KfK 3024
August 1980

MOSAIC (BERNET): Ein Modellierungssystem zur Leistungsprognose und -analyse für das BERNET-Rechnernetz

B. Wolfinger
Institut für Datenverarbeitung in der Technik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Datenverarbeitung in der Technik

KfK 3024

MOSAIC (BERNET) : Ein Modellierungssystem zur Leistungsprognose und - analyse für das BERNET-Rechnernetz

B. Wolfinger

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

Kurzfassung

Rechnernetze gehören sicher zu den komplexesten ingenieurmäßig zu bewältigenden Strukturen, und daher werden Entscheidungshilfen zur Unterstützung der Entwurfs- und Implementierungsphase sowie zur Optimierung des laufenden Betriebs von Rechnernetzen dringend benötigt.

Dieser Bericht stellt zunächst ein Projektierungshilfsmittel - das Modellierungssystem MOSAIC (Modellierungssystem zur Simulation allgemeiner Informationsflüsse in Computernetzen) - vor, das eine quantitative Analyse von Rechnernetzen unterstützt und Aussagen über Engpässe, Vergleiche von Designalternativen u.ä. erlaubt. Die Vorgehensweise bei der Entwicklung von MOSAIC war dergestalt, daß ein Grundmodell erarbeitet wurde, welches sich auf möglichst einfache Weise an verschiedenartige Rechnernetzcharakteristika (wie z.B. Hierarchien von Kommunikationsprotokollen, Rechnernetzkonfigurationen, Auftragsprofile) anpassen läßt.

Es wird anschließend ein Modellierungssystem eingeführt, das aus dem MOSAIC-Grundmodell hervorging und zur Simulation der Kommunikationsflüsse des BERNET-Rechnernetzes unter detaillierter Berücksichtigung der verwendeten Kommunikationsprotokolle herangezogen werden kann.

MOSAIC (BERNET): A modeling system for performance evaluation and prediction of the BERNET computer network

Abstract

The complexity of computer networks implies the need for decision aids to support the design and implementation of planned networks as well as to optimize the operation of existing networks.

This report introduces a tool - the modeling system MOSAIC (modeling system to simulate arbitrary information flows in computer networks) - which allows to analyze the behavior of computer networks, to determine bottlenecks, to compare design alternatives, etc. MOSAIC has been built up as a basic modeling system which can be adapted to cover different computer network characteristics (e.g. hierarchies of communication protocols, network configurations, load profile).

The adaption of this basic modeling system to simulate the communication flows within the BERNET computer network is shown in detail to demonstrate the application of MOSAIC to an existing computer network.

Inhaltsverzeichnis

		Seite
1.	Einleitung und Problemstellung	1
2.	Konzeptionelle Modelle zur Beschreibung von Kommunikationsflüssen in Rechnernetzen	3
3.	Das Modellierungssystem MOSAIC	10
3.1.	Struktur und Komponenten	10
3.2.	Programmbausteine zur Modellierung der Komponenten eines Rechnernetzes	16
3.2.1.	Modellierung hierarchisch organisierter Kommunikationssoftware	16
3.2.2.	Modellierung von Aufträgen in einem Rechnernetz	18
3.2.3.	Modellierung des Datenflusses in einem Rechnernetz	22
3.2.4.	Modell eines Rechnernetzknotens	24
3.2.5.	Generierung der Rechnernetzbelastung	25
3.3.	Konfigurierung eines ablauffähigen Simulationsprogramms	26
3.4.	Meßdatenerfassung	28
3.4.1.	Vermessungsgrößen	28
3.4.2.	Ergebnisvariablen	30
3.5.	Datenauswertung	32
4.	Abbildung des BERNET-Rechnernetzes auf MOSAIC	33
4.1.	Modellierung der physikalischen Charakteristika des BERNET	34
4.1.1.	Rechnernetztopologie	34
4.1.2.	Hardwareeigenschaften der Rechnernetzknoten	35
4.1.3.	Übertragungsleitungen	36
4.2.	Modellierung der Betriebssystemcharakteristika der Rechner des BERNET	37

		Seite
4.3.	Modellierung der Kommunikationssoftware des BERNET	38
4.3.1.	Gesamtaufbau der Protokollhierarchie	38
4.3.2.	Die modellierten Kommunikationsprotokolle und die Schichtenschnittstellen	41
4.3.2.1.	Basisprotokoll	41
4.3.2.2.	Das Protokoll HDLC	47
4.3.2.3.	Das Protokoll X.25 (Paketebene)	52
4.3.2.4.	Das Message Link Protokoll	60
4.3.2.5.	Das Remote Job Entry Protokoll	67
4.3.2.6.	Weitere Kommunikationsprotokolle	74
4.4.	Modellierung der Aufträge im BERNET-Rechnernetz	74
4.4.1.	Benutzertasks	74
4.4.2.	Schnittstellenaufträge	76
5.	Schlußwort	77
6.	Literaturverzeichnis	78
Anhang A:	Benutzeranleitung für MOSAIC(BERNET)	81
Anhang B:	Beispiel eines interaktiven Simulationsexperiments mit MOSAIC(BERNET)	107
	B1. Wesentliche Randbedingungen der gewählten BERNET-Konfiguration	108
	B2. Eingabedatensatz	111
	B3. Protokollierung des Experiments an einem Graphik-Display	116

1. Einleitung und Problemstellung

Durch große Fortschritte in der Halbleitertechnologie und in den Kommunikationstechniken haben sowohl lokale als auch überregionale Rechnernetze erheblich an Bedeutung gewonnen und werden einen starken Einfluß auf die zukünftigen Rechnerstrukturen ausüben. Obwohl viele der angestellten Überlegungen auch für lokale Rechnernetze gelten, wird sich der vorliegende Bericht auf die Analyse und Modellierung überregionaler - d.h. über öffentliche Fernmeldenetze kommunizierende - Rechnernetze konzentrieren. Die besondere Aktualität überregionaler Netze ergibt sich nicht zuletzt aus der Tatsache, daß auch in der Bundesrepublik mit dem baldigen bundesweiten Versuchsbetrieb des X.25 - Datenpaketvermittlungsdienstes der Deutschen Bundespost die Zeit schneller digitaler Rechnernetze begonnen hat.

Rechnernetze gehören sicher zu den komplexesten ingenieurmäßig zu bewältigenden Strukturen, und daher werden Entscheidungshilfen zur Unterstützung der Entwurfs- und Implementierungsphase sowie zur Optimierung des laufenden Betriebs von Rechnernetzen dringend benötigt. Im folgenden soll ein Projektierungshilfsmittel - das Modellierungssystem MOSAIC - vorgestellt werden, das eine quantitative Analyse von Rechnernetzen unterstützt und Aussagen über Engpässe, Vergleiche von Designalternativen u.ä. erlaubt. Die Vorgehensweise bei der Entwicklung von MOSAIC war dergestalt, daß ein Grundmodell erarbeitet wurde, welches sich auf möglichst einfache Weise an verschiedenartige Rechnernetzcharakteristika (wie z.B. Kommunikationsprotokolle, Rechnernetzkonfigurationen, Auftragsprofile) anpassen läßt.

Als wichtigste der aktuellen Anwendungen für MOSAIC wird hier die im Rahmen eines Kooperationsvertrags erfolgte Modellierung des BERNET-Rechnernetzes beschrieben *).

Im ersten Teil der Arbeit wird eine Menge von konzeptionellen Modellen zur Beschreibung des Leistungsverhaltens von Rechnernetzen eingeführt. Die Motivation, die der Definition dieser Modelle zugrundelag, wird dabei weitgehend ausgeklammert; es sei hierzu auf frühere Arbeiten des Autors /WOLF78-79a-79b/ verwiesen.

---

*) Das Modellierungssystem wurde in den Jahren 1978-1980 durch das Institut für Datenverarbeitung in der Technik des Kernforschungszentrums Karlsruhe im Auftrag des Senators für Wissenschaft und Forschung/Berlin erstellt.

Im zweiten Teil wird das Modellierungssystem MOSAIC, das der Auswertung der eingeführten konzeptionellen Modelle durch rechnergestützte Simulation dient, in seinen Grundfunktionen erläutert.

Den Schwerpunkt der Arbeit bildet eine Darstellung der Anpassungen und Erweiterungen des MOSAIC-Grundmodells, die zur Simulation der Kommunikationsflüsse des BERNET-Rechnernetzes erforderlich waren. Der Leistungsumfang des resultierenden Modellierungssystems MOSAIC(BERNET) wird aufgezeigt; seine Benutzung zur Durchführung von Simulationsexperimenten für das BERNET wird anhand eines Beispiels demonstriert.

2. Konzeptionelle Modelle zur Beschreibung von Kommunikationsflüssen in Rechnernetzen

Der Einsatz der Simulation zur Leistungsprognose und -analyse von Rechnernetzen setzt einerseits Modelle zur Beschreibung eines Rechnernetzes und andererseits ein Instrument zur Auswertung der Modelle unter verschiedenartigen Randbedingungen voraus. Insbesondere mit dem ersten dieser beiden Probleme haben wir uns in /WOLF78-79a-79b/ bereits eingehend auseinandergesetzt und generalisierte Modelle eingeführt, die sich durch ein breites Anwendungsspektrum auszeichnen und sich an diejenigen Rechnernetzcharakteristika anpassen lassen, die das interne Ablaufgeschehen in einem Rechnernetz maßgeblich beeinflussen (wie z.B. die zugrundeliegende Hard- und Softwarestruktur der Rechnernetzknoten sowie das Auftragsprofil).

Auf der Basis von Warteschlangenmodellen und sequentiellen Automaten wurde in den o.g. Arbeiten ein Architekturmodell für Rechnernetze eingeführt, das sich aus verschiedenen Modellbausteinen zusammensetzt und u.a. eine detaillierte Beschreibung erlaubt für

- die Kommunikationssoftware, die die Kommunikation in einem Rechnernetz regelt und die, wie die Rechnernetzdienste, in einer Hierarchie interagierender Schichten /vgl. ISO79/ organisiert ist;
- die Betriebsmittelzuteilung in den Rechnern (Rechnernetzknoten) eines Rechnernetzes;
- die Aufträge, die durch die Benutzer eines Rechnernetzes generiert werden.

Am Beispiel zweier kommunizierender Rechnernetzknoten DVS_1 und DVS_2 zeigt Abb. 2.1. die Bedienungsstationen, die bei der Formulierung eines Rechnernetzes als Warteschlangensystem grundsätzlich benötigt werden.

Die Benutzer in einem Rechnernetzknoten werden grundsätzlich als sequentielle Prozesse angesehen und durch Modellbausteine vom Typ "Benutzertask" repräsentiert. Benutzertasks können ausschließlich lokalen Charakter besitzen, oder sie können auch Rechnernetzdienste beanspruchen. Benutzertasks werden durch einen "Auftragsgenerator" erzeugt; pro Rechnernetzknoten wird genau ein Auftragsgenerator vorausgesetzt.

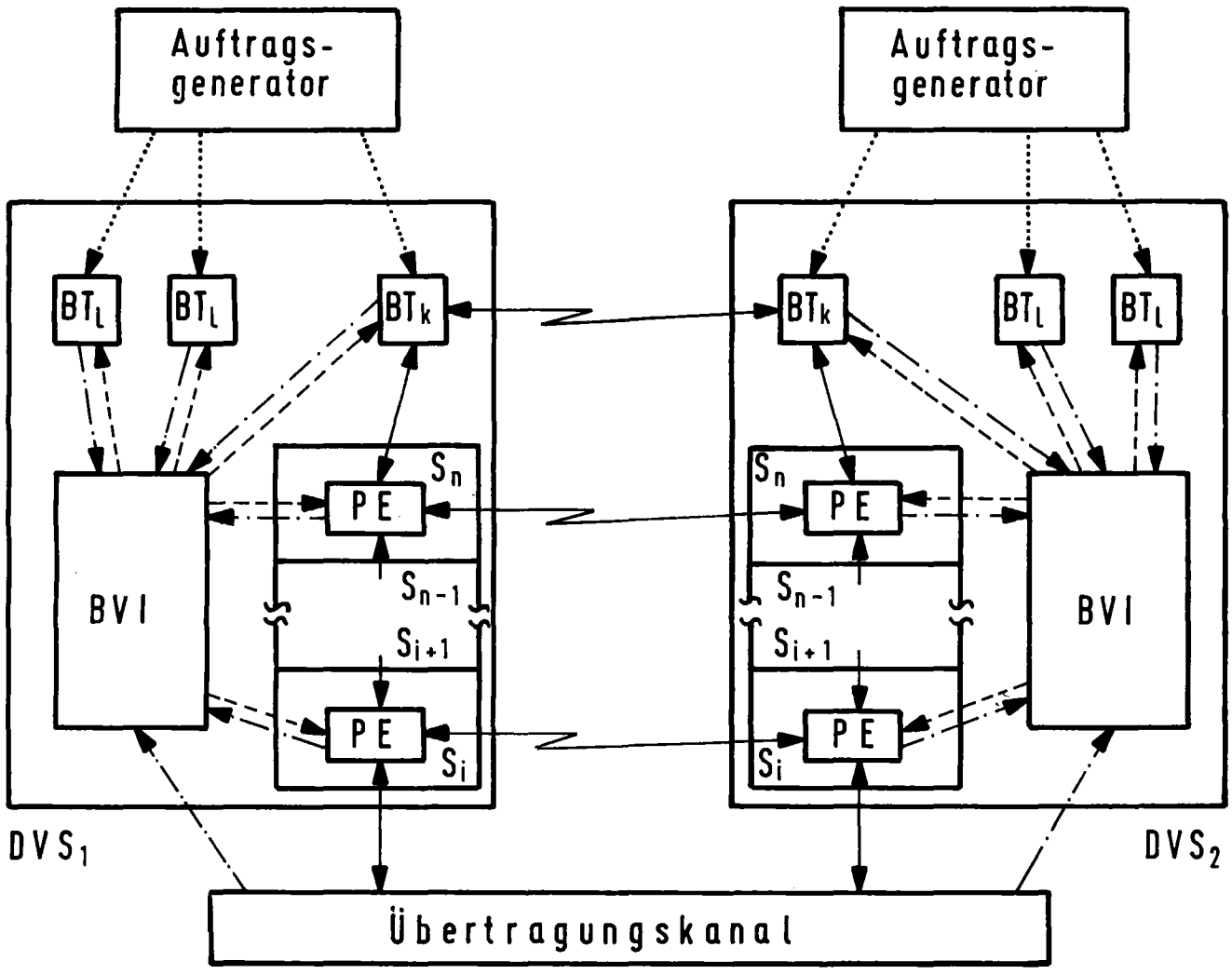


Abb. 2.1.: Die wichtigsten Komponenten eines Rechnernetzes am Beispiel zweier kommunizierender Rechnernetzknoten DVS_1 und DVS_2

[Notation :◄——► Austausch von Schnittstellenaufträgen und/oder Dateneinheiten

◄—≈—► Kommunikationsbeziehung zwischen korrespondierenden Protokolleinheiten/Benutzertasks

—·—·► Wunsch nach Neuverteilung der Rechnerbetriebsmittel

— — ► Zuteilung/Entzug von Rechnerbetriebsmitteln

·····► Generierung und Starten von Benutzertasks

BVI : Betriebsmittelverwaltungsinstanz

PE : Protokolleinheit

$BT_{l/k}$: Benutzertask (lokal/kommunizierend)

$S_1,\ldots,S_i,\ldots,S_n$: Schichten der Protokollhierarchie]

Die (Rechnernetz-) Dienste werden i.a.dezentral realisiert und dies bedingt die Durchführung von Kommunikation - unter Berücksichtigung eines Kommunikationsprotokolls - zwischen den an der Realisierung eines Rechnernetzdienstes beteiligten Instanzen (Kommunikationspartnern). Des weiteren sind Rechnernetzdienste gewöhnlich hierarchisch organisiert, d.h. ein Rechnernetzdienst einer Schicht wird unter Benutzung eines Dienstes einer tieferen Schicht realisiert. Somit kristallisieren sich zwei Aufgaben heraus für eine Instanz I(D), die den Rechnernetzdienst D einer Schicht abwickelt:

(1) die Kommunikation mit den an der dezentralen Realisierung des Dienstes D beteiligten Instanzen in den übrigen Rechnernetzknoten (Kommunikation zwischen sog. Korrespondenten),

(2) die Kommunikation mit den Instanzen benachbarter Schichten in demselben Rechnernetzknoten, die entweder den Dienst D beanspruchen oder die ihrerseits einen Dienst D' für I(D) bereitstellen (Kommunikation zwischen sog. Nachbarn).

Die Kommunikationspartner in einem Rechnernetz, die die Aufgaben (1) und (2) erfüllen, werden auf Modellbausteine vom Typ "Protokolleinheit" abgebildet. Werden in einer Schicht mehrere Dienste parallel bereitgestellt, so sind in dieser Schicht selbstverständlich mehrere Protokolleinheiten vorhanden.

Der Rechnernetzdienst der höchsten Schicht, deren Detaillierung nicht mehr angestrebt wird (im Beispiel der Abb. 2.1. die Schicht S_{i-1}), wird auf einen Modellbaustein vom Typ "Übertragungskanal" abgebildet. Ein Übertragungskanal trägt nur den wesentlichsten Eigenschaften des Dienstes Rechnung, den er repräsentiert (z.B. 'Overhead' der Schichten $S_1,\ldots,S_{i-1}$ in den Rechnernetzknoten DVS_1 bzw. DVS_2, Übertragungszeit und -modus beim Datenaustausch zwischen DVS_1 und DVS_2 u.ä.).

