen, welche alle für eine Leistungsbewertung wesentlichen Bedienungs- und Transportvorgänge in realitätsnaher Weise nachbilden.

### 2.5.1 Modellbildung

Verkehrsmodelle haben die Aufgabe, reale Ablaufgeschehen, wie Bedienungs- und Transportvorgänge in Systemen, realitätsnah mit Hilfe von einfachen und allgemein verständlichen Modellelementen nachzubilden.

Aufbauend auf grundlegenden Überlegungen werden in diesem Abschnitt Modelle entwickelt, welche zur quantitativen Analyse von Datenverbindungen in paketvermittelnden Netzen geeignet sind, /8,18/.

Leistungsbeeinflussende Parameter werden am Ende dieses Abschnittes vorgestellt.

2.5.1.1 Strukturen

Ein Verkehrsmodell weist prinzipiell folgende Strukturkomponenten auf:

- Bedieneinheiten (server): zur Nachbildung von Prozeßdauern für Verarbeitungs- und Übertragungsvorgänge,
- Warteschlangen (queues): zur Zwischenspeicherung von Anforderungen (z.B. Paketen) bei momentanem Engpaß in der Bedieneinheit,
- Netztopologie: zur Verbindung der einzelnen Bedieneinheiten bzw. der Warteschlangen untereinander.

Aufbauend auf diesen allgemeinen Komponenten lassen sich die nachfolgend aufgeführten speziellen Modelle, soweit sie im Rahmen dieser Arbeit von Bedeutung sind, angeben:

Prozessormodelle: Prozessormodelle sind einstufige Bedienungssysteme, die die typischen Merkmale von Realzeitprozessoren aufweisen (verschiedene Bearbeitungsphasen, Prioritätssteuerung, Ein-/Ausgabeverwaltung). Bei der Nachbildung von Verbindungen in Paketznetzen kommen Prozessormodelle für folgende hauptsächliche Verarbeitungen zur Anwendung:

- programmgesteuerte Verarbeitung (nichtunterbrechende Prioritäten),
- abfragegesteuerte Verarbeitung (polling).

Auf weitere prinzipielle Prozessormodelle und deren Erläuterung sei auf /18/ verwiesen.

Prozessormodell für programmgesteuerte Verarbeitung: Bild 2.10 zeigt ein Modell für programmgesteuerte Verarbeitung, d.h. für nichtunterbrechende Prioritäten.

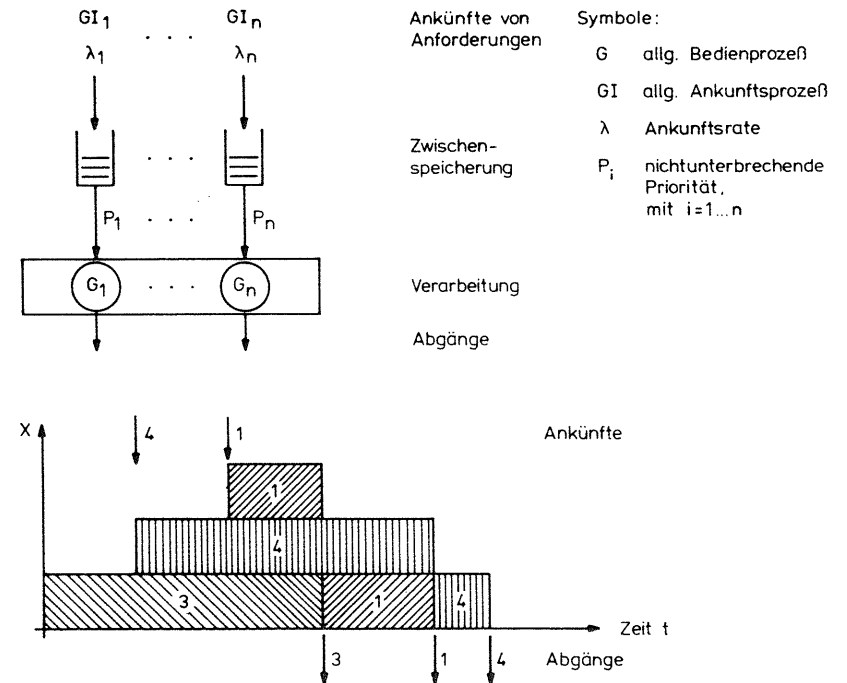


Bild 2.10: Prozessormodell für programmgesteuerte Verarbeitung (nichtunterbrechende Prioritäten)

Die prioritätenbehafteten Anforderungen besitzen individuelle Ankunftsverteilungen und Ankunftsrate sowie individuelle Bedienprozesse (siehe Kap.2.5.1.2). Das zeitliche Ablaufgeschehen ist in dem Zeitdiagramm in Bild 2.10 für die Anzahl  $X$  von Anforderungen in Gesamtmodell skizziert, wobei gilt:

- eine Bedienung wird nie unterbrochen,
- bei Bedienungsende erfolgt die Abfertigung nach höchster Priorität (kleinstem Index),

- bei mehreren Anforderungen gleicher Priorität erfolgt die Abfertigung nach beliebiger, prioritätenindividueller Strategie.

Prozessormodell für abfragegesteuerte Verarbeitung: Bild 2.11 zeigt das Modell für abfragegesteuerte Verarbeitung.

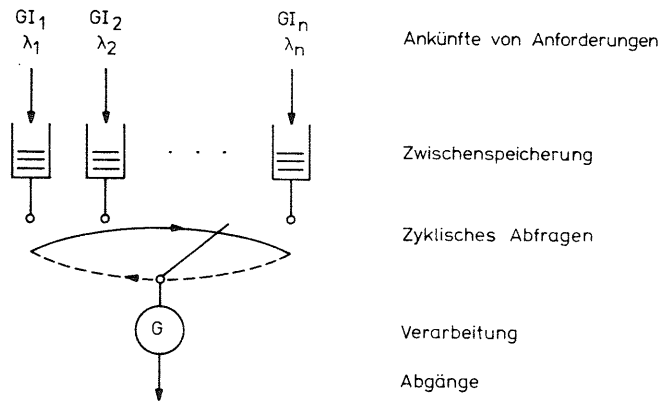


Bild 2.11: Prozessormodell für abfragegesteuerte Verarbeitung

Der Prozessor fragt nach jedem Bedienungsende, d.h. sobald er frei wird, die Zwischenspeicher ab. Dieses Abfragen kann z.B. in einfacher zyklischer Reihenfolge oder durch erhöhte Abfragefrequenz bevorrechtigter Gruppen (Realisierung von Prioritäten) erfolgen.

Derartige Prozessormodelle mit abfragegesteuerter Verarbeitung eignen sich z.B. zur Nachbildung von Multiplex- bzw. Demultiplexvorgängen.

Kanalmodelle: Kanalmodelle sind Nachbildungen des Betriebsmittels "Übertragungskanal", da dieser beim Steuer- bzw. Nutzdatenaustausch zum Systemengpaß werden kann. In Bild 2.12 ist ein Beispiel eines Kanalmodells skizziert.

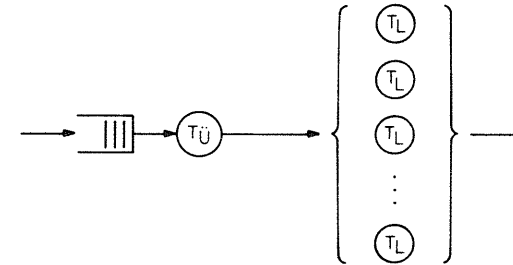


Bild 2.12: Kanalmodell

Die erste Bearbeitungsphase mit der Dauer T<sub>U</sub> steht hierbei stellvertretend für die Übertragungsdauer der Nachricht, d.h. für die Dauer des Aussendens, beginnend bei Bit 1 und endend beim letzten Bit der Nachricht. Anschließend belegt diese Nachricht eine der endlich vielen Phasen der Dauer T<sub>L</sub>, welche die physikalische Laufzeit auf dem Übertragungsmedium repräsentiert. Die Anzahl der Phasen T<sub>L</sub> ist hierbei durch die Anzahl Nachrichten, die sich gleichzeitig auf dem Kanal befinden können, gegeben.

Betrachtet man die Zuteilungsstrategie des Übertragungskanals, so lassen sich zwei für diese Arbeit wichtige Kategorien angeben:

- feste Zuteilung,
- abfragegesteuerte Zuteilung.

Weitere Zuteilungsmethoden sind aus /18/ zu entnehmen.

Beim Prinzip mit fester Zuteilung existiert ein individueller Kanal zwischen den beiden Stationen (fest zugeteilte Zeitlage innerhalb einer Zeitmultiplexleitung, Anschlußleitungen zu lokalen Dateneneinrichtungen, Verbindungsleitungen zwischen Netzknoten), während bei der abfragegesteuerten Zuteilung eine sog. Primärstation den Übertragungskanal steuert.

Kanalmodelle können zusätzlich noch folgende Merkmale aufweisen:

- Halb- bzw. Vollduplex-Betrieb,
- Datenflußsteuerung mittels Fenstermechanismus.

Derartige Eigenschaften von Kanalmodellen sind Merkmale des zur Anwendung kommenden Übertragungsprotokolls, wie z.B. HDLC (high level data link control, vgl. Kap.3.2.3).

Netzmodelle: Nach /18/ sind Netzmodelle solche Verkehrsmodelle, welche größere Steuerungskomplexe zusammenhängend beschreiben.

Logisch lassen sich Netzmodelle strukturieren in:

- Lastmodelle: zur Beschreibung von (Realzeit-) Anforderungen,
- Prozessormodelle: (s.o.) für allg. Verarbeitungsfunktionen,
- Kanalmodelle: (s.o.) für Übertragungsfunktionen,
- Anforderungs-Szenarios: (verbaler oder graphischer Form) zur Beschreibung der sequentiellen oder parallelen Inanspruchnahme von Betriebsmitteln.

### 2.5.1.2 Parameter

Ziel dieses Abschnittes ist die Beschreibung wichtiger Verkehrsparameter sowie struktureller und protokollabhängiger Systemparameter, soweit sie Einfluß auf die Leistungsfähigkeit von Datenverbindungen nehmen können.

Nachfolgend sind die wichtigsten Verteilungsfunktionen (VF) der Nachrichtenverkehrstheorie sowie deren Kurzbezeichnung, Mittelwert und Variationskoeffizient aufgeführt, /19/.

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

Tabelle 2.1: Wichtige VF der Nachrichtenverkehrstheorie

Verkehrsparameter: Bei der Modellbildung sind folgende Verkehrsbeziehungen durch geeignete Wahl der Parameter zu bestimmen:

- Ankunftsprozeß:
  - Mittelwert und Verteilungsfunktion (entsprechend Tabelle 2.1) für den Ankunftsabstand der einzelnen Anforderungen an das System, individuell vorgebar für jede im System vorhandene Verbindung;
  - Mittelwert und Verteilungsfunktion (entsprechend Tabelle 2.1) für die Nachrichtenblocklänge in Bits der einzelnen eintreffenden Anforderungen (Nachrichtenblöcke), individuell vorgebar für jede Verbindung;

- Bedienprozeß:

- Mittelwert und Verteilungsfunktion (entsprechend Tabelle 2.1) für die Bedienungsdauern, individuell vorgebar für jeden im System vorhandenen Prozessor bzw. jede Bearbeitungsphase.

Strukturparameter: Mittels dieser Parameter werden strukturelle Beziehungen und Eigenschaften innerhalb des Modells festgelegt, wie z.B.:

- Anzahl der Prozessoren bzw. Bearbeitungsphasen,
- Anzahl und Größe der Zwischenspeicher,
- Betriebsart (z.B. Halb- oder Vollduplex-Betrieb),
- Zuteilungsprinzipien (z.B. feste oder abfragegesteuerte Zuteilung),
- Prioritäten.

Protokollparameter: Diese Parameter spiegeln protokollspezifische Eigenschaften wider, wie z.B.:

- MODULO-Zählweise bei der Folgenummernsteuerung,
- Fenstergröße in Sende- und Empfangsrichtung,
- Wiederholstrategien.

### 2.5.2 Simulative Analyse

Simulationsstudien werden im allgemeinen dann benötigt, wenn komplexe Problemstellungen keiner praktikablen analytischen Lösungsmethode zugänglich sind, jedoch quantitative und qualitative Aussagen benötigt werden. Bei der simulativen Untersuchung wird das Ablaufgeschehen im System mittels eines entsprechenden Modells (siehe Kap.2.5.1) sowie durch eine Folge von Ereignissen mit Hilfe eines Programms auf einer Datenverarbeitungsanlage nachgebildet. Dieses Simulationsprogramm hat dabei im wesentlichen folgende Aufgaben, /19,20/:

- Abbildung des Modells auf entsprechende Programmkomponenten,
- Erzeugung und Bearbeitung von Ereignissen,
- Durchführung statistischer Messungen.

Vor jeder Untersuchung ist jedoch zu prüfen, welche Methode der Simulation eingesetzt werden kann bzw. muß, /19,21,22/.

Ruf- und Löschzahlenmethode: Diese Methode (Roulette Simulation, Monte-Carlo Simulation) bietet sich dann an, wenn für Ereignisgrößen keine Zeittreue gefordert wird, das Ablaufgeschehen allein durch Zustände und Zustandsübergangswahrscheinlichkeiten beschrieben werden kann und alle beteiligten Prozeßphasen die Gedächtnisfreiheit (Markoff-Eigenschaft) besitzen. Die Simulation der Zustandsübergangsfolgen erfolgt dabei entsprechend den Wahrscheinlichkeitsdichten für die Zustandsübergänge durch eine proportionale Unterteilung eines Zufallszahlenintervalls [0,1) und durch die Klassifizierung (Rufzahl, d.h. Eintreffen einer Anforderung; Löschzahl, d.h. Endigung in einer Bedieneinheit) einer mittels eines Zufallszahlengenerators ausgewürfelten Zufallszahl.

Zeittreue Simulationsmethode: Diese Simulationsmethode (event-by-event simulation, time-true simulation) findet dann Anwendung, wenn Zeitgrößen als Ergebniswerte gefordert werden, wie z.B. Wartezeiten, Durchlaufzeiten, Belegungszeiten.

Nach /21,22/ können hierbei wiederum folgende in Tabelle 2.2 aufgeführte Methoden unterschieden werden.

Allen zeittreuen Verfahren ist jedoch gemeinsam:

- die Abbildung des zu untersuchenden Systems in ein äquivalentes Verkehrsmodell mit entsprechenden Warteschlangen und Bedieneinheiten,
- die Organisation eines sog. "Ereigniskalenders", in welchem alle zukünftigen, d.h. vorgeplanten, Ereigniszeitpunkte eingetragen sind, wie z.B. Ankünfte, Bedienenden, Taktzeitpunkte,
- die Erzeugung zufallsabhängiger Zeiten nach vorgegebenen Verteilungsfunktionen (vgl. Tabelle 2.1),
- die Einführung einer Systemzeitvariablen, welche den aktuellen Zeitpunkt auf der simulierten Zeitachse angibt,



- die Berücksichtigung nur jener Zeitpunkte, zu denen eine Änderung im System stattfindet. Hierbei werden Systemzustandsvariable aktualisiert und eventuell neue, vorzuplanende Ereignisse ermittelt (Folgeereignisse) sowie Statistikmeßdaten erfaßt.

<u>Zeitreue Simulationsverfahren</u>
- belastungstreue Simulation (load-type simulation)
- belegungsphasentreue Simulation (subcall-type simulation)
- belegungstreue Simulation (call-type simulation)

Tabelle 2.2: Zeitreue Simulationsverfahren

Die verschiedenen, in Tabelle 2.2 aufgeführten Verfahren, unterscheiden sich hinsichtlich der Merkmale, die die vorkommenden Ereignisse charakterisieren, und der erzielbaren Ergebnisgrößen. Ereignisse können prinzipiell in zwei Kategorien unterteilt werden:

Ruf-Erstereignisse: (call events), diese entsprechen im allgemeinen den Anforderungswünschen an das System;

Teilruf-Ereignisse: (subcall events), dies sind all diejenigen Folge-Ereignisse, die ganz bestimmte Ablaufphasen innerhalb des Systems initiieren.

Nachfolgend sei kurz auf die Wirkungsweise und den Unterschied der einzelnen Verfahren eingegangen, /21/:

Belastungstreue Simulation: Dieses Verfahren generiert Ereignisse, die im Zeitabstand  $T_A$  im Modell eintreffen und diesem eine Belastung zuführen, welche der Bedienzeit  $T_H$  entspricht. Der Ankunftsabstand  $T_A$  wird entsprechend dem Ankunftsprozeß ermittelt, welcher durch Verteilungsfunktion des zufallsabhängigen Ankunftsabstands  $T_A: A(t)=P\{T_A \leq t\}$  und Mittelwert  $E[T_A]$  beschrieben ist. Entsprechendes gilt für die Bedienzeit  $T_H$  mittels  $H(t)=P\{T_H \leq t\}$  und  $E[T_H]$ .

Als Ergebnisse können Werte für die Belastung der Bedieneinheiten sowie Wartezeiten ermittelt werden.

Belegungsphasentreue Simulation: In dieser Version werden dem Modell Ereignisse zugeführt, die vom Wesen her Teilruf-Ereignissen entsprechen und im Zeitabstand  $T_A$  eintreffen. Jedes generierte Ereignis ist mit einer Typkennung versehen und führt dem Verkehrsmodell eine Belastung zu, die einer typindividuellen Bedienzeit entspricht. Diese Charakterisierung des Typs für jedes auftretende Ereignis erfolgt entsprechend der Auftrittswahrscheinlichkeit der möglichen Ereignistypen.

Bei der belegungsphasentreuen Simulation werden somit Ereignisse quantitativ richtig erzeugt; es existiert jedoch keine zeitliche Korrelation zwischen den einzelnen vorkommenden Ereignissen.

Ankunftsabstand  $T_A$  und typindividuelle Bediendauer  $T_H(Typ)$  gehorchen dabei vorgegebenen Verteilungsfunktionen  $A(t)=P\{T_A \leq t\}$  und  $H(t)=P\{T_H(Typ) \leq t\}$  mit Mittelwerten  $E[T_A]$  und  $E[T_H(Typ)]$ .

Als Ergebnisgrößen können typspezifische Werte ermittelt werden.

Belegungstreue Simulation: Dieses Simulationsverfahren ist das umfangreichste der drei zeittreuen Verfahren in Bezug auf Programieraufwand, Speicherplatz und Fülle erzielbarer Ergebnisse. Es ist jedoch auch das realitätsnächste und somit genaueste Verfahren, da vom momentanen Systemzustand abhängige Rückwirkungen auf den Ereignisgenerator berücksichtigt werden können. Bei diesem Verfahren wird die komplette Abarbeitung einer Anforderung innerhalb des Systems durch eine vorgegebene Kette von (Folge-) Ereignissen (Teilruf-Ereignissen) beschrieben. Somit besteht zwischen den einzelnen Teilruf-Ereignissen eine eindeutige (zeitliche) Verbindung. Anforderungen, welche die Ereignisketten aktivieren, werden im Zeitabstand  $T_A$  mit vorgegebener Verteilungsfunktion  $A(t) = P\{T_A \leq t\}$  und Mittelwert  $E[T_A]$  erzeugt.

Die einzelnen Teilruf-Ereignisse werden abhängig vom Systemzustand entsprechend der Folge innerhalb der Ereigniskette ermittelt. Jedes Ereignis, das dem Verkehrsmodell angeboten wird, ist dabei mit einer typindividuellen Bedienzeit behaftet.

Als Ergebnisse liefert dieses Verfahren sowohl ereignistyp- als auch anforderungsbezogene Werte. Darüberhinaus lassen sich auch Systemreaktionszeiten auf bestimmte Ereignisse ermitteln, wie z.B. Quittierzeiten, protokollbedingte Wartezeiten usw.

### 2.5.3 Mathematische Analyse

Ausgangspunkt für die mathematische Analyse mittels Wahrscheinlichkeitstheoretischer Methoden ist die Tatsache, daß bei der Leistungsuntersuchung von Datenverbindungen innerhalb paketvermittelnder Netze extrem komplexe Verkehrsmodelle entstehen (siehe Kap.5.1) in denen Vorgänge ablaufen, welche hinsichtlich der Eintreffzeitpunkte von Anforderungen bzw. Belegungszeiten von Betriebsmitteln so vielschichtig sind, daß sie als zufallsabhängige (stochastische) Prozesse aufgefaßt und mit diesen Mitteln beschrieben werden können. Ein stochastischer Prozeß ist dabei durch folgende drei Komponenten gekennzeichnet, /23/:

- Zustandsraum,
- Zeit,
- Abhängigkeit zwischen den Zufallsvariablen für verschiedene Zeiten.

Die Klasse der Verkehrsmodelle, die einer geschlossenen analytischen Lösungsmethode zugänglich sind, ist stark eingeschränkt. Haupthinderungsgrund ist dabei die Tatsache, daß der Ausgangsprozess eines Bediensystems in der Regel kein Erneuerungsprozeß ist, d.h. neben einer i.A. nicht bekannten Verteilungsfunktion zusätzlich Abhängigkeiten zwischen aufeinanderfolgenden Ereignissen aufweist.

Ziel dieses Abschnittes ist, grundlegende Verfahren zur Leistungsanalyse von Netzwerken sowie deren Anwendungsbereiche vorzustellen.

#### 2.5.3.1 Produktformlösung

Exakte Berechnungsmethoden sind bekannt für offene und geschlossene Netze mit negativ exponentiell verteilten Ankunftsabständen bzw. Bedienzeiten, /20,24,25/. Bei derartigen Markoffschen Ankunfts- und Bedienprozessen sowie bei speziellen Betriebsarten existieren für den resultierenden vieldimensionalen Zustandsraum geschlossene Lösungen für die Zustandswahrscheinlichkeiten ("Produktformlösung"). Die Produktformlösung ist ferner durch eine Reihe weiterer Eigenschaften gekennzeichnet, welche sich vorteilhaft bei einer sukzessiven Analyse einsetzen lassen. Die Produktformlösung ist ferner Basis für leistungsfähige numerische Verfahren zur Bestimmung charakteristischer Mittelwerte (siehe Mittelwertanalyse in Kap. 2.5.3.2).

Nachfolgend werden die wesentlichsten Ergebnisse für Warteschlangennetze mit Produktformlösung vorgestellt, /20,28,29/. Zugrunde gelegt seien offene bzw. geschlossene Warteschlangennetze, deren Struktur durch eine Verzweigungsmatrix  $Q = (q_{ij})$  festgelegt ist, vgl. Bild 2.13.

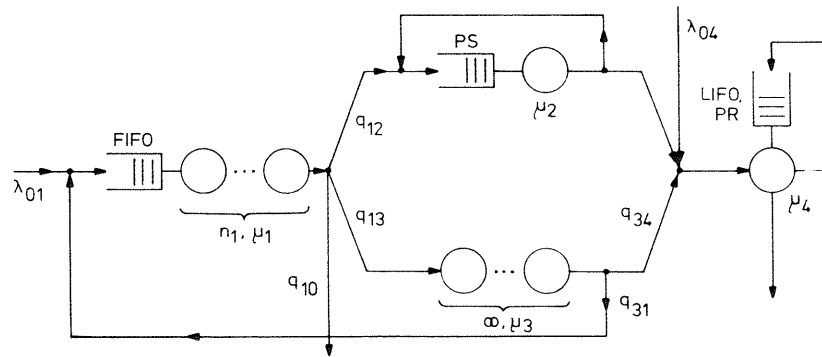


Bild 2.13: Beispiel eines offenen Warteschlangennetzes mit insgesamt 4 Bedienstationen

Bedienungssysteme können dabei einer der 4 nachfolgend beschriebenen Kategorien angehören:

1. Eine Bedieneinheit mit negativ-exponentieller Bedienzeit. Die Bedienungsrate darf abhängig von der momentanen Anzahl von Anforderungen in der Bedienstufe sein. Die Abfertigungsdisziplin für wartende Anforderungen ist FIFO.
2. Eine Bedieneinheit mit allgemeiner Bedienzeit. Alle sich momentan im Bedienungssystem befindlichen Anforderungen teilen sich die Bedienungskapazität zu gleichen Teilen nach dem Zeitscheibenverfahren auf ("processor sharing").
3. Hinreichend große Anzahl von Bedieneinheiten mit allgemeiner Bedienzeit.
4. Eine Bedieneinheit mit allgemeiner Bedienzeit. Dem Bedienungssystem neu hinzukommende Anforderungen erhalten sofort Bedienung, wobei eine eventuell sich gerade in Bedienung befindliche Anforderung sofort unterbrochen wird. Unterbrochene Anforderungen setzen später ihre Bedienung vom jeweils erreichten Bedienungszustand aus in umgekehrter Reihenfolge ihres Eintreffens fort (LIFO preemptive-resume).

Bei externen Ankünften, die bei offenen Netzen entsprechend exponentiell verteilten Ankunftsabständen erfolgen, darf die Gesamtankunftsrate von der Gesamtzahl von Anforderungen im Netz abhängig sein. Geschlossene Netze unterscheiden sich dadurch von offenen Netzen, daß bei ihnen die Anzahl der Nachrichten (z.B. Pakete) innerhalb der Verbindungen, in diesem Zusammenhang spricht man auch von Ketten, durch einen festen Wert, der sog. Populationsgröße, vorgegeben ist.

Wichtige Eigenschaften der Produktformlösung sollen im folgenden kurz dargestellt werden:

- Die Produktterme sind die Lösungen von getrennt betrachteten Wartesystemen mit negativ-exponentiell verteilten Eingangs- und Ausgangsprozessen. Eine Erklärung läßt sich nur im Falle offener und rückkopplungsfreier Netze mit negativ-exponentiell verteilten Bedienzeiten angeben, da in diesem Fall das Ausgangsprozeßtheorem (output theorem), [26,27], sukzessive angewendet werden kann. Dieses Theorem besagt, daß der Ausgangsprozeß eines Markoffschen Wartesystems wieder ein Poisson-Prozeß ist. Bei Netzen mit internen Rückkopplungen kann dieses Theorem nicht mehr angewendet werden, dennoch existiert dieselbe Lösung, wie wenn alle Prozesse die Markoff Eigenschaften hätten!
- Die Produktformlösung ist robust gegenüber der Verteilungsfunktions-Charakteristik der Bedienzeiten unter bestimmten Voraussetzungen. Diese Robustheit hängt damit zusammen, daß die einzelnen Bedienstufen unter den gegebenen Voraussetzungen bei Markoffschem Eingangsprozeß wieder einen Markoffschen Ausgangsprozeß erzeugen ("Markoff implies Markoff").
- Die Produktformlösung ist robust gegenüber Verzweigungsdisziplinen (Wegelenkung).
- Die allgemeinen Gleichgewichtsbedingungen können zerlegt werden in Teilgleichgewichtsbedingungen ("local balance").
- Die gesamte Netzumgebung einer einzelnen Stufe kann, in Bezug auf diese Stufe, (fluß-)äquivalent ersetzt werden durch eine einzige Komplementärstation ("Nortons Theorem", siehe Kap. 2.5.3.3.).

- Die Produktformlösung gestattet für geschlossene Netze effektive Rekursionsalgorithmen zur Bestimmung der in der Lösung vorkommenden Konstanten ohne die Berechnung aller einzelnen Zustandswahrscheinlichkeiten.

2.5.3.2 Mittelwertanalyse

Die in Kap.2.5.3.1 angegebenen üblichen Verfahren zur Lösung von Produktformlösungsnetzen beruhen in der Regel auf der Formulierung algebraischer Gleichgewichtsbedingungen für die mehrdimensionalen Zustandsverteilungen im Vektorraum des Systems, die schließlich mittels der bereits erwähnten Produktform bestimmt werden können. Einzig verbleibendes Problem ist die numerische Bestimmung der Normierungskonstanten. Diese Konstante hat im Fall offener Netze einen einfachen analytischen Ausdruck, wohingegen sie bei geschlossenen Netzen die Form einer Summe von Produkttermen annimmt.

In /30-33/ wird ein neues analytisches Verfahren - die Mittelwertanalyse (mean value analysis)- vorgestellt, welches die Berechnung von Mittelwerten, wie z.B. mittlere Warteschlangenlänge, mittlere Wartezeiten und Durchsatz in geschlossenen Netzen (sog. Produktformlösungsnetzen) zur Aufgabe hat. Der Vorteil der Mittelwertanalyse ist, daß die Bestimmung der Ergebnisse auf rekursive Weise ohne Bestimmung einer Normierungskonstanten erfolgen kann. Die Mittelwertanalyse basiert im wesentlichen auf zwei Beziehungen, dem "arrival theorem", /37,38/, und dem Gesetz von Little, /34-36/.

"Arrival Theorem": Dieses Theorem besagt, daß in einem geschlossenen Markoffschen System der Systemzustand, den eine beliebige Nachricht bei Eintreffen an einer beliebigen Bedienstation antrifft, dieselbe Wahrscheinlichkeitsverteilung besitzt, wie die Lösung der stationären Zustände desselben Systems mit einer um 1 Nachricht reduzierten Population.

Dieses Theorem scheint deshalb einleuchtend, da das Markoffsche System ohnehin gedachtnisfrei ist, /35,36/, und die eine entnommene Nachricht, die als neu ankommende Anforderung betrachtet wird, dann das System ohne sich selbst im Gleichgewicht sieht.

Gesetz von Little: (Little's Formula); dieses in der Leistungsbeurteilung von Kommunikationssystemen sehr häufig angewandte Gesetz setzt den Durchsatz  $\gamma$ , die mittlere Anzahl  $E[X]$  an Nachrichten im System und die mittlere Durchlaufzeit  $E[T]$  durch das System folgendermaßen zueinander in Beziehung:

$$E[X] = \gamma \cdot E[T] \quad ( 2.1 )$$

Im folgenden sollen die zur Anwendung kommenden Beziehungen und ihr Wirkungsbereich kurz zusammengefaßt und hergeleitet werden.

Netze mit einer geschlossenen Kette ohne Verzweigungen: Die Idee, die hinter der Mittelwertanalyse steht, läßt sich am einfachsten am Beispiel einer einzelnen geschlossenen Kette aufzeigen (siehe hierzu Bild 2.14).

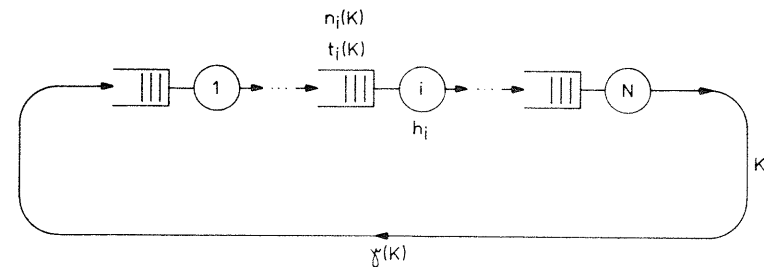


Bild 2.14: Netz mit einer einzelnen geschlossenen Kette ohne Verzweigungen

Für Bild 2.14 und die folgenden Beziehungen gelten die Vereinbarungen:

K	Population (Anzahl Nachrichten in der Kette)
N	Anzahl Stationen (Bedieneinheit (BE) + Warteschlange (WS))
$h_i$	mittlere Bediendauer in BE der Station i
$t_i(K)$	mittlere Aufenthaltsdauer in Station i (incl. BE)
$n_i(K)$	mittlere WS-Länge in Station i (incl. Nachricht in BE)
$\gamma(K)$	Durchsatz der Kette (bei Population K)

Gemäß Bild 2.14 bilden die N Stationen eine geschlossene Kette, in welcher K Nachrichten zyklisch zirkulieren.

Damit gilt, /30,33/:

$$t_i(K) = h_i + h_i \cdot \left\{ \begin{array}{l} \text{mittlere Anzahl Nachrichten,} \\ \text{welche bei Ankunft angetroffen} \\ \text{wurden} \end{array} \right\} \quad (2.2)$$

$$\gamma(K) = K / \sum_{i=1}^N t_i(K) \quad (2.3)$$

$$n_i(K) = \gamma(K) \cdot t_i(K) \quad (2.4)$$

- Gleichung (2.2) gilt für Ankunftszeitpunkte und besagt, daß die mittlere Aufenthaltsdauer in Station i gleich der Summe aus eigener Bediendauer plus der Gesamtbedienzeit aller bei Ankunft angetroffenen Nachrichten ist;
- Gleichung (2.3) erhält man durch Anwendung des Gesetzes von Little auf die gesamte Kette;
- Gleichung (2.4) entsteht durch Anwendung des Gesetzes von Little auf eine einzelne Station i,  $i=1,2,\dots,N$ .

Durch konsequente Anwendung des o.a. "Arrival Theorem" läßt sich (ohne weiteren Beweis) die Gleichung (2.2) angeben durch:

$$t_i(K) = h_i + h_i \cdot n_i(K-1) \quad (2.5)$$

Damit läßt sich zusammenfassend eine rekursive Lösung für diese einzelne geschlossene Kette angeben:

$$\text{Startwert: } n_i(0) = 0 \quad i=1,2,\dots,N \quad (2.6)$$

$$\text{Rekursion: } k=1,2,\dots,K$$

$$t_i(k) = h_i + h_i \cdot n_i(k-1); \quad i=1,2,\dots,N \quad (2.7)$$

$$\gamma(k) = k / \sum_{i=1}^N t_i(k) \quad (2.8)$$

$$n_i(k) = \gamma(k) \cdot t_i(k); \quad i=1,2,\dots,N \quad (2.9)$$

Netze mit einer geschlossenen Kette mit Verzweigungen: Die in den Gleichungen (2.6) bis (2.9) aufgeführte Rekursion läßt sich auf einfache Art und Weise zur Berechnung von Netzen mit einer geschlossenen Kette mit Verzweigungsmatrix  $Q=(q_{ij})$  und Wahrscheinlichkeiten  $q_{ij}$  für Verzweigen von Station i nach j umformen (vgl. Bild 2.15).

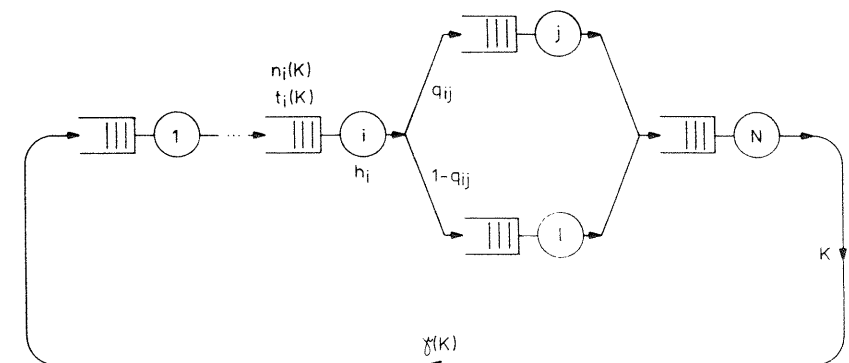


Bild 2.15: Netz mit einer einzelnen geschlossenen Kette und Verzweigung (Beispiel)

Im folgenden sei mit  $i^*$  eine beliebig gewählte (fest markierte) Station innerhalb des Netzes bezeichnet. Zusammen mit dem Vektor  $\theta = (\theta_i)$ , wobei die Meßgröße  $\theta_i$  die mittlere Anzahl Besuche einer Nachricht bei Station  $i$  zwischen zwei aufeinanderfolgenden Besuchen bei der markierten Station  $i^*$  angibt, folgt:

$$\theta_{i^*} = 1 \quad (2.10)$$

$$\theta = \theta \cdot Q \quad (2.11)$$

Da die Anzahl der Besuche direkt proportional zum Durchsatz ist, kann mit  $\theta_i$  ebenfalls der Durchsatz  $\delta_i$  der Station  $i$  in Einheiten von  $\delta_{i^*}$ , dem Durchsatz der markierten Station  $i^*$ , angegeben werden mit:

$$\delta_i = \theta_i \cdot \delta_{i^*}; \quad i=1,2,\dots,N \quad (2.12)$$

Für die Zeit zwischen 2 aufeinanderfolgenden Besuchen einer Nachricht bei Station  $i^*$  gilt:

$$t_z = \sum_{i=1}^N \theta_i \cdot t_i \quad (2.13)$$

Mit der vereinfachenden Schreibweise  $\delta^* = \delta_{i^*}$  und mit den Gleichungen (2.12) und (2.13) folgt schließlich die rekursive Lösung:

$$t_i(k) = h_i + h_i \cdot n_i(k-1) \quad (2.14)$$

$$\delta^*(k) = k / \sum_{i=1}^N \theta_i \cdot t_i(k) \quad (2.15)$$

$$n_i(k) = \delta^*(k) \cdot \theta_i \cdot t_i(k) \quad (2.16)$$

Definiert man die Zeit  $t_{i^*,i}$  als die mittlere Aufenthaltsdauer einer Nachricht in Station  $i$  zwischen 2 aufeinanderfolgenden Besuchen bei Station  $i^*$  sowie die Verkehrsintensität  $q_i$  jeweils zu:

$$t_{i^*,i} = \theta_i \cdot t_i \quad (2.17)$$

$$q_i = \theta_i \cdot h_i \quad (2.18)$$

so ergibt sich als Lösung für geschlossene Netze mit einer einzelnen Kette und Verzweigungen folgende Rekursion,  $k = 1,2,\dots,K$ :

$$t_{i^*,i}(k) = q_i \cdot (1 + n_i(k-1)) \quad (2.19)$$

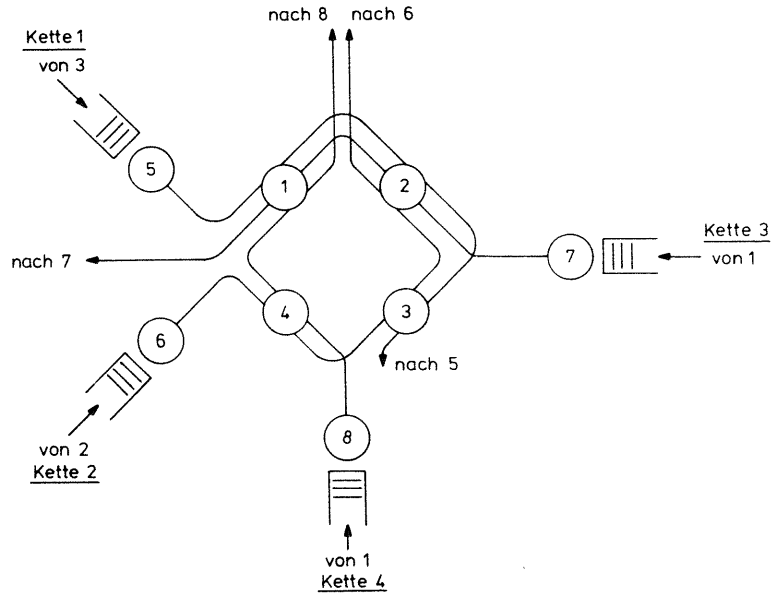
$$\delta^*(k) = k / \sum_{i=1}^N t_{i^*,i}(k) \quad (2.20)$$

$$n_i(k) = \delta^*(k) \cdot t_{i^*,i}(k) \quad (2.21)$$

Netze mit mehreren geschlossenen Ketten: Eine Verallgemeinerung der in den Gleichungen (2.6) bis (2.9) angegebenen Beziehungen führt auf den allgemeinen Fall von Warteschlangennetzen mit mehreren, voneinander getrennten, geschlossenen Ketten (siehe Bild 2.16).