In einem Rechnernetzknoten existieren implizite Interaktionen zwischen den Protokolleinheiten - genauer: zwischen den Protokollmoduln der Protokolleinheiten (s.u.) - und den Benutzertasks, die aus der Konkurrenz um gemeinsam benötigte Rechnerbetriebsmittel resultieren. Man kann einen Rechnernetzknoten somit - neben der an der Struktur der Kommunikationssoftware orientierten Sicht - auch sehen als System von um Betriebsmittel konkurrierenden parallelen Prozessen. Zur Überwachung und Durchführung der Betriebsmittelvergabe an diese Prozesse (im folgenden als "Tasks" bezeichnet) führen wir im Modell eine zentrale Verwaltungsinstanz, die "Betriebsmittelverwaltungsinstanz" ein, die innerhalb ihres Rechnernetzknotens insbesondere die Vergabe der Betriebsmittel Hauptspeicher und CPU organisiert.

Da sich die Struktur der Protokolleinheiten und der Betriebsmittelverwaltungsinstanz als von zentraler Bedeutung für das Leistungsverhalten eines Rechnernetzknotens erweist, lag der Schwerpunkt bei der Modellierung auf diesen beiden Komponenten. Beide Modellbausteine wurden als Warteschlangensysteme formuliert.

- Modell einer Protokolleinheit:

Die Modellierung einer Protokolleinheit geht aus von zwei Bedienungsstationen, dem "Sendeprotokollmodul" und dem "Empfangsprotokollmodul" (s. Abb. 2.2.). Dem Sendeprotokollmodul obliegen die sendeseitigen Aktivitäten einer Protokolleinheit, d.h. die Aktionen, die mit dem Absenden von Daten in Verbindung stehen; der Empfangsprotokollmodul ist für die Durchführung der empfangsseitigen Aktivitäten verantwortlich. Generell bestehen die Aktionen von Sende- und Empfangsprotokollmodul in der Analyse und Abarbeitung sogenannter "Schnittstellenaufträge", wobei die Bearbeitung eines Schnittstellenauftrags die Manipulation von Daten, sogenannter "Dateneinheiten", nach sich ziehen kann. Die Struktur von Dateneinheiten wird rekursiv definiert, wobei sich eine Dateneinheit an der Schnittstelle der Schichten S_i und S_{i-1} zusammensetzt aus "Kontrollinformation" der Schicht S_i, ggf. vereinigt mit Dateneinheiten höherer Schichten. Kontrollinformation dient dem gegenseitigen Austausch von Information zwischen korrespondierenden Protokolleinheiten derselben Schicht in verschiedenen Rechnernetzknoten zur Abwicklung eines Kommunikationsprotokolls. Neben den Schnittstellenaufträgen und Dateneinheiten existieren weitere passive Modellkomponenten, wie z.B. die "Zeitüberwachungsaufträge", die zeitkritische Aktivitäten von

Sende- und Empfangsprotokollmodul überwachen und bei ihrem Ablaufen (Timeout) die Generierung von Schnittstellenaufträgen implizieren können.

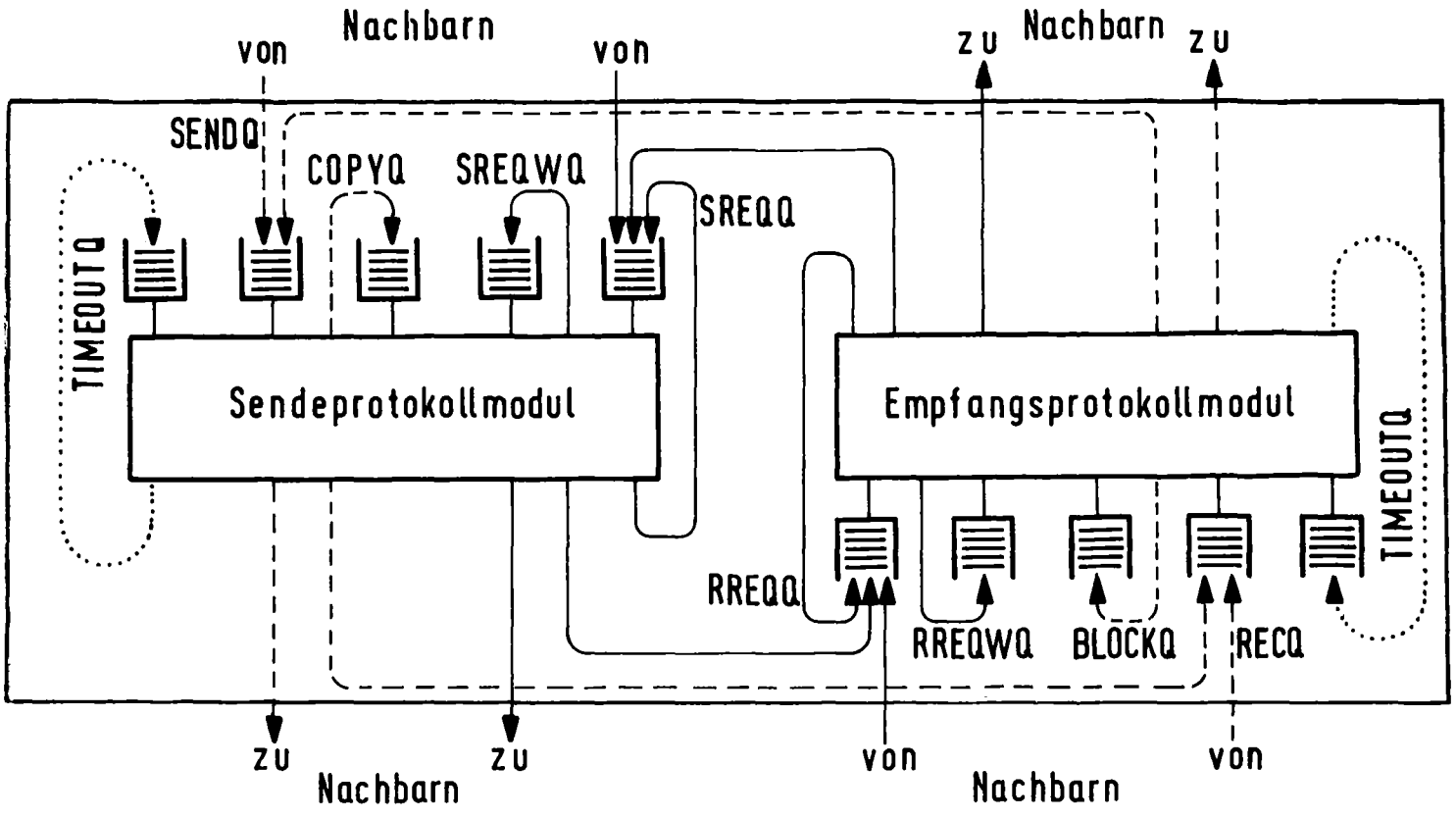


Abb. 2.2.: Warteschlangenmodell einer Protokolleinheit (implementiertes Basismodell)

[Notation: ——→ Auftragsfluß (Übergabe von Schnittstellenaufträgen)

– – – –→ Datenfluß (Übergabe von Dateneinheiten)

·····→ Zeitüberwachungsaufträge]

Die Auftrennung der Auftragswarteschlangen einer Protokolleinheit in die Warteschlangen mit aktuell zu bearbeitenden Elementen (Warteschlangen SREQQ und RREQQ) und in die Warteschlangen mit temporär blockierten Schnittstellenaufträgen (SREQWQ und RREQWQ) wurde aus Gründen der Effizienzerhöhung des Modells vorgenommen. Aus demselben Grund wurden auch beim Datenfluß die Warteschlangen der aufzubereitenden Dateneinheiten (SENDQ, RECQ) von den Warteschlangen mit aktuell nicht zu bearbeitenden Dateneinheiten (COPYQ, BLOCKQ) getrennt.

Die Schnittstellenaufträge und Dateneinheiten sind die Elemente, die durch Sende- und Empfangsprotokollmoduln sowie durch Benutzertasks "bedient" werden. Den Schnittstellenaufträgen kommt dabei eine besondere Bedeutung zu, denn durch sie wird der Ablauf der Protokolleinheit determiniert.

Benutzertasks und Protokollmoduln besitzen eine Doppelrolle: sie sind zum einen Bedienungsstationen für die o.g. passiven Modellbausteine (z.B. Schnittstellenaufträge und Dateneinheiten), jedoch zum anderen sind sie ihrerseits Elemente der Betriebsmittelverwaltungsinstanz-Warteschlangen, und somit fungieren Benutzertasks und Protokollmoduln auch selbst als passive Modellbausteine.

Im Gegensatz zu der Menge der Benutzertasks, die sich dynamisch verändern kann, werden die Anzahl der Protokolleinheiten und ihre gegenseitigen Beziehungen als konstant angenommen.

- Modell einer Betriebsmittelverwaltungsinstanz:

Als Modell für die Betriebsmittelverwaltungsinstanz wurde ein zweistufiges Warteschlangenmodell (s. Abb. 2.3.) gewählt. Die Warteschlangen dienen der Speicherung der sich um Rechnerbetriebsmittel (z.B. CPU, Hauptspeicher) bewerbenden Benutzertasks und Protokollmoduln. Das Warteschlangenmodell ist zweistufig, da die Warteschlangenelemente, z.B. die Sende- und Empfangsprotokollmoduln, ihrerseits wiederum Warteschlangen besitzen.

Die Warteschlangen besitzen folgende Bedeutung:

Q_{T0} = Warteschlange der Benutzertasks im dynamischen Taskzustand "Warten auf Hauptspeicher"

Q_{T1} = Warteschlange der Benutzertasks im dynamischen Taskzustand "Warten auf CPU"

Q_{T2} = Warteschlange der Benutzertasks im dynamischen Taskzustand "Blockiert"

Q_{PM} = Warteschlange der nicht CPU-belegenden Protokollmoduln

Q_{CPU} = Warteschlange der CPU-belegenden Tasks

Q_{IO} = Warteschlange der E/A-aktiven Tasks

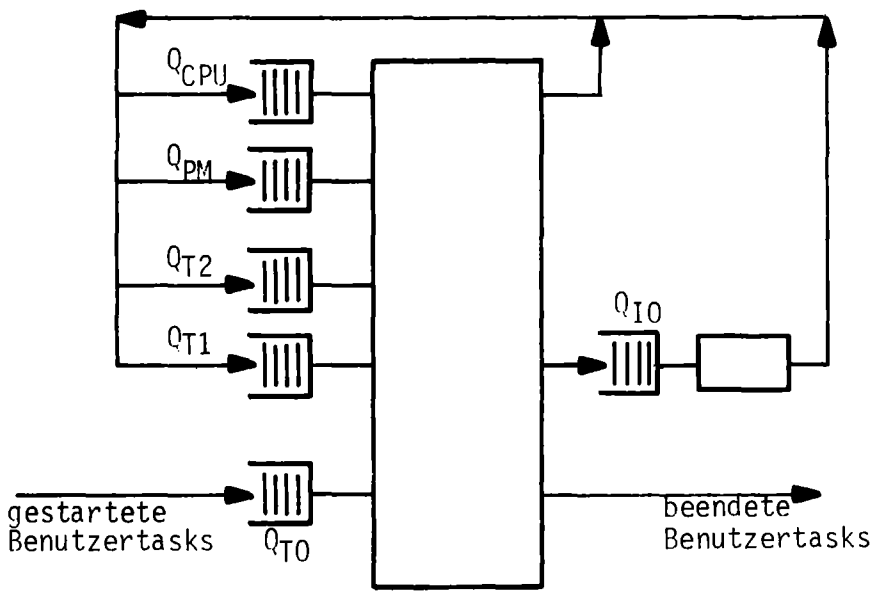


Abb. 2.3.: Warteschlangenmodell einer Betriebsmittelverwaltungsinstanz

3. Das Modellierungssystem MOSAIC

3.1. Struktur und Komponenten

Ziel der nachfolgenden Ausführungen wird es sein, ein Instrument bereitzustellen, das es erlaubt, aus einzelnen Modellbausteinen ein Gesamtmodell eines Rechnernetzes aufzubauen und das Verhalten des Gesamtmodells z.B. unter Einwirkung einer vorgegebenen Belastung zu studieren.

Im folgenden wird der Simulator MOSAIC (Modellierungssystem zur Simulation allgemeiner Informationsflüsse in Computernetzen) vorgestellt, der am Institut für Datenverarbeitung in der Technik des Kernforschungszentrums Karlsruhe entwickelt wurde und zur Auswertung der in Kapitel 2 dargestellten konzeptionellen Teilmodelle eines Rechnernetzes dient.

Für die Erstellung des Programmsystems MOSAIC wurde eine Reihe wesentlicher Anforderungen berücksichtigt:

(1) Anwendbarkeit des Simulators bei verschiedenartigen Randbedingungen (Rechnernetzcharakteristika, Auftragsprofile etc.);
(2) Einsatzmöglichkeit des Simulators für Untersuchungen mit unterschiedlichem Detaillierungsgrad;
(3) einfache Erweiterbarkeit des Simulators;
(4) benutzernahe Resultatdarstellung durch Einsatz graphischer Hilfsmittel;
(5) effiziente Experimentdurchführung durch Unterstützung interaktiver Simulationsexperimente;
(6) hohe Portabilität der zur Implementierung verwendeten Programmiersprache(n).

Besonders im Hinblick auf die Anforderungen (1) - (4) wurde für MOSAIC das Konzept eines Modellierungssystems erarbeitet. Die grundlegende Idee hierbei ist, die zur Simulation von Kommunikationsflüssen in Rechnernetzen benötigten Modellbausteine in Form eines Baukastensatzes bereitzustellen und aus diesen Bausteinen während der Programmausführung ein ablauffähiges Simulationsprogramm zu konfigurieren. Diese Methode erlaubt u.a. eine gegenseitig unabhängige Definition sowie den problemlosen Austausch von Modellbausteinen. Folgende Hauptkomponenten eines Modellierungssystems können unterschieden werden (vgl. Abb. 3.1.):

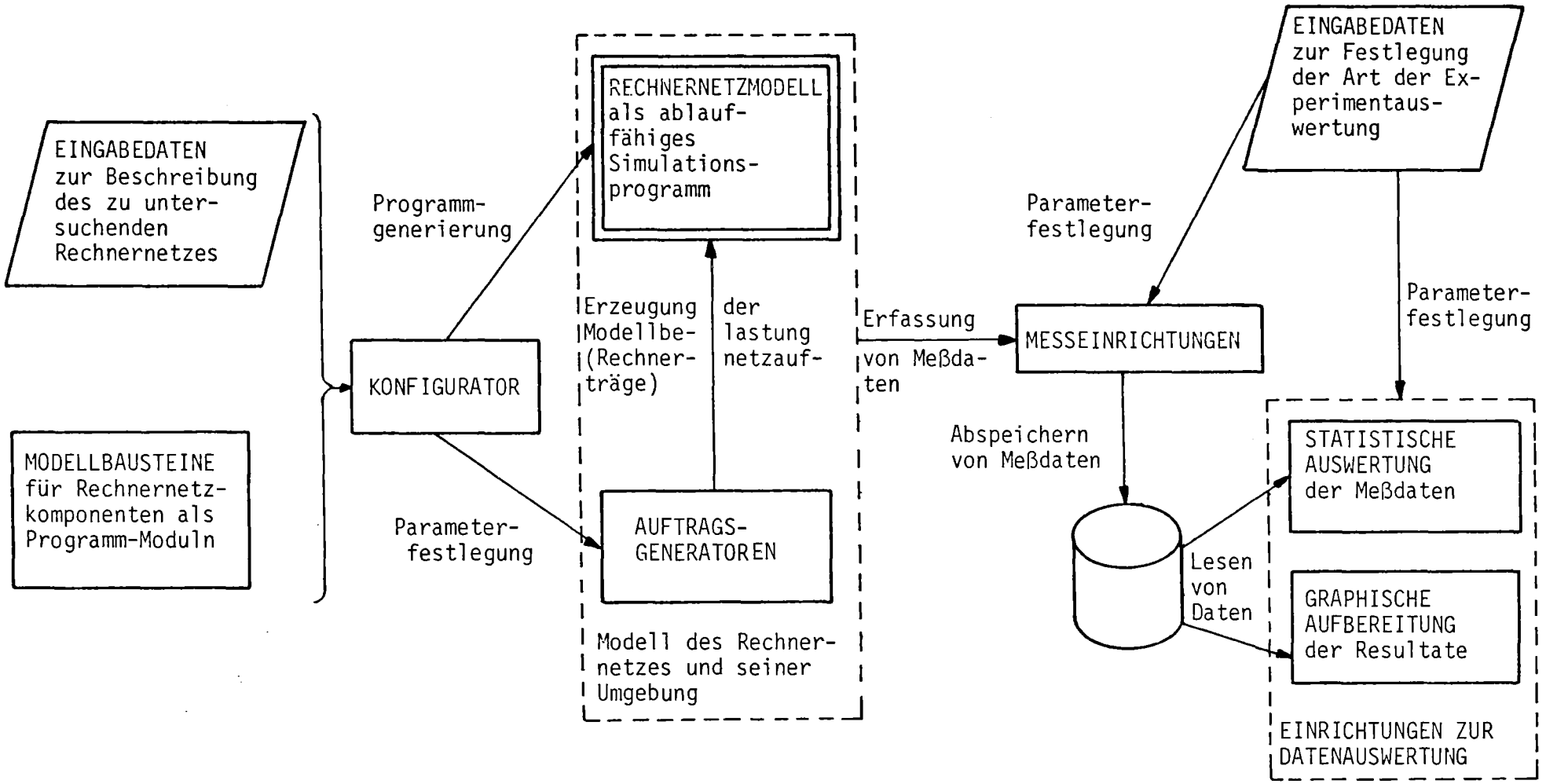


Abb. 3.1.: Zusammenwirken der Hauptkomponenten des Modellierungssystems MOSAIC während eines Simulationsexperiments

- Die Modellbausteine, die eine Umsetzung der in Kapitel 2 präsentierten Teilmodelle in die verwendete Programmiersprache darstellen;
- ein Konfigurator, der zu Beginn eines Simulationsexperimentes aus den Modellbausteinen unter Berücksichtigung der Eingabedaten das Rechnernetzmodell als ablauffähiges Simulationsprogramm erstellt;
- die Auftragsgeneratoren, die zur Erzeugung der Aufträge der Rechnernetzumgebung fungieren und während eines Simulationsexperimentes auf das Rechnernetzmodell einwirken;
- die Meßeinrichtungen, die dazu dienen, im Laufe eines Simulationsexperiments Modellvariablen (z.B. Leistungskenngrößen) zu vermessen und ggf. die erfaßten Meßdaten für eine spätere Auswertung abzuspeichern;
- die Einrichtungen zur Datenauswertung, die dedizierte Komponenten sowohl zur statistischen Auswertung der gesamten Meßdaten als auch zur graphischen Aufbereitung der ermittelten Experimentresultate umfassen.