Für die Lösung derartiger Netze gelten folgende Vereinbarungen:

- R Anzahl geschlossener Ketten
- r Hochindex für Größen, die sich auf Kette r beziehen;  
Bereich:  $r = 1,2,\dots,R$
- i Tiefindex für Größen, die sich auf Station i beziehen;  
Bereich:  $i = 1,2,\dots,N$
- IK Populationsvektor;  $IK = (K^1, K^2, \dots, K^r, \dots, K^R)$
- IK-e<sup>r</sup> Populationsvektor, wobei in Kette r eine Nachricht weniger angenommen wird;  $IK-e^r = (K^1, K^2, \dots, K^{r-1}, K^{r-1}, K^{r+1}, \dots, K^R)$



Beispiel:  $R = 4$   
 $Q(1) = (5, 1, 2, 3)$       $Q(3) = (7, 2, 1)$   
 $Q(2) = (6, 4, 3, 2)$       $Q(4) = (8, 4, 1)$

$N = 8$   
 $R(1) = (1, 3, 4)$       $R(5) = (1)$   
 $R(2) = (1, 2, 3)$       $R(6) = (2)$   
 $R(3) = (1, 2)$       $R(7) = (3)$   
 $R(4) = (2, 4)$       $R(8) = (4)$

Bild 2.16: Netz mit mehreren geschlossenen Ketten

$R(i)$  Menge derjenigen Ketten, welche über Station  $i$  verlaufen  
 $Q(r)$  Menge derjenigen Stationen, welche der Kette  $r$  angehören  
 $\delta^r$  Durchsatz einer (willkürlich markierten) Station in Kette  $r$

$h_i^r$  mittlere Bediendauer in der Bedieneinheit  $i$  für Nachrichten der Kette  $r$   
 $t_{i^*,i}^r$  mittlere Aufenthaltsdauer einer Nachricht der Kette  $r$  in Station  $i$  zwischen 2 aufeinanderfolgenden Besuchen einer willkürlich gewählten Station  $i^*$  dieser Kette  $r$   
 $n_i^r$  mittlere Anzahl Nachrichten der Kette  $r$  in Station  $i$   
 $\theta_i^r$  mittlere Anzahl Besuche einer Nachricht der Kette  $r$  in Station  $i$  zwischen 2 aufeinanderfolgenden Besuchen der willkürlich gewählten Station dieser Kette  $r$ .

Analog Gleichung (2.18) kann gefolgert werden:

$$g_i^r = \theta_i^r \cdot h_i^r \quad (2.22)$$

Für die mittlere WS-Länge  $n_i$  in Station  $i$  (incl. Nachricht in der BE) gilt:

$$n_i = \sum_{r \in R(i)} n_i^r \quad (2.23)$$

Mit diesen Vorgaben folgt für die rekursive Lösung von Netzen mit mehreren geschlossenen Ketten:

$$\text{Startwert: } n_i(0) = 0 \quad (2.24)$$

Rekursion:

$$t_{i^*,i}^r(IK) = \begin{cases} g_i^r \cdot (1 + n_i(IK - e^r)) & \text{Fall 1, 2} \\ g_i^r & \text{Fall 3} \end{cases} \quad (2.25)$$

$$\delta^r(IK) = K^r / \sum_{i \in Q(r)} t_{i^*,i}^r(IK) \quad (2.26)$$

$$n_i(IK) = \sum_{j \in R(i)} \delta^j(IK) \cdot t_{i^*,i}^j(IK) \quad (2.27)$$

Für die Fallunterscheidung in Gleichung (2.25) gilt:

- Fall 1: WS-Abfertigungsdisziplin durch Prozessor-Sharing oder unterbrechende Prioritäten mit last-in,first-out,
- Fall 2: WS-Abfertigungsdisziplin ist first-in,first-out, und Bediendauer ist negativ exponentiell verteilt,
- Fall 3: reine Zeitverzögerungsstufe, d.h. "infinite server"-Stufe.

Zu beachten bei der Rekursion nach Gleichung (2.24) bis (2.27) ist, daß sämtliche Zwischenwerte zwischengespeichert werden müssen und daß die Rekursionsschleife über den Vektor  $IK$  in aufsteigender Reihenfolge (der Ketten) erfolgen muß.

### 2.5.3.3 Dekomposition

Eine weitere Vorgehensweise zur Lösung komplexer Netze ist die der Dekomposition, /39-45/. Bei diesem Verfahren wird das gesamte Warteschlangennetz in Untersysteme aufgegliedert, die dann getrennt voneinander analysierbar sind.

Die Berechnung der Subsysteme kann dabei auf zweierlei Arten erfolgen:

Klassifizierung von Eingangs- und Ausgangsprozesse: Bei diesem Näherungsverfahren werden einzelne Warteschlangen oder auch ganze Subsysteme unter Berücksichtigung ihres jeweiligen Eingangs- und Ausgangsverhaltens isoliert voneinander analysiert. In der Regel werden dabei die Eingangs- und Ausgangsprozesse näherungsweise als Erneuerungsprozesse, /35/, angenommen oder durch ihre ersten beiden Momente - Mittelwert und Variationskoeffizient - beschrieben (vgl. hierzu Bild 2.17, aus /45/).

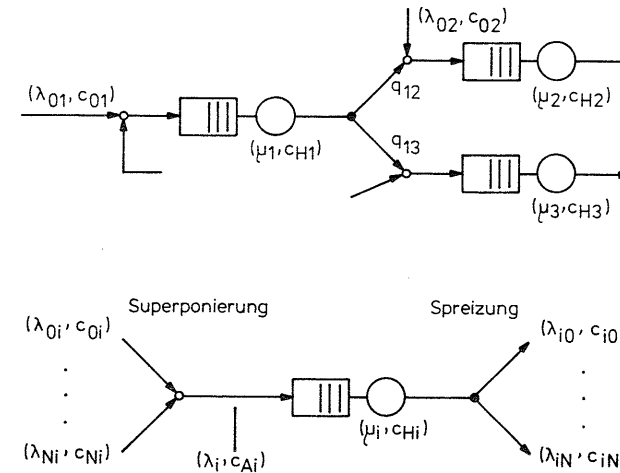


Bild 2.17: Dekomposition durch Klassifizierung von Eingangs- und Ausgangsprozesse

Im Falle offener Netze unterliegen die Datenflußraten  $\lambda_i$  einem System linearer Gleichungen, unter Beachtung der Flußerhaltungsregel bei jeder Station. Mittelwerte und Variationskoeffizienten können dann dem Ausgangsprozess entnommen werden, /46/.

Bei geschlossenen Netzen, insbesondere bei netzinternen Schleifen, als auch bei offenen Netzen mit Rückkopplungen ist eine iterative Lösung unumgänglich.

Zwei Basisoperationen sind bei dem Verfahren der Klassifizierung vonnöten:

- Superponierung: Überlagerung bzw. Zusammenführung mehrerer Ströme zu einem Gesamtstrom,
  - Spreizung: Dekomposition eines Stromes in mehrere Teilströme.
- Beiden Operationen liegt dabei die Hypothese zugrunde, daß die Teil- und Gesamtströme unabhängig voneinander sind und alle Ströme einem Erneuerungsprozeß unterliegen.



Hierarchische Dekomposition: Diese Technik beruht auf der Idee der Aggregation. Subsysteme werden definiert und durch einige wenige einfachere Netze oder, wenn überhaupt möglich, durch sog. single-server Stationen mit zugehörigen Bedienzeiten beschrieben. Das auf diese Weise resultierende System weist dieselbe Verkehrsflußrate (Durchsatz) auf wie das Originalsystem.

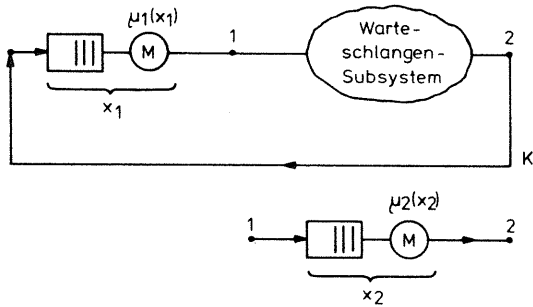


Bild 2.18: Dekomposition und Aggregation

Nach Bild 2.18 wird das zwischen den Punkten 1 und 2 liegende Warteschlangennetz in eine einzige Bedienstation aggregiert und diese als Ersatz zwischen diese Punkte 1 und 2 eingesetzt. Man spricht in diesem Zusammenhang auch vom Komplement zur Bedienstation 1, /33/, im folgenden 1-Komplement genannt.

Zum besseren Verständnis sowie zur detaillierteren Beschreibung der Komplementbildung sei auf folgendes Beispiel verwiesen (vgl. Bild 2.19).

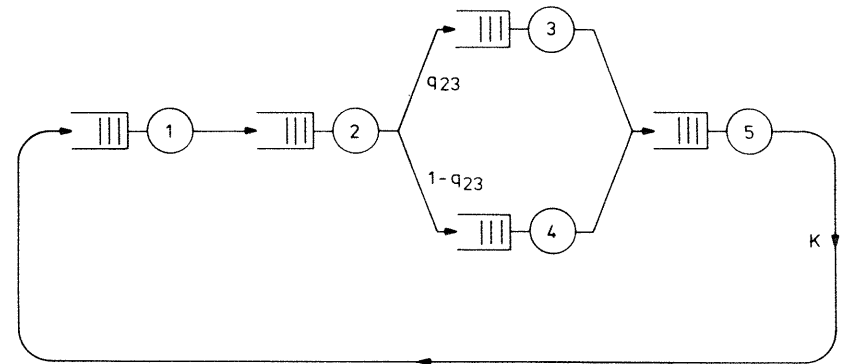


Bild 2.19: Markoffsches Netz mit 5 Bedienstationen und Verzweigungswahrscheinlichkeiten

Im folgenden sei angenommen, daß nur die Ergebnisse von Station 1 von Interesse seien. In diesem Fall kann das Gesamtsystem aus Bild 2.19 in 2 geschlossene Subsysteme zerlegt werden (Bild 2.20):

- das 1-Komplement und
- ein zyklisches System, bestehend aus Station 1 und einer 1-Komplementärstation, die das komplette 1-Komplement repräsentiert.

Allgemein ist das i-Komplement dasjenige geschlossene Markoffsche Netz, welches nach Überbrückung ("Kurzschluß") der Station i mit ansonsten unveränderten Parametern verbleibt.

Das Dekompositionstheorem, /47/, besagt nun, daß das statistische Verhalten, d.h. die Zustandsverteilung von Station 1 (im allgemeinen von Station i) im Ursprungssystem identisch ist mit demjenigen von Station 1 (i.a. Station i) im zyklischen Ersatzsystem (Bild 2.22 (a)), wobei für die Bedienrate  $\mu^*(x)$  der Komplementärstation gilt:

$$\mu^*(x) = \gamma^{(i)}(x) \quad ( 2.28 )$$

mit  $\gamma^{(i)}(x)$  als Durchsatz des  $i$ -Komplements bei einer Population von  $k = x$ ,  $k = 1, 2, \dots, K$ .

In diesem Zusammenhang wird die Komplementärstation auch als Norton-Äquivalent des  $i$ -Komplements bezeichnet, /47/.

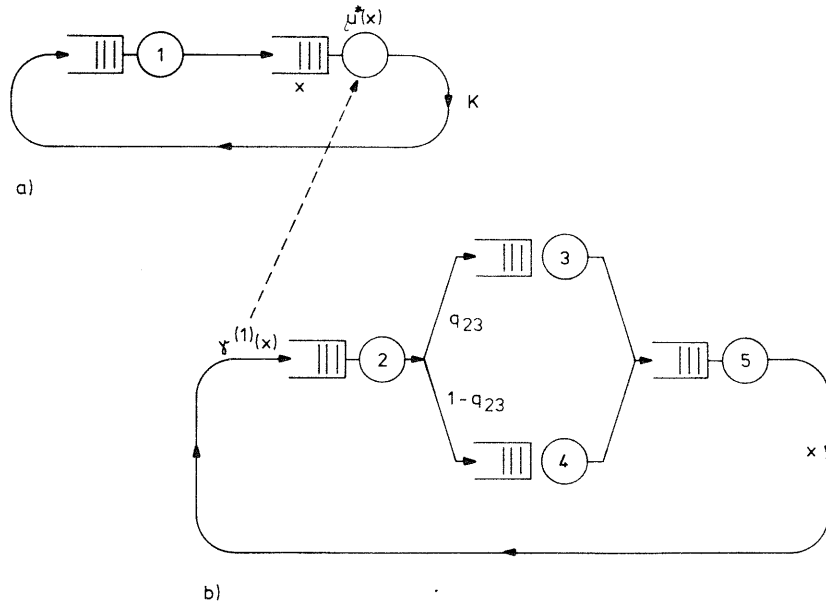


Bild 2.20: Dekomposition in zyklisches System mit 2 Bedienstationen (a) und 1-Komplement (b)

Die wiederholte Anwendung dieser Methode bei großen Systemen, sozusagen von oben nach unten, führt zur erwähnten hierarchischen Dekomposition und erlaubt damit eine schrittweise Verringerung der Komplexität des Gesamtsystems.

### 3. DIE CCITT-NETZZUGANGSSCHNITTSTELLE X.25

Mit Hilfe der vorliegenden Arbeit sollen sinnvolle und einfach realisierbare Möglichkeiten aufgezeigt werden, die es erlauben, die Leistungsfähigkeit mehrerer virtueller Verbindungen auf einem Übertragungsabschnitt innerhalb eines Kommunikationsnetzes zu ermitteln. Damit können dann auch Antworten auf Fragen über Dienstgüte, leistungsbeeinflussende Parameter und Festlegung der vom Benutzer wählbaren Parameter gegeben werden.

Die gezeigten Untersuchungen stützen sich dabei auf die von der CCITT (Comité Consultatif International Téléphonique et Télégraphique) verabschiedeten Empfehlung der Netzzugangsschnittstelle X.25, /2,48/:

"Interface between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks."

Die Wahl, die Untersuchungen an X.25 vorzunehmen, wurde weniger wegen der bereits vorliegenden Normung getroffen, sondern mehr aufgrund ihrer Allgemeingültigkeit. Aus diesem Grund sollen in diesem Kapitel die Netzzugangsschnittstelle X.25, ihre Strukturierung mit entsprechenden Funktionen sowie spezielle Mechanismen und leistungsbeeinflussende Parameter, jeweils im Hinblick auf o.a. erwähnte Allgemeingültigkeit, erklärt werden.

In diesem Zusammenhang sei auf die Beschreibung der CCITT-Empfehlung in /2/, /48/ und /49/ verwiesen sowie auf /50/ bis /52/ als Einführung in Funktionsweisen von X.25.

Alle in dieser Arbeit im Zusammenhang mit X.25 gemachten Aussagen beziehen sich ausschließlich auf die in /2/ und /48/ veröffentlichte Empfehlung. Änderungen, welche in der CCITT-Sitzungsperiode 1981 bis 1984 vorgenommen wurden und in der revidierten Version von X.25 zum Ausdruck kommen, bleiben bei den vorliegenden Untersuchungen unberücksichtigt.

### 3.1 Allgemeines

X.25 ist ein Standard für eine Netzzugangsschnittstelle zwischen einer Datenendeinrichtung (DEE) und einer Datenübertragungseinrichtung (DÜE) für Endgeräte, die im Paketmodus an öffentlichen Datennetzen arbeiten.

Die Datenendeinrichtung ist dabei die Einrichtung auf der Teilnehmerseite und fungiert als Quelle bzw. Senke für Datenpakete. Physikalisch gesehen handelt es sich bei der DEE entweder um Datenverarbeitungsanlagen oder intelligente Endgeräte (Terminals) (siehe Bild 3.1).

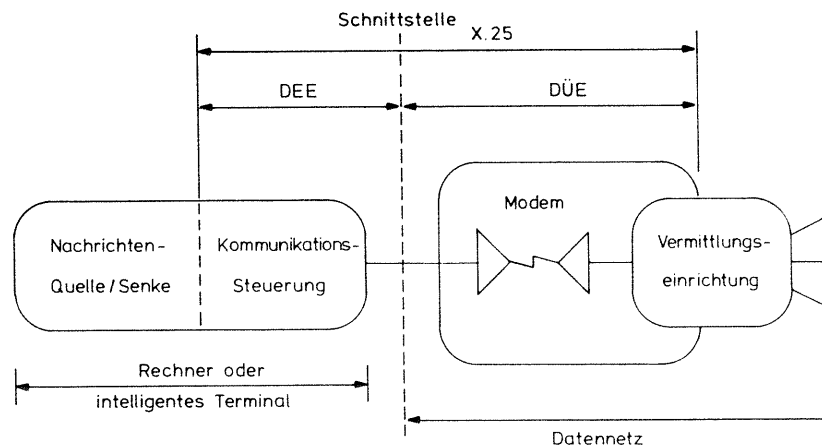


Bild 3.1: X.25-Netzzugangsschnittstelle

Die Datenübertragungseinrichtung ist streng genommen diejenige Einrichtung, die die Signale von der DEE in eine über die Teilnehmeranschlußleitung übertragbare Form konvertiert. Im Sinne von X.25 ist die DÜE jedoch als Zugangspunkt zum Datennetz anzusehen, an welchem die DEE angeschlossen ist (Bild 3.2).

Damit ergibt sich folgender im Bild 3.2 widergespiegelter Wirkungsbereich von X.25.

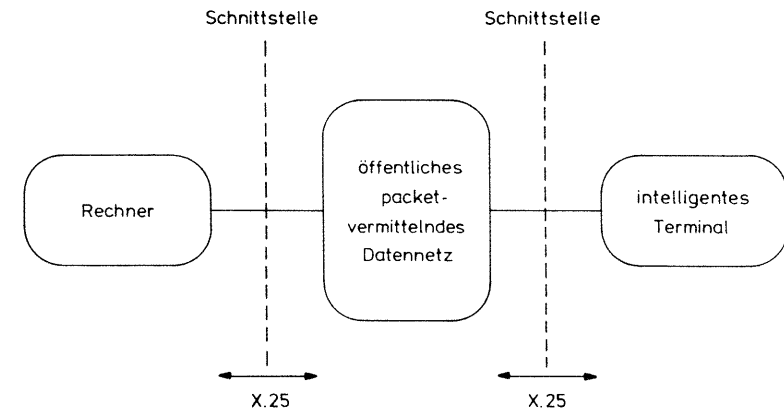


Bild 3.2: Wirkungsbereich von X.25

Die in der CCITT-Empfehlung X.25 standardisierten Protokolle umfassen dabei sowohl das Format der zu übertragenden Daten als auch die prozeduralen Festlegungen für den zeitlichen Ablauf.

Folgende hauptsächlichen Dienste werden durch X.25 bereitgestellt:

- Sicherung des Übertragungsabschnitts (Netzzugang) durch Fehlererkennungs- und Fehlerbehebungsmechanismen,
- Bildung mehrerer logischer Kanäle auf diesem einen Übertragungsabschnitt,
- Auf- und Abbau virtueller Verbindungen,
- Multiplexen/Demultiplexen mehrerer logischer Kanäle (virtueller Verbindungen) auf einen Übertragungsabschnitt,
- Folgenummernsteuerung zur Einhaltung der Paketreihenfolge,
- verbindungsindividuelle Datenflußsteuerung,
- Regeln zum Unterbrechen, Rücksetzen und Wiederanlauf,
- Regeln zur Bestimmung wahlfreier Benutzer- und Leistungsmerkmale.

### 3.2 Schichtenstrukturierung und Funktionen

Um die im vorigen Abschnitt aufgeführten Dienste erbringen zu können, sind die Funktionen und Prozeduren der Empfehlung X.25 hierarchisch gegliederten Schichten zugeordnet.

In diesem Abschnitt sollen die in X.25 vorgenommene Schichtenstrukturierung und die enthaltenen Funktionen, die die Dienste erbringen, näher erläutert werden.

#### 3.2.1 Logische Struktur und Protokollschichtung

Die CCITT-Empfehlung X.25 definiert drei getrennte und voneinander unabhängige Schichten, die entsprechend dem ISO-Referenzmodell (vgl. Kap.2.4.3) die Schichten 1 bis 3 repräsentieren:

- Schicht 1: Bitübertragungsschicht,
- Schicht 2: Sicherungsschicht,
- Schicht 3: Paketschicht.

Es ergibt sich somit eine logische Struktur wie in Bild 3.3 gezeigt.

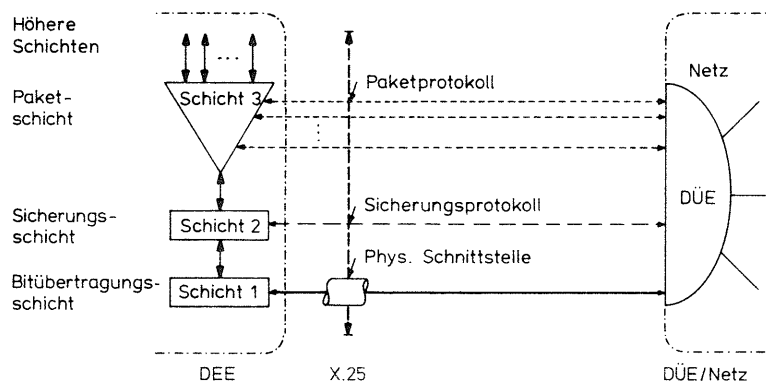


Bild 3.3: Logische Struktur von X.25

Innerhalb dieser Struktur benutzen die Prozeduren der einen Schicht jene Dienstleistungen, die ihnen von der nächst darunterliegenden Schicht angeboten werden (vgl. Kap.2.4.1). Die Art und Weise, wie diese Dienste erbracht werden, d.h. wie letztendlich diese darunterliegende Schicht implementiert ist, ist dabei ohne Belang. Dadurch können Schichten komplett gegen neue, anders realisierte ausgewechselt werden, sofern sie dieselben Funktionen erbringen.

Desweiteren wird in der Empfehlung X.25 eine Protokollschichtung vorgegeben, derart, daß entsprechend Kap.2.4.2 drei voneinander unabhängige Schichtenprotokolle definiert werden, d.h. Protokolle bzw. Regeln zum Informationsaustausch zwischen gleichrangigen Schichten der DEE und der DÜE:

- Schicht 1: Protokoll für die physikalische Übertragung, entsprechend der CCITT-Empfehlung X.21;
- Schicht 2: Protokoll zur Sicherung des Übertragungsabschnittes;
- Schicht 3: Protokoll zur Übertragung von Paketen.

Damit ergibt sich die in Bild 3.4 wiedergegebene Protokollschichtung.

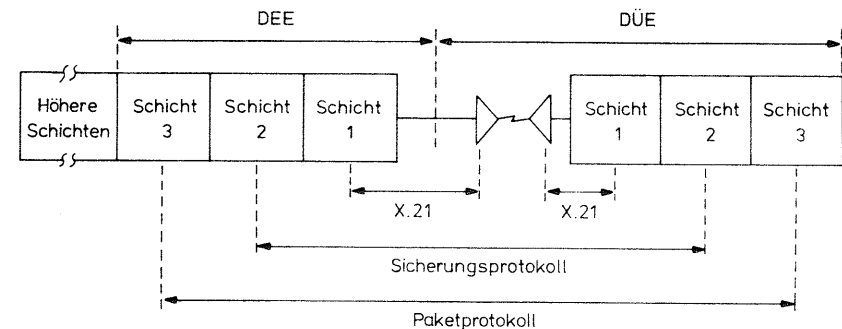


Bild 3.4: Protokollschichtung innerhalb von X.25

Im Zusammenhang mit Protokollen soll nicht unerwähnt bleiben, daß X.25 bezüglich der sog. Dienstprotokolle (vgl. Kap.2.4.2) bislang keinerlei Aussagen macht und es somit jedem Implementierer obliegt, diese geeignet zu definieren.

Bei X.25 akzeptiert jede Schicht die Informationen, die ihr die nächst höhere Schicht anliefert. Konkret heißt das, Schicht 3 akzeptiert Nachrichtenblöcke von Schicht 4, fügt protokollbedingte Steuerinformationen SI(3) hinzu und reicht diese komplettierte Dateneinheit, auf Ebene 3 "Paket" genannt, an Ebene 2 weiter (siehe Bild 3.5).

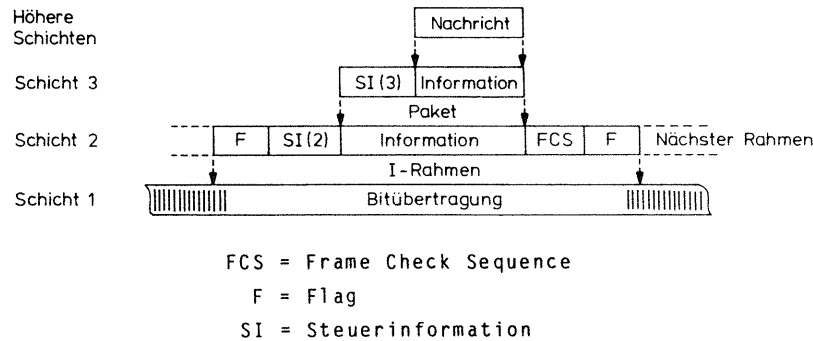


Bild 3.5: Informationstransfer durch die Schichten

Schicht 2 fügt ihrerseits ihren eigenen Steuerteil SI(2) hinzu und übergibt diesen Block, bei Ebene 2 "Rahmen" genannt, an Ebene 1 zur bitweisen Übertragung zur Gegenstation. Dort erfolgt nach fehlerfreier Übertragung die Entnahme der gesendeten Nachricht in umgekehrter Reihenfolge.

### 3.2.2 Bitübertragungsschicht

Die in X.25 definierte Bitübertragungsschicht spezifiziert die unterste Schicht 1 in vollkommener Übereinstimmung mit der CCITT-Empfehlung X.21, /53/.

Die physikalische Netzschnittstelle zwischen DEE und D0E entsprechend X.21 gilt für Endeinrichtungen im Synchronbetrieb (Benutzerklassen 3 bis 7 nach CCITT X.1, /54/). Sie findet gleichermaßen Anwendung in Datennetzen mit Durchschaltvermittlung (vgl. Kap. 2.1.1) als auch mit Paketvermittlung (vgl. Kap.2.1.2).

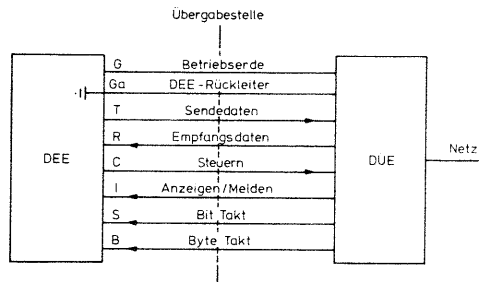
Im einzelnen spezifiziert X.21:

- die physikalischen Eigenschaften der Schnittstelle: Steckerform und -belegung (nach CCITT X.24, /55/: 8 Leitungen mit 15 Stiften),
- die elektrischen Eigenschaften (nach CCITT X.26 und X.27, /56,57/),
- die synchrone, bitserielle Übertragung,
- die Punkt-zu-Punkt, vollen-duplexe Betriebsweise,
- die Ablaufsteuerung für den Aufbau vermittelter Verbindungen sowie
- die Ablaufsteuerung für den Aufbau von Standverbindungen.

Bild 3.6 zeigt die eigentliche physikalische Schnittstelle mit ihren 8 Leitungen und deren Bedeutung.

In der CCITT-Empfehlung X.21 werden insgesamt 30 Zustände definiert, die neben den normalen Betriebsfunktionen auch alle Störfälle mit einschließen.

Auf weitere Einzelheiten und Details soll hier nicht näher eingegangen werden, da diese ohne Belang für die in dieser Arbeit vorgestellten Untersuchungen sind.



- G Betriebserde
- Ga DEE-Rückleiter
- T Sendedaten Übertragung Nutz- und Steuerdaten  
DEE → DUE
- R Empfangsdaten Übertragung Nutz- und Steuerdaten  
DUE → DEE
- C Steuerleitung Zustandsanzeige der DEE an die DUE
- I Meldeleitung Zustandsanzeige der DUE an die DEE
- S Bittakt Versorgung der DEE mit dem Takt des öffentlichen Netzes
- B Bytetakt Synchronisationszeichen bei zeichenweiser Übertragung (optional)

Bild 3.6: Physikalische Schnittstelle X.21

### 3.2.3 Sicherungsschicht

Die in X.25 spezifizierte Sicherungsschicht beschreibt den gesicherten Datenaustausch zwischen DEE und DUE, die für die Benutzerklassen 8 bis 11 (nach CCITT X.1, /54/) eingerichtet sind. Die in dieser Schicht dabei benutzte Terminologie ist dieselbe wie jene der ISO HDLC-Normen (vgl. /58-65/). Schicht 2 arbeitet jedoch nur mit einer Untermenge der in diesen Normen beschriebenen Optionen. Die Summe der Prozeduren in Schicht 2 wird global mit Link Access Procedures (LAPB) bezeichnet, wobei das Kürzel B für Balanced Class of Procedures (Class BA, Optionen 2 und 8 der ISO HDLC-Normen) steht.

Die Haupteigenschaften der Sicherungsschicht von X.25, und damit auch der HDLC Prozeduren, sind:

- bitorientierte, synchrone Datenübertragung,
- codeunabhängige (codetransparente) Übertragung,
- gesicherte Übertragung durch Folgenummersteuerung und Blockprüfzeichen (Fehlererkennung und Rahmenwiederholung),
- Datenaustausch im Vollduplex Betrieb,
- gleichzeitige Quittierung mehrerer Rahmen.

Die Schicht 2 führt somit den fehlerbehafteten Übertragungsabschnitt in einen (quasi) fehlerfreien, d.h. mit einer sehr geringen Restfehlerwahrscheinlichkeit behafteten Abschnitt, über.

Die zur Erlangung der aufgeführten Eigenschaften notwendigen Spezifikationen der Sicherungsschicht gliedern sich grob in drei Teile: Prozedurklassen, Rahmenstrukturen und Prozedurelemente.

Prozedurklassen: Die HDLC-Normen unterscheiden hier zwei Arten der Prozedurklassen (siehe hierzu Bild 3.7):

- Unbalanced Class of Procedures
- Balanced Class of Procedures.

Unbalanced Class of Procedures: Diese steuern Verbindungen zwischen einer Leitsteuerung (primary station) und einer oder mehreren Folgesteuern (secondary station):

- NRM: Normal Response Mode; die Folgesteuerung kann nur nach Aufforderung durch die Leitsteuerung senden und gibt nach Übertragungsende die Sendeberechtigung wieder ab;
- ARM: Asynchronous Response Mode; beide Stationen können ohne Aufforderung zu senden beginnen, wobei die Verbindungssteuerung der Leitstation obliegt.

Balanced Class of Procedures: Diese steuern Punkt-zu-Punkt Verbindungen zwischen Hybridstationen (combined station):

- ABM: Asynchronous Balanced Mode; die Steuerung der Verbindung ist durch beide Stationen möglich; es ist keine Sendeaufforderung erforderlich.

Bei den in der Sicherungsschicht von X.25 definierten Prozeduren handelt es sich ausschließlich um solche der zuletzt aufgeführten Klasse der "Balanced Class of Procedures".

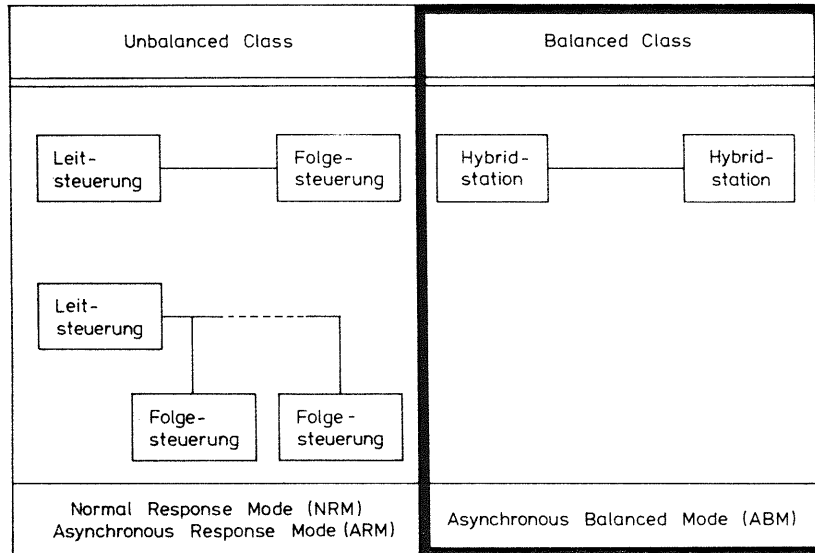


Bild 3.7: Prozedurklassen der HDLC-Normen

Bild 3.7 ( aus /20/) zeigt eine graphische Zusammenfassung der verschiedenen in den HDLC-Normen spezifizierten Prozeduren sowie (stark umrandet) den in X.25 Schicht 2 ausschließlich zur Anwendung kommenden Betrieb.

Rahmenstrukturen: Das grundlegende Element der HDLC-Prozeduren ist der Rahmen (frame). Der Datenaustausch zwischen den beteiligten Stationen erfolgt ausschließlich mit diesen Rahmen, wobei zwei prinzipielle Formate unterschieden werden (Bild 3.8):

- Rahmen für den Informationsaustausch,
- Rahmen für die Prozedursteuerung.

Flag	Adressfeld	Steuerfeld	Datenfeld	Blockprüfzeichen	Flag
01111110	8 Bit	8 Bit	Paketlänge	16 Bit	01111110

a)

Flag	Adressfeld	Steuerfeld	Blockprüfzeichen	Flag
01111110	8 Bit	8 Bit	16 Bit	01111110

b)

Bild 3.8: Rahmenstruktur: Informationsrahmen (a),  
Steuerrahmen (b)

Die Bestandteile der Rahmen sind:

- Flags zur Rahmenbegrenzung,
- Adresse zur Unterscheidung der Nachrichtenart:
  - Befehle enthalten die Adresse des Empfängers,
  - Meldungen enthalten die Adresse des Senders,
- Steuerfeld zur Kennzeichnung der Rahmenart und zum Übermitteln der Folgenummern,
- Datenfeld (nur bei Informationsrahmen): enthält das zu übertragende Paket aus Schicht 3, wobei Länge und Bitfolge beliebig sein können,
- Blockprüfzeichen: enthält die aus Adreß-, Steuer- und Datenfeld errechneten Kontrollbits für die Fehlererkennung auf der Empfängerseite.

Prozedurelemente: Die HDLC-Normen spezifizieren verschiedene Befehls- und Meldungstypen sowie die Interaktionen und Reaktionen, die nach Empfang bestimmter Rahmen abzulaufen haben.

Ohne Anspruch auf Vollständigkeit soll hier nur kurz auf einige wenige Details eingegangen werden:

- Folgenummernsteuerung/Fenstermechanismus (vgl. Kap.2.3.2 und Bilder 2.6 bis 2.8) mit:
  - Sendefolgenummer N(S) (in Kap.2.3.2 mit S bezeichnet),
  - Empfangsfolgenummer N(R) (in Kap.2.3.2 mit E bezeichnet),
  - MODULUS = 8,
  - Sendefenstergröße  $w_S = 7$ ,
  - Empfangsfenstergröße  $w_E = 1$ ,
- Fehlererkennung und -behebung entsprechend der aus /7/ entnommenen Tabelle 3.1:

Störungsursache	mindestens 1 Bit gefälscht; durch FCS erkannt; Rahmen wird verworfen		
Fehlererkennung durch	N(S)-Folgefehler (sequence error)	Kontrollpunktverfahren (checkpointing)	Time-Out
Erkennung bei	Empfänger	Sender	Sender
Fehlererkennung infolge	nicht der erwartete I-Rahmen: $N(S) \neq V(E)$	mit F-Bit nicht alle I-Rahmen vor P-Bit quittiert: N(R) mit F-Bit $\leq$ N(S) vor P-Bit	keine Quittung innerhalb vorgegebener Zeit
Reaktion	fehlenden Rahmen mit REJ od. SREJ anfordern (REJ-recovery, SREJ-recovery)	ab letztem quittierten Rahmen alles wiederholen (P/F-recovery)	sofort neues Kontrollpunktverfahren beginnen (checkpointing)

Tabelle 3.1: Fehlererkennung und -behebung

Weitere Einzelheiten und detaillierte Beschreibungen der Elemente und Funktionsabläufe innerhalb der Schicht 2 von X.25 sind aus /58/ bis /65/ zu entnehmen. /66/ bis /73/ stellt einen Teil der Veröffentlichungen dar, die die Analyse von HDLC zum Inhalt haben.

### 3.2.4 Paketschicht

Dieser Abschnitt soll sowohl den in X.25 vorgegebenen Leistungsumfang der Paketschicht als auch eine detaillierte Beschreibung der Prozeduren und Prozedurelemente aufzeigen, um den Forderungen, die an Schicht 3 gestellt werden, gerecht zu werden.

#### 3.2.4.1 Leistungsumfang

Die Empfehlung X.25 definiert für Schicht 3 die Pakettypen und -formate sowie die Zustandsübergänge, die durch den Empfang bzw. das Senden von Paketen verursacht werden. Die Prozeduren umfassen zudem den zeitlichen Ablauf der zu übertragenden Daten.

Durch die Paketschicht werden folgende Dienste bereitgestellt:

- Auf- und Abbau virtueller Verbindungen,
- Multiplexen/Demultiplexen mehrerer logischer Kanäle (und damit auch mehrerer virtueller Verbindungen),
- Datagramservice,
- Folgenummernsteuerung zur Einhaltung und Kontrolle der Paketreihenfolge, getrennt für jede Verbindung,
- verbindungsindividuelle Datenflußsteuerung am Netzzugang,
- wahlweise verbindungsindividuelle Ende-zu-Ende Datenflußsteuerung,
- Fehlererkennung und -behandlung,
- Wahl bestimmter Benutzer- und Leistungsmerkmale.

Im direkten Vergleich mit den in Kap.2.4.3 aufgezählten Aufgaben der einzelnen Schichten des ISO-Referenzmodells läßt sich jedoch unschwer erkennen, daß die Paketschicht von X.25 folgende, in Ebene 3 des Referenzmodells vorgesehene Funktionen nicht abdeckt:

- Verkehrslenkung beim Verbindungsaufbau,
- Vermittlung der Pakete während der Datenübertragungsphase.





Oktett	8 7 6 5	4 3 2 1	Bits
1	Kennzeichen für Grundformat	Logische Kanalgruppennummer	
2	Logische Kanalnummer		
3	Kennzeichen für Pakettyp		
4			
	:		

Bild 3.10: Paket-Grundstruktur

Die Oktetts 1, 2 und 3 eines jeden Paketes bilden den sog. Paketkopf.

Die Bits 5 und 6 des 1. Oktetts bestimmen dabei den Wertebereich, d.h. den MODULUS-Wert der Folgenummernsteuerung. Sie bilden somit das Kennzeichen für das Grundformat (general format identifier) entsprechend Tabelle 3.2 (aus /2/).

Die Bits 1 bis 4 von Oktett 1 kennzeichnen die logische Kanalgruppennummer, während Oktett 2 die logische Kanalnummer innerhalb der Gruppe angibt (vgl. Kap.3.2.4.2).