Als Implementierungssprache für die Erstellung des Modellierungssystems MOSAIC wurde - abgesehen von der Hauptkomponente "Graphische (Resultat-)Aufbereitung", deren Implementierung in FORTRAN erfolgte, - die höhere Programmiersprache SIMULA, s./DAHL68/, gewählt, zum einen um der Forderung nach hoher Portabilität zu genügen und zum anderen auch deshalb, weil das "CLASS"-Konzept in SIMULA die Definition in sich abgeschlossener gegenseitig unabhängiger Modellkomponenten erlaubt, die in beliebiger Anzahl reproduzierbar sind. (Eine "CLASS" in SIMULA bezeichnet eine syntaktische Einheit, die durch Definition eines strukturierten Datentyps und einer Menge darauf auszuführender Operationen festgelgt wird). In diesem Sinne unterstützt die Sprache SIMULA den Aufbau von Modellierungssystemen, da sich die benötigten Modellbausteine als "CLASS" darstellen lassen. Wie Vergleiche zwischen verschiedenen Simulationssprachen zeigen, besitzt SIMULA überdies eine wesentlich höhere Effizienz als Sprachen wie z.B. GPSS, wenn als Vergleichskriterium der CPU-Zeitbedarf für die Ausführung einander äquivalenter Programme gewählt wird (s. Vergleich in /NIED79/).

Das Modellierungssystem MOSAIC enthält die meisten der für eine Rechnernetzsimulation relevanten Komponenten und kann als Nukleus (Grundmodell) angesehen werden, der als Basis detaillierterer Modellierungssysteme dienen kann. Die Erweiterbarkeit sowie die Austauschbarkeit von Modellbausteinen, die MOSAIC dem zugrundeliegenden Modellierungssystemkonzept verdankt, erleichtern derartige Anpassungen von MOSAIC an unterschiedliche Rechnernetze.

Ehe die Hauptkomponenten des Modellierungssystems MOSAIC präziser dargestellt werden, soll in einer SIMULA-ähnlichen Notation ein Überblick über die Grobstruktur der aktuellen Implementierung von MOSAIC gegeben werden. (Eingefügte Kommentare sind dabei zwischen die Zeichen /* und */ eingeschlossen):

```
begin /*    Modellierungssystem MOSAIC

            Initialisierung (1.Teil):
            Definition und Vorbesetzung von Parametern zur globalen Fest-
            legung der Rechnernetzstruktur (z.B. Anzahl der Rechnernetz-
            knoten [NUMCOMP] , Anzahl der Arbeitsrechner [NUMHOST] ,
            Anzahl der Protokollschichten pro Rechnernetzknoten...) * /

      /*    Abschnitt A: MOSAIC-Hauptkomponente "GRAPHISCHE AUFBEREITUNG" */
            external FORTRAN procedure PLOTNE, PLOTQS, PLOTDI;
      /*    A: "GRAPHISCHE AUFBEREITUNG" (Ende)  */

      /*    Abschnitt B: MOSAIC-Hauptkomponente "STATISTISCHE AUSWERTUNG" */
            external procedure MEAN_VALUE, VARIANCE, ...;
      /*    B: "STATISTISCHE AUSWERTUNG" (Ende) */

      SIMULATION begin

            ref (LOADGENERATOR) array RLOADGEN(1:NUMCOMP);
                /*  Verweis auf die Auftragsgeneratoren in den Rechner-
                    netzknoten  */
            ref (COMPUTER) array RCOMP(1:NUMCOMP);
                /*  Verweis auf die Rechnernetzknoten   */
            ref (CHANNEL) array RCHAN(1:NUMCOMP, NUMHOST+1:NUMCOMP);
                /*  Verweis auf die Übertragungskanäle   */


            ....

      /*    Abschnitt C: MOSAIC-Hauptkomponente "MODELLBAUSTEINE"   */
            LINK class DATAUNIT ...;  /*  Dateneinheit   */
            LINK class HEADER ...;    /*  Kontrollinformation  */
```

```
      LINK class REQUEST ...;  /*  Schnittstellenauftrag  */
      PROCESS class CHANNEL ...;  /*  Übertragungskanal  */
      PROCESS class COMPUTER ...;  /*  Rechnernetzknoten  */
      PROCESS class TASK ...;  /*  Task  */
      TASK class USERTASK ...;  /*  Benutzertask  */
      class LAYER ...;  /* Protokollschicht  */
      class PROTUNIT ...;  /*  Protokolleinheit  */
      TASK class SENDPROTMOD ...;  /*  Sendeprotokollmodul  */
      TASK class RECPROTMOD ...;  /*  Empfangsprotokollmodul  */
      PROCESS class TIMER ...;  /*  Zeitüberwachungsauftrag  */
/*    C: "MODELLBAUSTEINE" (Ende)  */

/*    Abschnitt D: MOSAIC-Hauptkomponente "AUFTRAGSGENERATOR"  */
      PROCESS class LOADGENERATOR ...;
         /*  Komponente zur Generierung der Aufträge (Workload)
             innerhalb eines Rechnernetzknotens  */
/*    D:  "AUFTRAGSGENERATOR" (Ende)  */

/*    Abschnitt E: Definition von Prozeduren  */
      procedure GENTASK ...;   /*  Prozedur zur Generierung von
                                   Benutzertasks  */
      procedure GENPMREQ ...;  /*  Prozedur zur Generierung von
                                   Schnittstellenaufträgen  */

....

/*    E:  Prozeduren (Ende)  */

/*    Abschnitt F: MOSAIC-Hauptkomponente "KONFIGURATOR"  */
      procedure INITIALIZE ...;
/*    F:  "KONFIGURATOR" (Ende)  */

/*    Abschnitt G: MOSAIC-Hauptkomponente "MESSEINRICHTUNGEN"  */
      PROCESS class SNAPSHOT ...;   /*  Komponente zur Erfassung
                                        von Meßdaten  */
      PROCESS class PRINTDISC ...;  /*  Komponente zum Abspeichern
                                        von Meßdaten  */
```

```
        procedure TEST ...;   /*  Trace-Prozeduren, z.B. zum Zwecke
                                  der Programmverifikation    */
    /*  G: "MESSEINRICHTUNGEN" (Ende)    */


        /*  Hauptprogramm (Beginn)   */

        ...

        activate new PRINTDISC;    /*  Aktivierung der Meßeinrichtungen   */
        INITIALIZE;                /*  Aufruf des Konfigurators und Ini-
                                       tialisierung (2. Teil)    */
        activate new SNAPSHOT ...;    /*  Aktivierung der Datenerfassung    */
        for I: = 1 step 1 until NUMCOMP do
        begin RLOADGEN(I) : - new LOADGENERATOR ...;
            activate RLOADGEN(I);    /*  Aktivierung des Auftragsgene-
                                         rators im Rechnernetzknoten
                                         RCOMP(I)   */


        end;
        HOLD(SIMTIME);             /*  Durchführung des Simulationsex-
                                       periments für das Zeitintervall
                                       [0,SIMTIME]     */

        /*     Hauptprogramm (Ende)    */

    end of SIMULATION;

end of MOSAIC.
```

3.2 Programmbausteine zur Modellierung der Komponenten eines Rechnernetzes

Die Formulierung der konzeptionellen Teilmodelle eines Rechnernetzes als Warteschlangensysteme und als sequentielle Automaten diente als direkte Grundlage für die Implementierung der Modellbausteine in MOSAIC. Um das Zeitverhalten eines zu untersuchenden Systems A nachzubilden, können dabei sowohl den Zuständen als auch den Transitionen der sequentiellen Automaten Zeiten zugeordnet werden oder - was äquivalent ist - den Warteschlangensystemen können Bedienungszeiten für die Abfertigung von Warteschlangenelementen zugewiesen werden. Diese Zeiten sind im konkreten Anwendungsfall so zu wählen, daß sie der Ausführungsdauer der modellierten Hard- oder Software des zu untersuchenden Systems A entsprechen.

3.2.1. Modellierung hierarchisch organisierter Kommunikationssoftware

Zur Nachbildung der Kommunikationssoftware werden folgende Modellbausteine benötigt:

- die Protokollschichten, die ihrerseits die Protokolleinheiten beinhalten,
- die Protokolleinheiten, die jeweils einen Sende- und Empfangsprotokollmodul enthalten (Zusammenhang der entsprechenden Komponenten in MOSAIC s. Abb. 3.2.),
- die Übertragungskanäle.

(Die nachfolgenden Ausführungen setzen zu einem geringen Teil Grundkenntnisse der höheren Programmiersprache SIMULA voraus).

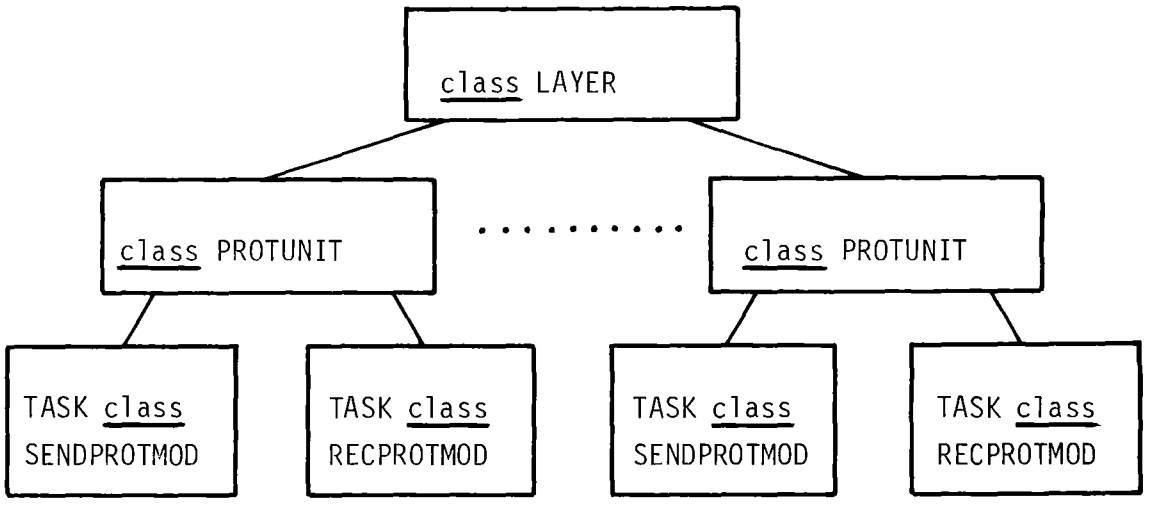


Abb. 3.2.: MOSAIC-Moduln zur Beschreibung von Kommunikationssoftware

Jede der Protokollschichten eines Rechnernetzes wird auf ein Exemplar der class LAYER abgebildet. Die class LAYER besteht ausschließlich aus einer Menge von Variablendeklarationen, die es erlauben, eine Protokollschicht zu charakterisieren, wie z.B.

- Anzahl der Protokolleinheiten in der Schicht,
- Nummer der Schicht, etc.

Jede Protokolleinheit einer Protokollschicht wird auf ein Exemplar der class PROTUNIT abgebildet. Die class PROTUNIT besteht nur aus den Deklarationen von Variablen, die zur Beschreibung von Protokolleinheiten benötigt werden, wie z.B.

- Kennzeichnung des zugrundeliegenden Kommunikationsprotokolls,
- Kennzeichnung der Korrespondenten,
- Verweise auf die Protokollmoduln innerhalb der Protokolleinheit,
- Nummer der Schicht, die die Protokolleinheit enthält, etc.

In der Implementierung von Sende- und Empfangsprotokollmoduln verbirgt sich u.a. das gesamte Kommunikationsprotokoll, nach dessen Regeln die Protokolleinheit mit ihren Korrespondenten kommuniziert. Wegen des hohen Detaillierungsgrads für Kommunikationsprotokolle in MOSAIC ergeben sich somit recht komplexe Implementierungen für die Protokollmoduln / vgl. WOLF80/.

Sende- und Empfangsprotokollmodul besitzen Teile, die unabhängig sind vom konkreten Kommunikationsprotokoll, sowie - zumeist als Prozeduren realisierte - protokollabhängige Teile. Die protokollabhängigen Teile wurden bislang u.a. detailliert für die Kommunikationsprotokolle:

- HDLC ("High level data link control" - Protokoll: balanced class of procedures / s. ISO77/),
- BSC ("Binary synchronous communication" - Protokoll),
- X.25-Level 3 (Paketebene der Protokollempfehlung X.25 /s. CCITT77/),
- MLP ("Message link" - Protokoll des Arbeitskreises PIX/s.HERT78/),
- RJE ("Remote job entry" - Protokoll des Arbeitskreises PIX/s. HEIN78/),

sowie ein

- Basisprotokoll /s. WOLF79b/.

Sowohl Sende- als auch Empfangsprotokollmoduln haben die Eigenschaften einer Task inne. Daher wurden beide Typen von Protokollmoduln als Unterklasse der PROCESS class TASK formuliert.

Ein Übertragungskanal zwischen zwei direkt benachbarten Rechnernetzknoten DVS_1 und DVS_2 wird abgebildet auf ein Exemplar der PROCESS class CHANNEL. Der CHANNEL dient der Modellierung der (niedrigen) Protokollschichten, deren innere Struktur nicht mehr von Interesse ist; hierzu hat der CHANNEL u.a. die Dateneinheitenverzögerung zu berücksichtigen, die beim Datenaustausch zwischen solchen Korrespondenten entsteht, die der tiefsten der modellierten Protokollschichten angehören.

Sowohl in DVS_1 als auch in DVS_2 können die Protokollmoduln der niedrigsten modellierten Schicht Schnittstellenaufträge und Dateneinheiten in die hierfür vorgesehenen Übertragungskanal-Warteschlangen einfügen. Des weiteren kann der Übertragungskanal durch jeden der beiden Rechnernetzknoten DVS_1 und DVS_2 angestoßen, d.h. in den Zustand aktiv versetzt werden. Bei jeder derartigen Aktivierung überprüft der Übertragungskanal, ob die aktuellen Randbedingungen die Bearbeitung eines Schnittstellenauftrags der Übertragungskanal-Warteschlangen erlauben. Falls diese Überprüfung zu einem positiven Resultat führt, wird die in die Übertragung involvierte Dateneinheit durch eine dedizierte Auftragsbearbeitungsinstanz verzögert und anschließend im Zielrechner gemeinsam mit einem Schnittstellenauftrag dem entsprechenden Empfangsprotokollmodul übergeben.

Mit dem Modell des Übertragungskanals sind die Modellbausteine vollständig, die wir zur statischen Beschreibung einer Protokollhierarchie benötigen. Die Elemente, die die Protokollhierarchie durchlaufen (wie z.B. Dateneinheiten, Schnittstellenaufträge) sollen im folgenden eingeführt werden.

3.2.2. Modellierung von Aufträgen in einem Rechnernetz

Um den Auftragsfluß zwischen hierarchisch benachbarten Protokolleinheiten bzw. innerhalb einer Protokollschicht und/oder zwischen Protokolleinheiten und Benutzertasks zu beschreiben (s. Kap. 2.), sind Modelle für Schnittstellen- und Zeitüberwachungsaufträge erforderlich.

Die Schnittstellenaufträge werden durch die LINK class REQUEST repräsentiert; die Notwendigkeit für die Verwendung der Oberklasse LINK ergibt sich daraus, daß Schnittstellenaufträge u.a. als Warteschlangenelemente fungieren und des-

halb die Eigenschaften besitzen sollten, die SIMULA für derartige Elemente voraussetzt. Variablen zur Charakterisierung eines Schnittstellenauftrags sind z.B.

- Auftragstyp,
- Priorität für die Auftragsbearbeitung,
- Verweis auf die Protokolleinheit oder Benutzertask, durch die der Auftrag generiert wurde,
- Kennzeichnung der durch den Schnittstellenauftrag angesprochenen Kommunikationsbeziehung,
- Verweis auf die evtl. in die Auftragsbearbeitung involvierte Dateneinheit, etc.

Die Zeitüberwachungsaufträge können sowohl durch Sende- als auch durch Empfangsprotokollmoduln zur Überwachung zeitkritischer Aktivitäten innerhalb der Protokollmoduln generiert werden; sie werden in der Protokollmodul-Warteschlange TIMEOUTQ abgespeichert. Zeitüberwachungsaufträge wurden als PROCESS class implementiert: dies bietet erstens die Möglichkeit, die Aufträge mit den Eigenschaften von Warteschlangenelementen zu versehen (-PROCESS ist Unterklasse von LINK-),und es erlaubt zweitens, daß sich Zeitüberwachungsaufträge selbst verzögern (und zwar um den Zeitraum, der zu überwachen ist). Durch die Technik der selbständigen Verzögerung von Zeitüberwachungsaufträgen erspart man sich eine spezielle Bedienungsstation zur Abarbeitung dieser Aufträge.

Ein Zeitüberwachungsauftrag wird sofort vernichtet, wenn sich die zu überwachende Aktivität im Protokollmodul ereignet; läuft der Zeitüberwachungsauftrag hingegen ab, ehe die erwartete Aktivität eingetreten ist, so wird ein entsprechender (interner) Schnittstellenauftrag erzeugt und in die Protokollmodul-Warteschlange SREQQ bzw. RREQQ eingefügt.

Beispiel "Überwachung einer Sendeaktivität im Sendeprotokollmodul":
Läuft der Zeitüberwachungsauftrag ab, ohne daß der Korrespondent eine Reaktion gezeigt hat, so wird ein Schnittstellenauftrag (Typ: SEND-COPY) generiert, der anzeigt, daß die erfolgte Übertragung zu wiederholen ist .