Das dritte Oktett des Paketkopfes gibt schließlich die eindeutige Kennzeichnung des verwendeten Pakettyps (packet type identifier) entsprechend der aus der CCITT-Empfehlung, /2/, entnommenen Tabelle 3.3,

General format identifier		Octet 1 Bits			
		8	7	6	5
Call set-up packets	Sequence numbering scheme modulo 8	0	X	0	1
	Sequence numbering scheme modulo 128	0	X	1	0
Clearing, datagram, flow control, interrupt, reset, restart and diagnostic packets	Sequence numbering scheme modulo 8	0	0	0	1
	Sequence numbering scheme modulo 128	0	0	1	0
Data packets	Sequence numbering scheme modulo 8	X	X	0	1
	Sequence numbering scheme modulo 128	X	X	1	0
Datagram service signal packets	Sequence numbering scheme modulo 8	1	0	0	1
	Sequence numbering scheme modulo 128	1	0	1	0
General format identifier extension		*	*	1	1

\* Undefined.

Note - A bit which is indicated as "X" may be set to either 0 or 1 as indicated in the text.

Tabelle 3.2: Kennzeichen für das Grundformat

Packet type		Octet 3 Bits							
From DCE to DTE	From DTE to DCE	8	7	6	5	4	3	2	1
<i>Call set-up and clearing</i>									
Incoming call	Call request	0	0	0	0	1	0	1	1
Call connected	Call accepted	0	0	0	0	1	1	1	1
Clear indication	Clear request	0	0	0	1	0	0	1	1
DCE clear confirmation	DTE clear confirmation	0	0	0	1	0	1	1	1
<i>Data and interrupt</i>									
DCE data	DTE data	X	X	X	X	X	X	X	0
DCE interrupt	DTE interrupt	0	0	1	0	0	0	1	1
DCE interrupt confirmation	DTE interrupt confirmation	0	0	1	0	0	1	1	1
<i>Datagram <sup>a)</sup></i>									
DCE datagram	DTE datagram	X	X	X	X	X	X	X	0
Datagram service signal		X	X	X	X	X	X	X	0
<i>Flow control and reset</i>									
DCE RR (modulo 8)	DTE RR (modulo 8)	X	X	X	0	0	0	0	1
DCE RR (modulo 128) <sup>a)</sup>	DTE RR (modulo 128) <sup>a)</sup>	0	0	0	0	0	0	0	1
DCE RNR (modulo 8)	DTE RNR (modulo 8)	X	X	X	0	0	1	0	1
DCE RNR (modulo 128) <sup>a)</sup>	DTE RNR (modulo 128) <sup>a)</sup>	0	0	0	0	0	1	0	1
	DTE REJ (modulo 8) <sup>a)</sup>	X	X	X	0	1	0	0	1
	DTE REJ (modulo 128) <sup>a)</sup>	0	0	0	0	1	0	0	1
Reset indication	Reset request	0	0	0	1	1	0	1	1
DCE reset confirmation	DTE reset confirmation	0	0	0	1	1	1	1	1
<i>Restart</i>									
Restart indication	Restart request	1	1	1	1	1	0	1	1
DCE restart confirmation	DTE restart confirmation	1	1	1	1	1	1	1	1
<i>Diagnostic</i>									
Diagnostic <sup>a)</sup>		1	1	1	1	0	0	0	1

Tabelle 3.3: Kennzeichen für Pakettyp

Im folgenden sollen zwei Pakettypen, die für die hier angeführten Untersuchungen von Bedeutung sind, bezüglich ihrer Funktion und ihres Formats eingehender erläutert werden.

Paket für Datentransfer (data packet): Mit Hilfe dieses Pakets werden nach vollendetem Verbindungsaufbau die Benutzernachrichten aus Schicht 4 über den Übertragungsabschnitt geleitet. Die logische Kanalgruppennummer sowie die logische Kanalnummer stehen fest, Adressen werden nicht mehr benötigt, da die virtuelle Verbindung, d.h. die Verknüpfung der an der Verbindung beteiligten logischen Kanäle, in den Steuerdatenbereichen der Vermittlungsknoten abgespeichert ist (vgl. Kap.2.2.1).

Das Datenpaket hat damit folgenden Gesamtaufbau (vgl. Bild 3.11, aus /2/):

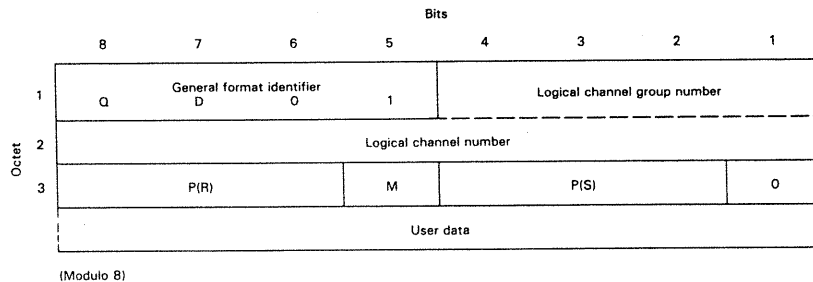


Bild 3.11: Datenpaket

Q-Bit (qualifier bit):

Unterscheidungsmerkmal zur Benutzung ein- und desselben logischen Kanals auf zweierlei Ebenen,

Q = 0 : normale Benutzerdaten,

Q = 1 : priorisierte Daten,

D-Bit (delivery confirmation bit):

Unterscheidungsmerkmal für die Art des Quittierverfahrens,

D = 0 : lokale Quittierung,

D = 1 : Ende-zu-Ende Quittierung,

M-Bit (more data bit):

Kennzeichnung zusammenhängender Paketfolgen,

P(R) (packet receive sequence number):

Paketempfangsfolgennummer für die Datenflußsteuerung (siehe Kap.3.2.4.4 und Kap.2.3.2),

P(S) (packet send sequence number):

Paketsendefolgennummer für die Datenflußsteuerung (siehe Kap.3.2.4.4 und Kap. 2.3.2),

Daten (user data):

von Schicht 4 transparent übernommene Daten, Längenbeschränkung siehe Kap.3.2.4.4.

Zur näheren Erläuterung der Bedeutung der Q-, D- und M-Bits sei auf /2/ verwiesen, da diese für die in dieser Arbeit vorgestellten Untersuchungen ohne weiteren Belang sind.

Paket zur Datenflußsteuerung (flow control packet): Der Tabelle 3.3 kann entnommen werden, daß zwischen zwei prinzipiell unterschiedlichen Pakettypen zur Datenflußkontrolle unterschieden werden kann:

- receive ready packet: RR-Paket,
- receive not ready packet: RNR-Paket.

RNR-Pakete werden in aller Regel dazu verwendet, der Gegenstation einen momentanen Zustand anzuzeigen, in welchem die das RNR-Paket sendende Station keine weiteren Pakete auf dem angezeigten logischen Kanal mehr empfangen kann.

RR-Pakete werden einerseits gesendet, um der Gegenstation zu melden, daß der bisherige RNR-Zustand dieses im RR-Paket angezeigten logischen Kanals wieder aufgehoben ist, die Station also wieder aufnahmefähig ist und andererseits, um korrekt empfangene Pakete zu quit-

tieren, momentan jedoch keine Benutzerdaten aus Schicht 4 für diesen logischen Kanal zum Senden anstehen und somit kein Datenpaket gesendet werden kann. Bild 3.12 (aus /2/) zeigt den Aufbau eines RR-Pakets, wobei die Funktionen der einzelnen Felder innerhalb der Oktetts den o.a. Beschreibungen entnommen werden können.

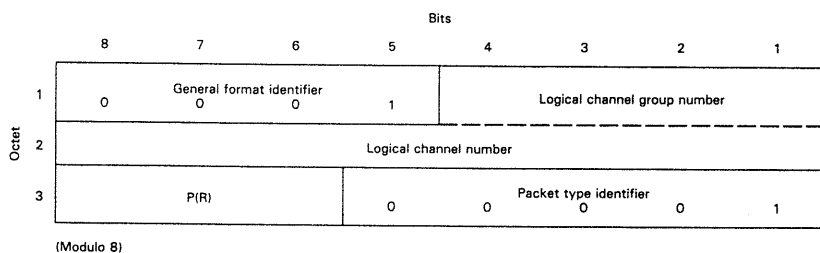


Bild 3.12: RR-Datenflußsteuerpaket

Jedes Paket, gleichgültig welchen Typs, wird nach seiner Fertigstellung in Schicht 3 beim Sendevorgang an Schicht 2 übergeben, die dieses dann völlig transparent innerhalb eines Informationsrahmens (vgl. Kap.3.2.3 und Bild 3.8(a)) gesichert überträgt. In umgekehrter Reihenfolge übergibt Schicht 2 ein korrekt empfangenes Paket an Schicht 3 weiter, die dann dieses individuell für jede logische Kanalnummer auswertet, weiterbearbeitet und eventuelle Daten an Schicht 4 weiterreicht.

#### 3.2.4.4 Datentransfer

Sende- und Empfangsfolgennummern: Die Datentransferphase, die sich unmittelbar an die Aufbauphase anschließt, dient dem Transport der von Schicht 4 angebotenen Benutzernachrichten innerhalb des Datenfeldes der Datenpakete (vgl. Bild 3.11). Schicht 3 bewerkstelligt hierzu eine individuelle und für jede virtuelle Verbindung unabhängige Datenflußsteuerung in Form von Sende- und Empfangsfolgennummern.

Die gemäß X.25 definierte Folgennummernsteuerung arbeitet entsprechend dem in Kap.2.3.2 erwähnten Verfahren mit Sendefolgennummern P(S) (in Kap.2.3.2 allgemein nur mit S bezeichnet) und Empfangsfolgennummern P(R) (in Kap.2.3.2 mit E bezeichnet) mit den dort erwähnten Bedeutungen und Funktionen. In der Regel werden diese Folgennummern mit MODULUS = 8 durchgezählt; eine Alternative erlaubt auch MODULUS = 128.

Es ist offensichtlich, daß beide Stationen, DEE und DOE, Folgezähler einrichten müssen, analog V(S) und V(E) aus Kap.2.3.2. Diese Folgezähler werden bei X.25 nur in Schicht 2 explizit definiert, nicht jedoch in Schicht 3.

Fenstermechanismus: Die Verwendung von Folgennummern dient in Schicht 2 vorrangig der Fehlererkennung und -behebung. Da virtuelle Verbindungen in Schicht 3 auf einem (quasi) fehlerfreien Übertragungsabschnitt existieren, dient die Folgennummernsteuerung in Schicht 3 in erster Linie der Datenflußsteuerung, d.h. der Vermeidung von Blockierungen, indem die Anzahl vom Netz akzeptierter Pakete limitiert wird. Aus diesem Grund wird für jede aufgebaute virtuelle Verbindung ein unabhängiger Fenstermechanismus, entsprechend Kap.2.3.2, eingerichtet. Die Fenstergröße  $w_S$  gibt dabei die maximale Anzahl an Paketen an, die die DEE bzw. die DOE zu jeder Zeit und für jeden logischen Kanal unquittiert ausstehen haben darf. Der Wert von  $w_S$  wird dabei entweder einmalig von der Postverwaltung beim Einrichten des X.25-Anschlusses vorgegeben, oder er kann beim Verbindungsaufbau vom Benutzer selbst gewählt werden. Dabei steht folgender Wertebereich zur Verfügung:

$$1 \leq w_S \leq \text{MODULUS}-1.$$

Der Wert  $w_E$  des Empfangsfolgefensers hingegen ist bei X.25 konstant 1, d.h. der Empfänger akzeptiert, für jede virtuelle Verbindung unabhängig, Pakete nur in streng zyklischer Reihenfolge (vgl. Kap.2.3.2).

In der Regel ist die Flußsteuerung der Schicht 3 von X.25 von lokaler Bedeutung. Die Annahme von Datenpaketen durch die lokale DÖE bedeutet nicht, daß die Ziel-DEE ebenfalls zugestimmt hat, diese Pakete zu übernehmen. Vielmehr bedeutet das Akzeptieren der Pakete, daß das Netz genügend Speicher zur Verfügung gestellt hat, so daß nun auf die Annahme durch die Ziel-DEE gewartet werden kann. Die Fenstergrößen  $w_S$  der Sendefenster werden also lokal gewählt und können somit auch an beiden Enden der virtuellen Verbindung verschieden sein.

Für den Transport von Quittungen, d.h. für das Senden der Empfangsfolgennummern  $P(R)$ , sieht die CCITT-Empfehlung mehrere Möglichkeiten vor:

- Datenpakete, d.h. gleichzeitiges Übertragen von Benutzerdaten und von Quittierinformation ("piggy-backing"),
- Steuerpakete, d.h. ausschließliches Übertragen von Quittierinformation:
  - RR-Paket: der Sender ist empfangsfähig, hat aber entweder keine Benutzerdaten für diese virtuelle Verbindung zum Senden, oder das Sendefenster ist für diese virtuelle Verbindung momentan geschlossen,
  - RNR-Paket: der Sender ist auf dem angezeigten logischen Kanal nicht mehr empfangsfähig, quittiert jedoch die noch empfangenen und aufgenommenen Pakete mit Sendefolgennummern bis  $P(S) = P(R) - 1$ .

Paketlänge: Entsprechend X.25 kann das Datenfeld im Datenpaket jede beliebige Anzahl Bits bis zu einer definierten oberen Grenze enthalten. Diese obere Grenze kann entweder während des Verbindungsaufbaus gewählt oder bei Einrichten des X.25-Anschlusses festgelegt werden. Die bevorzugte und empfohlene Länge des Datenfeldes umfaßt 128 Bytes. Zur Auswahl stehen die Alternativen: 16, 32, 256, 512 und 1024 Bytes.

### 3.3 Parameter

Dieser Abschnitt soll eine Übersicht über Parameter geben, die sowohl die Funktionen und das Verhalten als auch die Leistungsfähigkeit der Schnittstelle X.25 beeinflussen.

#### 3.3.1 Struktur- und Verkehrsparameter

Typische leistungsbeeinflussende Struktur- und Verkehrsparameter sind:

- Anzahl aufgebauter und aktiver virtueller Verbindungen,
- physikalische Laufzeit auf den Verbindungsleitungen zwischen DEE und DÖE,
- physikalische Übertragungsgeschwindigkeit (Taktfrequenz) auf der Leitung,
- Prozessorleistung für die Prozedursteuerungen in den Schichten 1 bis 3,
- Aufwandsverhältnis Hardware/Software,
- Verkehrsverhalten der höheren Schichten bezüglich der Erzeugung und Übergabe von Daten an bzw. der Abnahme von Daten von Schicht 3.

#### 3.3.2 Optionale Benutzer- und Leistungsmerkmale

Die CCITT-Empfehlung sieht für Schicht 3 einige optionale Benutzer- und Leistungsmerkmale vor, die sich unter Umständen sehr stark auf die Leistungsfähigkeit der Schnittstelle auswirken können.

Die Merkmale lassen sich dabei in zwei Kategorien einordnen, und zwar in

- solche Merkmale, die den gesamten Abschnitt DEE/DÖE anbelangen und zum Zeitpunkt des Einrichtens des X.25-Anschlusses fest vorgegeben werden, und in

- solche, die sich auf eine einzelne virtuelle Verbindung beziehen und während des Verbindungsaufbaus individuell mittels des Verbindungsaufbaupakets gewählt werden.

Einige typische optionale Benutzer- und Leistungsmerkmale sind:

- Wahl des MODULUS-Wertes für die Folgenummernsteuerung,
- Wahl der Fenstergröße  $w_s$ ,
- Wahl der maximalen Paketlänge,
- Wahl bestimmter Durchsatzklassen,
- Bildung geschlossener Teilnehmergruppen,
- Gebührenübernahme,
- Wahl der Fast Select Eigenschaften,
- Wahl des Quittierungsverfahrens (D-Bit Modifikation),
- Wahl des Datagram-Verbindungskonzepts.

#### 4. IMPLEMENTIERUNG DER CCITT-NETZZUGANGSSCHNITTSTELLE X.25

Um funktionale Eigenschaften der CCITT-Netzzugangsschnittstelle X.25 real aufzeigen zu können, wurden die beiden Schichten 2 und 3 der Dateneneinrichtung (DEE) und der Datenübertragungseinrichtung (DOE) von X.25 in Hard- und Software implementiert. Da X.25 während der Verbindungsaufbauphase über insgesamt zwei Dateneneinrichtungen, zwei Datenübertragungseinrichtungen sowie über das dazwischenliegende Netz hinweg arbeitet (vgl. hierzu /2/ und Bild 3.2), aus technischen Gründen jedoch nur eine Netzzugangsschnittstelle mit einer DEE und einer DOE realisiert werden konnte, mußten für den Signalisierungsaustausch während der Verbindungsaufbauphase leichte Modifikationen gegenüber X.25 vorgenommen werden. Die Implementierung erfolgte derart, daß an der DEE drei Teilnehmer angeschlossen werden können. Entsprechend X.25 bildet die DOE bereits einen Bestandteil des Datennetzes (vgl. Bild 3.1), da jedoch bei der vorliegenden Implementierung kein Netz zur Verfügung stand, wurde der Weg beschritten, daß auch an der DOE Anschlußmöglichkeiten für insgesamt drei Teilnehmer geschaffen wurden. Teilnehmer der DEE können dabei wahlfrei mit Teilnehmern der DOE kommunizieren, nicht jedoch mit Teilnehmern der eigenen Einrichtung. Für die Gegenrichtung gelten die entsprechenden Bedingungen.

Während der Daten- und Informationsaustausch zwischen gleichwertigen Schichten in X.25 eindeutig durch Schichtenprotokolle (vgl. Kap.2.4.2) festgelegt ist, hatte die Entwicklung des systeminternen Dienstprotokolls zwischen benachbarten Schichten (adjacent layer protocol; vgl. Kap.2.4.2) einen besonderen Stellenwert.

Ziel dieses Kapitels ist es, das zur Implementierung von X.25 gewählte Prozeßkonzept aufzuzeigen sowie oben angeführtes Dienstprotokoll zu erläutern. Desweiteren soll dieses Kapitel einen Einblick in die Implementierungsstruktur sowohl der Hardware als auch der Software vermitteln.

### 4.1 Prozeßkonzept

Um dem in Kap.2.4.1 erläuterten grundlegenden und speziell in X.25 (vgl. Kap.3.2.1) zur Anwendung kommenden Architekturprinzip der Schichtung insoweit Rechnung zu tragen, daß einerseits die Art und Weise der Dienstleistungserbringung für höhere Schichten unabhängig von der Realisierungsform der dienstleistungserbringenden Schicht sein soll und andererseits Schichten komplett gegen neue, anders realisierte austauschbar sein sollen, wurden bei der hier beschriebenen Realisierung von X.25 die beiden Schichten 2 und 3 auf getrennt arbeitenden Mikroprozessorsystemen implementiert, (vgl. hierzu auch /74/ bis /76/).

Dabei kamen für Schicht 2 der 8-Bit Mikroprozessor MC6800 mit 1MHz CPU-Takt und für Schicht 3 der leistungsstärkere 16-Bit Prozessor MC68000 mit 8MHz CPU-Takt, beide von der Firma Motorola, zur Anwendung, /77/ bis /79/.

Um einen systematisch strukturierten Aufbau der Netzchnittstelle X.25 zu erhalten, wurde ein Prozeßkonzept einer DEE bzw. DOE entwickelt mit klar definierten Eingangs- und Ausgangspunkten jedes einzelnen Prozeßmoduls, vgl. Bild 4.1.

Bild 4.1 zeigt sowohl das gesamte Prozeßkonzept, das der Implementierung von X.25 zugrundegelegt worden ist, als auch die Zugehörigkeit der einzelnen Prozesse zu den beiden Schichten und damit das Aufteilen derselben auf die beiden verwendeten Mikroprozessorsysteme. Eine Ausnahme bilden die beiden Prozesse "FCS-Erzeugung, Übertragung" und "FCS-Erkennung, Empfang", da diese auf einem separaten, bidirektional betreibbaren Kommunikationsbaustein ablaufen, der unabhängig von dem verwendeten, übergeordneten Prozessorsystem ist. Da dieser Baustein einerseits Aufgaben der Schicht 2 (Flagerzeugung und -erkennung, FCS-Erzeugung und -Erkennung) und andererseits Aufgaben der Schicht 1 (bitweises, seriell übertragen) ausführt, sind diese beiden Prozesse sowohl der Schicht 2 als auch der Schicht 1 zugeordnet.

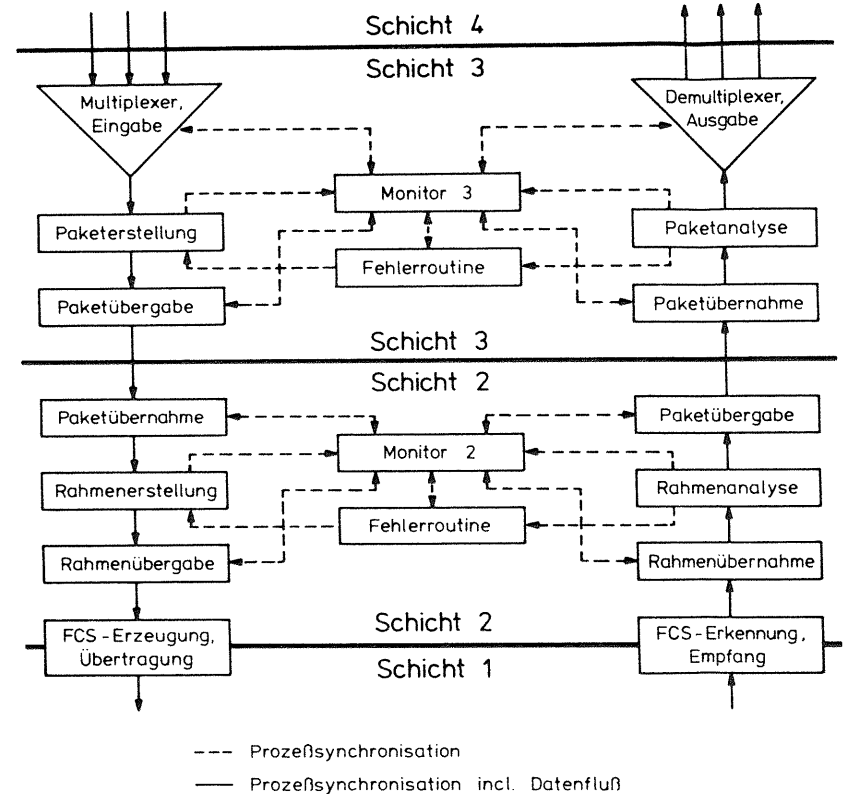


Bild 4.1: Prozeßkonzept einer DEE bzw. DOE

Auf eine detaillierte Aufgabenbeschreibung der weiteren Prozesse soll im Rahmen des hier beschriebenen Prozeßkonzepts verzichtet werden, da die in Bild 4.1 gewählten Bezeichnungen der einzelnen Prozesse zum Verständnis des Gesamtkonzepts ausreichen.

Die Aktivierung der Prozesse erfolgt entweder direkt über die Monitore oder über die Dienstzugangspunkte der Schichten 2 und 3 und die funktionelle Kopplung zwischen den Prozessen.

Da immer nur ein Prozeß pro Schicht auf dem Prozessor abläuft und jede Schicht ihren eigenen zugehörigen Speicher hat, treten keinerlei Konflikte auf.

In der Implementierung wurde jedem Prozeß ein separater Programmteil zugeordnet, so daß es ohne weiteres möglich ist, Prozeßrealisierungen durch andere auszutauschen, sofern Ein- und Ausstieg in bzw. aus dem jeweiligen Prozeß übereinstimmen.

#### 4.2 Dienstprotokoll

Wie eingangs bereits erwähnt, wurde bei der hier beschriebenen hard- und softwaremäßigen Realisierung von X.25 besonderer Wert auf das systemintern ablaufende Dienstprotokoll gelegt.

Um der heutigen Sicht formal spezifizierter Dienstprotokolle zu entsprechen, wurden in Anlehnung an /7/ und /11/ die in Tabelle 4.1 aufgeführten Grundsignale, im weiteren Verlauf mit Primitive bezeichnet, definiert, über welche der Datenaustausch zwischen den Schichten 2 und 3 erfolgt.

In der in dieser Arbeit vorgestellten Realisierung werden vier Klassen von Primitiven unterschieden:

- Request Primitive
- Indication Primitive
- Response Primitive
- Confirmation Primitive.

Request Primitive von Schicht 3 fordern dabei von Schicht 2 einen expliziten Dienst an, welcher über Confirmation Primitive bestätigt werden kann. Indication Primitive zeigen dagegen an, daß Schicht 2 von sich aus Schicht 3 einen bestimmten Dienst anbietet und auf dessen Abruf durch Response Primitive wartet.

Primitiv	von	nach
LAP-Connect Request	Schicht 3	Schicht 2
LAP-Connect Indication	Schicht 2	Schicht 3
LAP-Connect Response	Schicht 3	Schicht 2
LAP-Connect Confirmation	Schicht 2	Schicht 3
LAP-Disconnect Request	Schicht 3	Schicht 2
LAP-Disconnect Indication	Schicht 2	Schicht 3
LAP-Data Request	Schicht 3	Schicht 2
LAP-Data Indication	Schicht 2	Schicht 3
LAP-Error Indication	Schicht 2	Schicht 3

Tabelle 4.1: Grundsignale (Primitive) zwischen Schicht 2 und Schicht 3

Weiterhin ist aus Tabelle 4.1 ersichtlich, daß bei der gewählten Implementierung vier Gruppen von Primitiven bestehen, die

- den Auf- und Abbau von Schicht-2-Verbindungen betreffen,
- den Datenaustausch zwischen Schicht 2 und Schicht 3 bewirken und
- gezielte Fehlermeldungen an Schicht 3 weiterleiten.

Die Gruppe der "Data"-Primitive dient dazu, die von Schicht 3 generierten Pakete (entsprechend Tabelle 3.3) an Schicht 2 als deren Nutzinformation weiterzureichen - daher auch die zeitlich zuvor erforderliche Schicht-2-Aufbauinitialisierung mittels der Establish Request Primitive - bzw. dazu, die von Schicht 2 angebotene Nutzinformation, d.h. die angebotenen Pakete, an Schicht 3 weiterzuleiten.



Die technische Realisierung der in Tabelle 4.1 aufgeführten Primitive erfolgt für die Gruppe der Auf- und Abbau- sowie der Fehler-Primitive durch separate Steuerleitungen zwischen den den Schichten 2 und 3 zugeordneten Prozessorsystemen. Die verbleibenden Daten-Primitive sind durch entsprechend zu interpretierende Interrupts realisiert.

#### 4.3 Implementierungsstruktur

In diesem Abschnitt soll auf die in der Implementierung von X.25 zur Anwendung kommenden Strukturen bezüglich Hard- und Software näher eingegangen werden, um somit einen Einblick in die technische Realisierung zu vermitteln.

##### 4.3.1 Hardware

Die schaltungstechnischen Realisierungen der Schichten 2 und 3 von X.25 basieren aufgrund vorhandener Entwicklungssysteme auf den von der Firma Motorola entwickelten Mikroprozessorfamilien MC6800 für Schicht 2 bzw. MC68000 für Schicht 3, /77,78/.

Die Suche nach einem kommerziellen Anbieter, welcher leistungsfähige Baugruppen der Schaltungsfamilie MC68000 anbietet, führte zu dem Produkt MicroSys CPU 01, /79/, der Firma MicroSys GmbH, München. Diese fertige Platine erfüllt alle notwendigen technischen Voraussetzungen, die für die Realisierung der Schicht 3 von X.25 gefordert werden; die zusätzliche Entwicklung ergänzender Hardware konnte somit entfallen.

Die MicroSys CPU 01 umfaßt neben der 16-Bit CPU MC68000 und weiteren hochintegrierten Schaltungen den seriellen Schnittstellenbaustein MC68564 (ACIA, Asynchronous Communications Interface Adapter) und den parallelen Schnittstellenbaustein MC68230 (PIT, Parallel Interface/Timer), vgl. hierzu Bild 4.2.

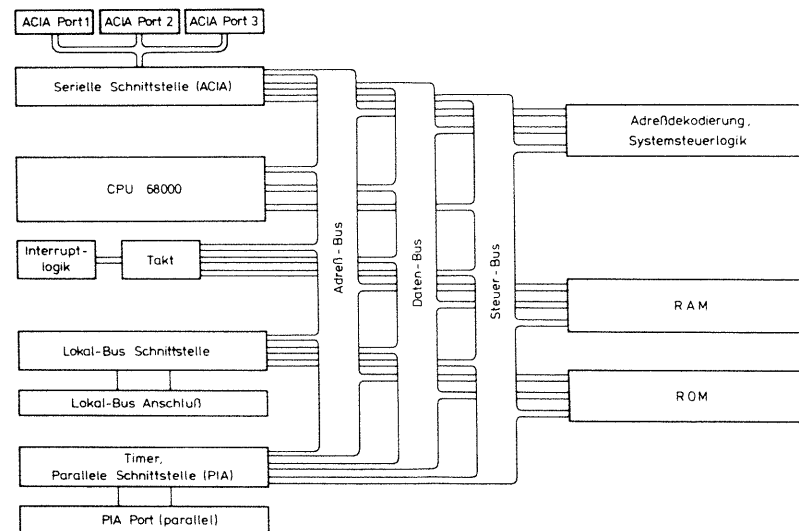


Bild 4.2: Hardwarestruktur in Schicht 3 einer DEE bzw. DOE

Diese Bausteine stellen dem Anwender zum Anschluß notwendiger Peripheriegeräte (Terminals, Drucker o.ä.) drei serielle Schnittstellen (ACIA Port 1 bis ACIA Port 3), je einen 8-Bit und 24-Bit Timer sowie 23 Ein-/Ausgabeleitungen (PIA, Parallel Interface Adapter) zur Verfügung.

Der gesamte Adreßbereich erstreckt sich über insgesamt 16 MByte, wovon 4 MByte über den lokalen Bus adressierbar sind. Die auf der Karte befindlichen RAM- und ROM-Bereiche betragen maximal 256 KByte bzw. 64 KByte.

Die realisierte Hardwarestruktur wird von Bild 4.2 wiedergegeben; aus Tabelle 4.2 sind die wichtigsten Leistungsmerkmale der MicroSys CPU 01 ersichtlich.

16-Bit CPU MC68000
- 16 Daten-, Adreß- und Stackregister (32 bit)
- 14 Adressierungsarten
- 16 MByte Adreßbereich
- 56 Befehlstypen
- 5 Datentypen: Bit, Byte, BCD, Word, Longword
- 256 Vektor-Interrupts
8 MHz CPU-Takt
Doppeleuropa-Format
Bis zu 64 KByte ROM On-Board
Bis zu 256 KByte RAM On-Board
- 128 KByte standard
- 256 KByte optional
3 serielle Schnittstellen
23 Ein-/Ausgabeleitungen (parallel)
2 Timer
- ein 8-Bit Timer
- ein 24-Bit Timer/Zähler
7 Interrupt-Ebenen für den lokalen Bus
Von außen bedienbare Steuertasten
- Reset
- Test
- Abort

Tabelle 4.2: Leistungsmerkmale der Schicht-3-Hardware

Die technische Realisierung der Schicht 2 basiert auf einer am Institut für Nachrichtenvermittlung und Datenverarbeitung entwickelten Platine im Einfacheuropa-Format für den 8-Bit Mikroprozessortyp MC6800. Diese Platine umfaßt dabei neben der CPU MC6800 und weiteren hochintegrierten Schaltungen den Timerbaustein MC6840 mit drei unabhängigen Timern, vgl. hierzu Bild 4.3.

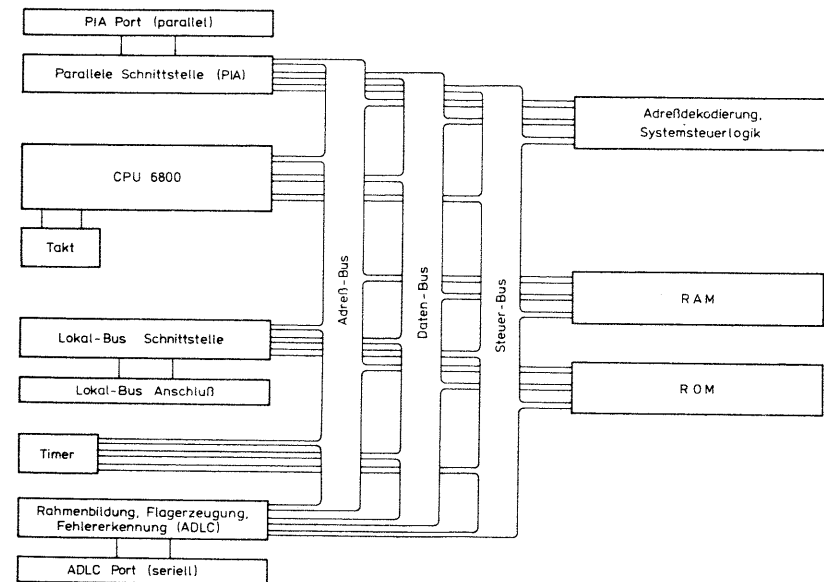


Bild 4.3: Hardwarestruktur in Schicht 2 einer DEE bzw. DOE

Um den gestellten Anforderungen zur Implementierung der Schicht 2 von X.25 auf einem Mikroprozessorsystem gerecht zu werden, wurde eine zweite Platine im Einfacheuropa-Format entwickelt, die über den Lokal-Bus Anschluß mit der CPU MC6800 zusammenarbeitet. Diese zweite Platine enthält einerseits zwei Bausteine für parallele Schnittstellen MC6820/21 (PIA) zur Kommunikation mit Schicht 3 über deren PIA-Port, vgl. Bild 4.2 und 4.3, und andererseits den hochintegrierten, bidirektional betreibbaren Kommunikationssteuerbaustein MC6854 (ADLC, Advanced Data Link Control). Die Hauptaufgaben dieses ADLC-Bausteines bestehen darin, Flags zu erzeugen und zu erkennen, die Rahmen der Schicht 2 abzuschließen und zu entnehmen, die FCS-Bitmuster zu erzeugen und zu prüfen, um somit Übertragungsfehler zu erkennen und an die CPU MC6800 zu melden.

8-Bit CPU MC6800
- 2 16-Bit-Register (Index- und Stackregister)
- 2 8-Bit-Register (Akkumulator A und B)
- 6(7) Adressierungsarten
- 64 KByte Adreßbereich
- Befehlssatz: 72 Instruktionen
- Interrupts: NMI (Non Maskable Interrupt)
IRQ (Interrupt Request)
SWI (Software Interrupt)
- 1 MHz CPU-Takt
2 Einfacheuropa-Format Platinen
Bis zu 8 KByte ROM On-Board
Bis zu 8 KByte RAM On-Board
2 parallele Schnittstellen
- je 16 Ein-/Ausgabeleitungen
- je 4 Kontroll-Leitungen
1 Kommunikationssteuerbaustein ADLC (parallel/seriell)
1 Timerbaustein (enthält 3 unabhängige Timer)
Adreß-, Daten-, Steuer-Bus (PEP-Karten kompatibel)
Von außen bedienbare Steuertaste
- Reset

Tabelle 4.3: Leistungsmerkmale der Schicht-2-Hardware

Die gesamte, zur Implementierung der Schicht 2 realisierte Hardwarestruktur ist aus Bild 4.3 zu erkennen; die Tabelle 4.3 gibt einen Einblick in die Leistungsfähigkeit dieser Schicht-2-Hardware.

Nach der Systeminitialisierung der beiden Schichten und einem erfolgten Schicht-3-Verbindungsaufbau über die zeitlich zuvor aufgebaute Schicht 2 gelangt die Benutzerinformation über denjenigen ACIA-Port, der der betreffenden Schicht-3-Verbindung und damit dem entsprechenden logischen Kanal zugeordnet ist, interruptgesteuert in den RAM-Baustein von Schicht 3.

Dabei werden die drei ACIA-Ports, die drei logisch getrennten und unabhängigen Verbindungen zugeordnet sind, durch ein spezielles Polling-Programm zyklisch abgefragt, ob Nachrichten zum Senden anstehen.

Es erfolgt nun die Schicht-3-spezifische Behandlung dieser Nachricht durch die CPU MC68000. Daran anschließend wird das komplette Paket, inklusive einem vorangestellten Längenindikator, über die PIA bytewise und interruptgesteuert an Schicht 2 weitergereicht.

Die CPU MC6800 der Schicht 2 nimmt ihrerseits das Paket mittels ihrer PIA bytewise und interruptgesteuert entgegen und speichert es in ihrem RAM-Bereich ab. Die Größe des Paketes ist der CPU durch die Auswertung des Längenindikators bekannt.

Nach erfolgter Schicht-2-spezifischer Behandlung wird der Rahmen vom Kommunikationssteuerbaustein ADLC interruptgesteuert, bytewise, an diesen ADLC gegeben, welcher u.a. den Rahmen durch Flags ergänzt und abgrenzt, das zur Fehlererkennung notwendige FCS-Bitmuster erzeugt und die bitserielle Übertragung auf dem Übertragungsmedium vornimmt.

Stehen keine Daten zum Übertragen an, sendet der ADLC-Baustein eine kontinuierliche Folge von Flags, so daß der Kanal zu jeder Zeit aktiv ist.

In entgegengesetzter Richtung empfängt der ADLC-Baustein den bitseriellen Datenstrom und berechnet das FCS-Kontrollfeld für empfangene Rahmen neu. Stimmt das empfangene FCS-Bitmuster mit dem neu berechneten überein, gibt der ADLC-Baustein ein Positiv-Signal an die CPU MC6800, die dann ihrerseits Schicht-2-spezifische Programme anstößt. Im Falle eines Negativ-Signals wird der zuvor empfangene Rahmen entsprechend der Norm verworfen.

Bei fehlerfreier Übertragung wertet die CPU die Rahmenfolgennummern des empfangenen Rahmens aus und nimmt eine entsprechende systeminterne Parameteraktualisierung vor. Eine sofortige Quittierung dieses korrekt empfangenen Rahmens erfolgt entweder durch Informationsrahmen, sofern Pakete aus Schicht 3 zum Senden anstehen, oder durch spezielle Stellerrahmen, vgl. hierzu Kap.3.2.3.

Bei korrektem Empfang eines Informationsrahmens und nach erfolgreicher Auswertung der Rahmenfolgennummern wird das im Informationsteil des Rahmens enthaltene Paket byteweise und interruptgesteuert mit vorangestelltem Längenindikator über die PIA an Schicht 3 weitergeleitet.

Die CPU MC68000 der Schicht 3 nimmt ihrerseits das Paket mittels ihrer PIA byteweise und interruptgesteuert entgegen und wertet die Paketfolgennummern für die systeminterne Parameteraktualisierung aus. Wie in Schicht 2 erfolgt auch in Schicht 3 eine sofortige Quittierung entweder in Form eines Datenpaketes, sofern weitere Benutzerinformationen über den ACIA-Port angeboten werden, oder durch spezielle Steuerpakete, vgl. Kap.3.2.4.3.