Eine weitere wichtige Klasse von Aufträgen, die in einem Rechnernetz zu bearbeiten sind, stellen die Benutzertasks dar. Naturgemäß wird das Modell der Benutzertasks ausschließlich für solche Simulationsstudien benötigt, deren Detaillierungsgrad die Benutzerebene (S_{n+1}) als modellrelevante

Schicht mit in die Betrachtung eines Rechnernetzes einbezieht.

Das Benutzertaskmodell (nähere Einzelheiten s. /WOLF79b/) basiert zunächst auf einer statischen Taskbeschreibung, die im wesentlichen in der Einführung einer Menge sog. interner Zustände und der Vorgabe von Übergängen zwischen diesen Zuständen besteht. Die internen Zustände dienen der Beschreibung der Sequenz der Teilaufträge, aus denen sich die Benutzertask zusammensetzt und die innerhalb "ihres" (Arbeits-) Rechners bzw. durch das gesamte Rechnernetz abgearbeitet werden. Eine zu starke Detaillierung der Struktur einer Benutzertask (etwa bis herab zur Instruktionsebene) erscheint allerdings für unsere Zwecke nicht sinnvoll, und daher wurden als repräsentative interne Zustände z.B. Teilaufträge gewählt wie

- ununterbrechbare CPU-Belegung einer vorgegebenen Zeitdauer,
- unterbrechbare CPU-Belegung einer vorgegebenen Zeitdauer,
- Belegung (lokaler) peripherer Geräte,
- Generierung eines Schnittstellenauftrages an eine Protokolleinheit der höchsten Protokollschicht S_n, u.ä. .

Ausgehend von der (erweiterbaren) Menge der internen Benutzertaskzustände ist eine Beschreibung des dynamischen Verhaltens von Benutzertasks möglich, die anzeigt, welchen internen Zustand eine Task aktuell erreicht hat und welche Rechnerbetriebsmittel sie momentan belegt bzw. als nächstes benötigen wird.

Die Implementierung einer Benutzertask geht auf eine Menge {T0, T1, ..., T6} von dynamischen Benutzertaskzuständen mit folgender Bedeutung zurück:

- T0 : Warten auf Hauptspeicher, keine Betriebsmittel allokiert;
- T1 : Warten auf CPU-Zuteilung, Hauptspeicher allokiert;
- T2 : Hauptspeicher und CPU allokiert, kein CPU-Entzug durch die Betriebsmittelverwaltungsinstanz angezeigt;
- T3 : s. T2, jedoch CPU-Entzug durch Betriebsmittelverwaltungsinstanz bereits angezeigt;
- T4 : Task temporär blockiert, Hauptspeicher allokiert;
- T5 : Warten auf bzw. aktuelle Durchführung von Ein-/ Ausgabe;
- T6 : Task beendet, keine Betriebsmittel allokiert.

Einen Überblick über die möglichen Transitionen zwischen Benutzertaskzuständen gibt Abb. 3.3. Die Bedingungen für die Zustandsübergänge sind zum Teil abhängig vom Typ der Benutzertask.

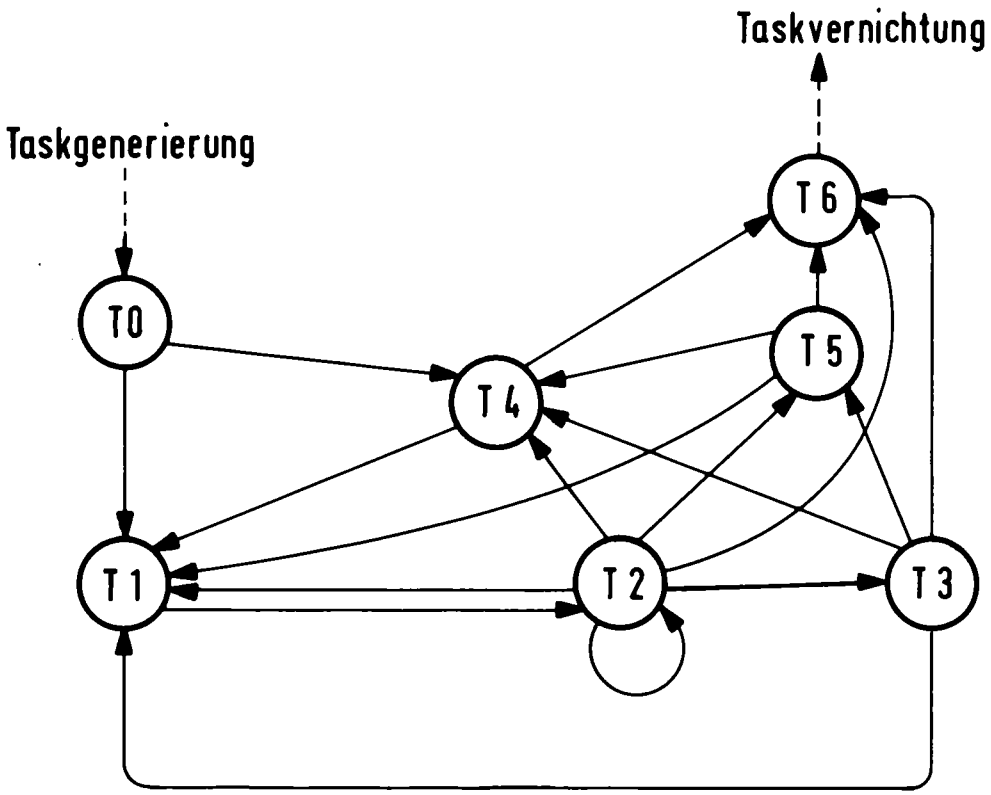


Abb. 3.3.: Mögliche Transitionen zwischen den dynamischen Zuständen einer Benutzertask

Zur Beschreibung ihres zeitlichen Verhaltens erwies es sich als angebracht, eine Benutzertask als PROCESS class zu formulieren, die u.a. Variablen beinhaltet zur Kennzeichnung der Art und des Umfangs der sukzessive benötigten Rechnerbetriebsmittel und (bei kommunizierenden Benutzertasks) Angaben enthält hinsichtlich der beanspruchten Rechnernetzdienste.

Sowohl die Protokollmoduln als auch die Benutzertask wurden als Unterklasse der PROCESS class TASK definiert, wodurch Sende- und Empfangsprotokollmoduln sowie Benutzertasks die Eigenschaften einer Task erhalten, d.h. mit den Taskvariablen versehen werden. Die Taskvariablen dienen insbesondere der Kennzeichnung

- des dynamischen Zustands einer Task,
- der Taskpriorität,
- der Unterbrechbarkeit während einer CPU- Belegung.

3.2.3. Modellierung des Datenflusses in einem Rechnernetz

Der Datenfluß in einem Rechnernetz resultiert aus dem Austausch von Dateneinheiten zwischen hierarchisch benachbarten Protokollmoduln. Die Dateneinheiten können u.a. als das Transporthilfsmittel für die Kontrollinformation angesehen werden, die Korrespondenten zum Zwecke der Kommunikation untereinander austauschen.

Eine Dateneinheit DE = $DE(S_{i-1})$ an der Schnittstelle zwischen den Schichten S_i und S_{i-1} besitzt einen Verweis auf die Kontrollinformation $KI(S_i)$ der Schicht S_i und Verweis auf die Dateneinheit(en) $DE_1, \ldots, DE_k$ höheren Schichten, die evtl. mitgeführt wird (werden) (Beispiel s. Abb. 3.4.). Die Bezeichnung $DE(S_{i-1})$ bzw. $KI(S_i)$ soll die Schichtenabhängigkeit von Dateneinheiten und Kontrollinformation andeuten.

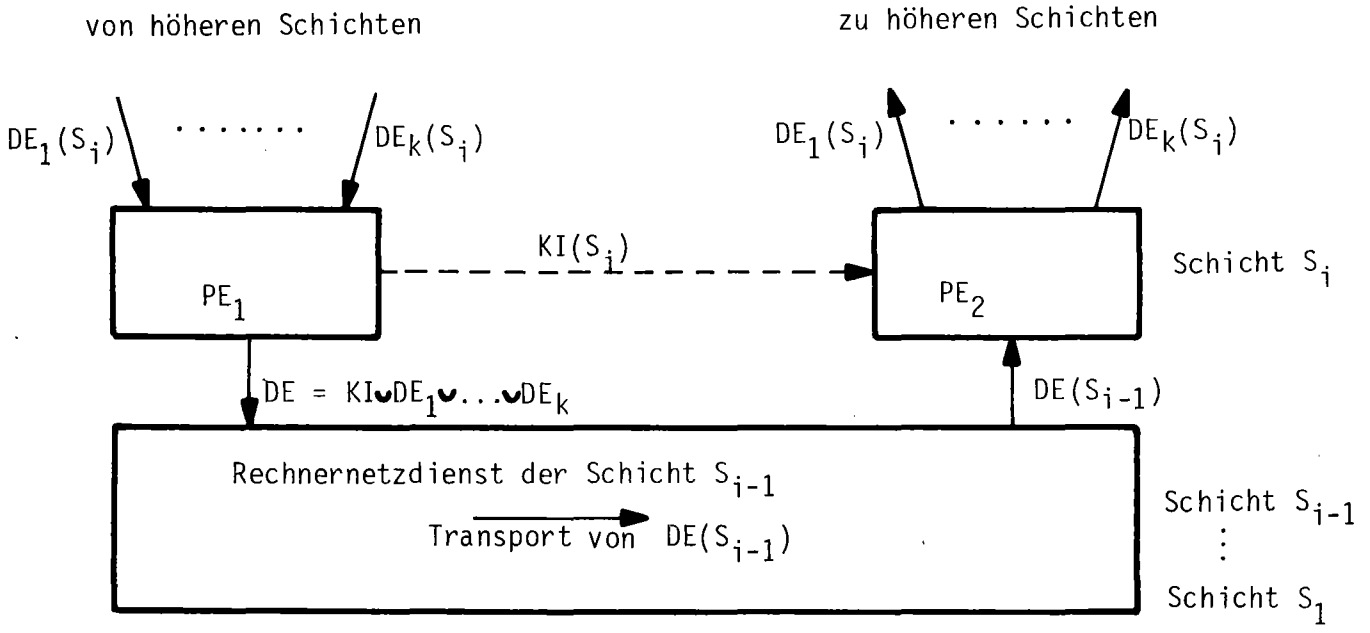


Abb. 3.4.: Datenfluß zwischen zwei korrespondierenden Protokolleinheiten PE_1 und PE_2 (Beispiel einer Blockbildung)

[Notation : ——→ realer Fluß von Dateneinheiten
– – – –→ virtueller Fluß von Dateneinheiten]

Was den Aufbau einer Dateneinheit betrifft, so existiert eine Warteschlange, die dazu dient, die mitgeführte Kontrollinformation sowie die Dateneinheiten höherer Schichten zu speichern. Im Beispiel der Abb.3.4. enthält die Warteschlange der Dateneinheit $DE(S_{i-1})$ die Kontrollinformation $KI(S_i)$ sowie die Dateneinheiten $DE_1(S_i)$, ... , $DE_k(S_i)$ [schematische Darstellung der gegenseitigen Verkettung s. Abb. 3.5.]. Für den Aufbau von DE_1, ... , DE_k läßt sich das für DE Gesagte iterativ anwenden.

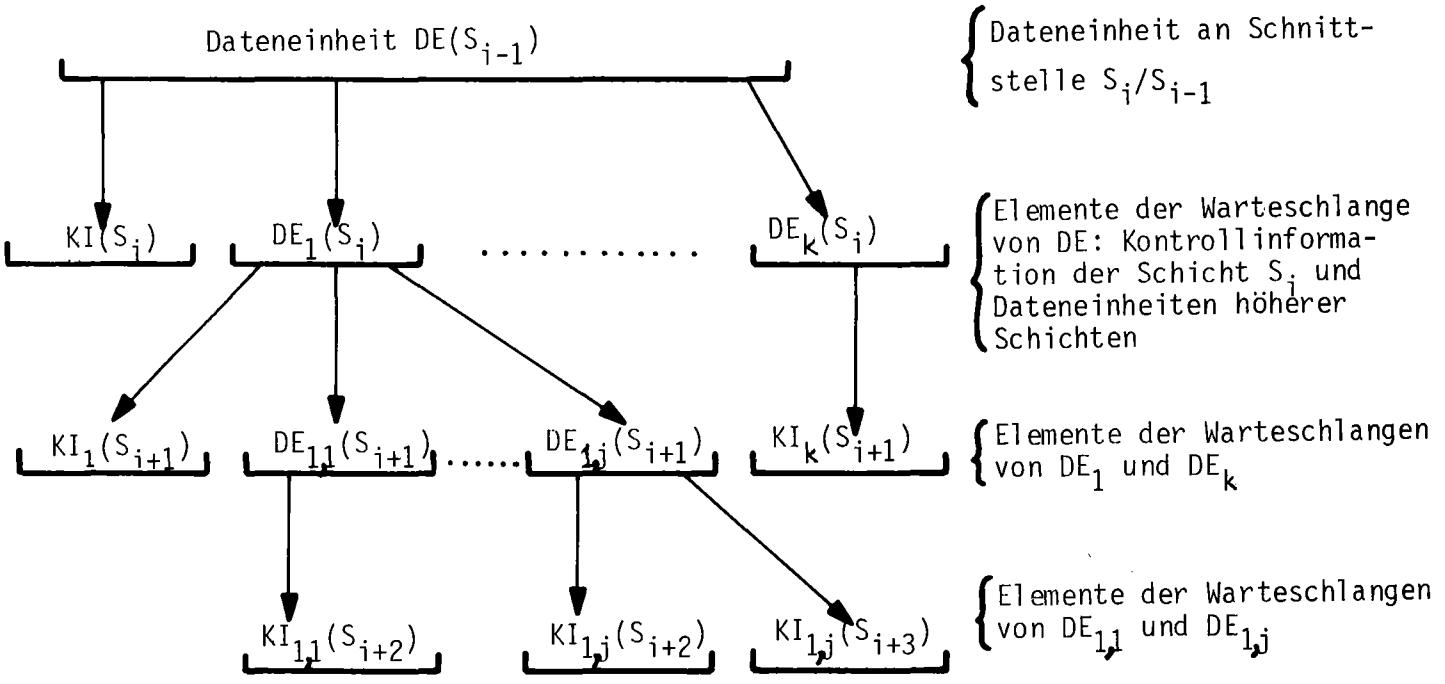


Abb. 3.5.: Beispiel für einen möglichen Aufbau der Dateneinheit DE beim Passieren der Schnittstelle S_i/S_{i-1} [DE s. Abb. 3.4.; Randbedingung : $1 \leq i \leq n-3$, wobei n = Anzahl der Protokollschichten]

Grundsätzlich sind folgende Strukturen einer Dateneinheit möglich:

- Dateneinheiten $DE(S_n)$ an der Schnittstelle zwischen Benutzerebene S_{n+1} und der höchsten Kommunikationsschicht S_n besitzen generell eine leere Warteschlange, da Benutzertasks keine Kontrollinformation untereinander austauschen.
- Der Inhalt der Warteschlange einer Dateneinheit $DE(S_{i-1})$ beim Überschreiten der Schnittstelle zwischen den Schichten S_i und S_{i-1} (für $i \leq n$)besteht im Normalfall aus Kontrollinformation $KI(S_i)$ der

Schicht S_i und den Kontrollinformations-Elementen für sämtliche der durchlaufenen höheren Schichten.

- Bei Zusammenfassung von Dateneinheiten zu einem "Dateneinheitenblock" (Blockbildung für Dateneinheiten s. /WOLF79a/) enthält die Warteschlange von DE Kontrollinformation der Schicht S_i und mehrere vollständige Dateneinheiten höherer Schichten (Abb. 3.5. setzt z.B. Blockbildung bei den Dateneinheiten DE und DE_1 voraus).
- Im Falle der Bildung eines "Dateneinheitenfragmentes" (Fragmentierung für Dateneinheiten s./WOLF79a/)innerhalb der Schicht S_i wird eine Dateneinheit DE' (S_i) einer höheren Schicht in Teilstücke zerlegt. Die Fragmentierung wird derart modelliert, daß die gesamte Warteschlange der Dateneinheit DE' kopiert wird und die Kopien in die Warteschlange einer neu zu generierenden Dateneinheit DE(S_{i-1}) eingefügt werden; die Länge von DE wird dabei entsprechend der Fragmentierung gekürzt. Die Dateneinheit DE(S_{i-1}), die nach der Fragmentierung an die Schicht S_{i-1} übergeben wird, beinhaltet in ihrer Warteschlange die Kontrollinformation KI(S_i) neben den Kopien der Warteschlangenelemente von DE'.
- Bei ausschließlicher Übertragung von Kontrollinformation KI(S_i) durch eine Dateneinheit DE(S_{i-1}) enthält die Warteschlange von DE als einziges Element KI.

Für die Modellierung folgt aus den obigen Ausführungen, daß eine Dateneinheit zum einen ausgestattet sein sollte mit einer Menge von Variablen zur Gesamtbeschreibung der Dateneinheit und zum anderen eine Warteschlange zu besitzen hat, um ein oder mehrere Kontrollinformations-Elemente und evtl. darüber hinaus Dateneinheiten abzuspeichern.

Dateneinheiten sowie Kontrollinformation wurden als LINK class implementiert, da beide als Warteschlangenelemente fungieren.

3.2.4. Modell eines Rechnernetzknotens

Das Modell eines Rechnernetzknotens wird in MOSAIC durch die PROCESS class COMPUTER repräsentiert. Die Hauptaufgabe des COMPUTERs besteht in der Betriebsmittelzuteilung (vgl. Teilmodell der Betriebsmittelverwaltungsinstanz) an diejenigen Tasks, die sich innerhalb des Rechnernetzknotens um Rechnerressourcen bewerben und die in den Rechnerwarteschlangen (s. Warteschlangenmodell der Abb. 2.3.) gespeichert sind. Der COMPUTER trägt wesentlichen Betriebssystemeigenschaften Rechnung durch die Art des Taskmanagements: Das aktuelle Modell erlaubt z.B. eine Betriebsmit-

telzuteilung nach einem Zeitscheibenverfahren oder prioritätsgesteuerte Betriebsmittelvergabe. Daneben lassen sich auch verschiedenartige Hardware-Charakteristika modellieren, wie CPU-Geschwindigkeit und Hauptspeicherausbau. (Die Leistungsfähigkeit einer CPU wird indirekt berücksichtigt durch entsprechende Vorgabe der (CPU-) Bedienzeiten für Tasks).

Ein Rechnernetzknoten besitzt zwei Arten von Verbindungen zu seiner Umgebung:

(a) die Verbindungen zu seinen "Benutzern" (Mensch, technischer Prozeß): die Aufträge, die die Benutzer für einen Rechnernetzknoten darstellen, werden durch den Auftragsgenerator (LOADGENERATOR, s.u.) erzeugt, der dem COMPUTER zugeordnet ist;

(b) die Verbindungen zu den physikalisch angeschlossenen Rechnernetzknoten: die Übertragungsleitung(en), die die physikalische Verbindung eines Rechnernetzknotens zu dem restlichen Rechnernetz herstellen, sind im Modell des CHANNEL enthalten.

Der COMPUTER erhält "Interrupts", d.h. er wird - in SIMULA-Terminologie- aktiviert, wenn

- ein Auftrag zur Bearbeitung erzeugt wurde (z.B. durch den Auftragsgenerator),
- eine Task ihren dynamischen Zustand verändert (d.h. Betriebsmittel freigibt bzw. anfordert),
- eine Eingabe über eine der angeschlossenen Übertragungsleitungen erfolgt ist.

3.2.5. Generierung der Rechnernetzbelastung

Die Umgebung (d.h. die Menge der Benutzer) eines Rechnernetzes generiert Aufträge, die durch einen oder mehrere Rechnernetzknoten zu bearbeiten sind. Im vorliegenden Modell existiert für jeden Rechnernetzknoten genau eine Komponente, der die Auftragserzeugung obliegt: die PROCESS class LOADGENERATOR.

Die Art der durch den LOADGENERATOR zu erzeugenden Aufträge ist abhängig vom Detaillierungsgrad einer Simulationsstudie:

(a) Beschränkt sich die Simulation auf eine Untersuchung der Kommunikationssoftware ohne Miteinbeziehung der in den Rechnernetzknoten ablaufenden Benutzertasks, so obliegt dem Auftragsgenerator die Erzeugung von Schnittstellenaufträgen für die Protokolleinheiten der höchsten modellierten Protokollschicht S_n;

(b) in Simulationsstudien hingegen, die die Benutzerschicht S_{n+1} als modellrelevante Schicht mit in die Betrachtung einbeziehen , handelt es sich bei den erzeugenden Aufträgen um Benutzertasks, welche ihrerseits Schnittstellenaufträge für die Protokolleinheiten der Schicht S_n generieren.

Eine Kombination der Fälle (a) und (b) wollen wir zulassen.

Die Ankunftsrate λ von Aufträgen/msec, die durch einen Auftragsgenerator erzeugt werden, wird gesteuert durch die Vorgabe der mittleren Zwischenankunftszeit MEAN [msec] für Aufträge, wobei gilt: $\lambda = 1/MEAN$. Das Verhältnis für die drei verschiedenen Klassen von Aufträgen "lokale Benutzertasks: kommunizierende Benutzertasks: Schnittstellenaufträge" wird durch Vorgabe von Wahrscheinlichkeiten für jede der Auftragsklassen gesteuert.

3.3. Konfigurierung eines ablauffähigen Simulationsprogramms

Aus den in Abschnitt 3.2. vorgestellten Teilmodellen, die den Aufbau eines vollständigen Rechnernetzmodells erlauben, wird zu Beginn eines Simulationsexperiments - unter Berücksichtigung der Menge der Eingabedaten - ein ablauffähiges Simulationsprogramm erzeugt. Diese Initialisierung obliegt dem Konfigurator (vgl. Abb. 3.1.), der weitgehend auf die Prozedur INITIALIZE abgebildet wurde. Um die Konfigurierung des Gesamtmodells aus den Teilmodellen flexibel zu gestalten, ist es notwendig, eine Reihe von Feldgrenzen (wie z.B. Anzahl der Rechnernetzknoten, der Protokollschichten u.ä.) dynamisch zu definieren. Dies kann erreicht werden durch Hinzunahme eines äußeren Blocks in MOSAIC, der den Simulations-Block beinhaltet.

Somit erhalten wir folgende Struktur des Modellierungssystems MOSAIC aus der Sicht der Modelldefinition:

```
begin /*    VORABINITIALISIERUNG, z.B.
            NUMCTYPE:  Anzahl der Klassen äquivalenter Rechner;
            NUMC(1:NUMCTYPE): Anzahl der Elemente in jeder der
                              Klassen äquivalenter Rechner;
            NUMCOMP:  Anzahl der Rechnernetzknoten;
            NUMLAY(1:NUMCTYPE): Anzahl der Protokollschichten für
                                sämtliche Rechner einer gemeinsamen
                                Rechnerklasse;
            etc.    */
            ...
            SIMULATION begin ...
```

Beispiele für dynamische Feldgrenzen

```
            ref (LOADGENERATOR) array RLOADGEN(1:NUMCOMP);
            ref (COMPUTER) array RCOMP(1:NUMCOMP);
            ...
            procedure INITIALIZE; ...
            INITIALIZE;   /* Aufruf zu Beginn des
                             Hauptprogramms */;
            ...
            end of SIMULATION
end of MOSAIC
```

Die Prozedur INITIALIZE besitzt drei Aufgaben:

(1) die Entgegennahme der Eingabedaten,
(2) den Aufbau des Gesamtmodells,
(3) die Ausgabe einer Kurzbeschreibung des Gesamtmodells zu Kontrollzwecken.

Der Konfigurator wird unterstützt durch die Prozedur PLOTNE, die eine graphische Definition der Rechnernetztopologie erlaubt /WOLF79c/ (zu einem Beispiel s. Anhang B). Eine detaillierte Beschreibung des ursprünglichen E/A-Interfaces von MOSAIC kann Anhang A entnommen werden.

Eine Arbeit im Hinblick auf eine Erweiterung der ursprünglichen graphischen Möglichkeiten bei der Konfigurierung eines Rechnernetzmodells in MOSAIC wurde jüngst abgeschlossen /MÜLL80/. Dieses E/A-Interface, das auf einer Kommandosprache basiert, gestattet dem Experimentator u.a. eine graphische Definition der Protokolleinheiten und insbesondere auch eine Festlegung sämtlicher Kommunikationsbeziehungen zwischen den Kommunikationspartnern in graphischer Form.

3.4. Meßdatenerfassung

3.4.1. Vermessungsgrößen

Zur Interpretation der Simulationsexperimente mit MOSAIC wird eine Vielzahl von Meßwerten erfaßt:

(a) Dateneinheiten:

- o Generierungszeitpunkt der Dateneinheit
- o Lebensdauer der Dateneinheit
- o für die durch Schicht S_i generierte Dateneinheit: Zeitpunkte des Überschreitens der Schnittstelle zwischen den Schichten S_i und S_{i-1}
- o Verweilzeit der (in S_i generierten) Dateneinheit in den Schichten $S_1, \ldots, S_{i-1}$

(b) Kontrollinformation:

- o Generierungszeitpunkt der Kontrollinformation
- o Lebensdauer der Kontrollinformation
- o für die durch Schicht S_i generierte Kontrollinformation: Zeitpunkte des Überschreitens der Schnittstelle zwischen den Schichten S_i und S_{i-1}
- o Verweilzeit der (in S_i generierten) Kontrollinformation in den Schichten $S_1, \ldots, S_{i-1}$

(c) Benutzertasks:

- o Generierungszeitpunkt der Benutzertask
- o bislang absorbierte CPU-Zeit

(d) Rechnernetzknoten:

- o Anzahl der
  - bislang im Rechner generierten Dateneinheiten (ohne Verwendung bereits existierender Dateneinheiten)
  - Dateneinheiten, die bislang in dem Rechnernetzknoten erzeugt wurden (inclusive der Dateneinheiten, die indirekt - d.h. durch Fragmentierung oder Blockbildung aus bereits existierenden Dateneinheiten - generiert wurden)
  - Dateneinheiten, die den Rechnernetzknoten bereits verlassen haben
  - Dateneinheiten, die im Rechnernetzknoten bislang eingetroffen sind

- Dateneinheiten, die durch den Auftragsgenerator erzeugt werden sollten, jedoch mangels Speicher nicht generiert werden konnten
- bislang generierten Benutzertasks
- aktuell im Rechner befindlichen kommunizierenden Benutzertasks
- Benutzertasks, die den Rechnernetzknoten bereits verlassen haben
- Benutzertasks, die aktuell die Protokolleinheiten der höchsten Protokollschicht benutzen
- Größe des aktuell für die Kommunikation bereitstehenden (freien) Speichers
- aktuelle Länge für jede der Betriebsmittelverwaltungsinstanz-Warteschlangen

(e) Sendeprotokollmoduln:

o Anzahl der

- Dateneinheiten, die bislang im Sendeprotokollmodul eingetroffen sind
- Dateneinheiten, die bislang den Sendeprotokollmodul verlassen haben

o Größe des aktuell für die Kommunikation belegten Speichers
o aktuelle Länge für jede der Sendeprotokollmodul-Warteschlangen

(f) Empfangsprotokollmoduln:

o Anzahl der

- Dateneinheiten, die bislang im Empfangsprotokollmodul eingetroffen sind und die ausschließlich Kontrollinformation für die erreichte Schicht enthielten
- Dateneinheiten, die bislang im Empfangsprotokollmodul eingetroffen sind und die nicht nur Kontrollinformation für die erreichte Schicht enthielten
- Dateneinheiten, die bislang den Empfangsprotokollmodul verlassen haben

o Größe des aktuell für die Kommunikation belegten Speichers
o aktuelle Länge für jede der Empfangsprotokollmodul-Warteschlangen
o Summe der Längen sämtlicher Dateneinheiten, die der Empfangsprotokollmodul bislang erhalten hat
o Summe der Längen für die Kontrollinformation, die der Empfangsprotokollmodul bislang erhalten hat

(g) Übertragungskanäle:

- o Größe des aktuell für die Kommunikation bereitstehenden (freien) Ausgabepuffers
- o aktuelle Länge für die beiden Übertragungskanal-Warteschlangen für Schnittstellenaufträge

(h) gesamtes Rechnernetz:

- o Anzahl der
  - bislang im gesamten Rechnernetz an der Schnittstelle zur Protokollschicht S_n generierten Dateneinheiten
  - für die Schicht S_n generierten Dateneinheiten, die sich aktuell im Rechnernetz aufhalten
  - für die Schicht S_n generierten Dateneinheiten, die das Rechnernetz bereits verlassen haben
  - bislang im gesamten Rechnernetz generierten Benutzertasks
  - lokalen Benutzertasks, die das Rechnernetz bereits verlassen haben
  - kommunizierenden Benutzertasks, die das Rechnernetz bereits verlassen haben
- o Gesamtlänge sämtlicher Dateneinheiten, die in der höchsten Protokollschicht durch ihren Empfänger vernichtet wurden

3.4.2. Ergebnisvariablen

Mit Hilfe der Vermessungsgrößen werden bislang folgende Ergebnisvariablen definiert:

(a) Dateneinheiten - Mittelwerte:

für das Gesamtrechnernetz:

- mittlere Reaktionszeit für Dateneinheiten auf Benutzerschicht (unter der Reaktionszeit sei das Zeitintervall vom Absenden einer Dateneinheit bis zum Erhalt der entsprechenden Antwort verstanden)
- mittlere Verweilzeit im Rechnernetz
- durchschnittliche Anzahl der (für die höchste Schicht S_n generierten) Dateneinheiten im Rechnernetz
- mittlere Länge der (für die Schicht S_n generierten) Dateneinheiten
- erzielter Durchsatz von Dateneinheiten der höchsten Schicht S_n

für jede Protokollschicht S_i:

- erzielter Durchsatz von (in S_i generierten und vernichteten) Dateneinheiten

für jede Protokolleinheit:

- mittlere Verweilzeit in den Protokollschichten S_1, ..., S_{i-1} für sämtliche Dateneinheiten, die eine Protokolleinheit der Schicht S_i von ihrem/ihren Korrespondenten bislang erhielt
- mittlere Lebensdauer für die Dateneinheiten, die eine Protokolleinheit der Schicht S_i von ihrem/ihren Korrespondenten bislang erhielt (erfaßt für die Menge der Dateneinheiten, die nicht ausschließlich Kontrollinformation für die Schicht S_i beinhalten)
- erzielter Datendurchsatz (bezogen auf die Dateneinheiten, die eine Protokolleinheit von ihrem/ihren Korrespondenten erhielt) [der Anteil der Dateneinheiten, die ausschließlich Kontrollinformation für die aktuell erreichte Schicht beinhalten, am Gesamtdurchsatz wird getrennt erfaßt (s.u.)]

(b) Kontrollinformation-Mittelwerte:

für jede Protokolleinheit:

- mittlere Verweilzeit in den Protokollschichten S_1, ..., S_{i-1} für die gesamte Kontrollinformation, die eine Protokolleinheit der Schicht S_i von ihrem/ihren Korrespondenten bislang erhielt
- mittlere Lebensdauer für die Dateneinheiten, die eine Protokolleinheit der Schicht S_i von ihrem/ihren Korrespondenten bislang erhielt (erfaßt für die Menge der Dateneinheiten, die ausschließlich Kontrollinformation für die Schicht S_i beinhalten)
- erzielter Durchsatz (bezogen auf die Kontrollinformation, die eine Protokolleinheit von ihrem/ihren Korrespondenten erhielt)

(c) Benutzertask-Mittelwerte:

für das Gesamtrechnernetz:

- mittlere Verweilzeit für die bislang vernichteten lokalen Benutzertasks
- mittlere Verweilzeit für die bislang vernichteten kommunizierenden Benutzertasks

(d) Aktivzeiten:

Summe aller Aktivzeiten für die Bedienstationen des Rechnernetzes seit Beginn des Simulationsexperiments

(e) Auslastungen:

Man erhält die Auslastungen sofort aus der Gleichung

Auslastung := Aktivzeit/Gesamtzeit

In einfacher Weise lassen sich bei Bedarf weitere Ergebnisvariablen aus den Vermessungsgrößen ableiten .

3.5. Datenauswertung

Die Datenauswertung für Simulationsexperimente mit dem Modellierungssystem MOSAIC umfaßt:

(1) die graphische Aufbereitung der ermittelten Experimentresultate sowie
(2) die statistische Auswertung der Vermessungsgrößen.

Für die graphische Resultataufbereitung beinhaltet MOSAIC die FORTRAN-Subroutines PLOTQS und PLOTDI, die die Benutzung eines TEKTRONIX-Storage-displays (z.B. T4012, T4014) voraussetzen. Die Subroutine PLOTQS erlaubt es, den Datenfluß im Rechnernetzmodell graphisch darzustellen; PLOTDI dient dazu, den zeitlichen Verlauf von Experimentresultaten aufzuzeichnen (zu einem Beispiel s. Anhang B). Diese beiden graphischen Komponenten sind in erster Linie dazu vorgesehen, die Interpretierbarkeit und das Verständnis von Simulationsexperimenten mit MOSAIC zu beschleunigen und zu erleichtern. Eine detailliertere Beschreibung der aktuellen graphischen MOSAIC-Komponenten wird in /WOLF79c/ gegeben.

Zur statistischen Auswertung von Vermessungsgrößen wurden die in /DROB74/ erläuterten - in SIMULA implementierten - Prozeduren in MOSAIC eingebettet. Diese Prozeduren zur Stichprobenauswertung lassen sich nach Bedarf zur Berechnung von Mittelwerten, Varianzen, Vertrauensintervallen u.ä. einsetzen.

4. Abbildung des BERNET-Rechnernetzes auf MOSAIC

Im Rahmen eines Kooperationsvertrages zwischen dem Kernforschungszentrum Karlsruhe (Institut für Datenverarbeitung in der Technik) und dem Land Berlin wird das im ersten Teil dieser Arbeit vorgestellte Modellierungssystem MOSAIC aktuell dazu herangezogen, eine detaillierte Simulation der Kommunikationsflüsse im BERNET-Rechnernetz durchzuführen.

Das BERNET-Rechnernetz wird zur Zeit in Berlin aufgebaut und verbindet die Großrechner einiger Berliner wissenschaftlicher Rechenzentren über ein X.25-Datenpaketvermittlungsnetz der DBP untereinander. Bei dem Paketvermittlungsnetz (BERPEX) handelt es sich um einen SL10-Vermittlungsrechner aus dem DATAPAC-Netz /CUNN77/. Für den Anschluß der Großrechner an das Paketvermittlungsnetz werden Prozeßrechner vom Typ AEG 80-40 als Netzeingangsprozessoren (NEP'en) verwendet. Zu einer Systemanalyse des BERNET-Rechnernetzes sei auf /STRA78a/,/STRA78b/ verwiesen.

Eines der wichtigsten Probleme bei der aktuellen Implementierung des BERNET-Rechnernetzes ist die Dimensionierung der Kommunikationsprotokolle und der Netzeingangsprozessoren (z.B. optimale Festlegung von Protokollparametern, von Speicherbereichen etc.). Um diesbezügliche Aussagen zu erhalten, wird ein Simulator benötigt, der eine integrierte Betrachtungsweise für die gesamte Hierarchie der Kommunikationsprotokolle erlaubt und überdies - zumindest für die Netzeingangsprozessoren - auch die Betriebsmittelzuteilung innerhalb der Rechnernetzknoten modelliert.