Nach der paketprotokollspezifischen Behandlung des Paketes wird - im Falle eines Datenpaketes - die im Datenfeld enthaltene Benutzerinformation interruptgesteuert über denjenigen ACIA-Port ausgegeben, der der betreffenden Verbindung zugeordnet ist.

#### 4.3.2 Software

In Kap.4.1 wurde bereits angedeutet, daß jedem in den Schichten 2 und 3 existierenden Prozeß, vgl. Bild 4.1, in der softwaremäßigen Realisierung genau ein separater Programmteil zugeordnet worden ist, soweit Prozesse nicht auf speziellen Bausteinen, wie z.B. dem ADLC, ablaufen. Dadurch ergibt sich eine realisierte Softwarestruktur, die ausschnittsweise in den Bildern 4.4 und 4.5 wiedergegeben ist.

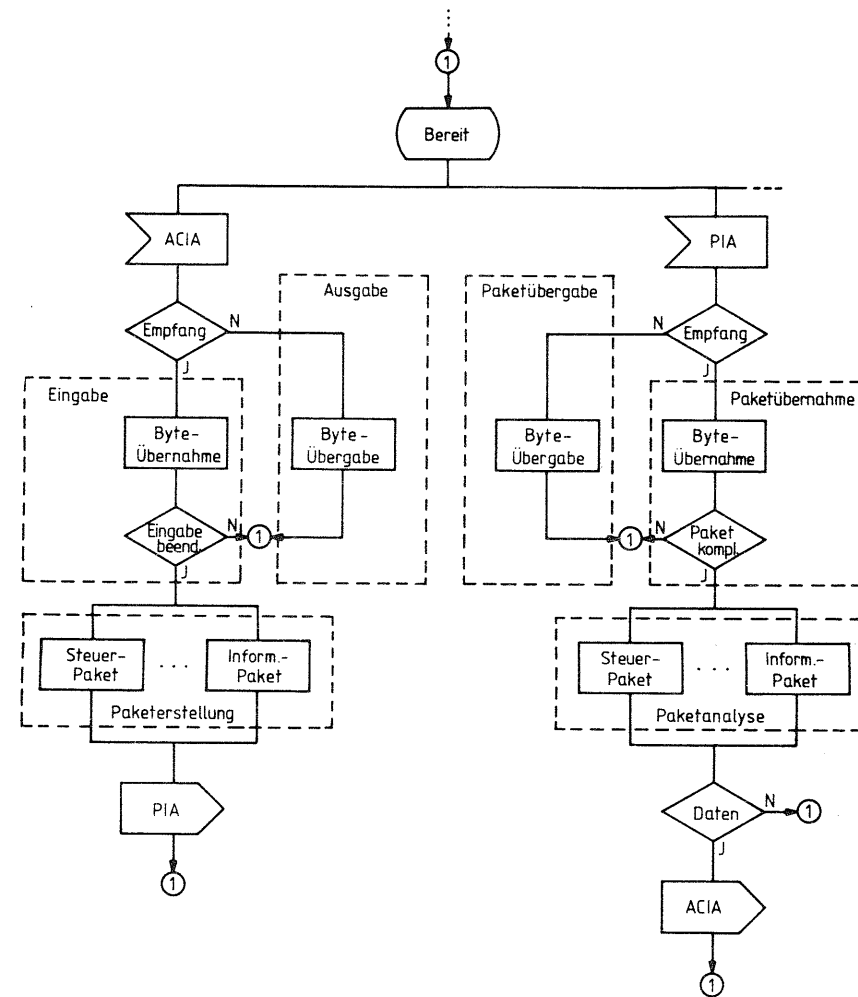


Bild 4.4: Softwarestruktur (Ausschnitt) in Schicht 3 einer DEE bzw. DOE

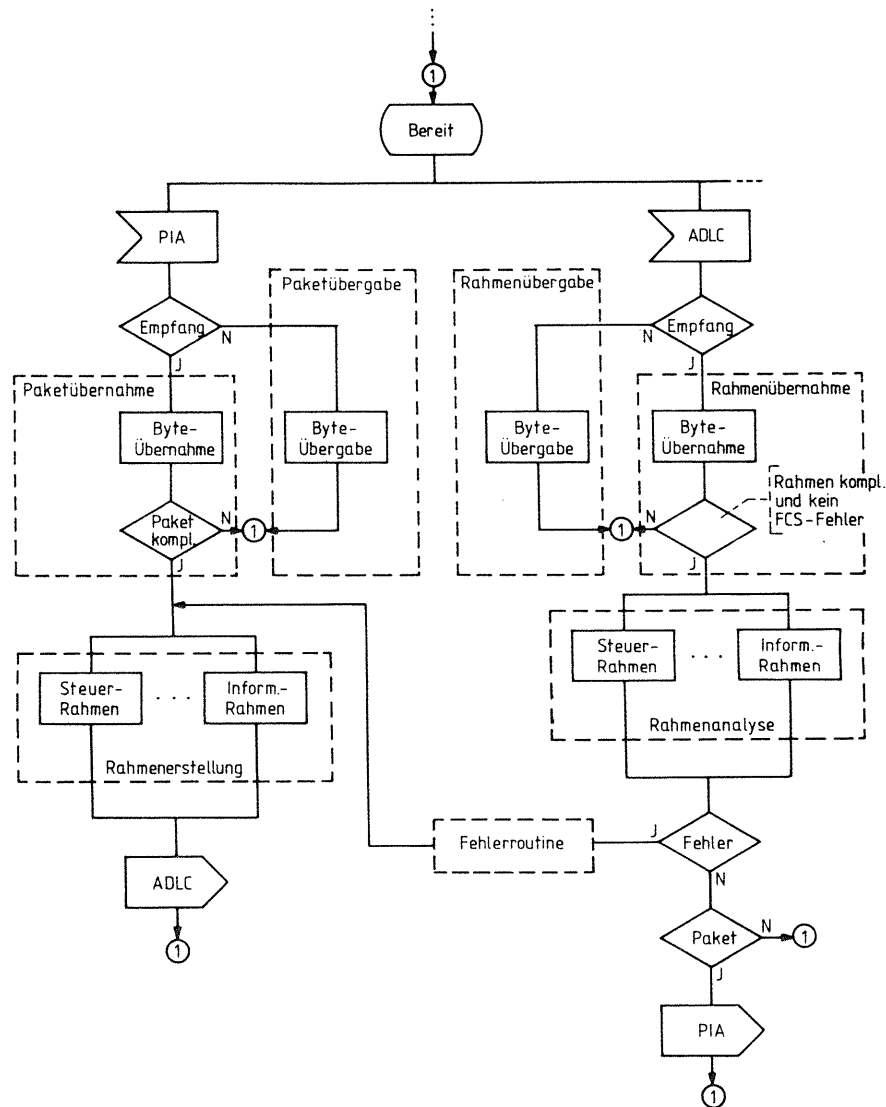


Bild 4.5: Softwarestruktur (Ausschnitt) in Schicht 2 einer DEE bzw. DOE

Hauptbestandteile der Software sind dabei die beiden Monitorprogramme, die neben wichtigen Überwachungsfunktionen auch Koordinationsfunktionen wahrnehmen und die somit für die Prozesssynchronisation der einzelnen, in den Schichten vorkommenden Prozesse verantwortlich zeichnen. Die Monitore wachen im einzelnen über die an die CPU gerichteten Interrupts und stoßen dann die entsprechenden, d.h. die Interrupts bearbeitenden Programme an.

Die Programme für den Eingabeprozess bzw. für die Übernahmeprozesse leiten das byteweise Weiterreichen der jeweiligen Daten in den zuständigen RAM-Bereich ein. Die Länge der Daten ist dabei bekannt, entweder durch vorheriges Auswerten eines anfänglich empfangenen Längenindicators oder durch separate, spezielle Meldungen, wie im Falle des ADLC.

Die Analyseprogramme werden aktiviert, wenn die Übernahmeprogramme die kompletten, aus tiefergelegenen Schichten angebotenen Daten im Speicher abgelegt haben. Die Analyse bezieht sich speziell auf das Auswerten der schichtspezifischen Folgennummern der empfangenen Daten und auf das Aktualisieren derjenigen Systemgrößen, die von den Erstellungsprogrammen benutzt werden. Treten bei der Analyse der Daten keine Fehler auf, wird das Demultiplexer/Ausgabeprogramm in Schicht 3 bzw. das Paketübergabeprogramm in Schicht 2 gestartet.

Werden bei der Analyse der empfangenen Daten Fehler entdeckt, wird als nächstes in der Programmfolge eine Fehlerroutine angestoßen, um schichtspezifische, fehlerbehebende Maßnahmen zu ergreifen. Die Fehlerroutinen nehmen ihrerseits ebenfalls Parameteraktualisierungen vor, um den Erstellungsprogrammen somit anzuzeigen, welche Daten mit welchen Größen sie als nächstes zu senden haben.

Die Paket- bzw. Rahmenerstellungsprogramme bereiten die zu sendende Information entsprechend den Schichtenprotokollen auf, soweit Daten zeitlich zuvor durch das Schicht-3-Multiplexer/Eingabeprogramm bzw. durch das Schicht-2-Paketübernahmeprogramm im jeweiligen Speicherbereich zur Verfügung gestellt wurden. Hierbei greifen die Erstellungsprogramme auf die internen Systemgrößen zurück, die durch die Analyse- bzw. Fehleroutineprogramme stets aktualisiert sind. Stehen keine Daten zum Senden bereit, wird in Schicht 3 ein Steuerpaket bzw. in Schicht 2 ein Steuerrahmen mit den jeweiligen aktualisierten Parametern fertiggestellt. Im Anschluß daran werden die Pakete bzw. Rahmen durch ein Übergabeprogramm an die darunterliegende Schicht, interruptgesteuert und byteweise, weitergereicht.

Die Übergabe- und Übernahmeprogramme an der internen Schnittstelle Schicht 2/Schicht 3 reichen die Daten ebenfalls interruptgesteuert und byteweise weiter. Sie arbeiten hierzu im Handshaking-Verfahren und bedienen sich der in Kap.4.2 vorgestellten Primitive.

Für die Eingabe- bzw. Ausgabeprogramme an der Schnittstelle zwischen Schicht 3 und Schicht 4 ist jeweils ein Polling-Mechanismus realisiert. Im Falle der Eingabe überprüft dieser Mechanismus die drei ACIA-Ports regelmäßig auf eventuelle Dateneingaben und übernimmt bei erfolgreicher Abfrage diese Daten in den dem ACIA-Port, und damit dem logischen Kanal zugeordneten RAM-Bereich. Für die Ausgabe werden die den ACIA-Ports zugeordneten Ausgabepuffer durch den zweiten Polling-Mechanismus abgeprüft und eventuell vorhandene Daten an Schicht 4 weitergereicht.

## 5. LEISTUNGSANALYSE DER PAKETSCHICHT VON X.25

Ziel der in dieser Arbeit vorgestellten Untersuchungen ist die Bestimmung der Leistungsfähigkeit virtueller Verbindungen auf einem X.25-Netzzugangsabschnitt.

Aus Gründen der Komplexität werden dabei nur Prozeduren und Protokollabläufe innerhalb der Paketschicht von X.25 betrachtet.

Die durch die Protokolle der Sicherungsschicht bedingte Leistungsbeeinflussung auf Ebene 3 bleibt bei der hier vorgestellten Leistungsanalyse unberücksichtigt. In anderen Arbeiten gewonnene Erkenntnisse aus Schicht 2 können jedoch global in den Ergebnissen der Leistungsbewertung von Schicht 3 berücksichtigt werden.

Da für eine realitätsgerechte Analyse eine saubere Modellbildung unumgänglich ist, wird im ersten Teil dieses Kapitels, unter Einbeziehung der in Kap.2.5.1.1 vorgestellten Modellstrukturen, ein Modell der X.25-Schnittstelle aufgezeigt und beschrieben.

Der zweite Teil befaßt sich dann mit Methoden der Leistungsuntersuchung, sowohl simulativer als auch mathematischer Art. Dabei werden bei der Simulationsbeschreibung Erkenntnisse aus Kap.5.2.1 aufgegriffen und vereinfachende Maßnahmen aufgezeigt.

Im mathematischen Teil wird eine auf das Modell bezogene hierarchische Dekomposition gezeigt sowie einerseits die Analyse der einzelnen Komponenten und die Aggregation gemäß Kap.2.5.3 vorgestellt als auch andererseits getroffene Approximationsannahmen erklärt.

### 5.1 Modellbildung der Schnittstelle

Unter Einbeziehung der in Bild 3.3 gezeigten logischen Struktur von X.25 läßt sich in einem ersten Schritt der Modellbildung die im folgenden Bild 5.1 aufgeführte Grobstruktur angeben, die sowohl die Protokolle und deren Schichtung als auch Implementierungsge-sichtspunkte, wie Anzahl von Prozessoren und Zwischenspeichern be-rücksichtigt.

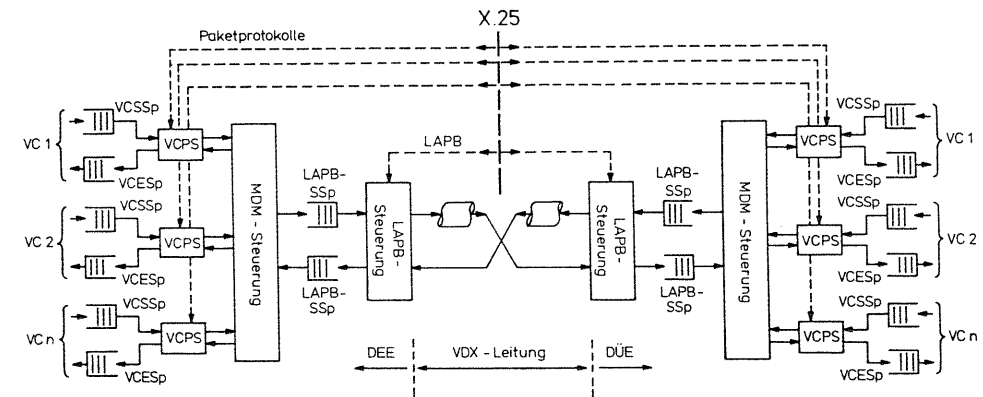
Für die folgende Modellbeschreibung werden die in Bild 5.1 und im "Abkürzungsverzeichnis" erläuterten Kürzel verwendet.

Bild 5.1 zeigt das Modell der DEE und der DÜE, die mittels einer Vollduplexleitung miteinander verbunden sind.

Die Schicht 1 wird hierbei durch die VDX-Leitung, dargestellt durch die Kanalsymbole, wiedergegeben.

Schicht 2 wird durch die LAPB-Steuerung repräsentiert, die mit Hilfe des LAPB-Protokolls mit ihrer zugeordneten Gegenseite kom-muniziert. Die Übergabestelle zwischen Schicht 2 und 3 wird durch die Zwischenspeicher LAPB-SSp und LAPB-ESp realisiert. LAPB-SSp und LAPB-ESp werden hierbei zur Schicht 2 zugehörig angesehen.

Die Schicht 3 besteht aus einer übergeordneten MDM-Steuerung sowie aus n verbindungsindividuellen VC-Protokollsteuerungen, wobei n die Anzahl der aktiven, d.h. Daten produzierenden, Verbindungen angibt. Jede VCPS kommuniziert mit ihrer zugeordneten Gegenseite mittels eines Paketprotokolls. Die Schnittstelle zwischen Schicht 3 und höheren Schichten wird durch die verbindungsindividuellen Zwischenspeicher VCSSp und VCESp realisiert, welche Bestandteil der Schicht 3 sind.



DEE	Datenendeinrichtung	SSp	Sendespeicher
DÜE	Datenübertragungseinrichtung	ESp	Empfangsspeicher
LAPB	Link Acc. Proc., Bal. Mode	VC	Virtuelle Verbindung
MDM	Multiplexer/Demultiplexer	VDX	Vollduplex
VCPS	VC-Protokollsteuerung		

Bild 5.1: Allgemeines Modell der Schnittstelle X.25

Hauptfunktionen der einzelnen Bestandteile des Modells sind:

- VCSSp:
  - Zwischenspeicherung ankommender Nachrichtenblöcke aus Schicht 4 zur Weiterbehandlung durch Schicht 3;
- VCPS:
  - Paketierung in Senderichtung, d.h. Erstellung kompletter Pakete aus den im VCSSp wartenden Blöcken durch Hinzufügen des Paketkopfes,
  - Depaketierung in Empfangsrichtung,
  - Datenflußsteuerung mittels Folgenummern und Fenstermechanismus,
  - Kontrolle der Paketreihenfolge,
  - Fehlererkennung und -behebung;

- MDM-Steuerung:
  - Multiplexen der Pakete in Senderichtung,
  - Einspeicherung in LAPB-SSp,
  - Demultiplexen der Pakete in Empfangsrichtung,
  - Ausspeicherung aus LAPB-ESp,
  - Bewertung der logischen Kanalnummern,
  - Weiterleitung der Pakete an entsprechende VCPS;
- LAPB-SSp:
  - Zwischenspeicherung von Paketen zur Weiterbehandlung durch Schicht 2;
- LAPB-Steuerung:
  - Bildung von Rahmen aus den im LAPB-SSp wartenden Paketen in Senderichtung,
  - Entnahme der Pakete aus fehlerfrei übertragenen Rahmen zum Einspeichern in LAPB-ESp in Empfangsrichtung,
  - Fehlererkennung und -korrektur,
  - Datenflußsteuerung mittels Folgenummern und Fenstermechanismus;
- VDX-Leitung:
  - Bitweise Übertragung,
  - Verzögerung durch physikalische Laufzeit;
- LAPB-ESp:
  - Zwischenspeicherung fehlerfrei übertragener Pakete zur Weiterbearbeitung durch Schicht 3;
- VCESp:
  - Zwischenspeicherung von Nachrichten für höhere Schichten.

Der Aspekt, die  $n$  VC-Protokollsteuerungen und die MDM-Steuerung in einem Prozessor zu implementieren sowie die LAPB-Steuerung einem weiteren Prozessor zuzuordnen, führt in einem zweiten Schritt der Modellbildung zum nachfolgend beschriebenen detaillierten Netzmodell, (vgl. hierzu auch Kap.4).

Die Realisierung der beiden Schichten auf getrennten Prozessoren kommt dabei der Erfüllung der Forderung nach Unabhängigkeit und Austauschbarkeit (siehe Kap.3.2.1 und Kap.4.1) der einzelnen Schichten zugute.

Diese Aufteilung entspricht auch den inzwischen kommerziell angebotenen Mikrorechnerkomponenten für die Implementierung der Schnittstelle X.25. Hierbei wird für die schnelle serielle Bearbeitung in Schicht 2 hauptsächlich auf Hardwarekomponenten zurückgegriffen, während für die blockweise Bearbeitung in Schicht 3 vielfach zusätzliche programmtechnische Realisierungen zur Anwendung kommen, (vgl. Kap.4).

Unter diesem Gesichtspunkt läßt sich unter Zuhilfenahme der Strukturen aus Kap.2.5.1.1 das in Bild 5.2 gezeigte detaillierte Warteschlangenmodell der Schnittstelle X.25 angeben.

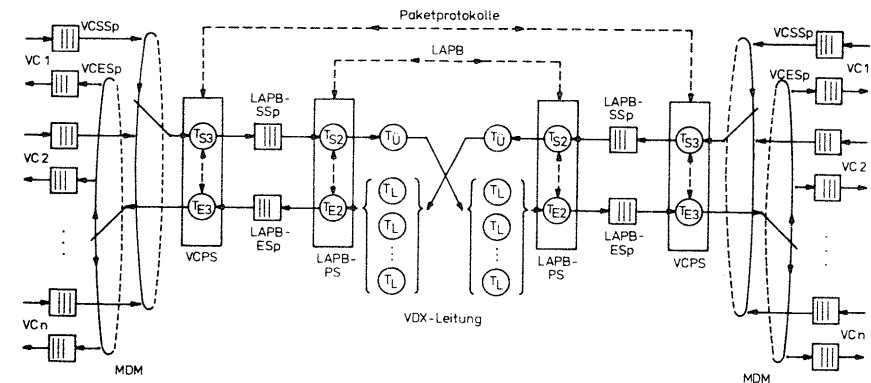


Bild 5.2: Detailliertes Warteschlangenmodell der Schnittstelle X.25

Die VC-Protokollsteuerungen und die MDM-Steuerung werden durch die Kombination aus Prozessormodell für programmgesteuerte (vgl. Bild 2.10) und Prozessormodell für abfragegesteuerte Verarbeitung (vgl. Bild 2.11) nachgebildet. Zwei Bearbeitungsphasen repräsentieren die für die Protokollabwicklung und für die MDM-Funktionen erforderlichen Bearbeitungszeiten, wobei die Empfangsphase der Dauer  $T_{E3}$  höhere Priorität gegenüber der Sendephase mit der Dauer  $T_{S3}$  besitzt.



Der Multiplexvorgang erfolgt in zyklischer Reihenfolge, während das Demultiplexen wahlfrei in Abhängigkeit des logischen Kanals, zu welchem das Paket gehört, erfolgt. Die LAPB-Steuerung wird entsprechend Kap.2.5.1.1 durch ein Prozessormodell für programmgesteuerte Verarbeitung mit einer Sendephase der Dauer  $T_{S2}$  und einer Empfangsphase mit der Dauer  $T_{E2}$  repräsentiert. Die Modellierung der VDX-Leitung erfolgt vollkommen analog zu dem in Bild 2.12 gezeigten Kanalmodell.

In Bezug auf wichtige Verkehrsabläufe sei in diesem Zusammenhang auf Kap.5.2.1 verwiesen.

## 5.2 Methoden zur Leistungsuntersuchung

In diesem Abschnitt kommen die in Kap.2.5 vorgestellten allgemeinen Untersuchungsmethoden zur spezifischen Leistungsanalyse der CCITT-Schnittstelle X.25 zur Anwendung.

Basis für diese Untersuchungen ist das im vorigen Abschnitt in Bild 5.2 vorgestellte detaillierte Netzmodell der Schnittstelle.

### 5.2.1 Simulationemethode

Für die simulative Analyse von X.25 wurde das exakt nachbildende Verfahren der belegungstremen Simulation aus der Gruppe der zeit-tremen Verfahren (vgl. Kap.2.5.2) gewählt.

In dem hierzu entwickelten Simulationsprogramm wurden sowohl die Struktur, entsprechend dem vereinfachten Netzmodell aus Bild 5.3, die Prozeduren und Protokollabläufe in Schicht 3 sowie die Verkehrsabläufe innerhalb der Schichten, zwischen den Schichten, zwischen den beiden Endeinrichtungen als auch das Verhalten zur Außenwelt bis ins Detail nachgebildet.

In diesem Zusammenhang sei auch auf die Veröffentlichungen /80/ und /81/ verwiesen, die sich ebenfalls mit der simulativen Leistungsbewertung von X.25 befassen.

Nicht unerwähnt sollte bleiben, daß die Auf- und Abbauphasen virtueller Verbindungen in dem Simulationsprogramm nicht nachgebildet wurden, da die Zeiten für den Auf- und auch den Abbau nur einen Bruchteil der gesamten Verbindungsdauer ausmachen und somit für die Leistungsbewertung nicht ins Gewicht fallen.

#### 5.2.1.1 Vereinfachende Annahmen

Aus Komplexitätsgründen als auch aus Gründen des begrenzten Speicherplatzes der Datenverarbeitungsanlage, die für die intensiven Simulationsstudien zur Verfügung stand, wurde die folgende Annahme getroffen:

die VDX-Leitung, die die DEE und DOE physikalisch miteinander verbindet, ist als ideale Leitung anzusehen.

Auf der Leitung treten somit keine Übertragungsfehler und keine durch äußere Einflüsse bedingten Fehler auf; der Kanal arbeitet somit fehlerfrei.

Diese Annahme ist insofern realistisch, als es sich bei X.25 um eine lokale Netzzugangsschnittstelle handelt, welche zukünftig mit relativ kurzen Anschlußleitungen auskommt und die somit keinen großen Umwelteinflüssen ausgesetzt sind.

Aus der getroffenen Annahme kann jedoch die vereinfachende Maßnahme gefolgert werden, die komplette Sicherungsschicht entfallen zu lassen.

Bild 5.3 zeigt das hieraus resultierende vereinfachte Netzmodell von X.25, das im folgenden allen Untersuchungen als Basis dient. Durch den Wegfall der Prozeduren, Funktionen und Verkehrsabläufe von Schicht 2 konnte der Komplexitätsgrad der Simulation verringert und damit überschaubar gehalten werden.

Trotzdem können Einflüsse aus Schicht 2 auf Schicht 3 global insofern berücksichtigt werden, daß (frei wählbare) Parameter, wie z.B. Übertragungsgeschwindigkeit oder Laufzeit, entsprechend Analyseergebnissen der Schicht 2 (z.B. aus /66/ bis /73/) angepaßt, d.h. modifiziert werden können.

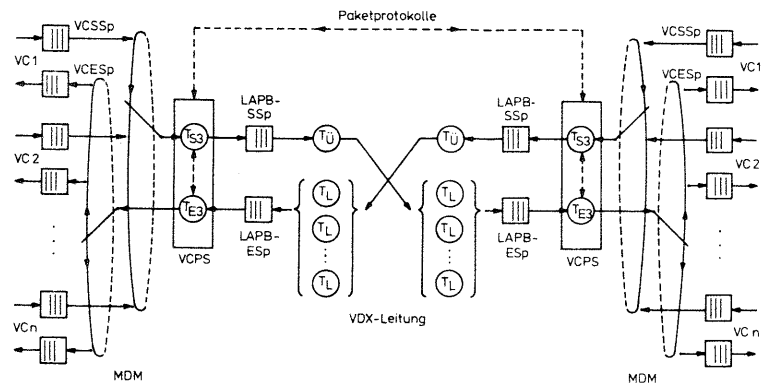


Bild 5.3: Vereinfachtes Warteschlangenmodell der Schnittstelle X.25

### 5.2.1.2 Hauptmechanismen an der Schnittstelle

Die durch das Netzmodell (vgl. Bild 5.3) vorgeschriebenen und in der Simulation berücksichtigten Mechanismen lassen sich prinzipiell wie folgt beschreiben:

#### Einspeichern:

- VCSSp: ankommende Nachrichtenblöcke, welche freie Plätze im VCSSp vorfinden, belegen einen Speicherplatz. Trifft ein Block auf einen vollen VCSSp, wird er abgewiesen und geht dem System verloren;
- VCESp: nur im Falle freien Speicherplatzes werden Blöcke aus dem LAPB-ESp in den VCESp umgespeichert, ansonsten bleiben die Pakete im LAPB-ESp gespeichert;
- LAPB-SSp: hat der Speicher freie Plätze zur Verfügung, können von der MDM-Steuerung Pakete eingeschrieben werden. Im Falle eines vollbelegten LAPB-SSp stoppt der Multiplexmechanismus, bis wieder freie Plätze vorhanden sind;
- LAPB-ESp: da der Steuermechanismus von Schicht 2 in der Simulation unberücksichtigt bleibt, muß die Größe des LAPB-ESp so groß gewählt werden, daß immer genug freie Plätze zur Verfügung stehen.

#### Ausspeichern:

Bei allen Warteschlangen kommt die Abfertigungsdisziplin FIFO (first-in, first-out) zur Anwendung.

#### Multiplexen:

Es können zwei Verfahren angewendet werden:

- zyklisches Multiplexen;
- prioritätengesteuertes Multiplexen, mit der Annahme, daß VC Nr.1 die höchste und VC Nr.n die niedrigste Priorität hat.

#### Demultiplexen:

Wahlfreies Demultiplexen, entsprechend der im Paketkopf enthaltenen logischen Kanalnummer.

#### Bearbeiten:

VCPS+MDM: Prozessormodell mit 2 Bearbeitungsphasen:

- Sendephase: mit konstanter Dauer  $T_S$ ,
- Empfangsphase: mit konstanter Dauer  $T_E$  sowie höherer, nicht-unterbrechender Priorität gegenüber der Sendephase.

Übertragen:

Wird der Übertragungskanal frei, d.h. ist das letzte Bit des vorangegangenen Rahmens übertragen (Phase mit  $T_0$  wird beendet), kann mit der Übertragung des nächsten Rahmens, der im LAPB-SSp wartet, begonnen werden. Die Dauer  $T_0$  ist dabei von der Paketlänge NBL des zu übertragenden Paketes sowie von der Übertragungsgeschwindigkeit  $v$  der Leitung abhängig:

- Datenpaket:

$$T_{0DP} = ( NBL + OVHD ) / v \quad ( 5.1 )$$

- Steuerpaket:

$$T_{0SP} = OVHD / v \quad ( 5.2 )$$

mit Steuerinformation (OVHD):

$$OVHD = SI(2)+SI(3) = 72 \text{ bit} \quad ( 5.3 )$$

Nach Ablauf der Phase  $T_0$  kann eine der nachfolgenden noch freien Phasen der konstanten Dauer  $T_L$  gestartet werden (vgl. Bild 5.3).

Hinweis: das vereinfachte Netzmodell von X.25 (nach Bild 5.3) berücksichtigt zwar nicht die speziellen Prozedursteuerungen von Schicht 2; der durch diese Schicht bedingte "Overhead" von 48 bit, d.i. der Steuerteil SI(2) (vgl. Bild 3.5), wird bei der Berechnung der paketindividuellen Übertragungszeit jedoch berücksichtigt, (siehe Gleichung (5.3)).

Protokollbedingte Mechanismen:

Die in X.25 definierte Folgenummernsteuerung, die für jede aufgebaute virtuelle Verbindung unabhängig und individuell erfolgt, kommt im Netzstrukturmodell nicht explizit zum Ausdruck, wird in der realisierten Form der Simulation jedoch bis ins Detail (entsprechend Kap.2.3.2 und 3.2.4.4) berücksichtigt.

5.2.1.3 Schnittstelle zur Umwelt

Die implementierungsspezifische Schnittstelle zur Umwelt wird bei der hier betrachteten Modellbildung durch die verbindungsindividuellen Zwischenspeicher VCSSp und VCESp realisiert (vgl. Bild 5.2 und 5.3).

Das eigentliche Verkehrsverhalten der Umwelt wird jedoch dadurch geprägt, wie stark sie das System mit neuen Daten belastet, d.h. wie groß sowohl der Nachrichtenblockankunftsabstand als auch der Umfang der Datenblöcke ist, sowie dadurch, wie schnell die Umwelt das System durch Entnahme erfolgreich übertragener Daten entlastet. Diesen unterschiedlichen Verhaltensweisen wird das Simulationsprogramm insofern gerecht, als einerseits sowohl die Verteilung des Ankunftsabstandes als auch die Verteilung der Nachrichtenblocklänge aus den in Tabelle 2.1 (Kap.2.5.1.2) aufgeführten Verteilungsfunktionen für jede aufgebaute, d.h. Daten produzierende, Verbindung unabhängig voneinander gewählt werden kann. Andererseits kann auch die Entnahme übertragener Blöcke dadurch realistisch nachgebildet werden, daß die Dauer zwischen zwei Entnahmepunkten von Blöcken aus dem VCESp für jede virtuelle Verbindung frei wählbar ist. Durch geeignete Wahl dieser speziellen Parameter ist somit eine detaillierte Nachbildung der unterschiedlichsten Verkehrsverhalten der Umwelt im weitesten Sinne möglich.

5.2.2 Analytische Methode

Die analytischen Lösungsverfahren zur Berechnung von Warteschlangennetzen gliedern sich prinzipiell in zwei Gruppen:

- exakte analytische Verfahren
- approximative Verfahren.

Exakte Lösungsverfahren sind auf alle Produktform-Warteschlangennetze, vgl. Kap.2.5.3.1, anwendbar. Die Berechnung der charakteristischen Größen kann dabei auf mehrere Arten erfolgen:

- 1) Durch direktes Lösen der Gleichungen. Diese Methode ist jedoch nur auf triviale Netzwerke mit wenigen Knoten und nur einer Kette anwendbar. Für mehrere Ketten wird der Zustandsraum mehrdimensional und damit numerisch sehr schwer lösbar.
- 2) Mittels verschiedener Faltungsalgorithmen. Die Lösung des Produktformansatzes kann dabei durch sequentielle Faltungsalgorithmen (sequential convolution) oder durch baumartige Faltungsalgorithmen (tree convolution) erfolgen.
- 3) Mit Hilfe der Mittelwertanalyse (vgl. Kap.2.5.3.2).

Allen exakten Verfahren gemeinsam ist jedoch, daß sie sehr umfangreich und sehr aufwendig in Bezug auf Speicherplatz und Rechenzeit sind.

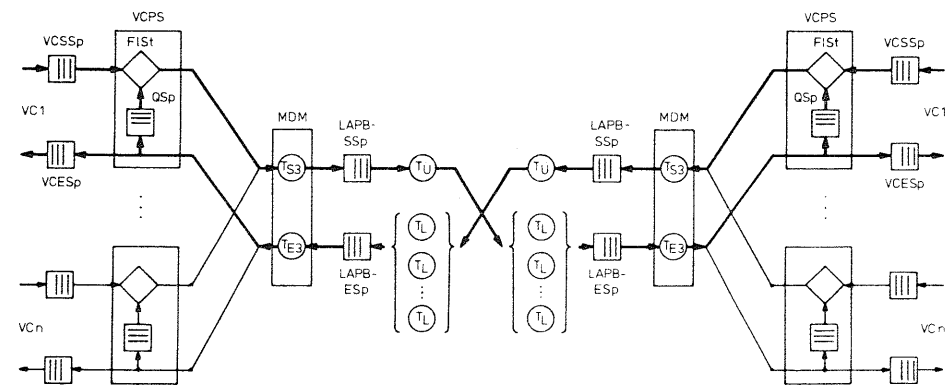
Die approximativen Verfahren gehen meist von exakten analytischen Verfahren, in der Regel von der in Kap.2.5.3.2 beschriebenen Mittelwertanalyse, aus. Durch vereinfachende Annahmen und durch geeignete Abschätzungen bestimmter in Wirklichkeit vorkommender Ereignisse, gelingt es, sowohl den Speicherplatzbedarf als auch die benötigte Rechenzeit um Größenordnungen zu reduzieren. Ein bekanntes approximatives Verfahren zur Lösung komplexer Warteschlangennetze ist die Dekompositionsmethode (vgl. Kap.2.5.3.3). Bei diesem Verfahren wird das gesamte Netzwerk derart in Teilsysteme zerlegt, daß die Interaktionen zwischen den Teilsystemen möglichst gering im Vergleich zu den Aktivitäten innerhalb der Teilsysteme sind. Die einzelnen Subsysteme können dann getrennt auf einfachere Art und Weise analysiert werden. Die wiederholte Anwendung der Dekompositionsmethode bei großen Netzwerken führt dann zur sogenannten hierarchischen Dekomposition und somit zur schrittweisen Verringerung der Komplexität des Gesamtsystems.

Für die mathematische Leistungsuntersuchung der CCITT-Schnittstelle X.25 wurde das Verfahren der hierarchischen Dekomposition, basierend auf der Idee der Systemaggregation (vgl. Kap.2.5.3.3), gewählt.

Dieses Kapitel soll einerseits den Weg der Dekomposition als auch andererseits die Analyse der einzelnen Komponenten aufzeigen.

### 5.2.2.1 Dekomposition und Aggregation

Aufbauend auf der allgemeinen Grobstruktur des Modells von X.25 (vgl. Bild 5.1) sowie des vereinfachten Warteschlangenmodells (entsprechend Bild 5.3) läßt sich ein vereinfachtes Strukturbild angeben, aus welchem sowohl der verbindungsindividuelle Flußsteuermechanismus als auch der allgemeine Nutz- und Steuerinformationsaustausch ersichtlich ist (siehe hierzu Bild 5.4).



FISst    Flußsteuerung  
 QSp    Quittungsspeicher

Bild 5.4: Nutz- und Steuerinformationsfluß

Die Modellierung des Fenstermechanismus innerhalb der VC-Protokollsteuerung VCPS besteht dabei im wesentlichen aus zwei Komponenten: der Flußsteuerung und dem Quittungsspeicher.

Der Quittungsspeicher wird bei dieser Modellvorstellung von der Flußsteuerung wie folgt verwaltet:

Wenn ein Nachrichtenblock im VCSSp wartet und mindestens eine Quittung im QSp vorhanden ist, wird der Nachrichtenblock zur weiteren Bearbeitung zum MDM weitergeleitet und gleichzeitig eine Quittung im QSp gestrichen. Mit anderen Worten, solange noch Quittungen im QSp vorhanden sind, können Pakete für diesen VC weitergesendet werden. Ist der QSp jedoch leer, dann stoppt die Flußsteuerung den Sendevorgang. Werden dagegen erfolgreich übertragene Pakete vom MDM der VCPS angeboten und quittieren diese Pakete den erfolgreichen Empfang auf der Gegenseite, so werden ebenso viele Quittungen, wie Pakete als korrekt empfangen bestätigt wurden, in den QSp eingetragen, und die Flußsteuerung kann wieder mit dem Senden beginnen. Auf diese Art und Weise läßt sich der in Kap.2.3.2 beschriebene und in X.25 für jeden logischen Kanal zur Anwendung kommende Fenstermechanismus sehr gut nachbilden.

Wird die maximale Anzahl möglicher Quittungen im QSp auf die Fenstergröße  $w_S$  beschränkt, so sind folgende Extrema möglich (vgl. hierzu auch Kap.2.3.2):

- $w_S$  Quittungen im QSp: momentanes Sendefenster ganz geöffnet,
- 0 Quittungen im QSp: momentanes Sendefenster geschlossen.

Bild 5.4 zeigt, durch einen kräftigen Linienzug stark hervorgehoben, stellvertretend für alle VC, den Nutzinformationsfluß in beide Richtungen, d.h. DEE-DOE und DOE-DEE, sowie den Steuerinformationsfluß innerhalb der an der Verbindung beteiligten VC-Protokollsteuerungen.

Aufgrund der detaillierten Ergebnisauswertung der intensiven Simulationsstudien konnte die Erkenntnis gewonnen werden, daß jedes übertragene Datenpaket durch Senden eines separaten Steuerpakets in Gegenrichtung als korrekt empfangen bestätigt wird.

Dies läßt den Schluß zu, daß auf allen Verbindungen für jede Nutzinformationsrichtung, d.h. sowohl für die Richtung von der DEE zur DOE als auch für die Richtung von der DOE zur DEE, in der jeweiligen Hinrichtung überwiegend Datenpakete und in der Rückrichtung überwiegend Steuerpakete (Quittierpakete) gesendet werden.

Mit anderen Worten, in der VC-Protokollsteuerung erfolgt eine "virtuelle Pakettypkonvertierung" derart, daß jedes empfangene Datenpaket sofort durch Senden eines Steuerpakets bestätigt wird. Dieser Sachverhalt kommt in Bild 5.5 insofern zum Ausdruck, als für die kräftig eingezeichnete Übertragungsrichtung DEE-DOE (d.h. Nutzinformation von der DEE zur DOE) für VC Nr.1 folgende geschlossene Kette entsteht:

Senden von Nutzdaten von der DEE zur DOE, virtuelle Pakettypkonvertierung in der VCPS der empfangenden DOE, Quittierung durch Senden von Steuerpaketen von der DOE zur DEE.

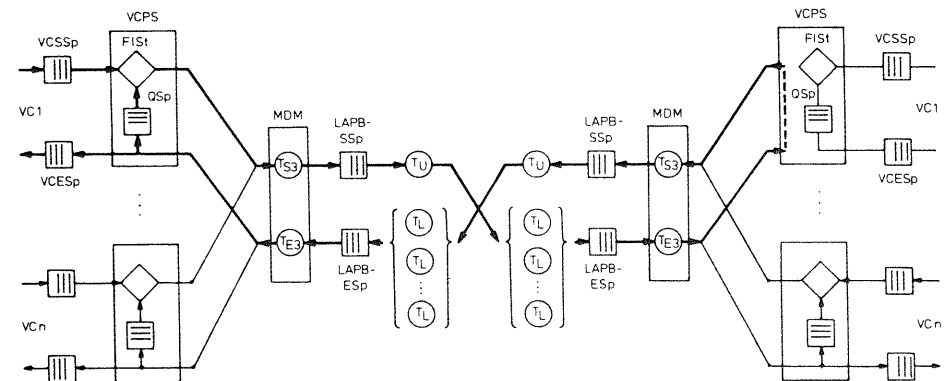


Bild 5.5: Geschlossene Kette durch virtuelle Pakettypkonvertierung

Für die Gegenrichtung entsteht eine äquivalente, spiegelverkehrte Kette:

Nutzinformation von der DOE zur DEE, virtuelle Pakettypkonvertierung in der VCPS der DEE, Quittierpakete von der DEE zur DOE.

Entsprechendes gilt für jede weitere aktive virtuelle Verbindung: je eine geschlossene Kette in Hin- und Rückrichtung.

Analog dem in Kap.2.5.3.3 allgemein beschriebenen Verfahren der hierarchischen Dekomposition wird auch in diesem speziellen Beispiel der Leistungsuntersuchung von X.25 das gesamte Netzmodell nach Bild 5.5 in zwei Subsysteme zerlegt, die getrennt voneinander analysierbar sind.

Für die vorliegende Arbeit wurde die Wahl des einen Subsystems auf die VC-Protokollsteuerung gelegt, um den Einfluß des Fenstermechanismus auf die Leistungsfähigkeit des Paketprotokolls ermitteln zu können.

Entsprechend den Bildern 2.19 und 2.20 aus Kap.2.5.3.3 bildet dann der Rest des Netzmodells das zweite Subsystem, auch als VCPS-Komplement bezeichnet. In den Ergebnissen dieses VCPS-Komplements kommen dann die Einflüsse zum Ausdruck, die dadurch bedingt sind, daß aufgrund der Multiplexfunktion in Schicht 3 Netzressourcen gemeinsam in Anspruch genommen werden. Die Analyse des VCPS-Komplements liefert schließlich ein aggregiertes System, bestehend aus einer Bedienstation mit zugehöriger Warteschlange und flußäquivalentem Verkehr. Mit Hilfe dieses aggregierten Systems und seines Verkehrsverhaltens ist dann das VCPS-Subsystem vollständig analysierbar. Bild 5.6 zeigt, durch starke Umrandung angedeutet, die Dekomposition in die beiden Subsysteme VCPS und VCPS-Komplement.

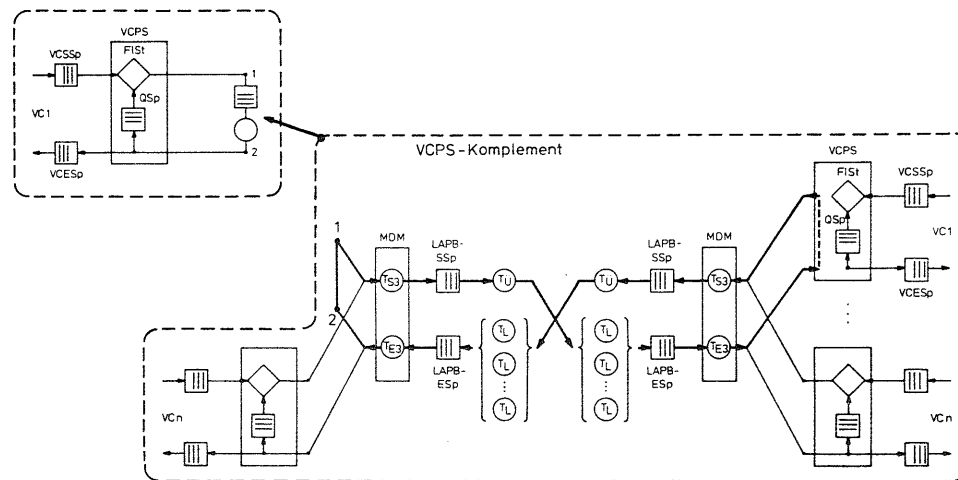


Bild 5.6: Hierarchische Dekomposition in VCPS und VCPS-Komplement

Im Zusammenhang mit der Analyse flußkontrollierter Datenverbindungen sei auch auf die Literaturen /82/ bis /87/ hingewiesen. Im Gegensatz zur vorliegenden Arbeit werden dort jedoch nur einzelne und nicht mehrere multiplexierte Verbindungen betrachtet, die alle individuell und unabhängig voneinander datenflußgesteuert sind.

#### 5.2.2.2 Analyse des Flußkontrollmechanismus

Dieses Kapitel hat die Analyse des in der VC-Protokollsteuerung realisierten Flußkontrollmechanismus zum Gegenstand (vgl. hierzu /82/).

Die hauptsächlich interessierenden Ergebnisgrößen sind hierbei einerseits die mittlere Wartezeit im VCSSp und andererseits der Durchsatz an Nachrichtenblöcken über die Schnittstelle.

Entsprechend der im vorigen Kapitel erläuterten Dekomposition und Aggregation ergibt sich analog dem Kap.2.5.3.3 und den Bildern 2.18 bis 2.20 folgendes Bild 5.7 als Basis für die Analyse des Fenstermechanismus.

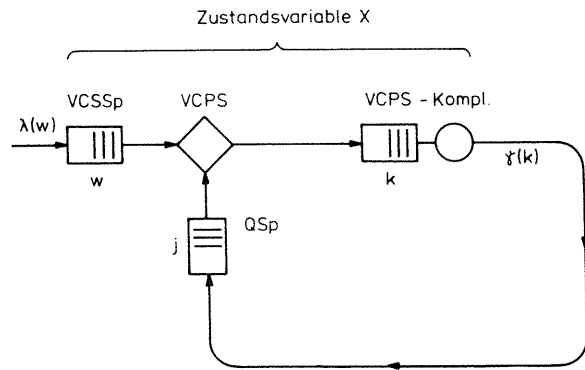


Bild 5.7: Modell für die Fenstermechanismusanalyse

Bild 5.7 besteht im wesentlichen aus den Speichern VCSSp für ankommende Nachrichtenblöcke und QSp für zur Verfügung stehende Quittungen (die Anzahl Quittungen im QSp repräsentiert die momentane Fenstergröße) sowie aus der VC-Protokollsteuerung VCPS und dem in der zweiten Bedieneinheit mit zugehöriger Warteschlange aggregierten System des VCPS-Komplements.

Zudem gelten folgende Vereinbarungen, mit W als Anzahl Warteplatze im VCSSp:

- w: Anzahl wartender Nachrichtenblöcke im VCSSp

$$0 \leq w \leq W \quad ( 5.4 )$$

- j: Anzahl Quittungen im QSp (momentane Fenstergröße)

$$0 \leq j \leq w_S \quad ( 5.5 )$$

- k: Anzahl Pakete im VCPS-Komplement

$$0 \leq k \leq w_S \quad ( 5.6 )$$

mit der Zusatzbedingung:

$$k + j = w_S \quad ( 5.7 )$$

Aus dem im vorigen Kapitel 5.2.2.1 beschriebenen Modell des Fenstermechanismus lassen sich folgende Beziehungen erkennen:

$$w = 0 \quad \text{für} \quad j > 0 \quad ( 5.8 )$$

$$w \geq 0 \quad \text{für} \quad j = 0 \quad ( 5.9 )$$

Zur Vereinfachung werden für die folgende Rechnung sowohl der Ankunftsprozeß als auch der Bedienprozeß als Markoff-Prozesse vorausgesetzt; ihre Ankunfts- und Bedienraten  $\lambda(w)$  bzw.  $\gamma(k)$  sind zustandsabhängig.

Für den Ankunftsprozeß wird zusätzlich vereinbart:

$$\lambda(w) = \begin{cases} \lambda & \text{für } w < W \\ 0 & \text{für } w = W \end{cases} \quad ( 5.10 )$$

Zur weiteren Vereinfachung werden die Größen w und k zu einer Zustandsvariablen X mit ihrem Wertebereich:

$$0 \leq x \leq w_S + W \quad (5.11)$$

zusammengefaßt, so daß sich der in Bild 5.8 gezeigte, zugehörige, eindimensionale Zustandsraum mit Übergangsraten aufzeigen läßt.

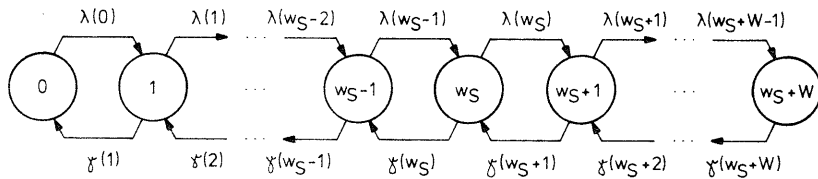


Bild 5.8: Zustandsraum

Innerhalb dieses eindimensionalen Zustandsraumes gilt dann für die Raten des Ankunftsprozesses, entsprechend Gleichung (5.10), und des Bedienprozesses:

$$\lambda(x) = \begin{cases} \lambda & \text{für } x < w_S + W \\ 0 & \text{für } x = w_S + W \end{cases} \quad (5.12)$$

$$\gamma(x) = \begin{cases} \gamma(k) & \text{für } x=k < w_S \\ \gamma(w_S) & \text{für } w_S \leq x \leq w_S + W \end{cases} \quad (5.13)$$

Die Enderate  $\gamma(k)$  des aggregierten Systems gilt für die hier beschriebene Analyse des Modells nach Bild 5.7 zunächst als gegeben. Sie wird entsprechend dem folgenden Kap.5.2.2.3 mit Hilfe der Mittelwertanalyse bestimmt. Die Größe k besagt dabei nichts weiter, als daß sich im VCPS-Komplement momentan k Pakete, mit  $0 \leq k \leq w_S$ , pro virtueller Verbindung befinden (siehe hierzu Kap.5.2.2.3).

Die Lösung des eindimensionalen Zustandsraumes ergibt sich durch Lösen des Gleichgewichtssystems für das statistische Gleichgewicht.

Die Wahrscheinlichkeit  $p(x)$  für das Auftreten eines beliebigen Zustandes läßt sich entweder geschlossen angeben durch

$$\frac{p(x)}{p(0)} = \frac{\lambda(0) \cdot \lambda(1) \cdot \lambda(2) \cdot \dots \cdot \lambda(x-1)}{\gamma(1) \cdot \gamma(2) \cdot \dots \cdot \gamma(x-1) \cdot \gamma(x)} \quad (5.14)$$

oder durch die folgende, rekursive Beziehung:

$$\frac{p(x)}{p(x-1)} = \frac{\lambda(x-1)}{\gamma(x)} \quad (5.15)$$

Mit Hilfe der Normierungsbedingung

$$\sum_{x=0}^{w_S+W} p(x) = 1 \quad (5.16)$$

können dann die Absolutwerte der Zustandswahrscheinlichkeiten berechnet werden.

Mittels der Zustandswahrscheinlichkeiten lassen sich schließlich unter spezieller Berücksichtigung der Gleichungen (5.4) bis (5.9) folgende Größen ermitteln:

- mittlere Anzahl  $\bar{w}$  gleichzeitig wartender Nachrichtenblöcke im VCSSp:

$$\bar{w} = \sum_{w=1}^W w \cdot p(w_S + w) \quad (5.17)$$

- mittlere Anzahl  $\bar{k}$  sich gleichzeitig im VCPS-Komplement befindender Nachrichtenblöcke:

$$\bar{k} = \sum_{k=1}^{w_S} k \cdot p(k) + w_S \cdot \sum_{k=w_S+1}^{w_S+W} p(k) \quad (5.18)$$



Die Gleichung (5.18) kann derart umgeformt werden, daß gilt:

$$\bar{k} = w_S - \sum_{k=0}^{w_S-1} (w_S-k) \cdot p(k) \quad (5.19)$$

mit dem Summenterm als mittlere momentane Fenstergröße  $\bar{j}$  (vgl. Gleichung (5.7)).

Der Durchsatz D des Subsystems ergibt sich schließlich zu:

$$D = \sum_{x=0}^{w_S+W} \gamma(x) \cdot p(x) = \lambda \cdot [1 - p(w_S+W)] \quad (5.20)$$

Entsprechend dem Gesetz von Little (vgl. Kap.2.5.3.2, Gleichung (2.1)) kann abschließend sowohl der Mittelwert der Wartezeit  $T_W$  im VCSSp als auch der Transitzeit  $T_T$  im VCPS-Komplement bestimmt werden:

$$E[T_W] = \bar{w} / D \quad (5.21)$$

$$E[T_T] = \bar{k} / D \quad (5.22)$$

Für die mittlere Transferzeit  $E[T_F]$  eines Nachrichtenblockes durch das System, d.h. für die Dauer zwischen Aufnahme im VCSSp und Verlassen der VC-Protokollsteuerung der Gegenseite, gilt:

$$E[T_F] = E[T_W] + E[T_T] \quad (5.23)$$

### 5.2.2.3 Analyse des Netzwerkeinflusses

Dieses Kapitel hat die Aggregierung des VCPS-Komplements entsprechend Bild 5.6 in eine einzige Bedienstation mit Warteschlange und flußäquivalentem Verkehr zur Aufgabe.

Ziel hierbei ist die Bestimmung der populationsgrößenabhängigen Durchsatzrate auf einem logischen Kanal, unter Berücksichtigung jener Einflüsse, die dadurch entstehen, daß aufgrund der Multiplexfunktion diverse Netzressourcen, wie Prozessoren und Speicher, gemeinsam in Anspruch genommen werden.

Die Ermittlung dieser Durchsatzraten erfolgt unter Zuhilfenahme der Mittelwertanalyse (vgl. Kap.2.5.3.2), unter gewissen Voraussetzungen, auf die im Verlauf dieses Kapitels näher eingegangen wird.

Entsprechend Bild 5.6 sowie unter der vereinfachenden Annahme, daß sich für jede virtuelle Verbindung und für jede Nutzinformationsrichtung eine geschlossene Kette mit je  $w_S$  Daten- und/oder Steuerpaketen ergibt, erhält man ein Netz mit mehreren geschlossenen Ketten (vgl. Bild 5.9).

Bild 5.9 zeigt schematisch die einzelnen Komponenten im VCPS-Komplement, die von den Paketen durchlaufen werden:

- ① und ④ : MDM: Teilmodell Multiplexer/Demultiplexer mit Bearbeitungszeiten  $T_S$  bzw.  $T_E$
- ② und ⑤ : LAPB-Steuerung: Teilmodell LAPB-Sendeeinrichtung mit Speicher LAPB-SSp und Übertragungszeiten  $T_{ODP}$  für Datenpakete bzw.  $T_{OSP}$  für Steuerpakete
- ③ und ⑥ : VDX-Leitung: Teilmodell Übertragungskanal mit Kanallaufzeit  $T_L$ .

Pro virtueller Verbindung ergeben sich zwei geschlossene Ketten, je eine für die beiden Nutzinformationsrichtungen.

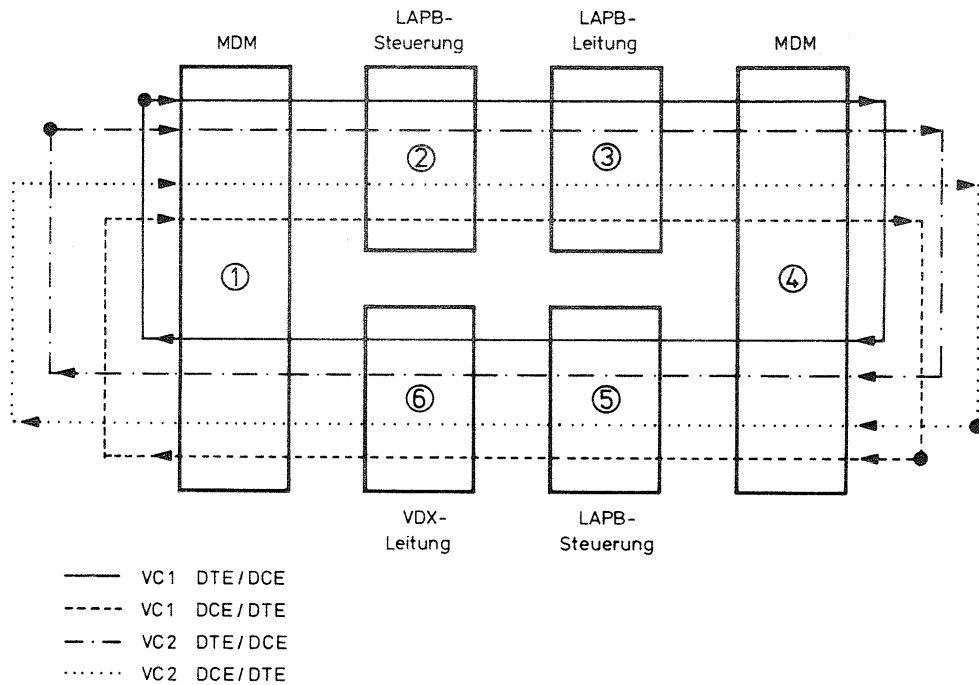


Bild 5.9: Netzmodell des VCPS-Komplements mit mehreren geschlossenen Ketten

Entsprechend der in Kap.2.5.3.2 definierten Nomenklatur ergibt sich in diesem speziellen Anwendungsfall:

- Anzahl R an geschlossenen Ketten:  
 $R = 2n$  ( 5.24 )

mit n = Anzahl virtueller Verbindungen und

- Anzahl N an Bedienstationen im Netz:  
 $N = 6$  ( 5.25 )

Die im folgenden verwendeten Indizes

- i = Nr. der betrachteten Bedienstation und
- r = Nr. der betrachteten Kette

haben dabei die Wertebereiche:

$$1 < i < N \quad ( 5.26 )$$

$$1 < r < R$$

Für die mittleren Bediendauern  $h_i^r$  in der Bedieneinheit i für Pakete der Kette r ergibt sich für das in Bild 5.9 gezeigte Beispiel zweier aufgebauter virtueller Verbindungen folgende Matrix:

Kette	Station					
	1	2	3	4	5	6
VC1: DEE/DOE	$T_S$	$T_{ODP}$	$T_L$	$T_E$	$T_{OSP}$	$T_L$
DOE/DEE	$T_E$	$T_{OSP}$	$T_L$	$T_S$	$T_{ODP}$	$T_L$
VC2: DEE/DOE	$T_S$	$T_{ODP}$	$T_L$	$T_E$	$T_{OSP}$	$T_L$
DOE/DEE	$T_E$	$T_{OSP}$	$T_L$	$T_S$	$T_{ODP}$	$T_L$

( 5.27 )

mit  $T_{ODP}$  und  $T_{OSP}$  entsprechend den Gleichungen (5.1) und (5.2).

Die Lösung dieses Netzmodells mit mehreren geschlossenen Ketten kann dabei auf mehrere Arten erfolgen, vgl. hierzu die Einleitung zu Kap.5.2.2. Für die Lösung des gegebenen speziellen Problems wurde aus Gründen der Komplexität ein auf der Mittelwertanalyse basierendes approximatives Verfahren gewählt.

Die exakte Lösung des Netzmodells mittels der in den Gleichungen (2.24) bis (2.27) beschriebenen Rekursion (vgl. Kap.2.5.3.2) und dem zu verwendenden Populationsvektor  $IK$ :

$$IK = (k^1, k^2, \dots, k^r, \dots, k^R), \quad (5.28)$$

mit  $0 \ll k^r \ll w_s$

läßt erkennen, daß es sich aufgrund dieses Populationsvektors um ein Rechenverfahren mit einem hohen Komplexitätsgrad  $C_{rek}$ , sowohl für den benötigten Speicherplatz als auch für die Anzahl der zu durchlaufenden Rekursionsschleifen, der folgenden Größenordnung handelt:

$$C_{rek} \approx \prod_{r=1}^R k^r \quad (5.29)$$

Dieser Sachverhalt legt die Überlegung nahe, die rechenintensive Rekursion durch ein schnelleres, approximatives Iterationsverfahren zu ersetzen (vgl. hierzu /30/).

Mittelwertanalyse mittels Iteration:

Es werde zunächst die Größe  $\mathcal{E}_i^r(IK)$  definiert, welche angibt, um wieviel die mittlere Anzahl von Anforderungen in Bedienstation  $i$  erhöht wird, wenn dem System im Zustand  $S(IK-e^r)$  eine zusätzliche Anforderung der Kette  $r$  hinzugefügt wird:

$$\mathcal{E}_i^r(IK) = n_i(IK) - n_i(IK-e^r) \quad (5.30)$$

Die Größen  $\mathcal{E}_i^r(IK)$  werden im folgenden als "Verteilwahrscheinlichkeiten" bezeichnet.

Damit läßt sich die Rekursion, entsprechend den Gleichungen (2.24) bis (2.27), in eine nur noch vom Populationsvektor  $IK$  allein abhängige Form umschreiben in:

$$t_i^r = \begin{cases} g_i^r \cdot (1 + n_i - \mathcal{E}_i^r) & \text{Fall 1,2} \\ g_i^r & \text{Fall 3} \end{cases} \quad (5.31)$$

$$\gamma^r = k^r / \sum_{i \in Q(r)} t_i^r \quad (5.32)$$

$$n_i = \sum_{j \in R(i)} \gamma^j \cdot t_i^j \quad (5.33)$$

Die Gleichungen (5.31) bis (5.33) stellen ein nichtlineares Gleichungssystem dar, wobei aus Gründen der Übersichtlichkeit der Populationsvektor weggelassen wurde, welches durch geeignete Wahl der Anfangswerte für  $\gamma^r$  und  $n_i$  mittels einer einfachen Iteration gelöst werden kann.

Genau betrachtet ist die Gleichung (5.30) nicht weniger komplex als das ursprüngliche rekursive Verfahren. In /30/ wird jedoch eine Approximation vorgestellt, die letztendlich die  $R$  Ketten in einer einzigen Kette vereinigt, welche dann effektiv durch die in den Gleichungen (2.6) bis (2.9) beschriebene einfache Rekursion gelöst werden kann.

Approximation von  $\mathcal{E}_i^r$ :

Wird eine Anforderung (Paket) aus einer Kette  $r$  entnommen, so wirkt sich diese Maßnahme auf alle Werte  $n_i^r$  aus.

Da jedoch die Beziehung gilt:

$$\sum_{i \in Q(r)} \{n_i^j(IK) - n_i^j(IK-e^r)\} = 0, \quad \forall j \neq r \quad (5.34)$$

wobei  $r$  diejenige Kette ist, der das Paket entnommen wurde, kann in erster Näherung angenommen werden, daß der Wert von  $\mathcal{E}_i^r$  am meisten durch die Kette  $r$  beeinflußt wird.

Diese Überlegung führt zu einer Approximation von  $\xi_i^r$  derart, daß der Wert von  $\xi_i^r$  aus Ergebnissen einer einzigen geschlossenen Kette mit korrigierten Parametern abgeschätzt wird.

Die Kapazität der Bedieneinheit  $i$ , die der Kette  $r$  zur Bearbeitung ihrer Anforderungen im Mittel noch verbleibt, ist gegeben durch

$$1 - \sum_{j \in R(i), j \neq r} \gamma^j \cdot h_i^j \quad (5.35)$$

Unter dem Aspekt der "fluid dynamic approximation" (vgl. hierzu /10/) muß somit auch die Bedienrate  $\mu_i^r$  dieser Bedieneinheit  $i$  für Kette- $r$ -Anforderungen entsprechend angepaßt werden:

$$\hat{\mu}_i^r = \mu_i^r \left( 1 - \sum_{j \in R(i), j \neq r} \gamma^j \cdot h_i^j \right) \quad (5.36)$$

Die mittlere, korrigierte Bediendauer ergibt sich somit zu:

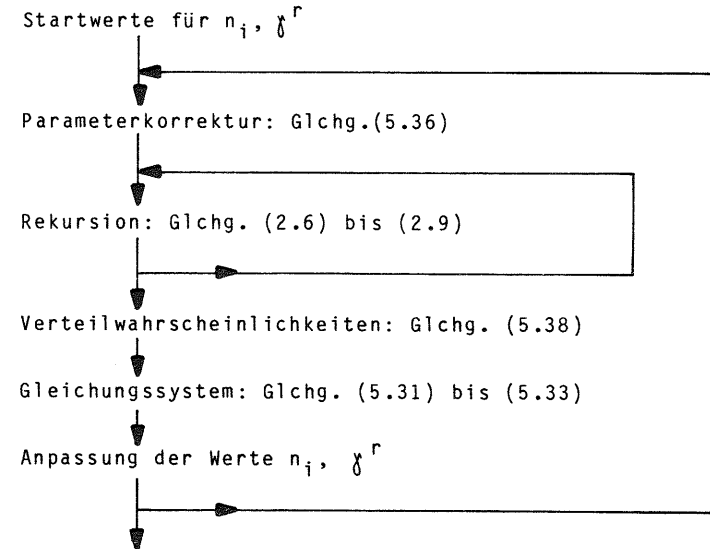
$$\hat{h}_i^r = h_i^r / \left( 1 + \gamma^r \cdot h_i^r - \sum_{j \in R(i)} \gamma^j \cdot h_i^j \right) \quad (5.37)$$

Mittels dieser angepaßten Bediendauern lassen sich in der durch die Gleichungen (2.6) bis (2.9) beschriebenen Rekursion die Größen  $n_i^r(K^r)$  bestimmen, die die mittlere Anzahl von Anforderungen in Station  $i$  angeben innerhalb eines aus einer einzigen geschlossenen Kette bestehenden Systems mit insgesamt  $|Q(r)|$  Stationen, korrigierten Parametern nach Gleichung (5.37) und der Populationsgröße  $K^r$ .

Für  $\xi_i^r$  folgt dann (vgl. Gleichung (5.30)):

$$\xi_i^r = n_i^r(K^r) - n_i^r(K^r-1) \quad (5.38)$$

Die Lösung des iterativ zu lösenden Gleichungssystems (5.31) bis (5.33) erfolgt durch zyklisches Durchlaufen folgender Gleichungen in der angegebenen Reihenfolge:



Die innerhalb des Gesamtzyklus aufgeführte Rekursion wird pro Kette im VCPS-Komplement, d.h. also insgesamt  $R$ -mal, durchlaufen. Der gesamte Zyklus wird so oft durchlaufen, bis die Abweichungen der beiden Größen  $n_i$  und  $\gamma^r$  zwischen zwei Durchläufen kleiner sind, als im Programm vorgebbare Grenzen. Damit steht als Hauptergebnis der Durchsatz  $\gamma^r$  der Ketten im VCPS-Komplement fest, womit schließlich die gewünschte Aggregation in eine einzelne Bedienstation mit einer Warteschlange und flußäquivalentem Verkehr  $\gamma(k)$  (vgl. Bild 5.7) vorgenommen werden kann.

Da bei der Dekomposition hauptsächlich die virtuelle Verbindung Nr.1 betrachtet wurde (vgl. Bild 5.6), gilt entsprechend Bild 2.20:

$$\gamma(k) = \gamma^1 \Big|_{K^1=k} \quad (5.39)$$

Der Komplexitätsgrad beträgt nunmehr nur noch:

$$C_{iter} \approx \sum_{r=1}^R K^r \quad ( 5.40 )$$

Im direkten Vergleich mit  $C_{rel}$  ergibt sich also bei der näherungsweisen Lösung von Warteschlangennetzen mit mehreren geschlossenen Ketten mittels der Iteration unter Verwendung der "Verteilwahrscheinlichkeiten" ein wesentlich geringerer Komplexitätsgrad und damit auch ein geringerer Bedarf an Schleifendurchläufen bzw. Programmlaufzeit.

#### 5.2.2.4 Zusammensetzung

Durch Zusammensetzen der in den Kapiteln 5.2.2.2 und 5.2.2.3 gewonnenen Ergebnisse für die beiden Subsysteme lassen sich die gewünschten Endergebnisse für die Analyse des Gesamtsystems ermitteln.

Dabei kommen zwei Möglichkeiten in Betracht, auf die hier kurz näher eingegangen werden soll:

- einfache Analyse und
- gewichtete Analyse.

#### Einfache Analyse:

Bei der einfachen Analyse werden die bei der Bewertung des VCPS-Komplements gewonnenen Ergebnisse unverändert für die Auswertung der Flußsteuerung übernommen.

Die mathematische Analyse der Schnittstelle X.25, genauer genommen die Bestimmung des Leistungsvermögens eines logischen Kanals (einer virtuellen Verbindung), hat hierbei folgenden prinzipiellen Verlauf:

Berechnung der Enderaten  $\gamma(k)$  des aggregierenden Systems mittels der Analyse des VCPS-Komplements, (vgl. Kap.5.2.2.3),

Ergebnis:

Enderaten  $\gamma(k)$  mit  $0 \leq k \leq w_S$



Analyse des Flußkontrollmechanismus unter Verwendung der nun bekannten Enderaten  $\gamma(k)$ , (vgl. Kap.5.2.2.2),

Ergebnisse:

mittlere Wartezeit  $E[T_W]$

mittlere Transitzeit  $E[T_T]$

mittlere Transferzeit  $E[T_F]$

Die einfache Analyse liefert im Vergleich mit simulativ ermittelten Werten beste Ergebnisse, solange sich das System im sog. gesättigten Zustand befindet. In diesem Fall stehen in den Speichern VCSSp immer Nachrichtenblöcke zum Senden an; ein Zustand, der bei hohem Verkehrsangebot pro virtueller Verbindung gegeben ist.

Der Grund für diese Voraussetzung ist darin zu sehen, daß bei der Ermittlung der Raten  $\gamma(k)$  mit Hilfe der Mittelwertanalyse des VCPS-Komplements davon ausgegangen wird, daß sich in allen R Ketten gleichzeitig k Pakete befinden, d.h. genau genommen gilt bei der einfachen Analyse:

$$\gamma^1 \Big|_{K^1=k} = \gamma^1 \Big|_{K^1=K^2=\dots=K^R=k} \quad ( 5.41 )$$

Dies ist jedoch nur unter der o.a. Voraussetzung, d.h. für gesättigten Verkehr, gewährleistet.

Für kleinere Angebotswerte kann jedoch nicht mehr generell davon ausgegangen werden, daß in allen R Ketten des VCPS-Komplements immer k Pakete gleichzeitig unterwegs sind.

Dies zeigt sich bei der Auswertung der Ergebnisse für kleinere Verkehrsangebote dadurch, daß die Werte für die Transferzeiten erheblich über denjenigen liegen, die simulativ ermittelt wurden. Da bei der einfachen Analyse von zuvielen sich gleichzeitig im VCPS-Komplement befindenden Paketen ausgegangen wird, entstehen längere Warteschlangen vor den gemeinsamen Netzressourcen, womit die größere Transferzeit zu erklären ist.

Dieser Sachverhalt veranlaßte die Entwicklung einer zweiten Möglichkeit, die Ergebnisse der beiden Subsysteme zusammenzusetzen.

Gewichtete Analyse:

Bei diesem Verfahren werden mittels der Analyse des VCPS-Komplements Durchsätze  $\delta(k,y)$  ermittelt, wobei gilt:

$$\delta(k,y) = \delta^1 \left| \begin{array}{l} k^1=k, k^2=k^3=\dots k^n=y, k^{n+1}=k, k^{n+2}=\dots k^R=y \\ \text{mit } 0 \leq k,y \leq w_S \end{array} \right. \quad (5.42)$$

Im Gegensatz zur einfachen Analyse (vgl. auch Gleichung (5.41)) werden bei der gewichteten Analyse unterschiedliche Populationsgrößen für die virtuelle Verbindung Nr.1, d.h. Ketten 1 und n+1, und für die restlichen Ketten berücksichtigt. Es wird also vorausgesetzt, daß für den logischen Kanal 1 in beiden Datenübertragungsrichtungen unabhängig voneinander je k Pakete und in den restlichen R-2 Ketten unabhängig voneinander je y Pakete unterwegs sind. Aus diesen Durchsätzen  $\delta(k,y)$  wird schließlich durch entsprechende Gewichtung die für die Analyse des Fenstermechanismus benötigte Enderate  $\delta(k)$  bestimmt.

Für die Gewichtung wird dabei wie folgt vorgegangen: Als Startwert wird aus den Durchsätzen  $\delta(k,y)$  der arithmetische Mittelwert gebildet:

$$\delta(k) = \frac{1}{w_S+1} \cdot \sum_{y=0}^{w_S} \delta(k,y) \quad (5.43)$$

Mittels dieser Enderaten  $\delta(k)$  werden bei der Analyse des Flußkontrollmechanismus die Zustandswahrscheinlichkeiten p(0) bis p(w<sub>S</sub>) ermittelt, so daß in einem 2.Schritt Gewichtungsfaktoren g(y) folgender Art bestimmt werden können:

$$g(y) = \frac{p(y)^{2n-2}}{\sum_{j=0}^{w_S} p(j)^{2n-2}} \quad 0 \leq y \leq w_S \quad (5.44)$$

Der Gleichung (5.44) liegt dabei die Überlegung zugrunde, daß jeweils genau y Pakete mit der Wahrscheinlichkeit p(y) unabhängig voneinander in jeder der 2n-2 verbleibenden Ketten sind.

Der Summenterm im Nenner von Gleichung (5.44) folgt aus der Normierungsbedingung:

$$\sum_{y=0}^{w_S} g(y) = 1 \quad (5.45)$$

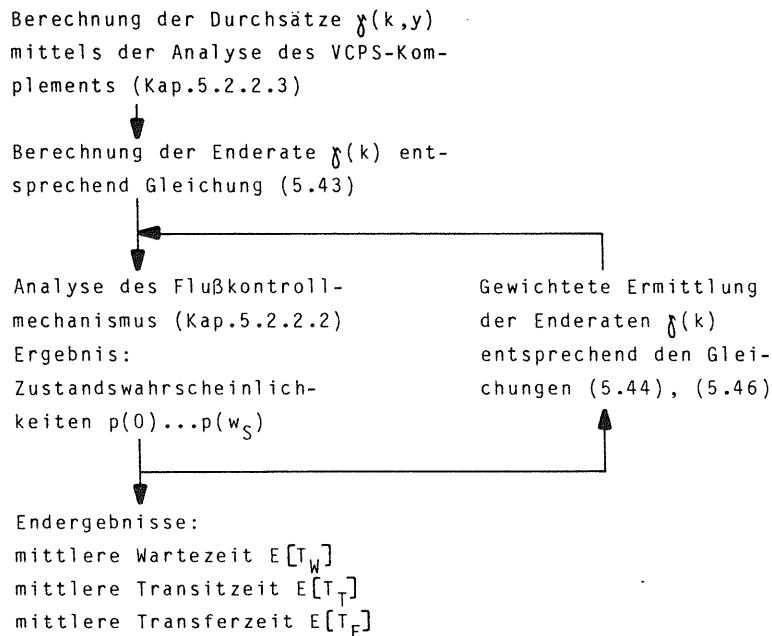
Mit diesen Gewichtungsfaktoren wird dann die Enderate  $\delta(k)$  neu berechnet zu:

$$\delta(k) = \sum_{y=0}^{w_S} g(y) \cdot \delta(k,y) \quad (5.46)$$

Die Anpassung der Enderaten  $\gamma(k)$  mittels der Gewichtungsfaktoren erfolgt so oft, bis die Abweichung kleiner als eine im Programm vorgebbare Grenze ist.

Die gewichtete Analyse berücksichtigt damit automatisch den sich durch den gegebenen Verkehr einstellenden Systemzustand in den mit Zustandswahrscheinlichkeiten gewichteten Enderaten  $\gamma(k)$ . Für geringen Verkehr ist, pauschal formuliert, der Zustand mit kleinen Werten von  $k$ , d.h. mit kleinen Populationsgrößen in den Ketten, wahrscheinlicher als der mit großen Werten. Dies wird in der gewichteten Ermittlung der Enderaten  $\gamma(k)$  berücksichtigt, so daß bei schwachem Verkehr keine so langen Warteschlangen vor gemeinsam genutzten Netzressourcen entstehen, womit die Transferzeiten deutlich abnehmen und den simulativ ermittelten Werten entsprechen (siehe Kap.6).

Die gewichtete Analyse hat schließlich folgenden Ablauf:



Es soll in diesem Zusammenhang nicht unerwähnt bleiben, daß eine weitere Verfeinerung in der Ermittlung der Durchsätze, z.B. der Form:

$$\gamma(k,y,z) = \gamma^1 \Big|_{K^1=k, K^2=y, K^3=\dots, K^R=z} \quad (5.47)$$

bis schließlich

$$\gamma^1(K^1, K^2, K^3, \dots, K^R) = \gamma^1(IK) \quad (5.48)$$

mit jeweils unterschiedlichen und unabhängigen Werten  $K^r$  zu absolut übereinstimmenden Ergebnissen, verglichen mit Simulationswerten, führen wird.

Letztendlich, d.h. mit Gleichung (5.48), wäre jedoch derselbe Komplexitätsgrad in Speicher- und Programmaufwand erreicht wie durch die in den Gleichungen (2.24) bis (2.27) beschriebene Rekursion mit dem durch Gleichung (5.28) definierten Populationsvektor  $IK$ .

### 5.3 Geltungsbereich der Analysemethoden

Die Aufgabe dieses Kapitels ist, Bereiche aufzuzeigen, in denen die Analysemethoden gute und gesicherte Ergebnisse erbringen.

Aufgrund der durch den überaus großen Komplexitätsgrad des Systems X.25 bedingten Annahmen und Vereinfachungen kann die Analyse, insbesondere die mathematische, nicht für jede gegebene Parameterkonstellation Ergebnisse liefern.

### 5.3.1 Simulative Analyse

Aufgrund der realitätsnahen, zeittreuen Abbildung des Ablaufgeschehens liefert die Simulationsmethode für jeden Parametersatz die sichersten Ergebnisse für die Ermittlung des Leistungsvermögens von X.25.

Die ausschließliche Anwendung der Simulation wird lediglich durch die enorm langen Programmlaufzeiten beschränkt, die bei ungünstig korrelierten Parameterwerten zur Ermittlung nur eines einzigen Ergebnisses bis zu mehreren Stunden betragen können.

In diesem Rahmen sei nochmals darauf hingewiesen, daß bei der in der vorliegenden Arbeit beschriebenen Programmimplementierung sämtliche Prozeduren, Funktionen und Verkehrsabläufe innerhalb der Schicht 2 unberücksichtigt geblieben sind.

Mit anderen Worten heißt das, daß sämtliche im nächsten Kapitel 6 erläuterten Ergebnisse das Systemverhalten zu optimistisch wiedergeben. Unter Berücksichtigung der Einflüsse von Schicht 2 können sämtliche Werte nur schlechter werden.