Ausgehend von dem MOSAIC-Grundmodell wurde durch entsprechende Festlegung der Eingabedaten und durch Modellerweiterungen (die Hauptarbeit bestand hier in der Anpassung der im Grundmodell implementierten Kommunikationsprotokolle an die im BERNET verwendeten Protokolle) eine MOSAIC-Version erzeugt, die auf die Leistungsanalyse des BERNET-Rechnernetzes zugeschnitten ist. Das resultierende Modellierungssystem soll als MOSAIC(BERNET) bezeichnet werden.

Im folgenden geht es nun darum, einige wichtige Eigenschaften des Modellierungssystems MOSAIC(BERNET) zusammenzustellen.

4.1. Modellierung der physikalischen Charakteristika des BERNET

4.1.1. Rechnernetztopologie

Bereits das MOSAIC-Grundmodell kann zur Modellierung beliebiger Rechnernetztopologien herangezogen werden, und dies gilt auch für MOSAIC(BERNET). Die einzige Restriktion, die z.B. der Untersuchung allzu umfangreicher Netztopologien im Wege steht, kann der Speicherplatz des Rechners darstellen, auf welchem die Simulation abläuft. Der für die Durchführung eines Simulationsexperiments benötigte Speicherplatz ist nämlich offensichtlich direkt abhängig (~ proportional) von der Anzahl der Rechnernetzknoten im Simulationsmodell.

Die Information hinsichtlich der Rechnernetztopologie ist in MOSAIC in sogenannten "Routing-Tabellen" abgespeichert, die in den Rechnernetzknoten geführt werden und die anzeigen, über welche(n) Vermittlungsrechner ein gegebener Zielrechner erreicht werden kann. Der aktuell implementierte Routing-Algorithmus berücksichtigt die "minimum number of hops" Strategie, d.h. der nächste Vermittlungsrechner wird derart ausgewählt, daß die Anzahl der zu durchlaufenden Rechnernetzknoten bis zum Zielrechner minimal wird. Existieren mehrere "gleichlange" Wege, so wird jeder der unterschiedlichen Wege mit derselben Wahrscheinlichkeit ausgewählt. Ein dynamisches Updating der Routing-Tabellen z.B. im Falle von Leitungsausfällen ist möglich. Bei graphischer Definition des Rechnernetzmodells (s. /WOLF79c/, /MÜLL80/) werden die Routing-Tabellen automatisch erzeugt und jedem Rechnernetzknoten zugeordnet.

Eine Klasse möglicher Topologien für das BERNET-Rechnernetz ist in Abb. 4.1. dargestellt. Insbesondere wird ersichtlich, daß Routing-Fragen für das BERNET aktuell unbedeutend sind.

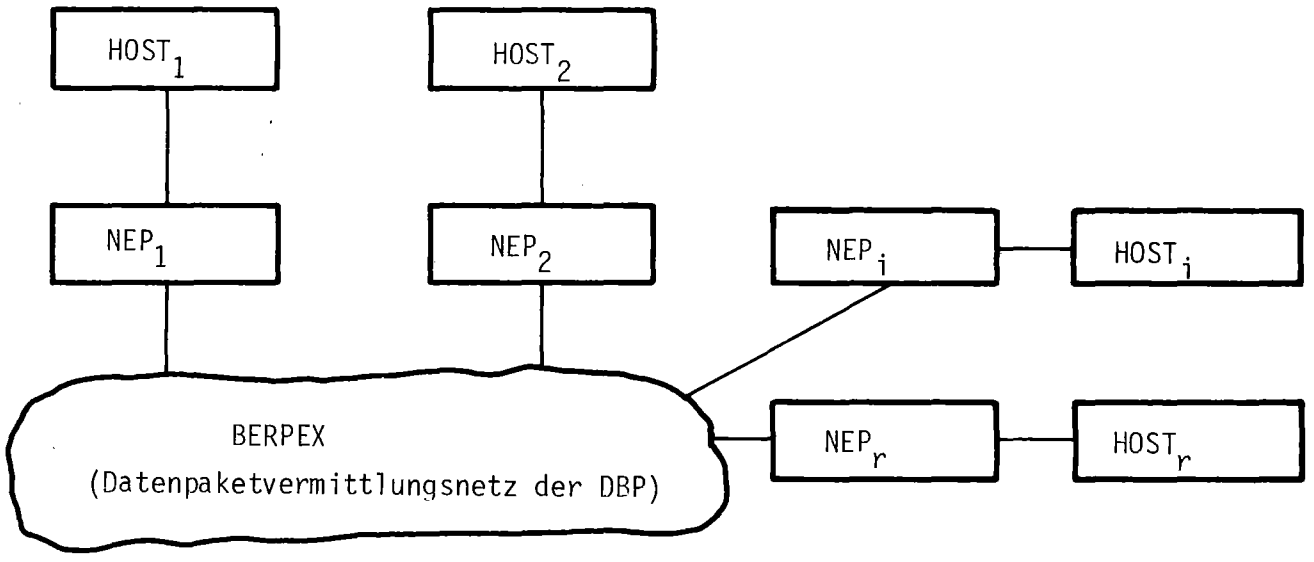


Abb. 4.1.: BERNET-Konfiguration mit r Arbeitsrechnern $HOST_1$, ..., $HOST_r$ (für die erste Ausbaustufe ist $r \leq 5$)

4.1.2. Hardwareeigenschaften der Rechnernetzknoten

Was die Hardwareeigenschaften betrifft, so wurde in erster Linie der Umfang des zur Verfügung stehenden (Kommunikations- und Haupt-)Speichers sowie die Leistungsfähigkeit der CPU berücksichtigt.

(a) Speicher:

(1) Kommunikationsspeicher:

In jedem Rechnernetzknoten kann optional die Verwaltung eines festen - für die Kommunikation reservierten - Speicherbereichs durchgeführt werden.
(Die Verwaltung eines Kommunikationsspeichers ist insbesondere für die Netzeingangsprozessoren relevant, s. "Buffermanager" im NEP).

(2) Hauptspeicher:

Die Speicherzuteilung an Tasks, die nicht Hauptspeicher-resident gehalten werden können, wird dadurch modelliert, daß einerseits den Tasks ein fest vorgebbarer Overhead für das Ein- bzw. Auslagern zugeschlagen wird und andererseits das I/O-Verhalten von

Tasks in geeigneter Weise geändert wird.
(Die Berücksichtigung der Hauptspeicherzuteilung ist notwendig, um das Paging-Verhalten von Netzeingangsprozessoren nachzubilden).

(b) CPU:

Die CPU-Geschwindigkeit (z.B. gegeben in [mittlere Anzahl von bearbeiteten Instruktionen pro Sekunde]) geht ein in die Zeiten, die für die Ausführung von Software vorgegeben werden.

Beispiel: Ein Softwaremodul S läuft auf einem Rechner R ab, und die Ausführungsdauer (CPU-Bedienzeit) für S auf dem Rechner R soll ermittelt werden, da eine Zeitmessung nicht möglich sei.

Dabei werde vorausgesetzt,

- S umfaßt K_1 Instruktionen,
- R führt im Mittel K_1 Instruktionen pro sec aus.

In diesem Fall ergibt sich die Ausführungsdauer A für S auf R näherungsweise aus $A = K_1/K_2$ [sec] .

Das Beispiel zeigt u.a., daß c-fache CPU-Geschwindigkeit identisch in das Modell eingeht wie eine Reduktion der Anzahl der auszuführenden Instruktionen um den Faktor c.(Die angedeutete Methode, die Ausführungsdauer von Softwaremoduln abzuschätzen, kann in den Fällen sehr wertvoll sein, in denen entsprechende Meßdaten nicht verfügbar sind!).

4.1.3. Übertragungsleitungen

Die Teilmodelle für die in BERNET verwendeten Übertragungsleitungen können durch geeignete Wahl der zur Verfügung stehenden Parameter direkt aus dem in MOSAIC enthaltenen Grundmodell eines Übertragungskanals erzeugt werden. Bereits durch eine derartige Parametervariation ist es möglich, sowohl die Vollduplex-Leitungen zwischen NEP'en und BERPEX mit Geschwindigkeiten von 1.2, 2.4, ..., 48.0 [KBit/sec] nachzubilden als auch die schnellen Kanäle zwischen den Arbeitsrechnern und den Netzeingangsprozessoren zu modellieren. Darüber hinaus kann auch die Zuverlässigkeit des Übertragungsmediums parametergesteuert variiert werden.

4.2. Modellierung der Betriebssystemcharakteristika der Rechner des BERNET

Was die verschiedenen Betriebssysteme in den BERNET-Rechnernetzknoten betrifft, so wurde versucht, in erster Linie das Taskmanagement in den Rechnern nachzubilden. Zur Modellierung der Betriebsmittelzuteilung an Tasks besitzt ein Benutzer des Modellierungssystems MOSAIC(BERNET) z.B. folgende Alternativen für Dispatcherstrategien:

- eine reine prioritätsgesteuerte CPU-Zuteilung an Tasks (s. beispielsweise Dispatcherstrategie des Betriebssystems MARTOS /AEG/ der AEG 80-40),
- eine CPU-Zuteilung nach einem Zeitscheibenverfahren mit beliebig einstellbarer Zeitscheibengröße, wobei Unterbrechungen von Tasks während einer Zeitscheibe mit einer gegebenen Wahrscheinlichkeit erlaubt sind (s. beispielsweise Dispatcher-Strategie im Betriebssystem BS 2000 /SIEM/ von SIEMENS-Großrechnern).

Für das Interrupthandling wird vorausgesetzt, daß der CPU-Zeitbedarf für die Bearbeitung von Interrupts akkumuliert und immer dann berücksichtigt wird, wenn die Betriebsmittelverwaltungsinstanz über die Zuteilung von CPU entscheidet.

Der zeitliche Overhead, der durch die Aktionen des Betriebssystems rechnerintern verursacht wird, kann dadurch in das Modell eingebracht werden, daß man

- eine lokale Benutzertask hoher Priorität während des gesamten Simulationsexperiments auf dem entsprechenden Rechner ablaufen läßt, oder
- den Betriebssystemoverhead direkt in der Betriebsmittelverwaltungsinstanz berücksichtigt (z.B. den Entscheidungszyklus einer Betriebsmittelverwaltungsinstanz mit entsprechendem CPU-Bedarf versieht).

4.3. Modellierung der Kommunikationssoftware des BERNET

4.3.1. Gesamtaufbau der Protokollhierarchie

Als Organisationsprinzip für die Kommunikationssoftware und -hardware wurde in BERNET - wie allgemein gebräuchlich - eine hierarchische Struktur gewählt. Dies impliziert, daß eine Hierarchie von Rechnernetzdiensten existiert, wie sie bereits bei der Einführung des Modellierungssystems MOSAIC in Kapitel 3 vorausgesetzt wurde. Jeder Schicht S_i können ein oder mehrere Kommunikationsprotokolle zugeordnet werden, unter deren Berücksichtigung der (die) Dienst(e) der Schicht S_i bereitgestellt wird (werden). Als Kommunikationsprotokolle werden in BERNET weitgehend Protokollempfehlungen verschiedener Normungsgremien (ISO, CCITT, PIX) verwendet.

Für nachfolgende Betrachtungen ist es sinnvoll, die Kommunikationsprotokolle des BERNET, die in MOSAIC(BERNET) Berücksichtigung fanden, vier Protokollschichten S_1, ..., S_4 zuzuordnen. (Eine Darstellung dieser rechnerspezifischen Zuordnung der Kommunikationsprotokolle auf die einzelnen Schichten - incl. physikalische Schicht S_0 und Benutzerschicht S_5 - ist durch Tabelle 4.2. gegeben.)

Die Modellierung einer Hierarchie von Kommunikationsprotokollen in MOSAIC erfolgt auf der Basis von Protokolleinheiten. Korrespondierende Protokolleinheiten derselben Schicht in verschiedenen Rechnernetzknoten kommunizieren unter Berücksichtigung eines gemeinsamen Kommunikationsprotokolls miteinander und stellen einen Rechnernetzdienst bereit; benachbarte Protokolleinheiten desselben Rechnernetzknotens kommunizieren über eine gemeinsame Schichtenschnittstelle. Die korrespondierenden Protokolleinheiten einer Schicht werden in einer sogenannten PUCLASS zusammengefaßt.

Bei der Definition der Protokollhierarchie eines Rechnernetzes steht die Vorgabe der Korrespondenten- und Nachbarbeziehungen somit im Vordergrund, wobei eine Teilaufgabe in der Festlegung der PUCLASS'es besteht. Um diese Arbeit für die Modellierung des BERNET zu vereinfachen, zeigt Abb. 4.3. die Gesamtsicht (Protokolleinheitensicht) des BERNET, aus der die Nachbar- und Korrespondentenzuordnung ersichtlich wird (incl. eines Vorschlags für ein Numerierungsschema der PUCLASS'es: HDLC wird abgewickelt in PUCLASS 102,...,106; X.25(L3) in PUCLASS 202,...,206; MLP in PUCLASS 301; CCP in PUCLASS 101,107,...,110; SCP in PUCLASS 201,207,...,210; RJEP in PUCLASS 401 sowie FTP in PUCLASS 402).

Tabelle 4.2.: Hierarchie der Kommunikationsprotokolle im BERNET-Rechnernetz

[Notation : CCP: Channel Communication-Protokoll
HDLC: HDLC-Protokoll
SCP: Subchannel-Protokoll
X.25(L3): X.25-Protokoll (Level 3)
MLP: Message Link-Protokoll
FTP: File Transfer-Protokoll
RJEP: Remote Job Entry-Protokoll
BT: kommunizierende Benutzertasks]

Rechnertyp / Schicht	Arbeitsrechner	Netzeingangs-prozessor	BERPEX
Benutzerschicht S_5	BT	-	-
Anwendungsorientierte Schicht S_4	FTP, RJEP	FTP, RJEP	-
Transportschicht S_3	-	MLP	-
Netzschicht S_2	SCP	SCP, X.25(L3)	X.25(L3)
Leitungsschicht S_1	CCP	CCP, HDLC	HDLC
phys. Schicht S_0	CHANNEL	CHANNEL	CHANNEL

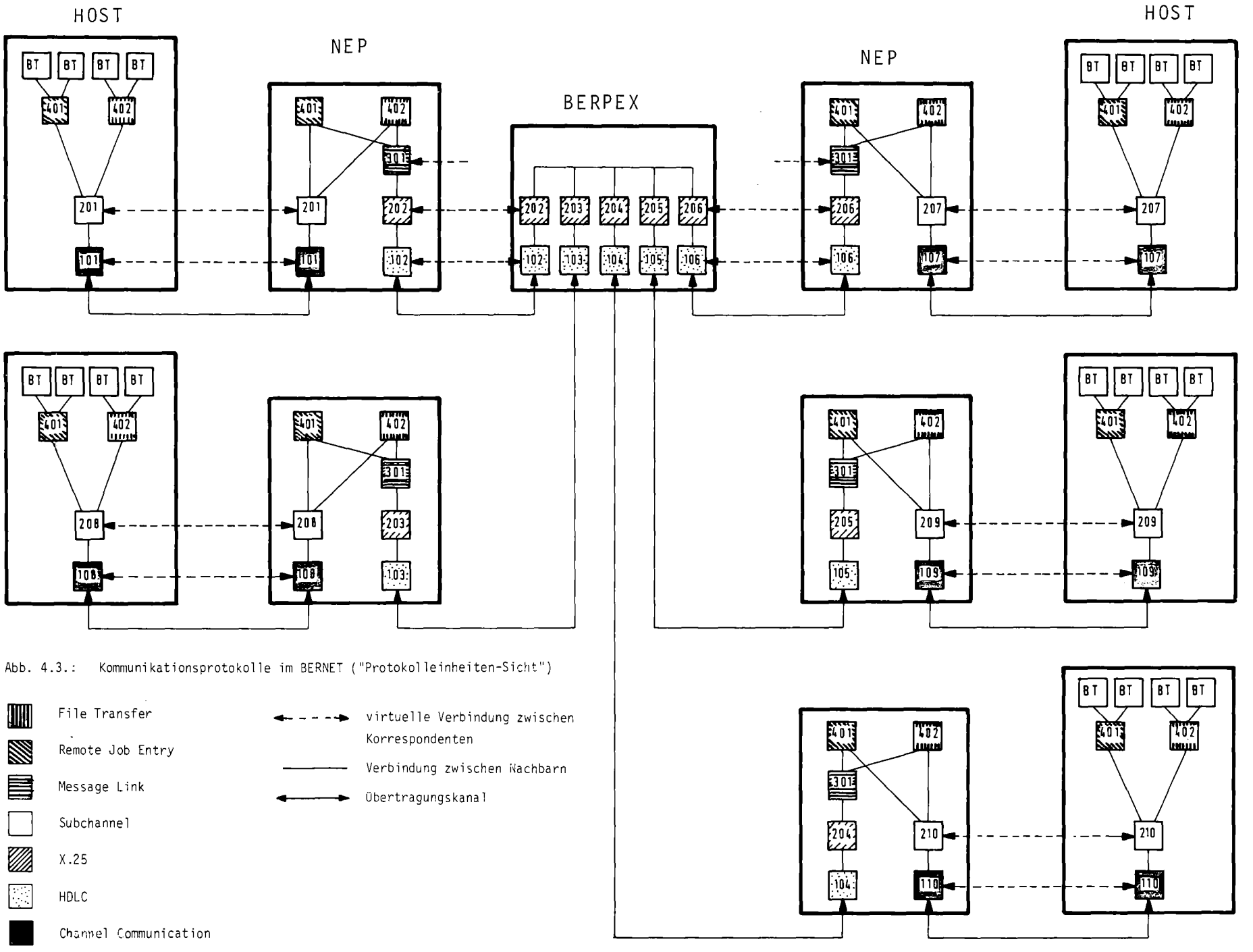


Abb. 4.3.: Kommunikationsprotokolle im BERNET ("Protokolleinheiten-Sicht")

Der an weiteren Einzelheiten hinsichtlich des Aufbaus der BERNET-Protokollhierarchie interessierte Leser sei auf Anhang B verwiesen; anhand eines Beispieles ist die Modellierung der Protokollhierarchie des BERNET in MOSAIC dort ausführlich dargestellt.

4.3.2. Die modellierten Kommunikationsprotokolle und die Schichtenschnittstellen

Die in MOSAIC(BERNET) enthaltenen Modelle für Kommunikationsprotokolle werden in der Folge unter Verwendung eines einheitlichen Beschreibungsschemas dargestellt.

4.3.2.1. Basisprotokoll

(1) Literatur:

Das Basisprotokoll, das als Grundlage für die verschiedenen Protokollimplementierungen diente, wurde in /WOLF79b/ eingeführt.

(2) Benötigte Schnittstellenaufträge:

Falls zwei Korrespondenten PE und PE' innerhalb der Schicht S_i unter Benutzung des Basisprotokolls miteinander kommunizieren, dann werden Schnittstellenaufträge folgender Typen generiert und interpretiert, um mit den benachbarten Schichten zu kommunizieren:

- OPEN CONNECTION [Kurzform: OPEN] mit den Parametern CORR und USER, der als Aufforderung dient, eine (logische) Kommunikationsbeziehung aufzubauen, über welche der Benutzer (Nachbar) USER mit seinem Korrespondenten CORR anschließend kommunizieren kann.
- CLOSE CONNECTION [Kurzform: CLOSE] mit den Parametern CORR und USER, der als Aufforderung dient, eine existierende (logische) Kommunikationsbeziehung abzubauen, über welche der Benutzer USER mit seinem Korrespondenten CORR bislang kommunizieren konnte.

- SEND DATAUNIT [Kurzform: SEND-DE] mit den Parametern CORR, USER und DU, der als Aufforderung dient, eine Dateneinheit DU für den Benutzer USER zu einem seiner Korrespondenten CORR weiterzuleiten.
- RECEIVE DATAUNIT [Kurzform: REC-DE] mit den Parametern CORR, USER und DU, der dazu dient, der benachbarten Protokolleinheit CORR die Ankunft der Dateneinheit DU anzuzeigen, die durch die Protokolleinheit USER generiert und abgesandt wurde.

Zu jedem der Schnittstellenaufträge gibt es mit OPEN RESP,CLOSE RESP, SEND-DE RESP und REC-DE RESP die entsprechende Rückmeldung.

(3) Schnittstelle, die der nächsthöheren Schicht S_{i+1} angeboten wird:

Die Schnittstellenaufträge, die von den Protokolleinheiten der Schicht S_{i+1} abgesetzt, d.h. an die Protokolleinheiten der Schicht S_i übergeben werden, sind:

OPEN, CLOSE, SEND-DE und REC-DE RESP.

Die Schnittstellenaufträge, die von den Protokolleinheiten der Schicht S_i an die nächsthöhere Schicht übergeben werden, sind:

REC-DE, OPEN RESP, CLOSE RESP und SEND-DE RESP

(s. hierzu Abb. 4.4.)

(4) Schnittstelle, die von der nächsttieferen Schicht S_{i-1} vorausgesetzt wird:

Die Schnittstelle zwischen PE bzw. PE' und der Schicht S_{i-1} basiert genau auf denselben Schnittstellenaufträgen wie die Schnittstelle zur Schicht S_{i+1}.

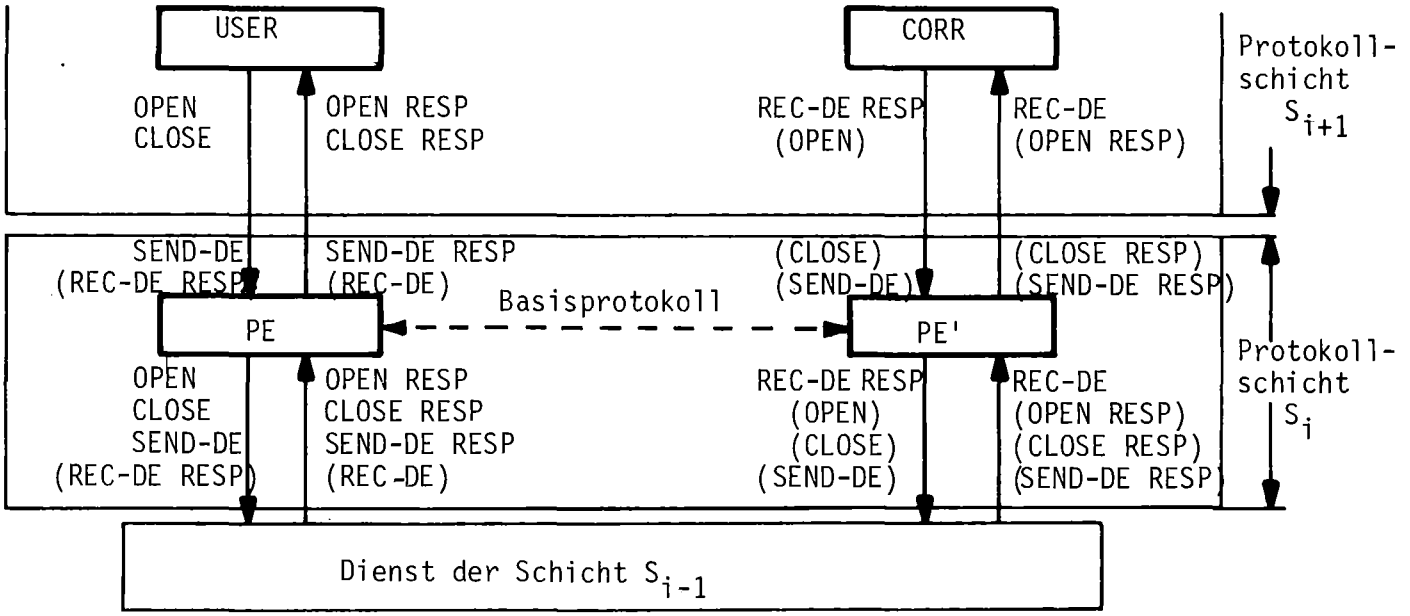


Abb. 4.4.: Schnittstelle, die im Falle des Basisprotokolls der nächsthöheren Schicht angeboten wird
[Schnittstellenaufträge ohne Klammern bei Initiative durch die Protokolleinheit USER; Schnittstellenaufträge in Klammern bei Initiative durch CORR]

(5) Benötigte Dateneinheiten:

Folgende Typen von Dateneinheiten werden für die Kommunikation zwischen PE und PE' benötigt:

- Dateneinheit, die nur Daten der Schicht S_{i+1} mitführt [de] (Typ = 1)
- positive Quittung / keine Daten der Schicht S_{i+1} werden mitgeführt [ack] (Typ = 2)
- negative Quittung / Keine Daten für Schicht S_{i+1} [nak] (Typ = 3)
- positive Quittung und Daten für Schicht S_{i+1} [ack & de] (Typ = 4)
- negative Quittung und Daten für Schicht S_{i+1} [nak & de] (Typ = 5)
- Initialisierungsaufforderung an den Korrespondenten [init] (Typ = 6)
- Antwort für eine Initialisierungsaufforderung [init resp] (Typ = 7)
- Terminierungsaufforderung an den Korrespondenten [term] (Typ = 8)
- Antwort für eine Terminierungsaufforderung [term resp.] (Typ = 9)

(6) Aktionen einer Protokolleinheit (PE bzw. PE') des Basisprotokolls:

(a) Erhalt von OPEN durch einen Nachbarn USER (s. Abb. 4.5.):

- Verbindung zu dem entsprechenden Korrespondenten PE' über die nächsttiefere Schicht S_{i-1} wird hergestellt;
- dem Korrespondenten PE' wird eine Initialisierungsaufforderung (init) übermittelt;
- nach Erhalt einer Antwort (init resp.) von PE' wird ein Schnittstellenauftrag OPEN RESP an die Protokolleinheit USER übergeben, durch die der ursprüngliche OPEN-Schnittstellenauftrag generiert worden war.

(b) Erhalt von CLOSE durch den Nachbarn USER (s. Abb. 4.6.):

- dem Korrespondenten wird eine Terminierungsaufforderung (term) übermittelt;
- nach Erhalt einer Antwort (term resp.) von PE' wird die existierende Verbindung zu PE' in der nächsttieferen Schicht S_{i-1} abgebaut;
- dem Nachbarn USER wird ein CLOSE RESP-Schnittstellenauftrag übergeben.

(c) Erhalt von SEND-DE durch den Nachbarn USER (s. Abb. 4.7.):

- die referenzierte Dateneinheit (d) wird an den entsprechenden Korrespondenten PE' über die nächsttiefere Schicht S_{i-1} übertragen (Fragmentierung, Blockbildung und Flußkontrolle unter Benutzung eines "Window"-Mechanismus /WOLF79b/ sind optional möglich);
- dem Nachbarn USER wird nach Beendigung des Sendevorgangs ein SEND-DE RESP übergeben.

(d) Erhalt von REC-DE durch einen Nachbarn der nächsttieferen Schicht S_{i-1} (s. Abb. 4.7.):

- die referenzierte Dateneinheit (d), die durch S_{i-1} an PE' übergeben wurde, ist zu interpretieren und gegebenenfalls

an die Protokolleinheit CORR der nächsthöheren Schicht weiterzuleiten (Reassemblierung, Entblocken - d.h. Auftrennen eines Dateneinheitenblocks in die ursprünglichen Dateneinheiten - und Quittungserstellung unter Benutzung eines "Window"-Mechanismus /WOLF79b/ sind optional möglich);

- dem Nachbarn der nächsttieferen Schicht wird für jede erhaltene Dateneinheit ein REC-DE RESP übergeben.

(7) Zeitliches Verhalten:

Das zeitliche Verhalten bei der Bearbeitung der Schnittstellenaufträge OPEN, CLOSE, SEND-DE und REC-DE kann den Abb. 4.5. - 4.7. entnommen werden.

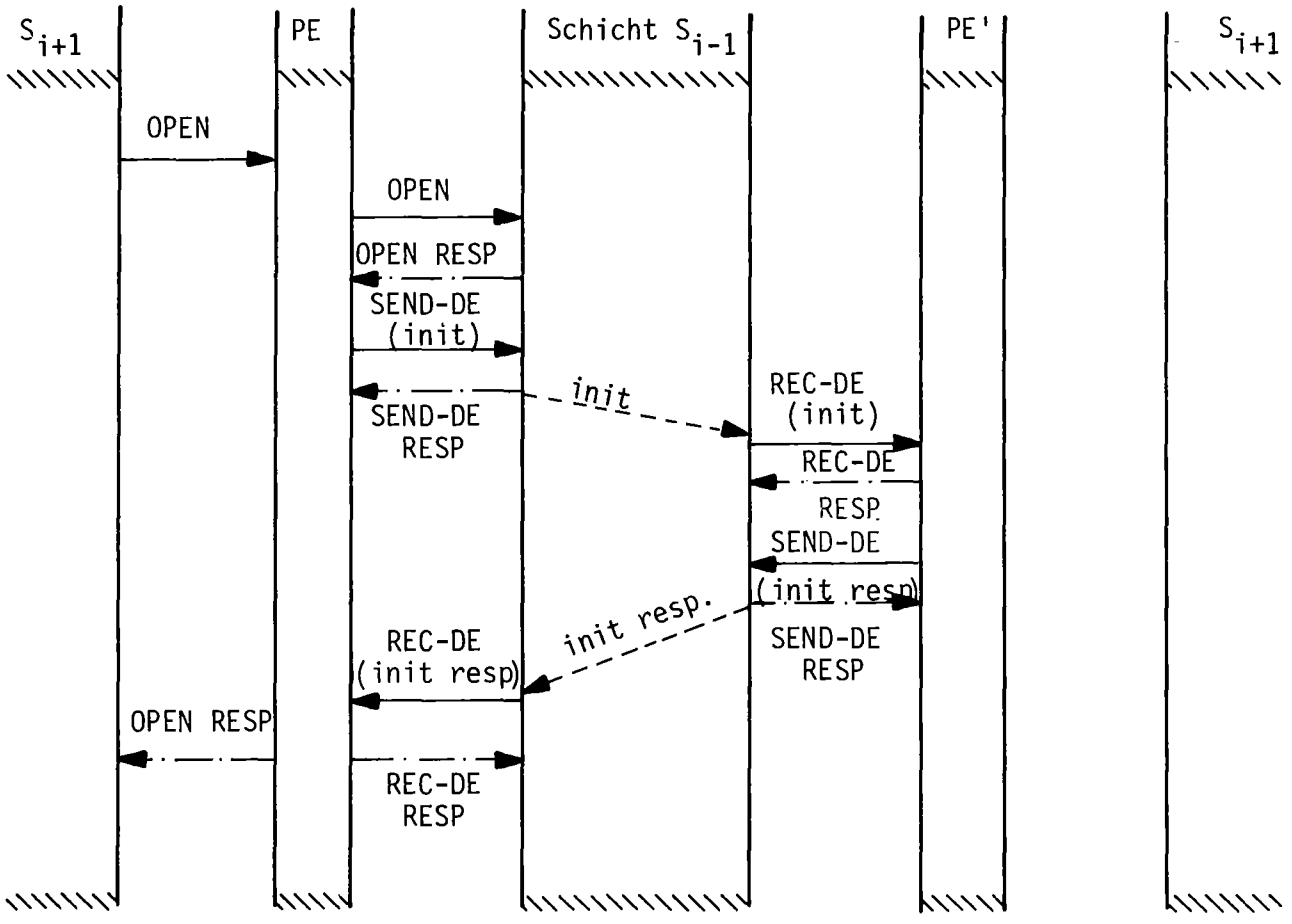


Abb. 4.5.: Abarbeitung eines OPEN-Schnittstellenauftrags im Falle des Basisprotokolls

(Notation: ——► / -·-·► Austausch von Schnittstellenaufträgen (Kommunikationsprimitive/Rückmeldungen) zwischen benachbarten Schichten

– – – ► Transport von Dateneinheiten durch einen Dienst der Schicht S_{i-1})

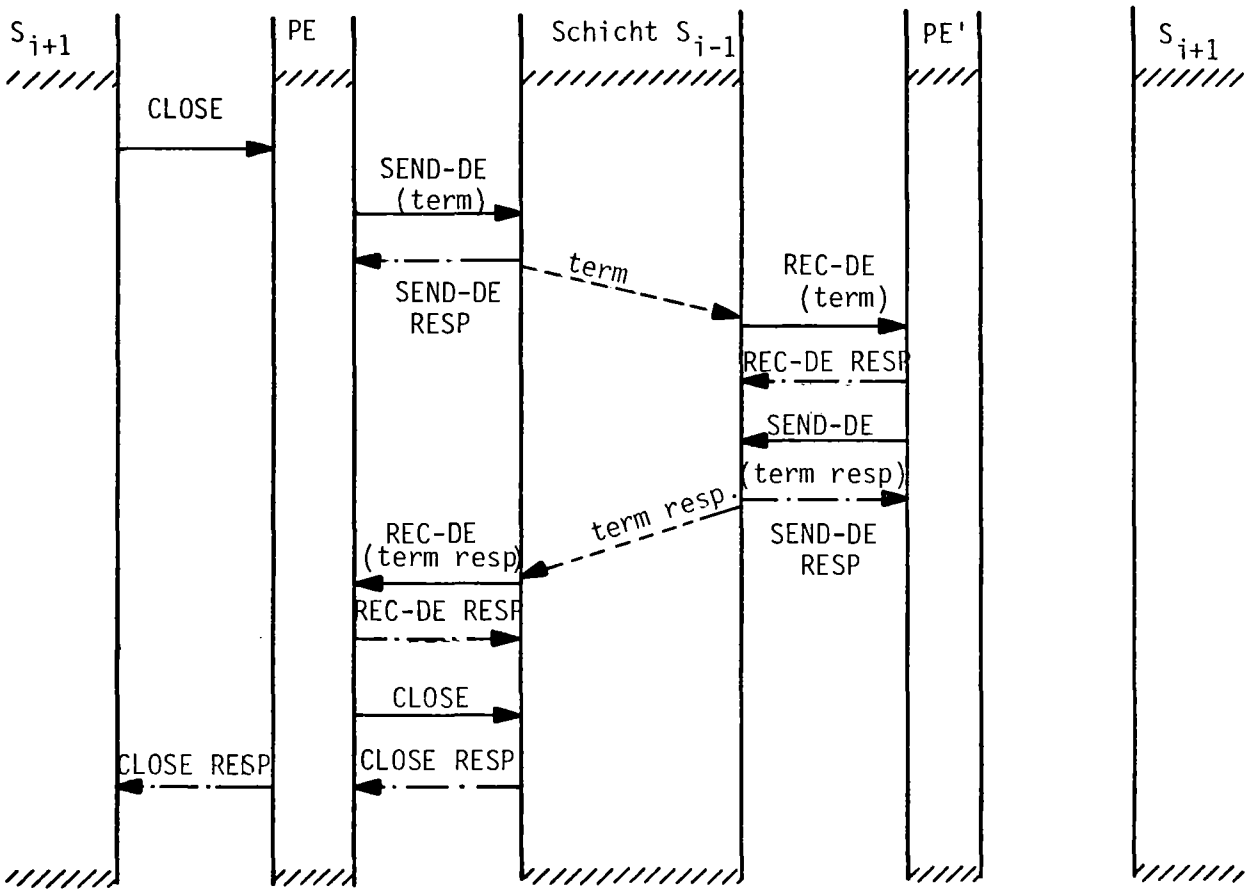


Abb. 4.6.: Abarbeitung eines CLOSE-Schnittstellenauftrags im Falle des Basisprotokolls