### 5.3.2 Mathematische Analyse

Die in den Kapiteln 5.2.2.2 und 5.2.2.3 beschriebene mathematische Analyseverfahren liefert dann gute und simulativ abgesicherte Ergebnisse, wenn die in den Analysen der Subsysteme implizit getroffenen Annahmen und Voraussetzungen erfüllt sind.

Im einzelnen sind dies:

1. einheitliche Fenstergröße  $w_S$  für alle  $n$  virtuellen Verbindungen,
2. einheitlicher, negativ exponentiell verteilter Nachrichtenblockankunftsabstand für alle  $n$  virtuellen Verbindungen,

3. einheitliche und konstante Blocklänge für alle  $n$  Verbindungen,
4. zyklische Abfertigung der  $n$  Sendespeicher VCSSp durch die MDM-Steuerung.

Da die Punkte 1. und 4. in der Regel durch Implementierungsaspekte erfüllt sind und die Punkte 2. und 3. ebenfalls meistens der Realität sehr nahe kommen, läßt sich o.a. mathematische Analyse für die meisten real vorkommenden Verhältnisse zur Ermittlung der Leistungsfähigkeit virtueller Verbindungen auf einem X.25 Netzzugangsabschnitt in geeigneter Weise verwenden.



## 6. ERGEBNISSE DER LEISTUNGSFÄHIGKEIT

Ziel dieses Kapitels ist es, leistungsbeeinflussende Parameter aufzuzeigen, um Aufschluß über die Güte virtueller Verbindungen zu geben. Der Schwerpunkt der Untersuchungen liegt dabei auf der Ermittlung des Durchsatzes an Datenpaketen und der Bestimmung der mittleren Transferzeit.

Um die Zuverlässigkeit der in Kap.5.2.2 beschriebenen Berechnungsmethode zu überprüfen, wurden die Simulationsergebnisse in den Diagrammen mit eingezeichnet. Auf das zusätzliche Markieren der bei jedem Simulationslauf ebenfalls ermittelten Vertrauensintervalle für eine statistische Aussagesicherheit von 95% wurde generell verzichtet; einerseits aus Gründen der Übersichtlichkeit und andererseits aufgrund der Tatsache, daß in den meisten Fällen die Breite des Vertrauensintervalls in der Größenordnung der verwendeten Strichstärken lag.

Das Kapitel schließt mit Empfehlungen für die optimale Einstellung von Parametern sowie mit einem Vergleich der vorgestellten Ergebnisse mit Messungen im realen DATEX-P Datennetz der Deutschen Bundespost.

### 6.1 Leistungsbeeinflussende Parameter

Wie in Kap.3.3 tabellarisch aufgeführt, existiert - bedingt sowohl durch die verbale Spezifikation der CCITT-Empfehlung X.25 als auch durch spezielle Implementierungen und reale Anwendungen - eine Fülle an Parametern, die die Leistungsfähigkeit der Schnittstelle in zum Teil erheblichem Maße beeinflussen.

Da das Aufzeigen des Einflusses aller nur denkbaren (und sinnvollen) Parameterkonstellationen auf die Leistungsfähigkeit von X.25 den Rahmen dieser Arbeit sprengen würde, beschränkt sich die vorliegende Untersuchung auf die Leistungsbeeinflussung durch folgende Struktur- und Verkehrsparameter:

- Anzahl  $n$  aufgebauter und aktiver virtueller Verbindungen (VC)
- physikalische Laufzeit  $T_L$  auf der Verbindungsleitung (inklusive den Verarbeitungszeiten in den Schichten 1 und 2)
- Fenstergröße  $w_S$
- Verkehrsangebot  $A$  an die Schnittstelle
- Verteilungsfunktion und Mittelwert des Nachrichtenblockankunftsabstandes
- Verteilungsfunktion und Mittelwert der Nachrichtenblocklänge NBL.

Die folgenden Parameter wurden in den hier vorgestellten Untersuchungen unverändert belassen:

- Physikalische Übertragungsgeschwindigkeit auf der Verbindungsleitung  $v = 48000\text{bit/s}$
- Prozessorzeiten in Schicht 3  $T_S = T_E = 1\text{ms}$
- Anzahl Wartepplätze im VCSSp  $W = 50$ .

### 6.2 Durchsatz-Untersuchungen virtueller Verbindungen

In diesem Abschnitt werden Ergebnisse der analytischen und simulativen Ermittlung des Durchsatzes virtueller Verbindungen auf einem Übermittlungsabschnitt, welcher entsprechend X.25 gesteuert wird, vorgestellt und beschrieben. Der Durchsatz  $D$  ist die maximale Anzahl an Informationsbits, welche pro Zeiteinheit auf einer virtuellen Verbindung übertragen werden kann.

Durchsatzuntersuchungen sind bei solchen Anwendungen von großem Interesse, bei denen größere Datenmengen über einen längeren Zeitraum übertragen werden sollen.

Für die Durchsatzberechnungen wurden stets g e s ä t t i g t e Verkehrssituationen betrachtet, bei denen für jede virtuelle Verbindung und für jede Hauptübertragungsrichtung zu jeder Zeit Nachrichtenblöcke zum Senden anstanden.

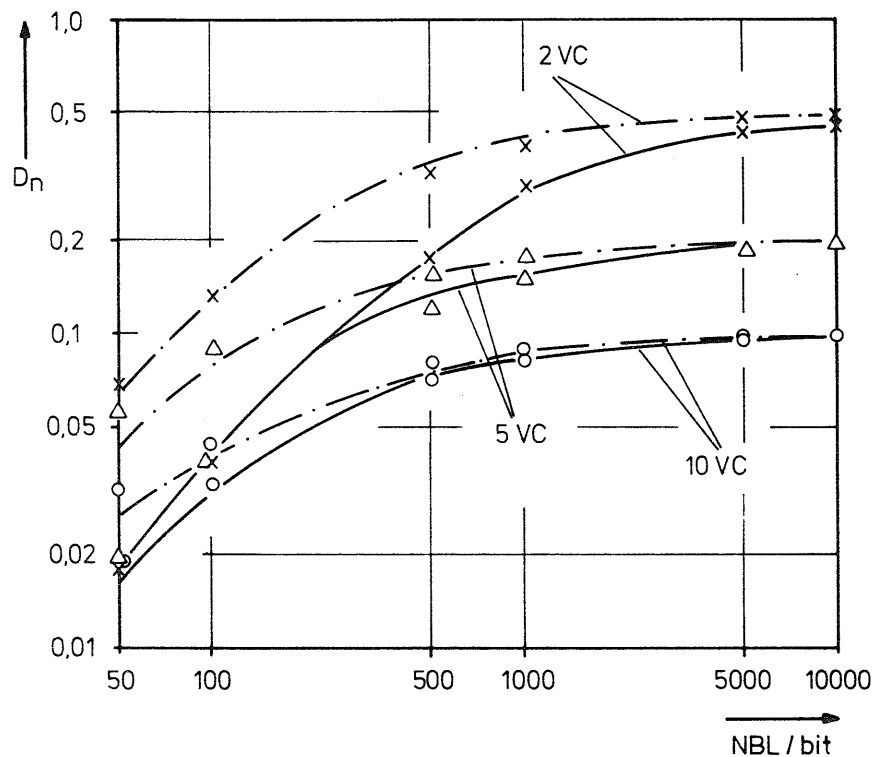


Bild 6.1: Normierter Durchsatz  $D_n$  als Funktion der Nachrichtenblocklänge  $NBL$ ; ( $T_L=50ms$ ,  $v=48000bit/s$ )  
 Berechnung: —  $w_S=2$   
 - - -  $w_S=7$   
 Simulation: × 2 VC  
 Δ 5 VC  
 ○ 10 VC

Bild 6.1 zeigt den normierten Durchsatz  $D_n$  einer virtuellen Verbindung als Funktion der konstanten Nachrichtenblocklänge  $NBL$ , der Anzahl  $n$  virtueller Verbindungen (VC) an der Schnittstelle sowie der einheitlichen Sender-Fenstergröße  $w_S$ . Der normierte Durchsatz  $D_n$  ist dabei der Durchsatz  $D$  einer virtuellen Verbindung bezogen auf die Übertragungsgeschwindigkeit  $v$ .

Die Laufzeit  $T_L=50ms$  schließt ferner konstante Verzögerungen für Verarbeitung, Umspeichern usw. der Schichten 1 und 2 ein. Desweiteren ist es mit diesem Wert möglich, Vergleiche mit früheren Leistungsuntersuchungen, /66-73/, bezüglich Schicht 2 anzustellen.

Die Kurven zeigen das typische Durchsatzverhalten von Übertragungsprotokollen ohne Fehlererkennung und Wiederholungsmechanismen, (vgl. Kap.5.2.1.1). Bei kurzen Nachrichtenblocklängen ist der Durchsatz gering, da der Anteil an zusätzlichen Bits für Steuerzwecke ( $SI(2)=48bit$ ,  $SI(3)=24bit$ , jeweils für  $MODULO=8$ ) relativ hoch ist. Bei langen Blocklängen ist dieser Anteil vernachlässigbar und die Kurven streben ihrem jeweiligen Grenzwert  $D_{n,gr}$  zu:

$$D_{n,gr} = 1/n \quad (6.1)$$

Der Einfluß des Fenstermechanismus auf den Durchsatz ist besonders deutlich bei den Kurven für  $n=2$  VC zu erkennen: bei kleinen Blocklängen ist der Durchsatz sehr gering, da der Sender aufgrund des Fenstermechanismus und der Fenstergröße  $w_S=2$  sehr oft und schnell ( $v=48kbit/s$ ) den oberen Fensterrand erreicht und den Sendevorgang unterbricht (vgl. Kap.2.3.2). Eine Vergrößerung des Sendefensters um den Faktor 3,5, d.h. auf  $w_S=7$ , ergibt für Nachrichtenblocklängen  $NBL < 300bit$  eine Durchsatzverbesserung ebenfalls um den Faktor 3,5. Dieser Verbesserungseffekt verliert sich zunehmend mit größer werdender Blocklänge.

Bei mehreren virtuellen Verbindungen (z.B.  $n=10$ ) bringt eine Vergrößerung des Sendefensters keine Verbesserung in dem Maße wie bei 2 VC, da aufgrund des zwangsläufig längeren Multiplexzyklus (vgl. Kap.5.2.1.2) schon bei kleiner Fenstergröße die Wahrscheinlichkeit recht hoch ist, bereits vor Erreichen des oberen Fensterrandes Quittungen mit neuen Empfangsfolgennummern zu empfangen.

Weitere interessante Aufschlüsse über das Verhalten der Schnittstelle X.25 ergeben Untersuchungen über den erzielbaren Durchsatz bei unterschiedlichen Laufzeiten.

Bild 6.2 verdeutlicht den normierten Durchsatz  $D_n$  einer virtuellen Verbindung als Funktion der Laufzeit  $T_L$  (inklusive Verarbeitungszeiten in Schichten 1 und 2), der Anzahl  $n$  virtueller Verbindungen sowie der Fenstergröße  $w_S$ . Die Laufzeit  $T_L$  ist dabei auf die Übertragungszeit  $T_{ODP}$  eines Datenpakets normiert. Für die Nachrichtenblocklänge wurde einheitlich der Wert  $NBL=1000\text{bit}$  gewählt, da diese Größenordnung dem Wert der Standardpaketgröße entspricht, die in X.25 empfohlen wird.

Die Kurven zeigen auch hier das typische Durchsatzverhalten, indem der Durchsatz an Datenpaketen für lange Laufzeiten ( $T_L/T_{ODP} > 2$ ) abnimmt, da die durch die hohen Laufzeiten bedingten langen Quittierzeiten den Sender an den oberen Fensterrand stoßen lassen und somit den Datenfluß unterbrechen. Dieser Effekt ist sehr gut daran zu erkennen, daß eine Vergrößerung des Sendefensters eine spürbare Verbesserung im Durchsatzverhalten der Schnittstelle mit sich bringt. Diese Durchsatzsteigerung ist jedoch umso schwächer, je mehr virtuelle Verbindungen sich die Übertragungskapazität teilen. Für kurze Laufzeiten ( $T_L/T_{ODP} < 0,2$ ) ergibt eine Sendefenstervergrößerung keine Durchsatzsteigerung, da aufgrund der kurzen Zeiten und der kleinen Fenstergröße auch bei wenigen virtuellen Verbindungen und damit kurzem Multiplexzyklus die Wahrscheinlichkeit relativ hoch ist, bereits vor Erreichen des oberen Fensterrandes Quittungen mit neuen Empfangsfolgennummern zu empfangen.

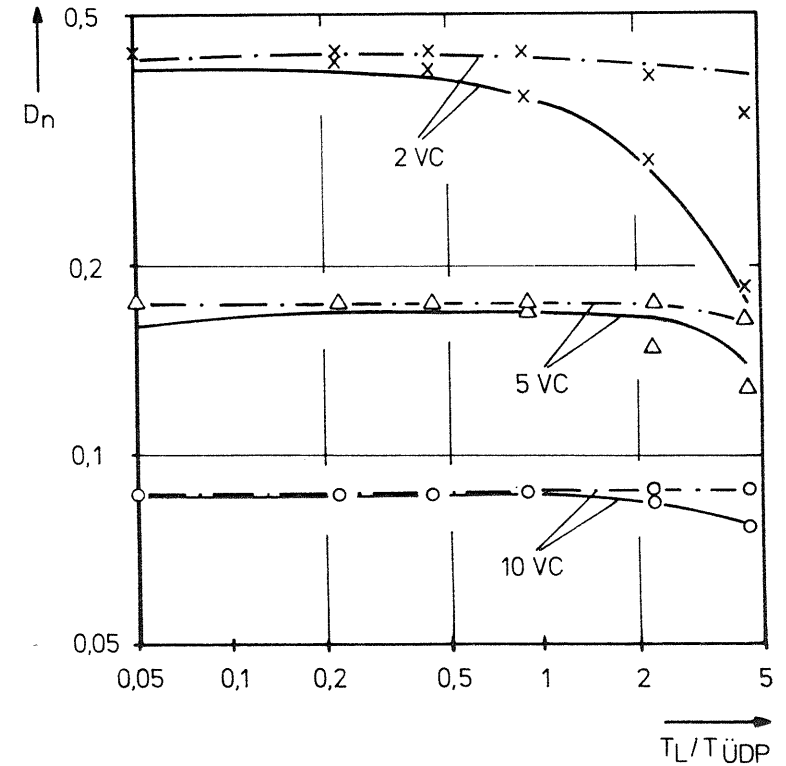


Bild 6.2: Normierter Durchsatz  $D_n$  als Funktion der Laufzeit  $T_L$ ;  
 ( $NBL=1000\text{bit}$ ,  $v=48000\text{bit/s}$ ,  $T_{ODP}=22,3\text{ms}$ )  
 Berechnung: —  $w_S=2$   
 - · - ·  $w_S=7$   
 Simulation: × 2 VC  
 △ 5 VC  
 ○ 10 VC

Wie in Kap.3.3.2 erwähnt, sieht die CCITT-Empfehlung X.25 einige optionale Benutzer- und Leistungsmerkmale vor, die die Leistungsfähigkeit in zum Teil erheblichem Maße beeinflussen können. Ein derartiges, leistungsbeeinflussendes, optionales Merkmal ist die verbindungsindividuelle Wahl der Fenstergröße  $w_S$ .

Es lag daher nahe, das Durchsatzverhalten der Schnittstelle mit mehreren virtuellen Verbindungen, aber unterschiedlichen Fenstergrößen  $w_S$  zu untersuchen. Hierzu wurde die folgende Systemkonfiguration gewählt:

Insgesamt werden  $n=5$  virtuelle Verbindungen betrachtet, wobei sowohl die beiden ersten als auch die beiden letzten Verbindungen (VC-Nr. 1,2,4,5) einheitlich die Fenstergröße  $w_S=2$  benutzen, während die verbleibende (mittlere) virtuelle Verbindung (VC-Nr. 3) durch die größere Fenstergröße  $w_S=7$  bevorzugt wurde.

Bild 6.3 zeigt sowohl den normierten Durchsatz  $D_n$  der bevorzugten Verbindung (VC-Nr. 3;  $w_S=7$ ) als auch den arithmetischen Mittelwert der erzielten Durchsätze der anderen Verbindungen (VC-Nr. 1,2,4,5;  $w_S=2$ ) als Funktion der konstanten Nachrichtenblocklänge NBL. In diesem Zusammenhang sei angemerkt, daß die Kurven ausschließlich durch Auswertung von Simulationsergebnissen gewonnen wurden (vgl. hierzu Kap.5.3).

Auch in diesem speziellen Anwendungsfall zeigen die Kurven typisches Durchsatzverhalten, d.h. geringer Durchsatz für kleine Blocklängen und Grenzwerte für den maximal erreichbaren Durchsatz bei langen Nachrichtenblocklängen. Der Einfluß der Bevorzugung durch ein größeres Fenster ist in dem Diagramm deutlich zu erkennen. Eine Vergrößerung des Sendefensters von VC-Nr.3 auf den Wert  $w_S=7$ , d.h. um den Faktor 3,5, ergibt im Vergleich zu den nicht bevorzugten Verbindungen für diese virtuelle Verbindung eine Durchsatzverbesserung ebenfalls um den Faktor 3,5 über den gesamten Wertebereich der Nachrichtenblocklängen.

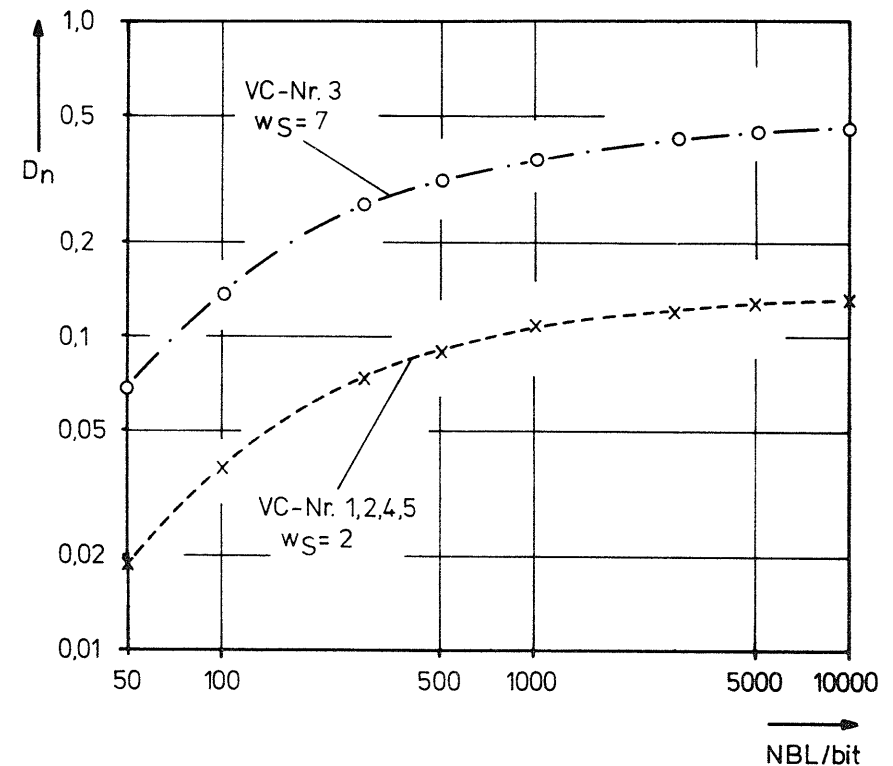


Bild 6.3: Normierter Durchsatz  $D_n$  als Funktion der Nachrichtenblocklänge NBL;  
 ( $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ )  
 Simulation: x - - - 4 VC mit  $w_S=2$   
 (VC-Nr. 1,2,4,5)  
 O - · - · 1 VC mit  $w_S=7$   
 (VC-Nr. 3)

Dies bedeutet aber auch, daß die Verbindung VC-Nr.3 denselben Durchsatz hat wie 3,5 "parallelgeschaltete" Verbindungen mit nur  $w_S=2$ . Die vorgegebene Systemkonfiguration hat somit dasselbe Durchsatzverhalten wie eine X.25-Schnittstelle mit insgesamt 7,5 virtuellen Verbindungen und je einem Sendefenster  $w_S=2$ .

Aus dieser Überlegung heraus ergeben sich für diese spezielle Systemkonfiguration folgende Grenzwert-Durchsätze, (vgl. Gleichung 6.1):

$$D_{n,gr} = \begin{cases} 1/7,5 & \text{für VC-Nr. 1,2,4,5} \\ 3,5/7,5 & \text{für VC-Nr. 3} \end{cases} \quad (6.2)$$

Unter Beibehaltung der gleichen Systemkonfiguration bei verschiedenen Laufzeiten erhält man das in Bild 6.4 gezeigte unterschiedliche Durchsatzverhalten.

Das durch Auswertung von Simulationsergebnissen gewonnene Bild 6.4 zeigt, analog zu Bild 6.3, sowohl den normierten Durchsatz  $D_n$  der bevorzugten virtuellen Verbindung Nr.3 als auch den arithmetischen Mittelwert der erzielten Durchsätze der anderen Verbindungen als Funktion der Laufzeit  $T_L$ , wobei  $T_L$  auf die Datenpaket-Übertragungszeit  $T_{UDP}$  normiert ist. Wie bei Bild 6.2 wurde auch hier die Nachrichtenblocklänge einheitlich bei  $NBL=1000\text{bit}$  belassen, um der Empfehlung in der X.25-Spezifikation gerecht zu werden.

Die Kurven zeigen ebenfalls dadurch das typische Durchsatzverhalten, daß mit größer werdenden Laufzeiten lange Quittierzeiten entstehen, die den Sender an den oberen Fensterrand stoßen und den Datenfluß unterbrechen lassen und somit der Durchsatz deutlich abnimmt. Auch hier bewirkt die Vergrößerung des Sendefensters von VC-Nr.3 auf den Wert  $w_S=7$ , im Vergleich zu den anderen, nicht bevorzugten Verbindungen, eine sichtbare Durchsatzverbesserung um denselben Faktor 3,5 über den gesamten Wertebereich der untersuchten Laufzeiten, da der Sender 3,5-fach mehr Datenpakete absenden kann, bevor er, bedingt durch den Fensteranschlag, den Sendevorgang unterbricht.

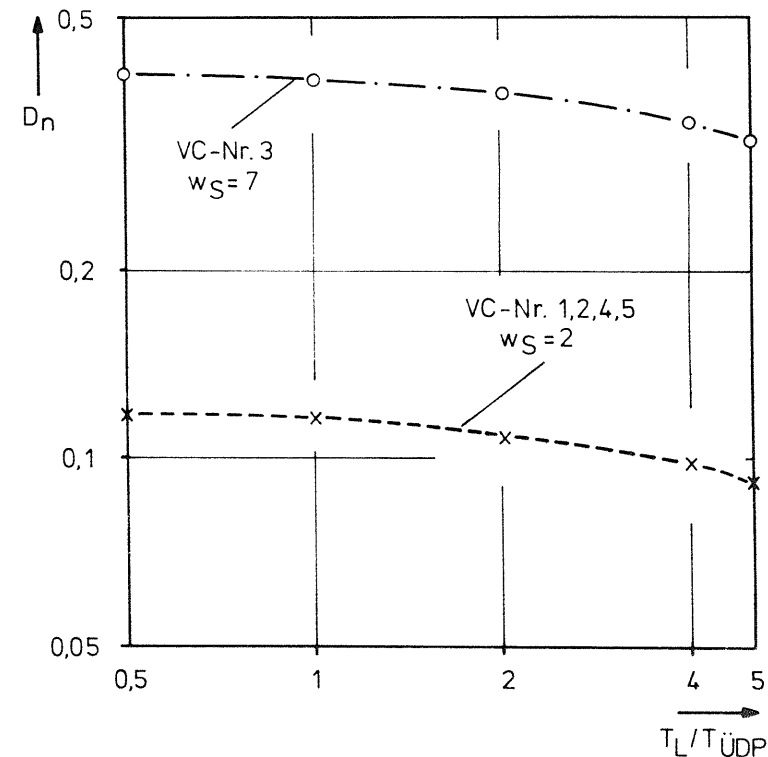


Bild 6.4: Normierter Durchsatz  $D_n$  als Funktion der Laufzeit  $T_L$ ; (NBL=1000bit,  $v=48000\text{bit/s}$ ,  $T_{UDP}=22,3\text{ms}$ )  
Simulation:  $\times$  - - - 4 VC mit  $w_S=2$  (VC-Nr. 1,2,4,5)  
 $\circ$  - - - 1 VC mit  $w_S=7$  (VC-Nr. 3)

Bemerkenswert an den Diagrammen ist, daß für die in der Empfehlung X.25 spezifizierten Standardparameterwerte:

- Standardpaketgröße 128 byte ( $\hat{=}$  NBL=1000bit)
- Standardfenstergröße  $w_S=2$

bereits gute bis sehr gute Durchsätze erzielt werden können.

Deutliche Durchsatzverbesserungen mittels Vergrößerung des Sendefensters lassen sich nur dann erzielen, wenn

- sich momentan wenige virtuelle Verbindungen ( $n \ll 5$ ) die Übertragungsstrecke teilen oder
- die Laufzeiten größer als die Datenpaketübertragungszeit sind ( $T_L/T_{ODP} \gg 1$ ) oder
- nur wenige unter vielen virtuellen Verbindungen durch diese gezielte Fenstervergrößerung bevorzugt werden.

Schließlich wird noch auf die gute Übereinstimmung zwischen analytisch und simulativ ermittelten Ergebnissen hingewiesen.

### 6.3 Transferzeit-Untersuchungen virtueller Verbindungen

Im Gegensatz zu den in Kap.6.2 gesättigten Verkehrssituationen werden nun *u n g e s ä t t i g t e* Fälle betrachtet, d.h. der physikalische Übertragungskanal ist geringer ausgelastet, und die Anzahl der zu übertragenden Nachrichtenblöcke der einzelnen virtuellen Verbindungen schwankt statistisch. Derartige Situationen sind charakteristisch für interaktiven Datenverkehr, für den die sog. Transferzeit eine wichtige Verkehrsgröße darstellt.

Die Transferzeit ist das Zeitintervall zwischen dem Ankunftszeitpunkt eines Nachrichtenblockes im VC-Sendespeicher der Sendeseite und dem Zeitpunkt der Entnahme dieses Blockes aus dem VC-Empfangs Speicher der Empfangsseite.

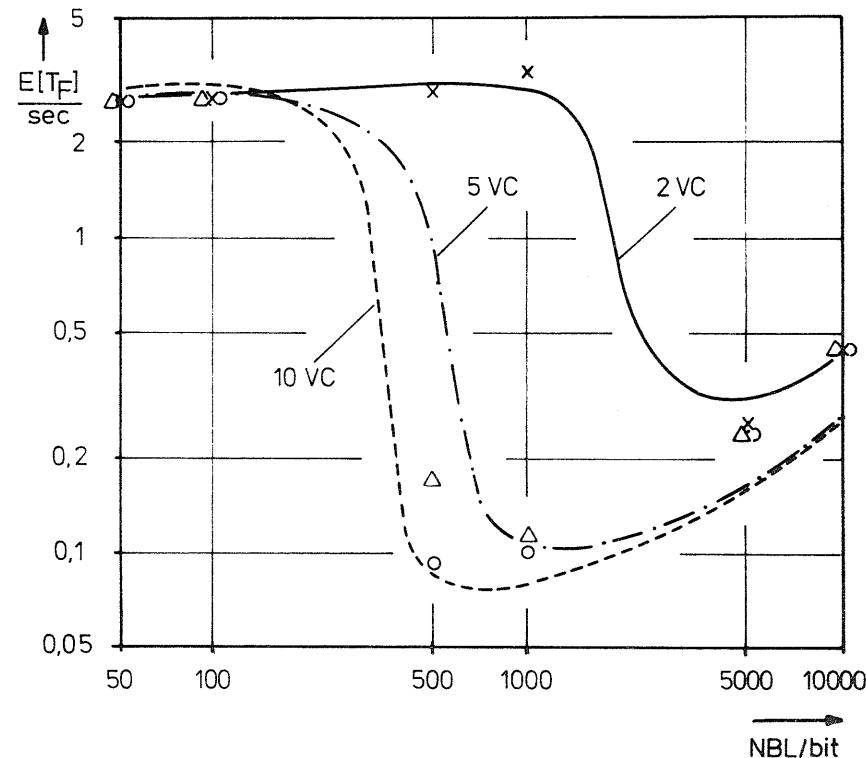


Bild 6.5: Mittlere Transferzeit  $E[T_F]$  als Funktion der Nachrichtenblocklänge NBL;  
 ( $A=0,6$ ,  $T_L=50ms$ ,  $v=48000bit/s$ ,  $w_S=2$ )

- Berechnung: — 2 VC  
 - - - 5 VC  
 - · - · 10 VC
- Simulation: x 2 VC  
 Δ 5 VC  
 ○ 10 VC

Bild 6.5 zeigt die mittlere Transferzeit  $E[T_F]$  für Datenpakete einer virtuellen Verbindung als Funktion der konstanten Nachrichtenblocklänge NBL und der Anzahl n virtueller Verbindungen bei einheitlicher Fenstergröße  $w_S=2$  und konstantem Gesamtangebot  $A=0,6$ .

Das Angebot A ist hierbei definiert als die gesamte, auf die Übertragungsgeschwindigkeit v bezogene Datenmenge aller virtuellen Verbindungen, die pro Zeiteinheit der Schnittstelle angeboten wird:

$$A = \frac{\lambda_{VC} \cdot n \cdot NBL}{v} \quad (6.3)$$

Für kurze Nachrichtenblocklängen ( $NBL < 300\text{bit}$ ) ist die Transferzeit extrem hoch und wird ausschließlich durch die Wartezeit im VC-Sendespeicher VCSSp und somit durch die Länge dieses Speichers bestimmt. Dieser Effekt wird deutlich, wenn Gleichung (6.3) in Betracht gezogen wird: aufgrund des festen und über den gesamten Wertebereich der Nachrichtenblocklängen konstanten Angebotes ergibt sich für den mittleren Nachrichtenblockankunftsabstand  $a_{VC}$  pro virtuelle Verbindung folgende Beziehung:

$$a_{VC} = \frac{1}{\lambda_{VC}} = \frac{n \cdot NBL}{A \cdot v} \quad (6.4)$$

Für kleine Blocklängen entstehen sehr kurze (konstante) Ankunftsabstände, die in Verbindung mit der kleinen Fenstergröße  $w_S=2$  dafür Sorge tragen, daß der VCSSp stets gefüllt ist und ein ankommender (und erfolgreich im VCSSp aufgenommener) Nachrichtenblock eine hohe Wartezeit hat, die der Länge des VCSSp direkt proportional ist. Die Fenstergröße  $w_S=2$  reicht nicht aus, diese in den kurzen Zeitabständen angebotene Datenmenge schnell genug abarbeiten zu können.

Mit zunehmender Blocklänge NBL wird entsprechend Gleichung (6.4) der mittlere Ankunftsabstand ebenfalls länger, so daß der Fensterkontrollmechanismus diese Datenmenge, trotz gleichbleibendem Gesamtangebot, besser abarbeiten kann. Im VC-Sendespeicher treten dadurch deutlich kürzere Warteschlangenlängen auf und die Transferzeit nimmt stark ab.

Für sehr lange Nachrichtenblöcke ( $NBL > 5000\text{bit}$ ) treten wiederum sehr hohe Kanalbelegungszeiten auf, die zwangsläufig die Wartezeit der Rahmen im LAPB-SSp erhöhen.

Die Kurven für die Transferzeiten haben somit einen Wendepunkt und weisen zwischen extrem kleinen bzw. großen Nachrichtenblocklängen deutliche Minima auf. Die Lage des Wendepunktes und des Minimums, d.h. der Bereich der stärksten Abnahme der Transferzeit, ist dabei abhängig von der Anzahl n aktiver virtueller Verbindungen und verschiebt sich mit wachsendem n nach links. Der Grund dafür ist einerseits darin zu sehen, daß aufgrund des länger werdenden Multiplexzyklus auch bei kleiner Fenstergröße die Wahrscheinlichkeit groß wird, bereits vor Erreichen des oberen Fensterrandes neue Quittungen zu empfangen und andererseits darin, daß der mittlere Ankunftsabstand entsprechend Gleichung (6.4) mit wachsendem n ebenfalls größer wird, was eine Abnahme der Warteschlangenlänge im VCSSp zur Folge hat, sowie in der Tatsache, daß die Summenfenstergröße mit wachsendem n ebenfalls ansteigt.

Bild 6.6 zeigt, analog Bild 6.5, die mittlere Transferzeit  $E[T_F]$  als Funktion der konstanten Nachrichtenblocklänge NBL und der Anzahl n virtueller Verbindungen bei einheitlichem, vergrößertem Fenster  $w_S=7$  und konstantem Gesamtangebot  $A=0,6$ .

Bild 6.6 zeigt einerseits dieselben Tendenzen wie das Transferzeitverhalten mit  $w_S=2$  (vgl. Bild 6.5) und andererseits die deutlichen Transferzeitverbesserungen im Bereich kleiner und mittlerer Nachrichtenblocklängen ( $50\text{bit} < NBL < 5000\text{bit}$ ), die durch Vergrößerung des Sendefensters erzielbar sind.

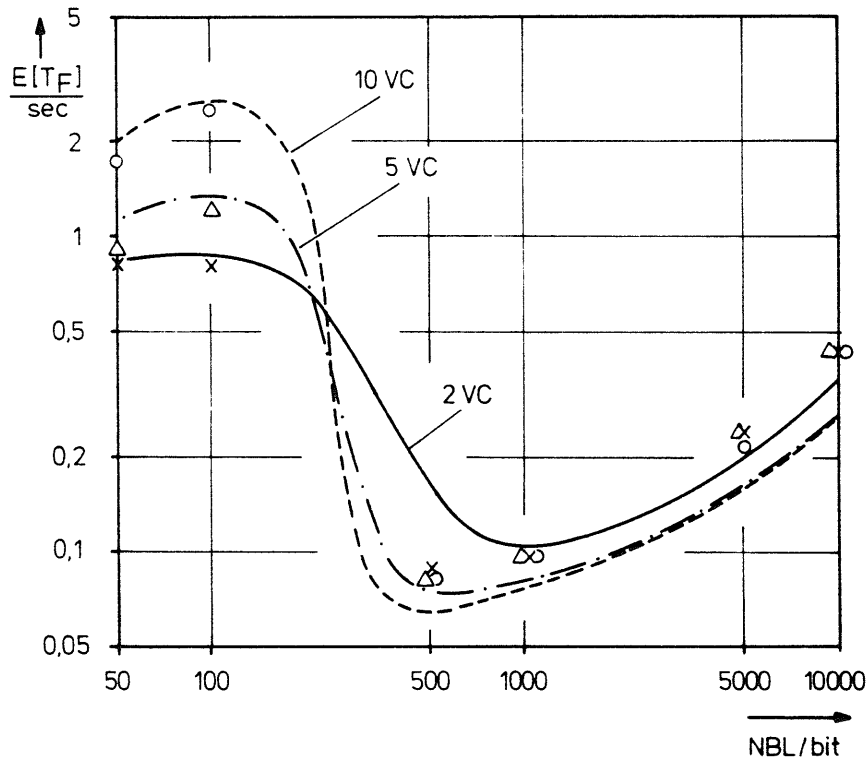


Bild 6.6: Mittlere Transferzeit  $E[T_F]$  als Funktion der Nachrichtenblocklänge NBL;  
 ( $A=0,6$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ ,  $w_S=7$ )

Berechnung: — 2 VC  
 - · - · - 5 VC  
 - - - 10 VC  
 Simulation: × 2 VC  
 Δ 5 VC  
 ○ 10 VC

Die Fenstergröße  $w_S=7$  reicht auch bei kleinen Nachrichtenblocklängen aus, die in den kurzen Zeitabständen angebotene Datenmenge in kürzeren Zeiten abzuarbeiten.

Mit zunehmender Nachrichtenblocklänge NBL nimmt auch der mittlere Ankunftsabstand zu (entsprechend Gleichung (6.4)), so daß der Sender relativ mehr Zeit hat, die angebotene Datenmenge abzuarbeiten. Die Warteschlangenlänge im VCSSp und somit die Transferzeit nehmen deutlich ab.

Im Vergleich zu Bild 6.5 verschoben sich die Lagen des Wendepunktes und des Minimums der Transferzeitkurven noch weiter nach links in Richtung kleinerer Blocklängen, da aufgrund der Fenstervergrößerung die Wahrscheinlichkeit zunimmt, bereits vor Erreichen des oberen Fensterrandes neue Quittungen zu empfangen, die ein Weiterdrehen des Sendefensters bewirken.

Der Effekt, daß für kleine Nachrichtenblocklängen ( $NBL < 300\text{bit}$ ) die Transferzeit mit der Anzahl  $n$  virtueller Verbindungen ansteigt, resultiert daraus, daß mit zunehmender Zahl an Verbindungen und großem Fenster  $w_S$  die Warteschlange im Wartespeicher LAPB-SSp von Schicht 2 anwächst und somit die Transferzeit insgesamt größer wird. Diese im LAPB-SSp entstehenden Wartezeiten zeigen sich in den Transferzeitkurven auch darin, daß die Transferzeit mit zunehmender Blocklänge (im Bereich kleiner Längen) aufgrund der nun längeren Kanalbelegungszeit die Transferzeit ansteigt.

Weitere Aufschlüsse über die Leistungsfähigkeit der Schnittstelle X.25 sollen die folgenden Diagramme geben, die die Ergebnisse von Transferzeituntersuchungen bei unterschiedlichen Laufzeiten aufzeigen.



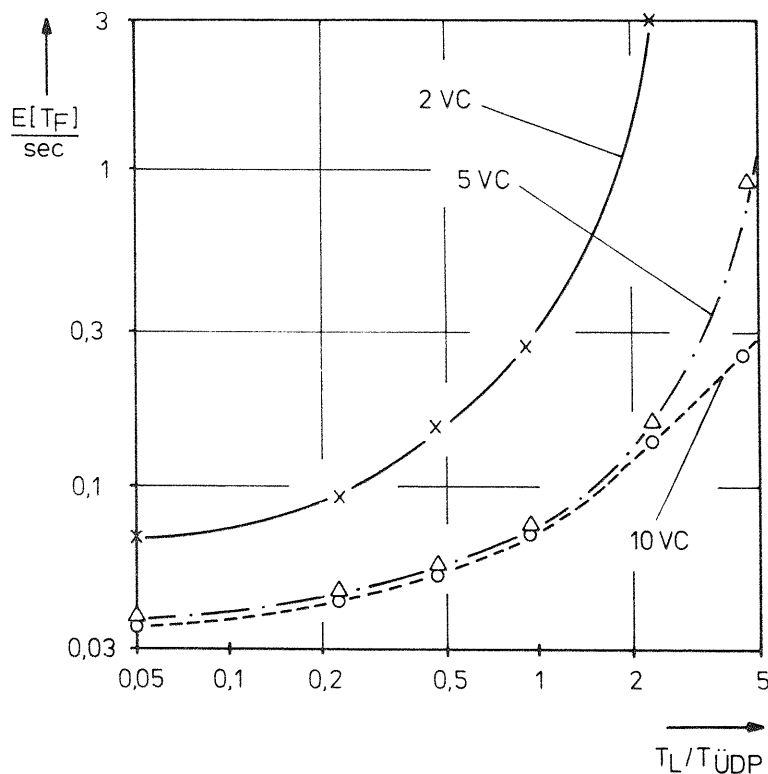


Bild 6.7: Mittlere Transferzeit  $E[T_F]$  als Funktion der Laufzeit  $T_L$ ; ( $A=0,6$ ,  $NBL=1000\text{bit}$ ,  $v=48000\text{bit/s}$ ,  $T_{UDP}=22,3\text{ms}$ ,  $w_S=2$ )

Berechnung: — 2 VC  
 - - - 5 VC  
 - - - 10 VC

Simulation: x 2 VC  
 Δ 5 VC  
 ○ 10 VC

Bild 6.7 veranschaulicht die mittlere Transferzeit  $E[T_F]$  für Datenpakete einer virtuellen Verbindung als Funktion der Laufzeit  $T_L$  (inklusive der Verarbeitungszeiten in Schicht 1 und 2) und der Anzahl  $n$  virtueller Verbindungen bei Fenstergröße  $w_S=2$  und konstantem Angebot  $A=0,6$ . Die Laufzeit  $T_L$  ist dabei auf die Übertragungszeit  $T_{UDP}$  eines Datenpaketes der Größe  $NBL=1000\text{bit}$  normiert.

Das Diagramm zeigt das typische Transferzeitverhalten für Datenverbindungen mit Fensterkontrollmechanismus. Für kleine Laufzeiten entstehen, selbst bei einer kleinen Anzahl virtueller Verbindungen, kurze Quittierzeiten, die zu einer ebenfalls kleinen Transferzeit führen.

Für größere Laufzeiten wird das Transferzeitverhalten deutlich schlechter insofern, als der Sender aufgrund der nun zwangsläufig längeren Quittierzeiten permanent am oberen Fensterrand anstößt und den Datenfluß unterbricht, wodurch zusätzlich zu der langen Laufzeit noch erhebliche Wartezeiten im VC-Sendespeicher auftreten.

Wie Bild 6.7 erkennen läßt, verbessert sich das Transferzeitverhalten der Schnittstelle beim Anschließen mehrerer Verbindungen wesentlich, da aufgrund des länger werdenden Multiplexzyklus und der grösseren Summen-Fenstergröße die Wahrscheinlichkeit zunimmt, Quittungen zu empfangen, bevor der obere Fensterrand erreicht ist. Es entsteht somit ein gleichmäßigerer Datenfluß, der zu geringeren Wartezeiten im VC-Sendespeicher und somit kleinerer Transferzeit führt als bei weniger virtuellen Verbindungen.

Bild 6.8 zeigt, wie das vorherige Bild 6.7, das Transferzeitverhalten  $E[T_F]$  einer virtuellen Verbindung als Funktion der Laufzeit  $T_L$  und der Anzahl  $n$  aktiver Verbindungen mit einheitlichem, vergrößertem Fenster  $w_S=7$  und ebenfalls konstantem Angebot  $A=0,6$ .

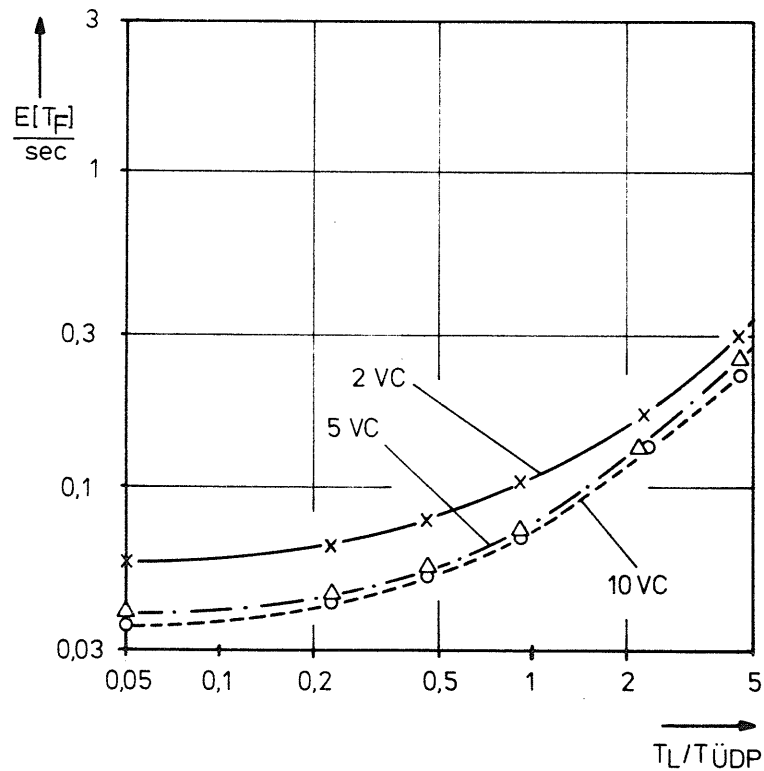


Bild 6.8: Mittlere Transferzeit  $E[T_F]$  als Funktion der Laufzeit  $T_L$ ;  
 ( $A=0,6$ ,  $NBL=1000\text{bit}$ ,  $v=48000\text{bit/s}$ ,  $T_{ÜDP}=22,3\text{ms}$ ,  $w_S=7$ )

Berechnung: ——— 2 VC  
 - - - - 5 VC  
 - - - - 10 VC

Simulation: x 2 VC  
 Δ 5 VC  
 ○ 10 VC

Das Diagramm veranschaulicht die erzielbaren Verbesserungen, die durch eine Vergrößerung des Sendefensters erreichbar sind. Diese Fenstervergrößerung bewirkt nun auch bei größeren Laufzeiten, und damit zwangsläufig längeren Quittierzeiten, einen stetigeren Datenfluß, der sich in kleineren Warteschlangen im VCSSp und somit besserer Transferzeit als bei  $w_S=2$  äußert.

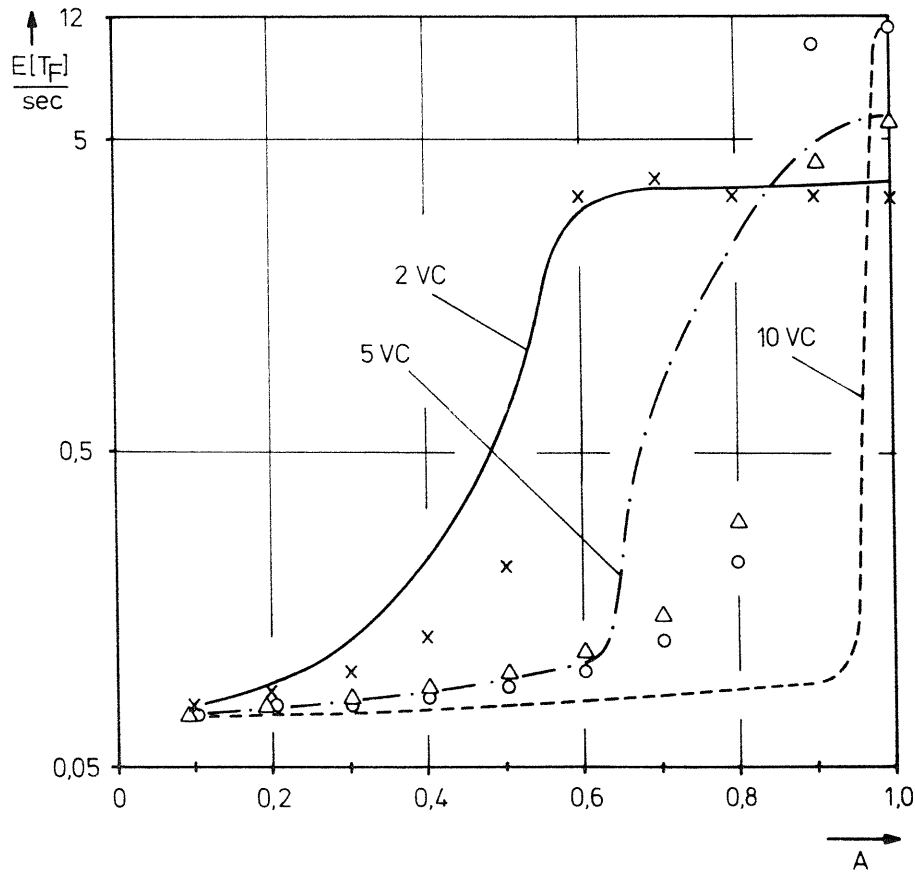
Der Effekt der Transferzeitverbesserung verliert sich zunehmend mit wachsender Anzahl virtueller Verbindungen, da bei vielen Verbindungen der damit zusammenhängende Multiplexzyklus bereits für eine Transferzeitminderung sorgt und eine Fenstervergrößerung somit nicht mehr die gewünschte Verbesserung erbringt.

Interessante Gesichtspunkte liefern Untersuchungen über das Transferzeitverhalten der Schnittstelle X.25 unter verschiedenen Lastsituationen, d.h. bei unterschiedlichen Angebotswerten.

Aus Bild 6.9 kann die Transferzeit  $E[T_F]$  einer virtuellen Verbindung als Funktion des Angebotes  $A$  und der Anzahl  $n$  an Verbindungen entnommen werden. Die Nachrichtenblocklänge  $NBL$  und die einheitliche Fenstergröße  $w_S$  wurden dabei konstant auf den Werten  $NBL=1000\text{bit}$  bzw.  $w_S=2$  belassen.

Es sei in diesem Zusammenhang nochmals darauf hingewiesen, daß unter dem Angebot  $A$  die gesamte, auf die Übertragungsgeschwindigkeit bezogene Datenmenge aller virtueller Verbindungen, verstanden wird, die pro Zeiteinheit der Schnittstelle angeboten wird (vgl. Gleichung (6.3)).

Wie das Bild erkennen läßt, ist die Transferzeit im Bereich kleiner Angebotswerte ( $A < 0,5$ ) sehr gering. Die Fenstergröße  $w_S=2$  reicht aus, diese relativ geringe Datenmenge schnell genug abzuarbeiten, um die Transferzeit in akzeptablen Grenzen zu halten.



Für 2 virtuelle Verbindungen steigt die Transferzeitkurve im Bereich mittleren Angebotes ( $0,5 < A < 0,7$ ) stark an, da hier der Fensterkontrollmechanismus mit  $w_S=2$  nicht mehr in der Lage ist, dieses Angebot an Datenmengen zügig genug abzufertigen. Es entstehen erhebliche Warteschlangenlängen in den VC-Sendespeichern, die zu hohen Warte- und damit auch zu hohen Transferzeiten führen. Eine weitere Steigerung der Transferzeit für noch höhere Angebotswerte ( $A > 0,7$ ) ist nicht zu beobachten, da in diesem Bereich die Transferzeit ausschließlich durch die Wartezeit im VCSSp und damit der Größe dieses Speichers bestimmt wird.

Für steigende Anzahlen virtueller Verbindungen zeigt das Diagramm deutliche Transferzeitverbesserungen im Bereich mittlerer Angebote; dies rührt daher, daß sich mit wachsender Anzahl  $n$  an Verbindungen der steile Anstieg in der Transferzeit nach rechts in Richtung größerer Angebote verschiebt.

Der Multiplexvorgang kommt bei mehreren Verbindungen, wie oben bereits erwähnt, dem Fenstermechanismus insofern entgegen, als mit wachsendem  $n$  zwangsläufig der Zyklus länger wird. Damit wird die Wahrscheinlichkeit ebenfalls größer, Quittungen zu empfangen, bevor der obere Fensterrand erreicht ist. Es entsteht ein stetigerer Datenfluß, der zu geringeren Warteschlangenlängen im VCSSp führt und der somit die Transferzeit über einen größeren Bereich an Angebotswerten in akzeptablen Grenzen hält.

Die Grenzwerte für die Transferzeiten für sehr hohe Angebotswerte ( $A=1,0$ ) steigen mit zunehmender Anzahl  $n$  an virtuellen Verbindungen an, da hierdurch mehr Rahmen im Wartespeicher LAPB-SSP der Schicht 2 auf ihre Übertragung warten und somit zusätzliche Wartezeiten entstehen, die die Transferzeit erhöhen. Die Begrenzung der Transferzeiten erfolgt bei  $n=2$  Verbindungen durch den begrenzten Speicherplatz im VCSSp, bei  $n=5$  bzw.  $n=10$  durch die begrenzte Kanalkapazität.

Bild 6.9: Mittlere Transferzeit  $E[T_F]$  als Funktion des Angebotes  $A$ ;  
 ( $NBL=1000\text{bit}$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ ,  $w_S=2$ )

- Berechnung: — 2 VC  
 - · - · 5 VC  
 - - - 10 VC
- Simulation: x 2 VC  
 Δ 5 VC  
 ○ 10 VC

Wie dem Bild 6.9 ferner zu entnehmen ist, differieren die Bereiche des Transferzeitanstiegs zwischen Berechnung und Simulation, da die Berechnung nur Mittelwerte und nicht durch Protokoll und Multiplexmechanismen bedingte statistische Schwankungen in Betracht ziehen kann. Weitergehende gegenseitige Einflüsse der Verbindungen aufeinander sowie die o.a. statistischen Schwankungen können nur in der Simulation berücksichtigt werden, wodurch die Differenzen in den Ergebnissen entstehen können.

Aus Bild 6.10, das ebenfalls wie Bild 6.9 das Transferzeitverhalten  $E[T_F]$  einer virtuellen Verbindung als Funktion des Angebotes  $A$  zeigt, können die Verbesserungen entnommen werden, die sich bei einer Vergrößerung des Sendefensters auf den Wert  $w_S=7$  erzielen lassen.

Diese Verbesserungen treten deutlich bei  $n=2$  virtuellen Verbindungen im Bereich mittleren Angebotes ( $0,5 < A < 0,7$ ) in Erscheinung, da nun der Fenstermechanismus auch bei nur 2 Verbindungen in der Lage ist, die angebotene Datenmenge zügiger abzuarbeiten. Der Verbesserungseffekt im Transferzeitverhalten verliert sich zunehmend mit steigender Anzahl  $n$  an Verbindungen, da auch hier der dadurch länger werdende Multiplexzyklus bereits für eine Transferzeitminderung sorgt.

Für die anzahlindividuellen Grenzwerte der Transferzeiten für sehr hohe Angebote ( $A=1,0$ ) gelten dieselben Überlegungen wie unter Bild 6.9 beschrieben.

Ebenso differieren auch bei diesem Diagramm die Bereiche maximalen Transferzeitanstiegs zwischen Berechnung und Simulation aus o.g. Gründen.

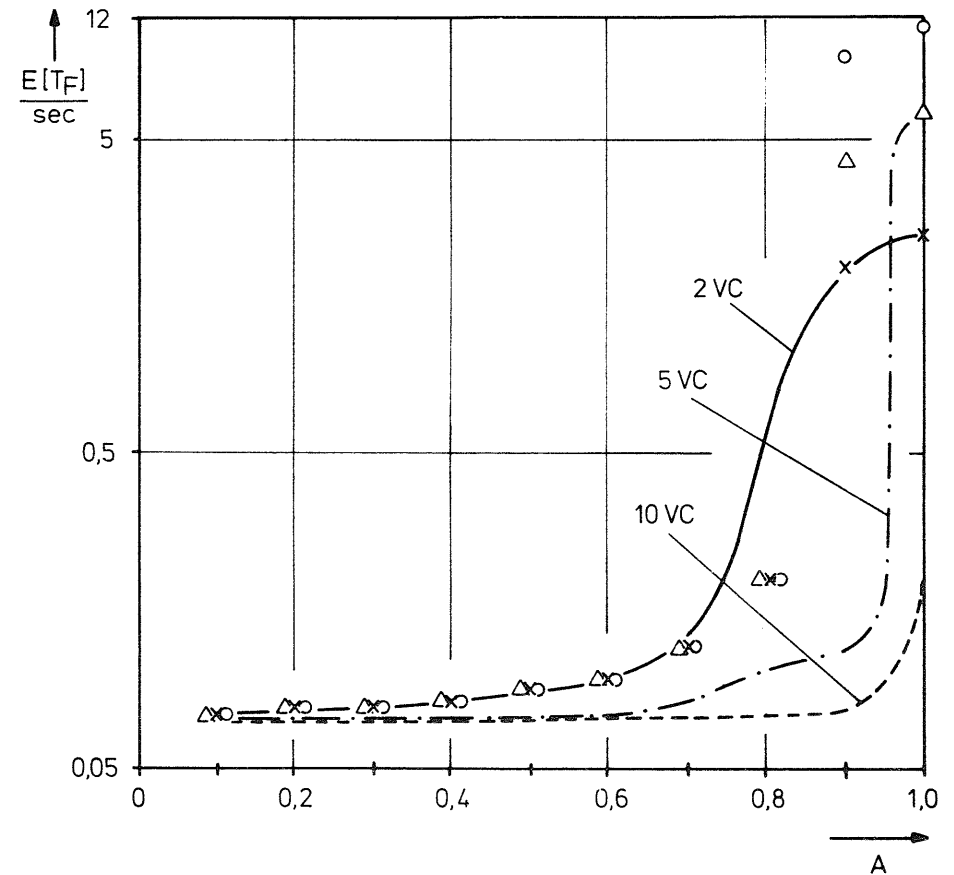


Bild 6.10: Mittlere Transferzeit  $E[T_F]$  als Funktion des Angebotes  $A$ ;  
 ( $NBL=1000\text{bit}$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ ,  $w_S=7$ )

- Berechnung: — 2 VC  
 - · - · - 5 VC  
 - - - 10 VC
- Simulation: x 2 VC  
 Δ 5 VC  
 ○ 10 VC

Auch im Falle der Transferzeituntersuchungen lag es nahe, das Verhalten der Schnittstelle X.25 als Funktion unterschiedlicher, optionaler Leistungsmerkmale zu analysieren, wobei der Schwerpunkt im besonderen auf den Einfluß unterschiedlicher Fenstergrößen bei mehreren virtuellen Verbindungen gelegt wurde. Hierzu wurde dieselbe Systemkonfiguration wie in Kap.6.2 beschrieben gewählt:  
 n=5 virtuelle Verbindungen, VC-Nr. 1,2,4,5 mit je  $w_S=2$  und VC-Nr.3 mit bevorzugtem  $w_S=7$ .

Bild 6.11 zeigt, als Funktion der konstanten Nachrichtenblocklänge NBL, die mittlere Transferzeit  $E[T_F]$  der bevorzugten Verbindung (VC-Nr.3;  $w_S=7$ ) sowie den Mittelwert der Zeiten der anderen Verbindungen (VC-Nr.1,2,4,5;  $w_S=2$ ).

Für kleine Nachrichtenblocklängen ( $NBL \ll 50\text{bit}$ ) ist bei gegebenem konstantem Angebot  $A=0,6$  die Anzahl der zu übertragenden Pakete am größten (vgl. Gleichung (6.4)). In diesem Bereich befindet sich der Sender auch für  $w_S=7$  am oberen Fensterrand, wodurch die Warteschlangen und die Wartezeiten im VCSSp anwachsen. Diese hohen Wartezeiten bestimmen im wesentlichen den Wert der Transferzeit.

Wie weiteren Teilergebnissen der Simulation zu entnehmen ist, tritt bei der bevorzugten Verbindung (VC-Nr.3;  $w_S=7$ ) ab einer Nachrichtenblocklänge  $NBL > 90\text{bit}$  kein Fensteranschlag am oberen Fensterrand mehr auf. Dieser Effekt zeigt sich in der deutlichen Abnahme der Transferzeit. Für die nicht bevorzugten Verbindungen (VC-Nr.1,2,4,5;  $w_S=2$ ) treten erst ab  $NBL > 360\text{bit}$  Verbesserungen im Transferzeitverhalten auf, da ab dieser Blocklänge der Fenstermechanismus die angebotene Datenmenge ohne Anstoßen am oberen Fensterrand abarbeiten kann.

Für größere Nachrichtenblocklängen bestimmen die hierzu proportionalen Übertragungszeiten  $T_{ODP}$  und die damit zusammenhängenden Wartezeiten im LAPB-SSp ausschließlich die Transferzeit.

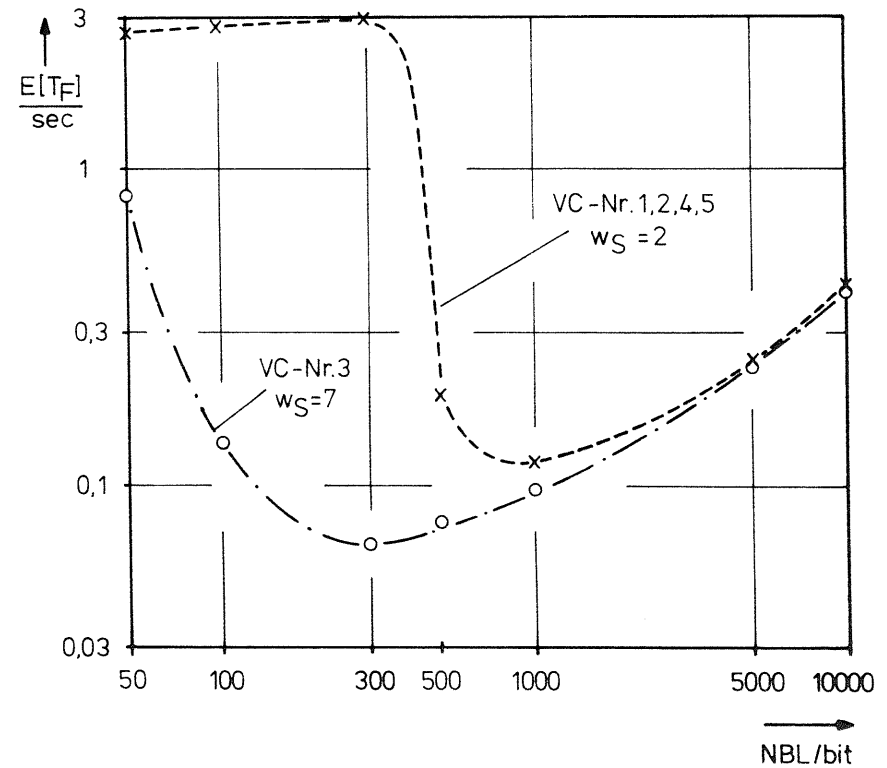


Bild 6.11: Mittlere Transferzeit  $E[T_F]$  als Funktion der Nachrichtenblocklänge NBL;  
 ( $A=0,6$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ )  
 Simulation: x - - - 4 VC mit  $w_S=2$   
 (VC-Nr. 1,2,4,5)  
 o - · - · 1 VC mit  $w_S=7$   
 (VC-Nr. 3)

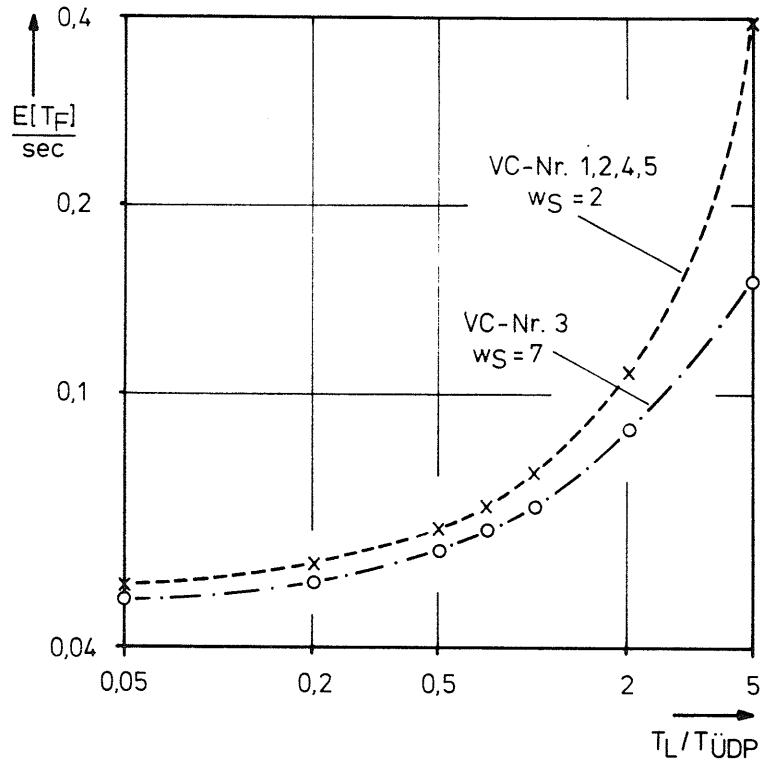


Bild 6.12: Mittlere Transferzeit  $E[T_F]$  als Funktion der Laufzeit  $T_L$ ;  
 ( $A=0,6$ ,  $NBL=1000\text{bit}$ ,  $v=48000\text{bit/s}$ ,  $T_{UDP}=22,3\text{ms}$ )  
 Simulation: x - - - 4 VC mit  $w_S=2$   
 (VC-Nr. 1,2,4,5)  
 o - · - · 1 VC mit  $w_S=7$   
 (VC-Nr. 3)

Da diese Zeiten nur noch vom Kanal bzw. von der Nachrichtenblocklänge abhängig sind, müssen sich die beiden Transferzeitkurven für große Blocklängen ( $NBL > 5000\text{bit}$ ) annähern.

Bild 6.12 zeigt das Transferzeitverhalten  $E[T_F]$  einerseits der bevorzugten Verbindung Nr.3 mit  $w_S=7$  und andererseits der restlichen Verbindungen Nr.1,2,4,5 mit  $w_S=2$  als Funktion der Laufzeit  $T_L$ . Die Laufzeit ist normiert auf die Datenpaketübertragungszeit  $T_{UDP}$ , wobei die Nachrichtenblocklänge über den gesamten Wertebereich von  $T_L$  konstant bei  $NBL=1000\text{bit}$  und das Gesamtangebot ebenfalls konstant bei  $A=0,6$  belassen wurde.

Analog Bild 6.7 zeigt auch dieses Diagramm das typische Transferzeitverhalten, indem mit zunehmender Laufzeit  $T_L$  die Transferzeit überproportional anwächst. Aufgrund der langen Lauf- und damit zusammenhängend auch Quittierzeiten stoßen die Sender permanent am oberen, den Datenfluß unterbrechenden Fensterrand an. Es entstehen somit zusätzlich zu den langen Laufzeiten erhebliche Wartezeiten im VCSSp.

Wie aus Bild 6.12 weiterhin entnommen werden kann, zeigt die Kurve der bevorzugten Verbindung ein deutlich verbessertes Transferzeitverhalten, da hier das vergrößerte Fenster mit  $w_S=7$  für geringere Wartezeiten und damit auch kleinere Transferzeiten sorgt als mit  $w_S=2$ .

Für die folgenden Untersuchungen wurden die wichtigsten Kombinationen der Verteilungsfunktionen für den zufallsabhängigen Ankunftsabstand  $T_A$  und für die Nachrichtenblocklänge  $NBL$  simuliert. Dabei standen die beiden gebräuchlichsten Verteilungsfunktionen - die Exponential-Verteilung (negex.) und die konstante Verteilung (konst.) - im Vordergrund der Untersuchungen.

Der Grund ist darin zu sehen, daß die negativ-exponentielle Verteilung für den Ankunftsabstand dem allgemeinen (menschlichen) Benutzerverhalten und die konstante Verteilung dem (getakteten) Rechnerverhalten am ehesten entsprechen. Die Exponentialverteilung für die Länge der Nachrichtenblöcke entspricht den üblicherweise im interaktiven Verkehr angebotenen Datenmengen, während die konstante Verteilung für die Blocklängen wiederum typisches Rechnerverhalten am besten widerspiegelt.

Die Bilder 6.13, 6.14 und 6.15 zeigen das Transferzeitverhalten  $E[T_F]$  der Schnittstelle X.25 als Funktion des Angebotes  $A$  sowie der möglichen Kombinationen für die Verteilungen des Ankunftsabstands  $T_A$  und der Nachrichtenblocklänge NBL. Da für die Verteilungen des Ankunftsabstandes und der Nachrichtenblocklänge je die negativ-exponentielle und die konstante Verteilung untersucht wurden, bestehen insgesamt vier Kombinationsmöglichkeiten.

Bild 6.13 berücksichtigt hierbei insgesamt  $n=2$  virtuelle Verbindungen mit je  $w_S=2$ , Bild 6.14 liegen  $n=5$  Verbindungen mit je  $w_S=2$  zugrunde und Bild 6.15 gibt das Verhalten mit  $n=10$  Verbindungen und  $w_S=2$  wieder.

Alle Kurven der Bilder 6.13 bis 6.15 zeigen denselben typischen Verlauf wie bereits in Bild 6.9 gezeigt und beschrieben: geringe Transferzeiten für kleinere Angebotswerte und hohe Transferzeiten für hohe Angebote. Der zwischen diesen Extremen liegende steile Anstieg verschiebt sich dabei mit wachsender Anzahl  $n$  an Verbindungen zunehmend nach rechts aufgrund des länger werdenden und die Transferzeit positiv beeinflussenden Multiplexzyklus.

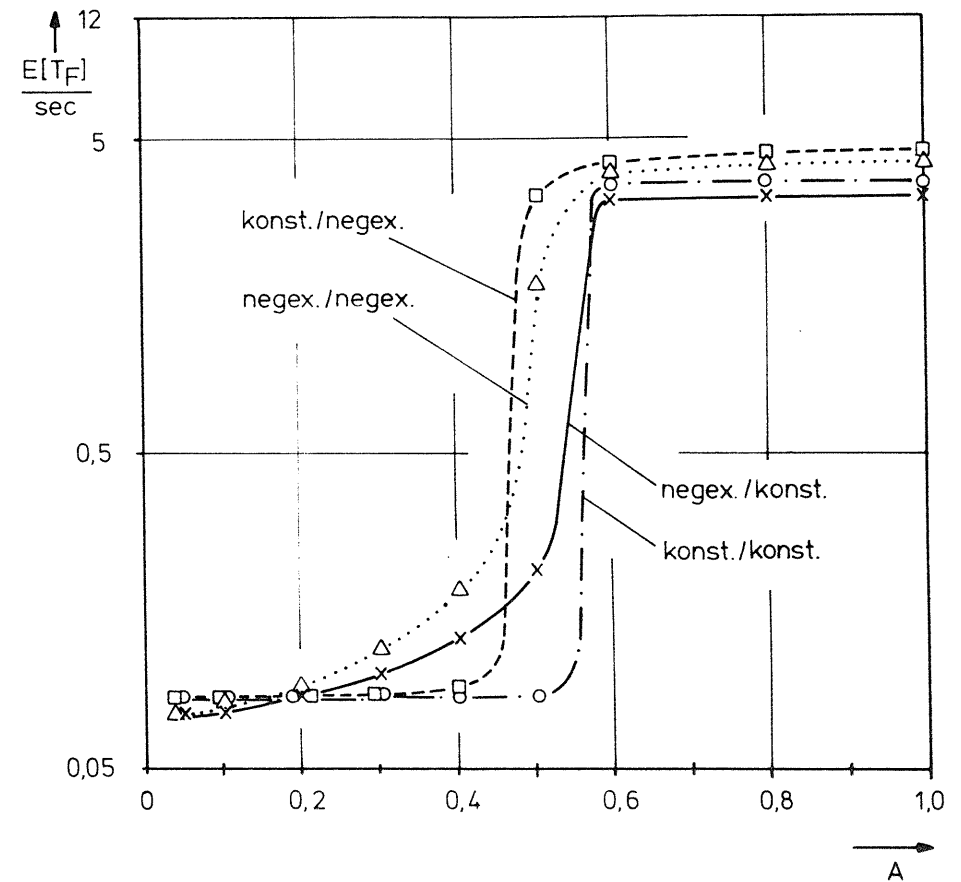


Bild 6.13: Mittlere Transferzeit  $E[T_F]$  als Funktion des Angebots  $A$ ;  
 ( $E[NBL]=1000\text{bit}$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ , 2 VC,  $w_S=2$ )

Simulation	Vtlg. $T_A$	Vtlg. NBL
x ———	negex.	konst.
△ ·····	negex.	negex.
□ - - -	konst.	negex.
○ - · - ·	konst.	konst.

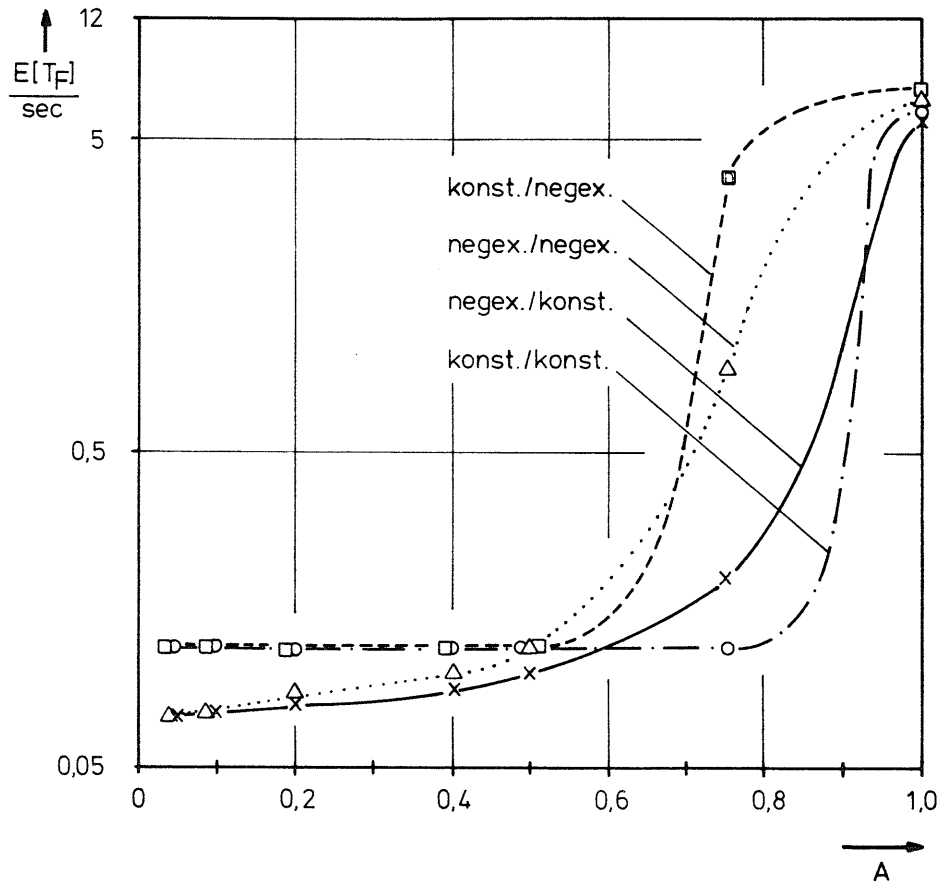


Bild 6.14: Mittlere Transferzeit  $E[T_F]$  als Funktion des Angebots  $A$ ;  
 ( $E[NBL]=1000\text{bit}$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ , 5 VC,  $w_S=2$ )

Simulation	Vtlg. $T_A$	Vtlg. NBL
x —	negex.	konst.
Δ ·····	negex.	negex.
□ - - -	konst.	negex.
○ - · - ·	konst.	konst.

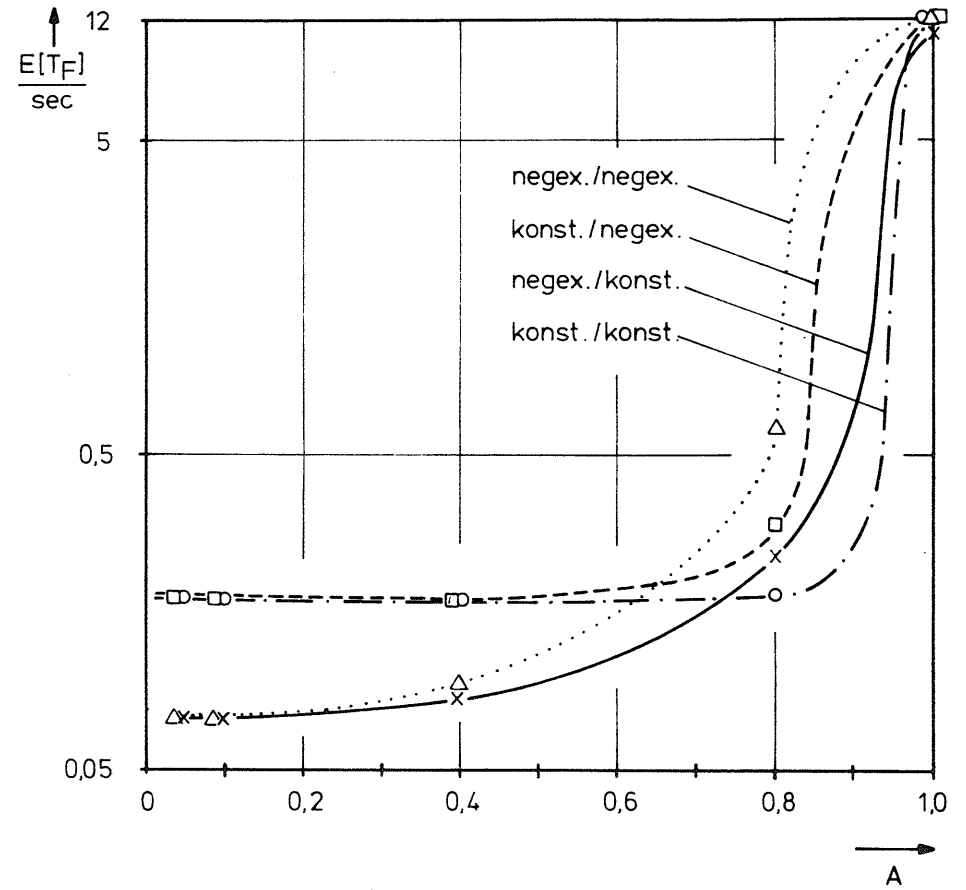


Bild 6.15: Mittlere Transferzeit  $E[T_F]$  als Funktion des Angebots  $A$ ;  
 ( $E[NBL]=1000\text{bit}$ ,  $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ , 10 VC,  $w_S=2$ )

Simulation	Vtlg. $T_A$	Vtlg. NBL
x —	negex.	konst.
Δ ·····	negex.	negex.
□ - - -	konst.	negex.
○ - · - ·	konst.	konst.



Bezüglich der unterschiedlichen Kombinationen von Ankunftsabstands- und Blocklängenverteilung treten im Bereich kleiner Angebotswerte ( $A < 0,2$ ) deutliche Unterschiede in den Transferzeiten auf. Diese Zeiten sind dann am höchsten, wenn der Ankunftsabstand  $T_A$  konstant verteilt ist. Die Transferzeitunterschiede nehmen mit steigender Anzahl  $n$  an Verbindungen zu, da zusätzliche Wartezeiten im LAPB-SSp von Schicht 2 auftreten.

Dieser Sachverhalt ist auf die gegebene Realisierung des verwendeten Simulationsprogrammes zurückzuführen. In diesem Programm wird zu Beginn der Simulation in einem Initialisierungsteil in jeder der insgesamt  $n$  VC-Sendespeicher ein Nachrichtenblock eingetragen. Aufgrund des zyklischen Multiplexvorganges und des konstanten Ankunftsabstandes entstehen somit Gruppenankünfte (mit ebenfalls konstantem Abstand) im LAPB-SSp der Schicht 2. Diese deterministischen Gruppenankünfte bewirken eine höhere Wartezeit im LAPB-SSp als bei negativ-exponentiell verteiltem Ankunftsabstand. Mit wachsender Anzahl  $n$  an virtuellen Verbindungen nimmt die Gruppengröße und somit auch die Wartezeit im LAPB-SSp zu, vgl. hierzu die Bilder 6.13 bis 6.15 mit je  $A < 0,2$ .

Für zunehmendes Angebot steigen die Transferzeiten aufgrund erheblicher Wartezeiten im VCSSp bzw. im LAPB-SSp stark an. Wie bereits bei Bild 6.9 zu erkennen war, steigen auch hier die Grenzwerte der Transferzeiten mit zunehmender Anzahl  $n$ , da hierdurch mehr Rahmen im LAPB-SSp auf ihre Übertragung warten und somit zusätzliche Wartezeiten entstehen. Die Begrenzung der Transferzeiten erfolgt, wie bei Bild 6.9 erwähnt, für  $n=2$  durch den Speicherplatz im VCSSp, für  $n=5$  bzw.  $n=10$  durch den Kanal.

Bei hohem Angebot nehmen die Einflüsse der Verteilungen ab, wobei zu erwähnen ist, daß die Differenzen der Absolutwerte relativ gering sind. Mit zunehmender Anzahl der Verbindungen verringern sich die relativen Differenzen in den Transferzeiten von 20% bei  $n=2$  auf 7% bei  $n=10$  VC.

Die stärksten Unterschiede im Transferzeitverhalten treten im Bereich des maximalen Anstiegs auf, da hier die statistischen Schwankungen und die gegenseitigen Einflüsse am stärksten zum Tragen kommen.

In diesem Zusammenhang sei nochmals darauf hingewiesen, daß aufgrund der in Kap.5.3 gemachten Aussagen die Kurven der Bilder 6.11 bis 6.15 ausschließlich durch Auswertung von Simulationsergebnissen gewonnen wurden.

Aufgrund der zum Teil relativ geringen Unterschiede im Transferzeitverhalten bei unterschiedlichen Kombinationen von Ankunftsabstands- und Blocklängenverteilung, dürfte sich das in Kap.5.2.2 vorgestellte Berechnungsverfahren auch dann zur Ermittlung der Tendenzen, bzw. Minima und Maxima im Transferzeitverhalten eignen, wenn andere Kombinationen, als nur exponentialverteilte Ankunftsabstände und konstant verteilte Nachrichtenblocklängen vorliegen.

Aus den vorgestellten Transferzeituntersuchungen ist unschwer zu erkennen, daß für die in der CCITT-Empfehlung X.25 angegebenen Standardparameterwerte:

- Paketgröße 128 byte
- Fenstergröße  $w_S=2$

unter fast allen Bedingungen sehr gute Transferzeiten auf dem Netzzugangsabschnitt erzielt werden können, solange sich mehrere virtuelle Verbindungen ( $n \geq 5$ ) den Übertragungsabschnitt teilen und die angebotene Datenmenge nicht zu hoch wird ( $A < 0,8$ ).

Werden insgesamt weniger virtuelle Verbindungen auf dem Netzzugangsabschnitt betrieben, empfiehlt es sich, von der optionalen Fenstergröße  $w_S=7$  Gebrauch zu machen, um ebenfalls in den Genuß guter bis sehr guter Transferzeiten zu kommen.

#### 6.4 Einstellung von Parametern

In Kap.3.3.2 wurden einige optionale Benutzer- und Leistungsmerkmale vorgestellt, die die Leistungsfähigkeit von X.25 erheblich beeinflussen können.

Die Untersuchungen der vorherigen Kapitel 6.2 und 6.3 dienten u.a. dazu, das Leistungsverhalten von X.25 bezüglich der beiden wichtigsten optionalen Merkmale aufzuzeigen:

- Wahl der Fenstergröße  $w_S$
- Wahl der Paketlänge.

Dieses Kapitel soll einerseits Hilfestellungen geben, diese beiden Leistungsmerkmale optimal einstellen zu können und andererseits kurz auf ein weiteres, ebenfalls optionales Leistungsmerkmal eingehen:

- Wahl der Durchsatzklasse.

Die für die beiden Leistungsmerkmale Paketlänge (genauer gesagt handelt es sich hierbei um das Benutzerdatenfeld innerhalb eines Datenpaketes) und Fenstergröße  $w_S$  in X.25 empfohlenen Standardwerte 128byte und  $w_S=2$  liefern insgesamt gesehen beste Ergebnisse bezüglich Durchsatz und Transferzeit, wenn folgende Voraussetzungen (zusätzlich) erfüllt sind:

- mehrere virtuelle Verbindungen ( $n \geq 5$ ) teilen sich die Übertragungskapazität
- Laufzeit  $T_L \ll$  Übertragungszeit eines Datenpaketes  $T_{UDP}$
- angebotene Datenmenge pro Zeiteinheit  $<$  0,8-fache der Übertragungsgeschwindigkeit  $v$ .

Sollten unter den eben genannten Bedingungen weniger als 5 Verbindungen auf dem Übertragungsabschnitt betrieben werden ( $n < 5$ ), empfiehlt es sich, den Wert der Fenstergröße auf  $w_S=7$  festzusetzen.

Ebenso ist eine Fenstervergrößerung auf  $w_S=7$  angebracht, wenn die Laufzeit  $T_L$  mehr als das Doppelte der Datenpaketübertragungszeit  $T_{UDP}$  betragen sollte.

Für die Wahl anderer Paketlängen als die empfohlene Standardgröße zeichnet sich der Bereich

- $300\text{bit} < \text{NBL} < 3000\text{bit}$

ab, in welchem ebenfalls eine gute Leistungsfähigkeit von X.25 gewährleistet werden kann.

Ein weiteres optionales Leistungsmerkmal, welches in der Empfehlung X.25 vorgesehen ist, betrifft die Wahl der Durchsatzklasse pro virtueller Verbindung. Die Durchsatzklasse gibt dabei an, welche Nutzinformation im Mittel der Zeit über diese eine bestimmte Verbindung gesendet werden kann. Ihre Dimension ist bit/s. Sie ist abhängig von:

- der Übertragungsgeschwindigkeit  $v$  des physikalischen Übertragungsmediums
- der Fenstergröße  $w_S$
- der Paketlänge
- der Anzahl  $n$  aktiver virtueller Verbindungen
- den Netzressourcen u.a.m.

Die Durchsatzklasse ist somit ein Richtmaß für den Durchsatz. Aufgrund des statistischen Aufteilens der Übertragungskapazität und weiterer Ressourcen kann es ohne weiteres sein, daß die Durchsatzklasse (kurzfristig) überschritten wird. Es kann jedoch auch nicht dafür garantiert werden, daß die gewählte Durchsatzklasse in 100% der Zeit erreicht wird.

Die folgenden Bilder 6.16 und 6.17 sind identisch mit den Diagrammen bezüglich des Durchsatzverhaltens (vgl. Bilder 6.1 und 6.2), mit dem Unterschied, daß jetzt die Ordinate die entsprechenden, in X.25 vorgesehenen Durchsatzklassen angibt.

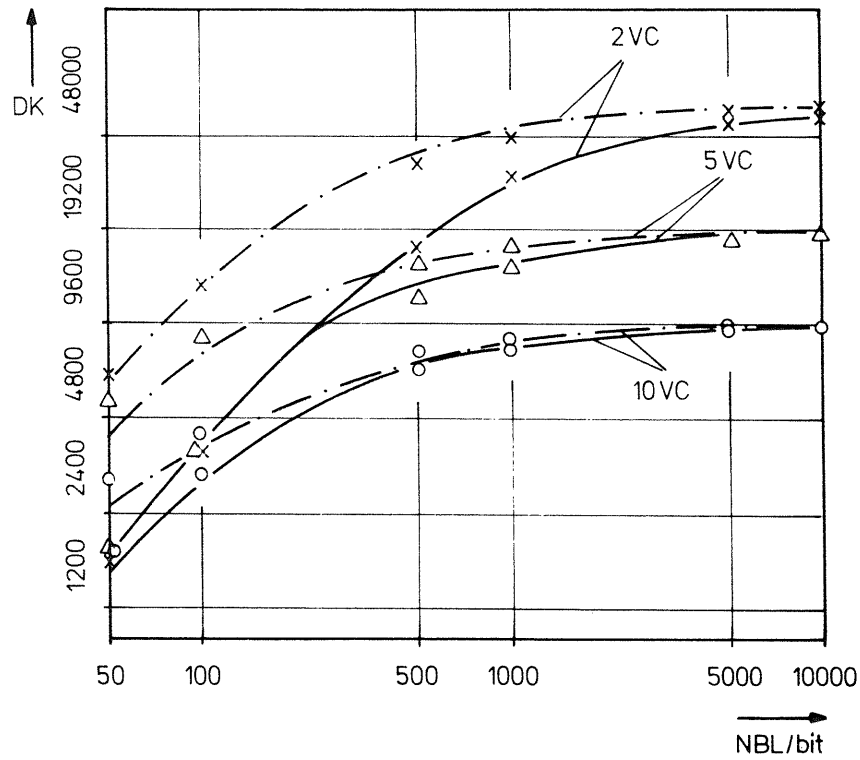


Bild 6.16: Durchsatzklasse DK als Funktion der Nachrichtenblocklänge NBL; ( $T_L=50\text{ms}$ ,  $v=48000\text{bit/s}$ )  
 Berechnung: —  $w_s=2$   
               - - -  $w_s=7$   
 Simulation: x 2 VC  
               △ 5 VC  
               ○ 10 VC

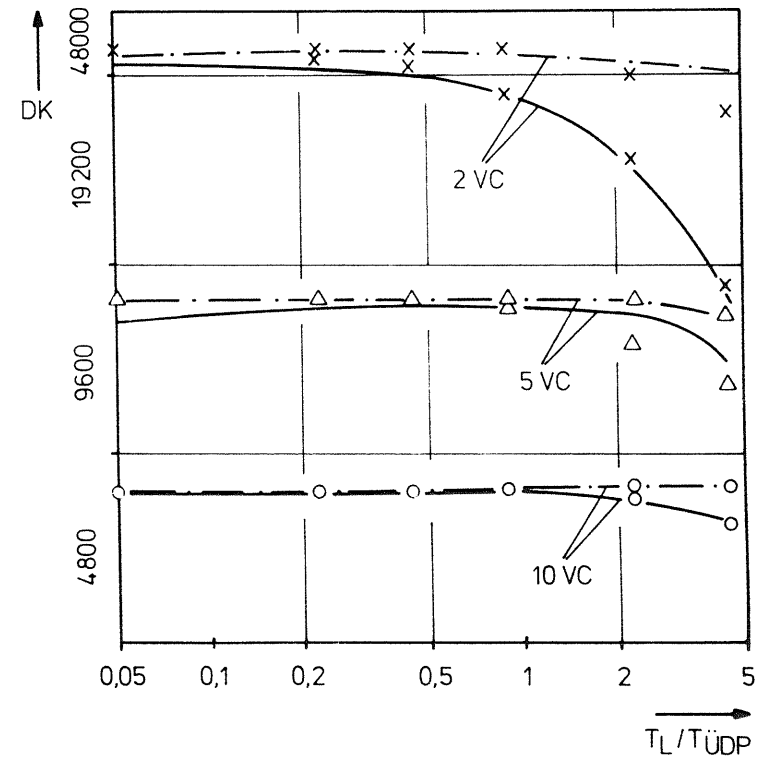


Bild 6.17: Durchsatzklasse DK als Funktion der Laufzeit  $T_L$ ; ( $NBL=1000\text{bit}$ ,  $v=48000\text{bit/s}$ ,  $T_{ÜDP}=22,3\text{ms}$ )  
 Berechnung: —  $w_s=2$   
               - - -  $w_s=7$   
 Simulation: x 2 VC  
               △ 5 VC  
               ○ 10 VC

Da die Durchsatzklasse, wie oben erwähnt, den maximal gewünschten Durchsatz angibt, wird ihr Bereich nach oben hin durch den Wert der entsprechenden Durchsatzklasse bestimmt. Die untere Schranke ihres Bereiches bildet der Wert der nächst niederen Durchsatzklasse.

Die Bilder stellen somit ein geeignetes Hilfsmittel dar, die Durchsatzklasse bei gegebenen Voraussetzungen (Parameterkonstellationen) zu ermitteln.

Den richtigen Gebrauch der Bilder 6.16 und 6.17 sollen die folgenden Beispiele veranschaulichen.

Bild 6.16 zeigt die Durchsatzklasse DK als Funktion der konstanten Nachrichtenblocklänge NBL. Teilen sich  $n=2$  virtuelle Verbindungen mit je  $w_S=2$  den Übertragungsabschnitt mit einer festen Nachrichtenblocklänge  $NBL=1000\text{bit}$ , so beträgt die höchste Durchsatzklasse DK, die entsprechend Bild 6.16 für die beiden Verbindungen gewählt werden kann,  $DK=19200\text{bit/s}$ . Bei einer Blocklänge von nur  $NBL=500\text{bit}$ , kann höchstens  $DK=9600\text{bit/s}$  gewählt werden. Sind für  $n=2$  Verbindungen höhere Durchsatzklassen gewünscht, z.B.  $DK=48000\text{bit/s}$ , so läßt Bild 6.16 erkennen, daß bei  $NBL=1000\text{bit}$  ein vergrößertes Fenster  $w_S=7$  zu wählen ist oder daß bei gleichbleibender Fenstergröße längere Blocklängen  $NBL > 3000\text{bit}$  anzustreben sind. Entsprechendes gilt auch für  $n=5$  bzw.  $n=10$  Verbindungen.

Bild 6.17 zeigt die Durchsatzklasse DK als Funktion der Laufzeit  $T_L$ . Beträgt z.B. die Laufzeit  $T_L$  die einfache Datenpaketübertragungszeit  $T_{ODP}$  ( $T_L/T_{ODP}=1,0$ ), können bei  $n=2$  virtuellen Verbindungen die Durchsatzklassen  $DK=19200\text{bit/s}$  mit  $w_S=2$  oder  $DK=48000\text{bit/s}$  mit  $w_S=7$  gewählt werden. Für  $n=5$  virtuelle Verbindungen kommt nur  $DK=9600\text{bit/s}$  in Frage;  $n=10$  VC läßt sogar nur  $DK=4800\text{bit/s}$  zu.

#### 6.5 Messungen im DATEX-P Netz der Deutschen Bundespost

Zur Erweiterung ihres Dienstleistungsangebotes im Bereich der Datenübermittlung entschied sich die Deutsche Bundespost im Dezember 1978 für die Einführung eines paketvermittelnden Datexdienstes. Hierdurch sollte der steigenden Nachfrage nach Datenfernverarbeitungs- und offener Kommunikation innerhalb eines Datenübermittlungssystems entsprochen werden. Darüber hinaus sollte die bessere Ausnutzung der Übertragungswege, die diese Technik ermöglicht, dem Teilnehmer für bestimmte Anwendungsfälle eine wirtschaftlichere Lösung bieten als z.B. private Netze. 1980 wurde mit dem Aufbau eines bundesweiten Netzes begonnen, bestehend aus 17 Vermittlungsstellen, einem Netzkontrollzentrum und einer Auslandskopfvermittlungsstelle, /88,89/.

In /90/ wurde über Messungen im DATEX-P Netz berichtet, wobei das Hauptaugenmerk auf die beiden Dienstgütekriterien Transferzeit und Durchsatz gelegt wurde, da diese in paketvermittelnden Netzen von größerer Bedeutung sind als in Durchschaltvermittlungsnetzen, vgl. Kap.2.1. Die Ermittlung der Meßwerte erfolgte während der Hauptverkehrszeiten über längere Zeiträume hinweg. Der Hauptverkehr in Datennetzen tritt dabei erfahrungsgemäß zu den Zeiten 10:00 Uhr bis 11:00 Uhr vormittags und 15:00 Uhr bis 16:00 Uhr nachmittags auf.

Weitere Meßergebnisse bezüglich der Leistungsfähigkeit von DATEX-P können den Veröffentlichungen /91,92/ entnommen werden, wobei in /92/ die Messungen ebenfalls zu Vergleichszwecken herangezogen wurden.

Der Grund, Messungen im DATEX-P Netz mit den in der vorliegenden Arbeit gezeigten Ergebnissen zu vergleichen, ist darin zu sehen, daß im DATEX-P Netz der Deutschen Bundespost hauptsächlich das Protokoll X.25 zur Anwendung kommt.

Bei den in /90/ bis /92/ vorgestellten Meßergebnissen handelt es sich um Resultate bundesweiter bzw. internationaler Messungen, die somit mehrere Übermittlungsabschnitte und damit auch die Prozeßzeiten der zwischengeschalteten Paketvermittlungsstellen berücksichtigen.

Die Absolutwerte der Messungen bewegen sich dadurch in anderen Bereichen wie die in der vorliegenden Arbeit vorgestellten Simulations- und Analyseergebnisse, da diese das Leistungsvermögen auf nur einem Übermittlungsabschnitt widerspiegeln.

Trotz dieser vermeintlichen Ungleichheiten zwischen den Messungen im DATEX-P Netz und den Ergebnissen dieser Arbeit lassen sich interessante Vergleiche und Gemeinsamkeiten aufzeigen.

Bezüglich des Durchsatzes zeigen die Messungen im DATEX-P Netz prinzipiell dasselbe Verhalten, wie in Bild 6.2 wiedergegeben. Mit zunehmender Anzahl an Übermittlungsabschnitten - dies entspricht einer Zunahme der Laufzeit  $T_L$  - nimmt der Durchsatz deutlich ab. Ebenso zeigen die Messungen den in Bild 6.1 gezeigten Verlauf insofern, als der Durchsatz pro virtueller Verbindung mit der Anzahl  $n$  an Verbindungen abnimmt bzw. mit zunehmender Fenstergröße  $w_S$  deutlich verbessert werden kann.

Hinsichtlich der Transferzeit konnte das in Bild 6.7 gezeigte Verhalten durch entsprechende Messungen im realen DATEX-P Netz belegt werden.

## 7. ZUSAMMENFASSUNG, AUSBLICK UND KRITIK

Die in der vorliegenden Arbeit aufgezeigten Untersuchungen haben zum Ziel, diejenigen Faktoren herauszukristallisieren, welche die Leistungsfähigkeit virtueller Datenverbindungen auf einem X.25-Übermittlungsabschnitt im wesentlichen bestimmen und beeinflussen. Es sollte insbesondere der Einfluß dieser leistungsbestimmenden Faktoren auf die beiden in Datenvermittlungssystemen sehr wichtigen Verkehrsgrößen Durchsatz und Transferzeit quantitativ ermittelt werden.

Für diese Aufgabenstellung wurde zunächst ein komplexes Verkehrsmodell entwickelt, welches das Ablaufgeschehen auf einem X.25-Netz Zugangsabschnitt in allen Schichten nachbildet und beschreibt. Hauptkomponenten dieses Modells sind dabei sowohl der Vollduplexkanal als auch Speicher für Pakete und Rahmen sowie Prozessoren zur Behandlung derselben und zur Protokollabwicklung.

Dieses Verkehrsmodell diente als Basis für die Entwicklung und Erstellung eines Simulationsprogrammes, in welchem alle Feinheiten der diversen Prozeduren der CCITT-Empfehlung X.25 implementiert sind. Desweiteren diente das Verkehrsmodell als Ausgangspunkt für die mathematische Analyse der Leistungsfähigkeit von X.25. Unter Zuhilfenahme der Dekompositionsmethode wurden zwei Teilmodelle erstellt, wobei das eine den Einfluß des leistungsbeeinflussenden Fensterkontrollmechanismus widerspiegelt, während das andere den Einfluß wiedergibt, der dadurch entsteht, daß mehrere virtuelle Verbindungen diverse Netzressourcen, wie Prozessoren und Speicher, gemeinsam in Anspruch nehmen.

Beide Teilmodelle wurden getrennt voneinander der Mittelwertanalyse unterworfen. Ergebnisse, die die Leistungsfähigkeit des gesamten Verkehrsmodells und damit von X.25 betreffen, wurden durch geeignetes Verknüpfen der Teilergebnisse erzielt.

Die beiden im Rahmen der Leistungsuntersuchung der CCITT-Empfehlung X.25 interessierenden Verkehrsgrößen - Durchsatz und Transferzeit - werden dabei hauptsächlich beeinflusst durch die Anzahl virtueller Verbindungen, die Größen der Speicher, die Bearbeitungszeiten der Pakete bzw. der Rahmen, die Übertragungsgeschwindigkeit des Kanals und seiner Laufzeit, den Ankunftsabstand und die Länge der zu übertragenden Daten, die Größe des Sendefensters sowie durch die in den Protokollen festgelegten Regeln zum gesteuerten Datenaustausch.

Die im Rahmen der vorliegenden Arbeit vorgestellte Methode zur Leistungsuntersuchung der komplexen und mehrschichtigen Protokollvorgänge in X.25 soll auch als Anhaltspunkt dienen, zukünftige oder auch bereits bestehende, ähnlich gelagerte Normen, wie z.B. die CCITT-Empfehlung X.75 oder der Pacing-Mechanismus, auf dieselbe Art und Weise untersuchen zu können.

In diesem Zusammenhang soll nicht unerwähnt bleiben, daß der Nachteil der gezeigten Methode für die Leistungsuntersuchung mittels Dekomposition und Mittelwertanalyse darin besteht, daß die Art der Vorgehensweise nur für symmetrische Parameterkonstellationen möglich ist und daß starke Abweichungen zwischen simulativ und mathematisch erzielten Ergebnissen auftreten können, da die mathematische Analysemethode nicht in der Lage ist, protokollspezifische Feinheiten in allen Details nachzubilden, wie dies die simulative Methode vermag.

Ein weiteres Ziel dieser Arbeit war die Implementierung der CCITT-Schnittstelle X.25 in Hard- und Software in den Schichten 2 und 3 auf zwei voneinander unabhängig arbeitenden Prozessorsystemen. Dieser Realisierung diente ein detailliert ausgearbeitetes Prozeßkonzept als Basis, welches einerseits die Austauschbarkeit der Schichten durch die klare Definition eines Dienstprotokolls erlaubt und welches andererseits eine leichte Änderbarkeit zuläßt, da jedem Prozeß genau ein Programmteil mit definiertem Ein- und Ausgangspunkt zugeordnet wurde.

Ein letztes und nicht weniger wichtiges Ziel dieser Arbeit war die Entwicklung und Vorstellung eines Verfahrens zur Leistungsbeurteilung virtueller Datenverbindungen, welches bei der praktischen Dimensionierung von X.25-gesteuerten Netzzugangsabschnitten eingesetzt werden kann.

LITERATURVERZEICHNIS

- /1/ P.R. Gerke:  
Neue Kommunikationsnetze. Prinzipien, Einrichtungen,  
Systeme.  
Springer-Verlag, Berlin/Heidelberg/New York, 1982.
- /2/ Recommendation X.25:  
Interface between Data Terminal Equipment (DTE) and Data  
Circuit Terminating Equipment (DCE) for Terminals Opera-  
ting in the Packet Mode on Public Data Networks.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /3/ Recommendation X.75:  
Terminal and Transit Call Control Procedures and Data  
Transfer System on International Circuits between Packet-  
Switched Data Networks.  
CCITT Yellow Book, Vol.VIII-3, Geneva, 1980.
- /4/ J. Atkins:  
Path Control: The Transport Network of SNA.  
IEEE Transactions on Communications, Vol.Com-28, No.4,  
April 1980.
- /5/ P. Bocker:  
Datenübertragung, Band I/II.  
Springer-Verlag, Berlin/Heidelberg/New York, 1976/1977.
- /6/ Daten- und Textkommunikation.  
Begriffe.  
NTG 1203, Entwurf 1981
- /7/ P. Kühn:  
Nachrichtenvermittlung (Rechner- und Datenkommunikation).  
Hochschulkolleg, Universität Stuttgart, 1984.

- /8/ W. Bux:  
Ober die verkehrsgerechte Dimensionierung der Vermittlungs-  
knoten in Daten- und Rechnernetzen.  
Dissertationsschrift, Universität Stuttgart, 1980.
- /9/ D.W. Davis, D.L.A. Barber:  
Communications Networks for Computer.  
J. Wiley & Sons, London/New York, 1973.
- /10/ L. Kleinrock:  
Queuing Systems. Volume II: Computer Applications.  
J. Wiley & Sons, London/New York, 1976.
- /11/ DIN/ISO-Norm 7498:  
Kommunikation offener Systeme. Basis Referenzmodell.  
Mai 1982.
- /12/ P. Schmitz, U. Hasenkamp:  
Rechnerverbundsysteme. Offene Kommunikationssysteme auf  
der Basis des ISO-Referenzmodells.  
Carl Hanser Verlag, München/Wien, 1981.
- /13/ H.J. Burkhardt:  
Kommunikation offener Systeme - Stand und Perspektiven  
der Normungsarbeit -.  
Der GMD-Spiegel, 2/81, (Teil I),  
Der GMD-Spiegel, 3/81, (Teil II).
- /14/ H.J. Burkhardt:  
Architektur offener Kommunikationssysteme - Stand der  
Normungsarbeit -.  
Tagungsband "Kommunikation in verteilten Systemen",  
Workshop der Gesellschaft für Informatik,  
Springer Verlag Berlin/Heidelberg/New York, 1979.

- /15/ H. Zimmermann:  
OSI Reference Model - The ISO Model of Architecture for  
Open Systems Interconnection.  
IEEE Transactions on Communications, Vol.Com-28, No.4,  
April 1980.
- /16/ J. Houldsworth:  
Standards for open network operation.  
Computer Communications, Vol.1, No.1, Febr. 1978.
- /17/ P. Green:  
An Introduction to Network Architectures and Protocols.  
IEEE Transactions on Communications, Vol.Com-28, No.4,  
April 1980.
- /18/ P.J. Kühn:  
Verteilte Mikrorechner-Steuerungen in Nachrichtenvermitt-  
lungssystemen - Strukturen, Organisation und Verkehrs-  
analyse.  
in:  
Dienstintegration in künftigen Kommunikationsnetzen.  
H.L. Hartmann (Hrsg.)  
B.G. Teubner, Stuttgart, 1982.
- /19/ P. Kühn:  
Wartezeitprobleme der Daten- und Nachrichtenverkehrs-  
theorie.  
Vorlesung, Universität Stuttgart, 1981.
- /20/ P. Kühn:  
Methoden der Verkehrsanalyse.  
Hochschulkolleg, Universität Stuttgart, 1983.
- /21/ W. Weiß:  
Ober Struktur, Betriebsweise und Verkehrsleistung eines  
dienstintegrierten Vermittlungssystems für Fernsprech-  
und Datenverkehr.  
Dissertationsschrift, Universität Stuttgart, 1984.

- /22/ Nachrichtenverkehrstheorie. Begriffe.  
Empfehlung NTG 0903, 1983.
- /23/ P. Kühn:  
Analyse zufallsabhängiger Prozesse in Systemen zur Nach-  
richtenvermittlung und Nachrichtenverarbeitung.  
30.Bericht über verkehrstheoretische Arbeiten,  
Institut für Nachrichtenvermittlung und Datenverarbeitung,  
Universität Stuttgart, 1981.
- /24/ J.R. Jackson:  
Networks of waiting lines.  
Operations Research, Vol.5, 1957.
- /25/ W.J. Gordon, G.F. Newell:  
Closed queuing systems with exponential servers.  
Operations Research, Vol.15, 1967.
- /26/ P.J. Burke:  
The Output of a Queuing System.  
Operations Research, Vol.4, 1956.
- /27/ E. Reich:  
Waiting Times when Queues are in Tandem.  
Ann. Math. Statist., Vol.28, 1957.
- /28/ F. Baskett, F.M. Chandy, R.R. Muntz, F.G. Palacios:  
Open, Closed and Mixed Networks of Queues with Different  
Classes of Customers.  
Journ. of Assoc. for Computer Machines, Vol.22, 1975.
- /29/ M. Reiser, H. Kobayashi:  
Queuing Networks with Multiple Closed Chains: Theory and  
Computational Algorithms.  
IBM Journ. Res. and Development, May 1975.



- /30/ M. Reiser:  
Mean-Value Analysis of Queuing Networks. A New Look at an Old Problem.  
Kongreßband "Performance of Computer Systems",  
Wien, Österreich, Febr. 1979,  
North Holland Publ. Company, 1979.
- /31/ M. Reiser, S.S. Lavenberg:  
Mean-Value Analysis of Closed Multichain Queuing Networks.  
Journ. of Assoc. for Computer Machines, Vol.27, No.2,  
April 1980.
- /32/ M. Reiser:  
Mean-Value Analysis and Convolution Method for Queue-  
Dependent Servers in Closed Queuing Networks.  
Performance Evaluation, No.1, 1981.
- /33/ M. Reiser:  
Performance Evaluation of Data Communication Systems.  
Proceedings of the IEEE, Vol.70, No.2, Febr. 1982.
- /34/ J.C. Little:  
A Proof of the Queuing Formula:  $L = \lambda \cdot w$ .  
Operations Research, Vol.9, No.3.
- /35/ L. Kleinrock:  
Queuing Systems. Volume I: Theory.  
J. Wiley & Sons, New York, 1975.
- /36/ G. Bolch, I.F. Akyildiz:  
Analyse von Rechensystemen.  
B.G. Teubner, Stuttgart, 1982.
- /37/ S.S. Lavenberg, M. Reiser:  
Stationary State Probabilities of Arrival Instants for  
Closed Queuing Networks with Multiple Types of Customers.  
Journ. of Applied Prob., Vol.27, 1980.

- /38/ K.C. Sevcik, I. Mitrani:  
The Distribution of Queuing Network States at Input and  
Output Instants.  
Kongreßband "4th Intern. Sympos. Modelling and Performance  
Evaluation of Computer Systems".  
North Holland Publ. Company, 1979.
- /39/ P.J. Kühn:  
Analysis of Complex Queuing Networks by Decomposition.  
Kongreßband "8th Internat. Teletraffic Congress",  
Melbourne, Australien, 1976.
- /40/ P.J. Kühn:  
Approximate Analysis of General Queuing Networks by  
Decomposition.  
IEEE Transactions on Communications, Vol.Com-27, No.1,  
Jan. 1979.
- /41/ P.J. Kühn:  
Analysis of Switching System Control Structures by  
Decomposition.  
Kongreßband "9th Internat. Teletraffic Congress",  
Torremolinos, Spanien, 1979.
- /42/ P.J. Kühn:  
Analysis of Switching System Control Structures by  
Decomposition.  
AEÜ, Band 34, Heft 2, 1980.
- /43/ P.J. Kühn:  
Multiqueue Systems with Nonexhaustive Cyclic Service.  
The Bell System Technical Journal, Vol.58, No.3,  
März 1979.

- /44/ P.J. Kühn:  
Performance of ARQ-Protocols for HDX-Transmission in Hierarchical Polling Systems.  
Performance Evaluation, No.1, 1981.
- /45/ P.J. Kühn:  
Modelling and Analysis of Computer Networks - Decomposition Techniques, Transient Analysis and Protocol Implications.  
Kongreßband "Conference on Computer Communications '84", Amsterdam, Holland, 1984.
- /46/ C.D. Pack:  
The Output of an M/D/1 Queue.  
Operations Research, Vol.23, No.4, July/August 1975.
- /47/ K.M. Chandy, U. Herzog, L. Woo:  
Parametric Analysis of Queuing Networks.  
IBM Journ. Res. Develop., Jan. 1975.
- /48/ CCITT-Recommendation X.25.  
CCITT Study Group VII, Doc. AP VII-No.7-E, June 1980.
- /49/ Die V.- und X. Empfehlungen des CCITT.  
Elektronik, Heft 21, 1979.
- /50/ M.S. Sloman:  
X.25 explained.  
Computer Communications, Vol.1, No.6, Dec. 1980.
- /51/ W. Tietz:  
Was ist X.25?  
Elektronische Rechenanlagen, 21.Jhrg., Heft 3, Juni 1979.
- /52/ J.L. Snare:  
An Introduction to the CCITT-Recommendation X.25.  
Telecomm. Journ. Australia, Vol.33, No.2, 1983.

- /53/ Recommendation X.21: Interface between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Synchronous Operation on Public Data Networks.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /54/ Recommendation X.1: International User Classes of Service in Public Data Networks.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /55/ Recommendation X.24: List of Definitions for Interchange Circuits between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) on Public Data Networks.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /56/ Recommendation X.26: Electrical Characteristics for Unbalanced Double-Current Interchange Circuits for General Use with Integrated Circuit Equipment in the Field of Data Communications.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /57/ Recommendation X.27: Electrical Characteristics for Balanced Double-Current Interchange Circuits for General Use with Integrated Circuit Equipment in the Field of Data Communications.  
CCITT Yellow Book, Vol.VIII-2, Geneva, 1980.
- /58/ Data Communication - High Level Data Link Control Procedures - Frame Structure.  
International Standard IS 3309.
- /59/ Data Communication - High Level Data Link Control Procedures - Elements of Procedures (independent numbering).  
International Standard IS 4335.

- /60/ HDLC - Proposed Additional Elements of Procedures.  
Document ISO/TC 97/SC 6 - No.1300.
- /61/ HDLC - Proposed Unbalanced Classes of Procedures.  
Document ISO/TC 97/SC 6 - No.1339.
- /62/ HDLC - Proposed Balanced Classes of Procedures.  
Document ISO/TC 97/SC 6 - No.1444.
- /63/ HDLC - Proposed Enhancement to DIS 4335.  
Document ISO/TC 97/SC 6 - No.1445.
- /64/ Bitorientierte Steuerungsverfahren zur Datenübertragung.  
Teil 1: HDLC, Aufbau des Datenübertragungsblockes.  
April 1980.  
Teil 2: HDLC, Elemente der Steuerungsverfahren.  
Entwurf 1983.  
Teil 3: HDLC, Klassen der Steuerungsverfahren.  
Entwurf 1983.  
DIN 66 221.
- /65/ HDLC - Obermittlungsvorschrift zwischen Hybridstationen.  
April 1983.  
DIN 66 222.
- /66/ H.L. Truong:  
Über die Leistungsfähigkeit von HDLC-gesteuerten Daten-  
verbindungen.  
Dissertationsschrift, Universität Stuttgart, 1980.
- /67/ W. Bux, K. Kümmerle, H.L. Truong:  
Results on the Performance of Balanced HDLC-Procedures.  
Kongreßband "National Telecommunication Conference",  
Birmingham, USA, 1978.

- /68/ W. Bux, H.L. Truong:  
A Queuing Model for HDLC-Controlled Data Links.  
Kongreßband "Internat. Symp. on Flow Control in Computer  
Networks",  
Versailles, Frankreich, 1979.
- /69/ W. Bux, H.L. Truong:  
High Level Data Link Control - Traffic Considerations.  
Kongreßband "9th Internat. Teletraffic Congress",  
Torremolinos, Spanien, 1979.
- /70/ W. Bux, K. Kümmerle, H.L. Truong:  
Balanced HDLC-Procedures: A Performance Analysis.  
IEEE Transactions on Communications, Vol.Com-28, No.11,  
Nov. 1980.
- /71/ W. Bux, K. Kümmerle, H.L. Truong:  
Data Link Control Performance - Results Comparing HDLC  
Operational Modes.  
IBM Research Report RZ 1027, 1980.
- /72/ W. Bux, K. Kümmerle, H.L. Truong:  
HDLC Performance - Comparison of Normal Response Mode  
and Asynchronous Balanced Mode of Operation.  
Kongreßband "National Telecomm. Conference",  
Houston, USA, 1980.
- /73/ W. Bux, K. Kümmerle, H.L. Truong:  
Data Link-Control Performance: Results Comparing HDLC  
Operational Modes.  
Computer Networks, No.6, 1982.
- /74/ C. Chlap, H.P. Dittler:  
Entwicklung eines X.25-Bausteins.  
Forschungsbericht BMFT-FB-DV 81-002,  
Universität Karlsruhe, 1981.

- /75/ G.v. Bochmann, T. Joachim:  
Development and Structure of an X.25 Implementation.  
IEEE Transactions on Software Eng., Vol.SE-5, No.5,  
May 1979.
- /76/ J.P. Johnson, J.M. Heap:  
Designing an X.25 network-to-host interface.  
Computer Communications, Vol.1, No.1, 1978.
- /77/ M 6800 Microprocessor Applications Manual.  
Motorola, 1st edition, 1975.
- /78/ MC 68000 - 16-bit Microprocessor User's Manual.  
Motorola, 3rd edition, 1982.
- /79/ MicroSys CPU 01 - Anwenderhandbuch.  
MicroSys electronics GmbH, 1.dtsche Ausgabe, 1983.
- /80/ A. Giessler:  
Throughput Guarantee in X.25 Networks.  
Kongreßband "Flow Control in Computer Networks",  
Versailles, Frankreich, 1979.
- /81/ A. Giessler, A. Jägemann, E. Mäser:  
Simulation of an X.25 Network Providing Throughput  
Guarantees.  
Kongreßband "Performance of Data Communication Systems  
and Their Applications",  
Paris, Frankreich, 1981.
- /82/ M. Reiser:  
Admission Delays on Virtual Routes with Window Flow  
Control.  
Kongreßband "Performance of Data Communication Systems  
and Their Applications",  
Paris, Frankreich, 1981.

- /83/ M. Reiser:  
Calculation of Response-Time Distributions in Cyclic  
Exponential Queues.  
Performance Evaluation, No.1, 1981.
- /84/ M. Reiser:  
A Queuing Network Analysis of Computer Communication  
Networks with Window Flow Control.  
IEEE Transactions on Communications, Vol.Com-27, No.8,  
August 1979.
- /85/ M. Schwartz:  
Performance Analysis of the SNA Virtual Route Pacing  
Control.  
IEEE Transactions on Communications, Vol.Com-30, No.1,  
Jan. 1982.
- /86/ L.W. Yu, J. Majithia:  
An Analysis of One Direction of Window Mechanism.  
IEEE Transactions on Communications, Vol.Com-27, No.5,  
May 1979.
- /87/ J. Labetoulle, G. Pujolle, N. Mikou:  
A Study of Flows in an X.25 Environment.  
Kongreßband "Flow Control in Computer Networks",  
Versailles, Frankreich, 1979.
- /88/ D. Runkel, W. Tietz:  
Die Technik des paketvermittelten Datennetzes DATEX-P.  
Der Fernmelde-Ingenieur, 35.Jahrgang, Heft 8, 1981.
- /89/ W. Tietz, D. Runkel:  
Der paketvermittelte Datendienst DATEX-P.  
Der Fernmelde-Ingenieur, 36.Jahrgang, Heft 6, 1982.

- /90/ K.P. Steinruck:  
Transferzeit- und Durchsatzmessungen im DATEX-P-Netz.  
Kolloquiumsvortrag am Forschungsinstitut der Deutschen  
Bundespost, Darmstadt, Oktober 1984.
- /91/ R. Runkel:  
Experience with and performance measurements of the packet  
switching network in the Federal Republic of Germany.  
Kongreßband "6th International Conference on Computer Com-  
munication", London, Sept. 1982.
- /92/ H. Mühlenbein, F. Limburger:  
Simulation of Flow and Congestion Control Algorithms.  
Kongreßband "Performance of Computer-Communication Systems".  
Zürich, Schweiz, 1984.