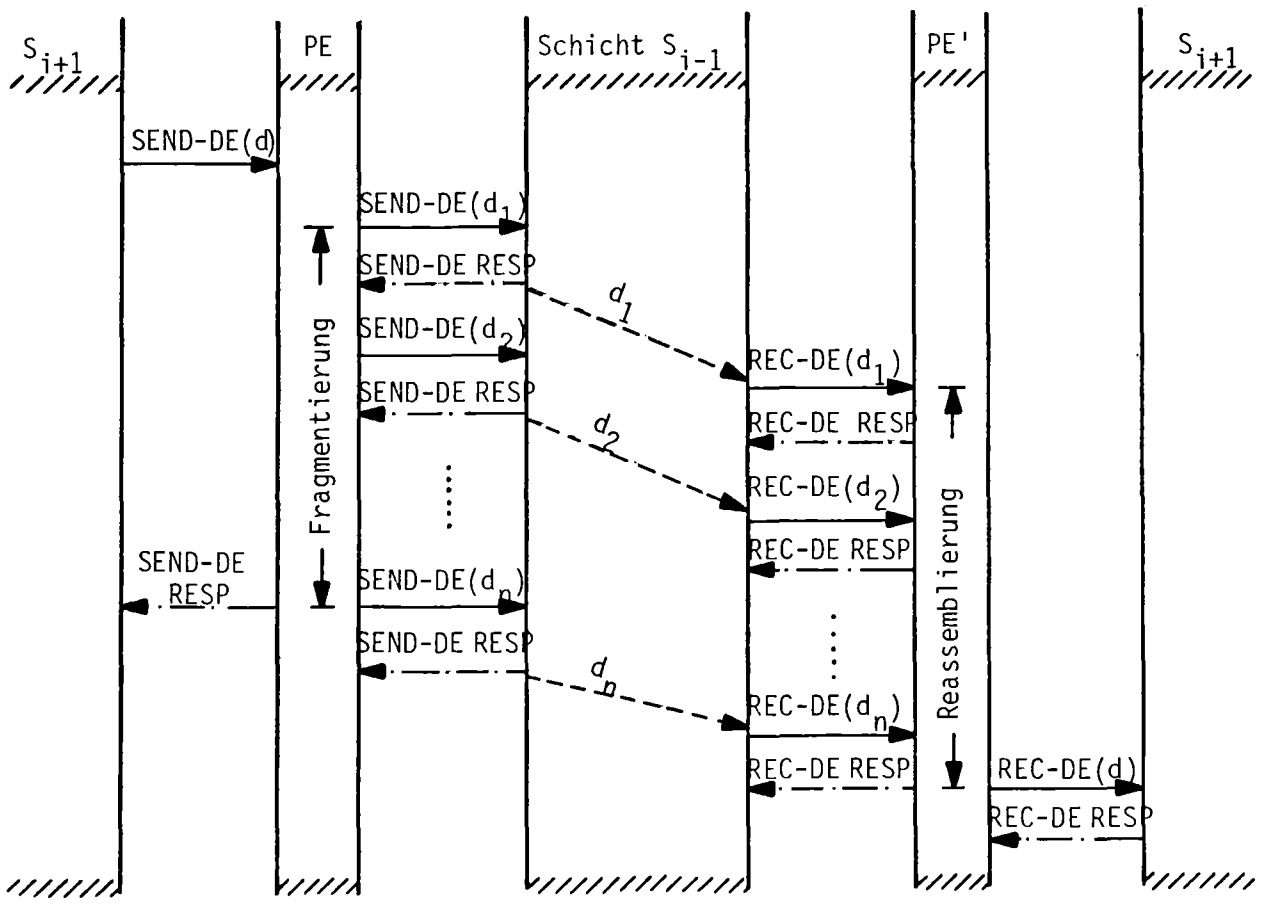


Abb. 4.7.: Abarbeitung eines SEND-DE-Schnittstellenauftrags in PE und Abarbeitung eines REC-DE-Schnittstellenauftrags in PE' im Fall des Basisprotokolls (Beispiel : Fragmentierung)

(8) Allgemeine Bemerkungen:

Das Basisprotokoll ist in MOSAIC durch NUMPROTOCOL = 0 gekennzeichnet.

4.3.2.2. Das Protokoll HDLC

(1) Literatur:

Für die Implementierung des HDLC-Protokolls in MOSAIC wurde /ISO77/ zugrundegelegt; insbesondere wurden "balanced operation" und "asynchronous response mode" vorausgesetzt.

(2) Benötigte Schnittstellenaufträge:

Es werden die Schnittstellenaufträge OPEN, CLOSE, SEND-DE und REC-DE sowie die entsprechenden Rückmeldungen benötigt. (OPEN und CLOSE werden dabei ausschließlich empfangen und nicht generiert).
Zur Semantik dieser Schnittstellenaufträge s. Basisprotokoll.

(3),(4) Schnittstelle, die der nächsthöheren Schicht S_{i+1} angeboten wird und Schnittstelle, die von der nächsttieferen Schicht S_{i-1} vorausgesetzt wird:

Die Schnittstellen zwischen einer HDLC-Protokolleinheit PE und den benachbarten Schichten sind in Abb. 4.8. dargestellt.

(5) Benötigte Dateneinheiten:

Die unterschiedlichen Typen von HDLC-"Frames" wurden auf folgende Typen von Dateneinheiten abgebildet (Typ-Angabe in Klammern):

- information transfer frame:

  i-frame (1001)

- supervisory frames:

  receive ready = rr (1002), receive not ready = rnr (1003),
  reject = rej (1004)

- unnumbered frames:

  set asynchronous response mode = sarm (1005), disconnect = disc (1006),
  unnumbered acknowledgement = ua (1007), command reject = cmdr (1008)

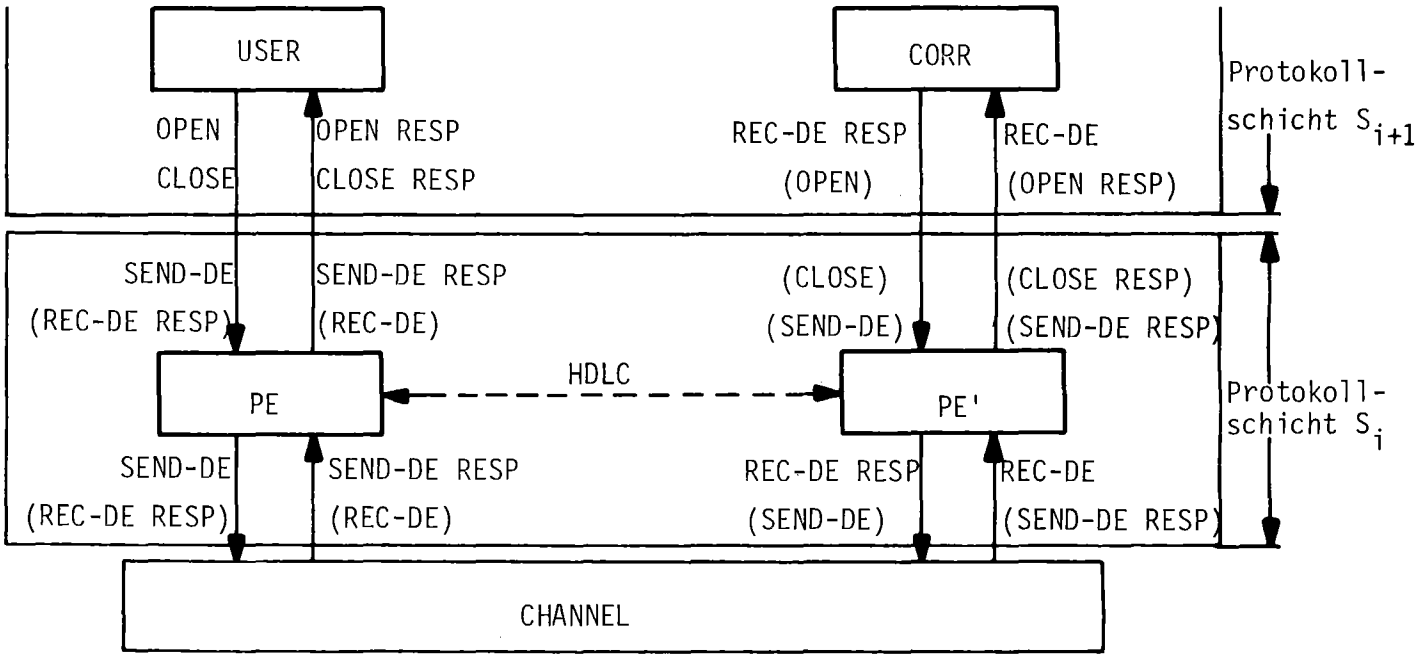


Abb. 4.8.: Schnittstellen, die zwischen HDLC-Protokolleinheiten und benachbarten Protokollschichten existieren
[Schnittstellenaufträge ohne Klammern bei Initiative durch die Protokolleinheit USER; Schnittstellenaufträge in Klammern bei Initiative durch CORR]

(6) Aktionen einer HDLC-Protokolleinheit (PE bzw. PE'):

(a) Erhalt von OPEN durch einen Nachbarn USER (s. Abb. 4.9.):

- falls Datenaustauschphase noch nicht erreicht:
  Protokollinitialisierung gemäß HDLC-Protokoll erfolgt;
- nach Beendigung der Initialisierung mit PE':
  OPEN RESP wird an die Protokolleinheit USER übergeben, durch die der ursprüngliche OPEN-Schnittstellenauftrag generiert worden war.

(b) Erhalt von CLOSE durch den Nachbarn USER (s. Abb. 4.10.):

- gegebenenfalls Protokollterminierung gemäß HDLC-Protokoll;
- nach Beendigung der Terminierung mit PE':
  CLOSE RESP wird an die Protokolleinheit USER übergeben, durch die das CLOSE generiert worden war.

(c) Erhalt von SEND-DE durch einen Nachbarn USER (s. Abb. 4.11.):

- die referenzierte Dateneinheit (d) wird in ein i-frame einge-

bettet, und dieses i-frame wird an den Korrespondenten PE' unter Benutzung des CHANNEL übermittelt (Flußkontrolle mit beliebigem "Window", z.B. 8 oder 128, wird durchgeführt);

- dem Nachbarn USER wird nach Beendigung des Sendevorgangs ein SEND-DE RESP übergeben.

(d) Erhalt von REC-DE durch einen Nachbarn der nächsttieferen Schicht S_{i-1} (s. Abb. 4.11):

- die referenzierte Dateneinheit (z.B. i-frame), die durch S_{i-1} (d.h. CHANNEL) an PE' übergeben wurde, ist zu interpretieren und gegebenenfalls an die Protokolleinheit CORR der nächsthöheren Schicht weiterzuleiten (⟶ eventuelle Quittungserstellung für das i-frame gemäß HDLC-Protokoll);
- dem Nachbarn der nächsttieferen Schicht wird für jede erhaltene Dateneinheit ein REC-DE RESP übergeben.

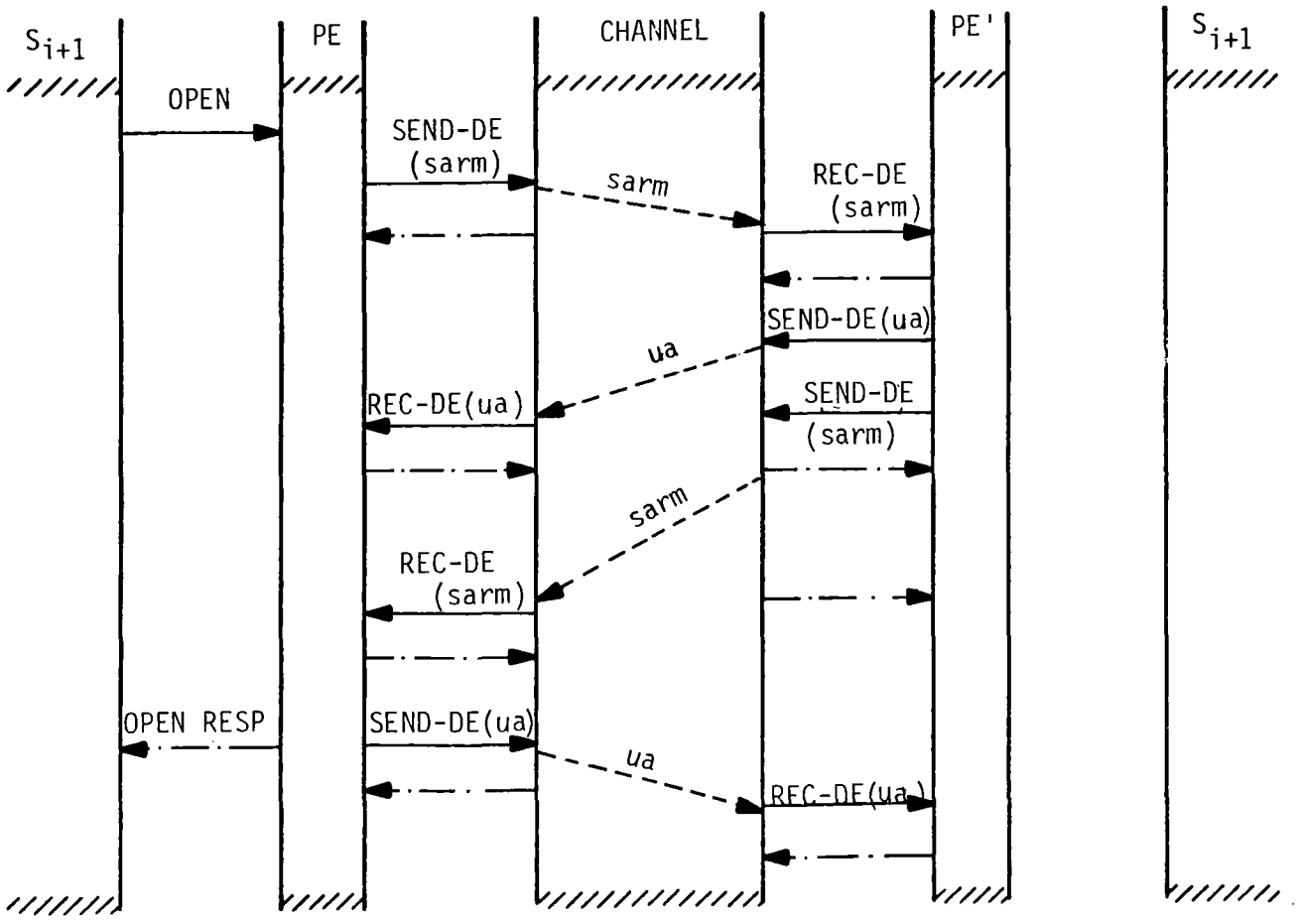


Abb. 4.9.: Abarbeitung eines OPEN-Schnittstellenauftrags (HDLC)

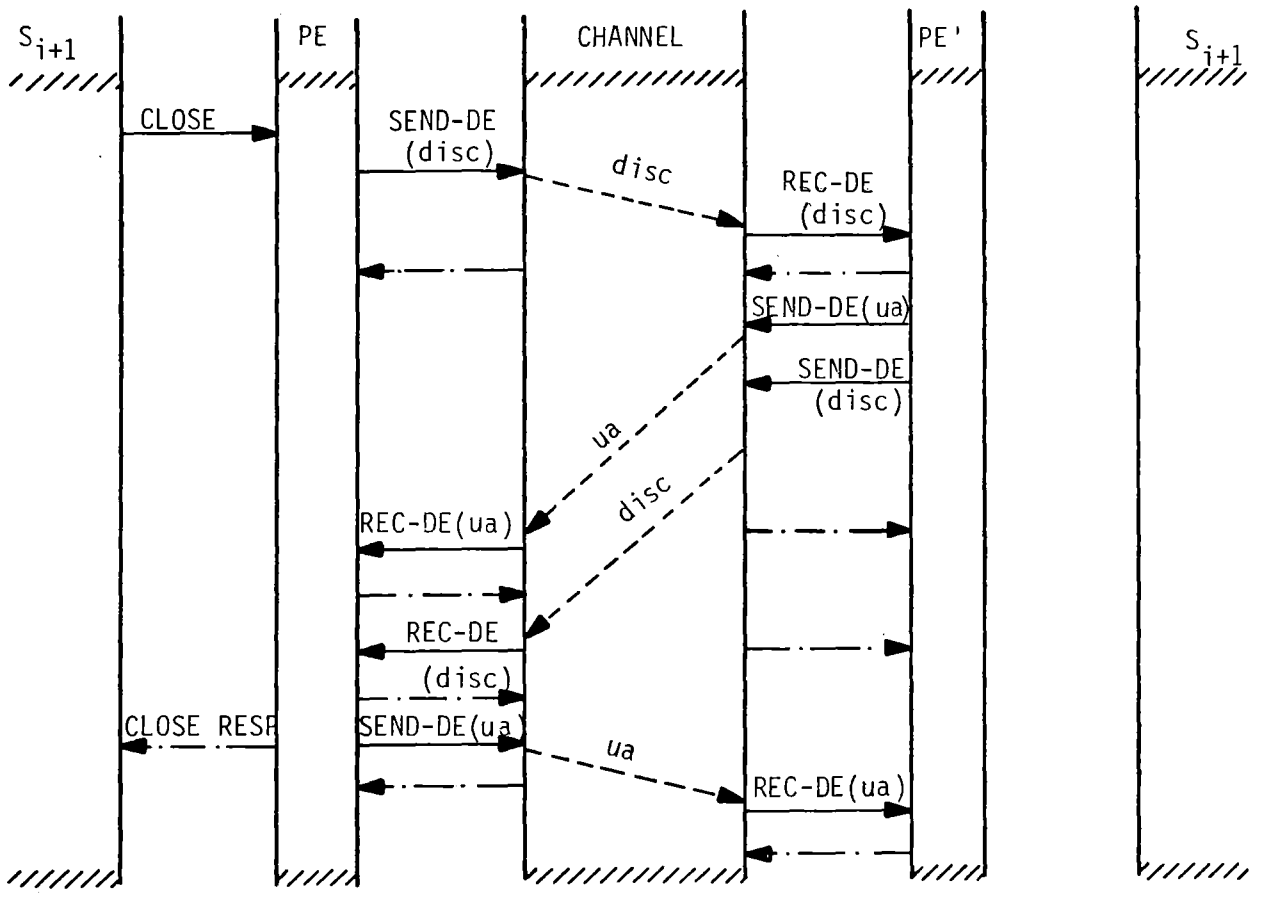


Abb. 4.10.: Abarbeitung eines CLOSE-Schnittstellenauftrags (HDLC)

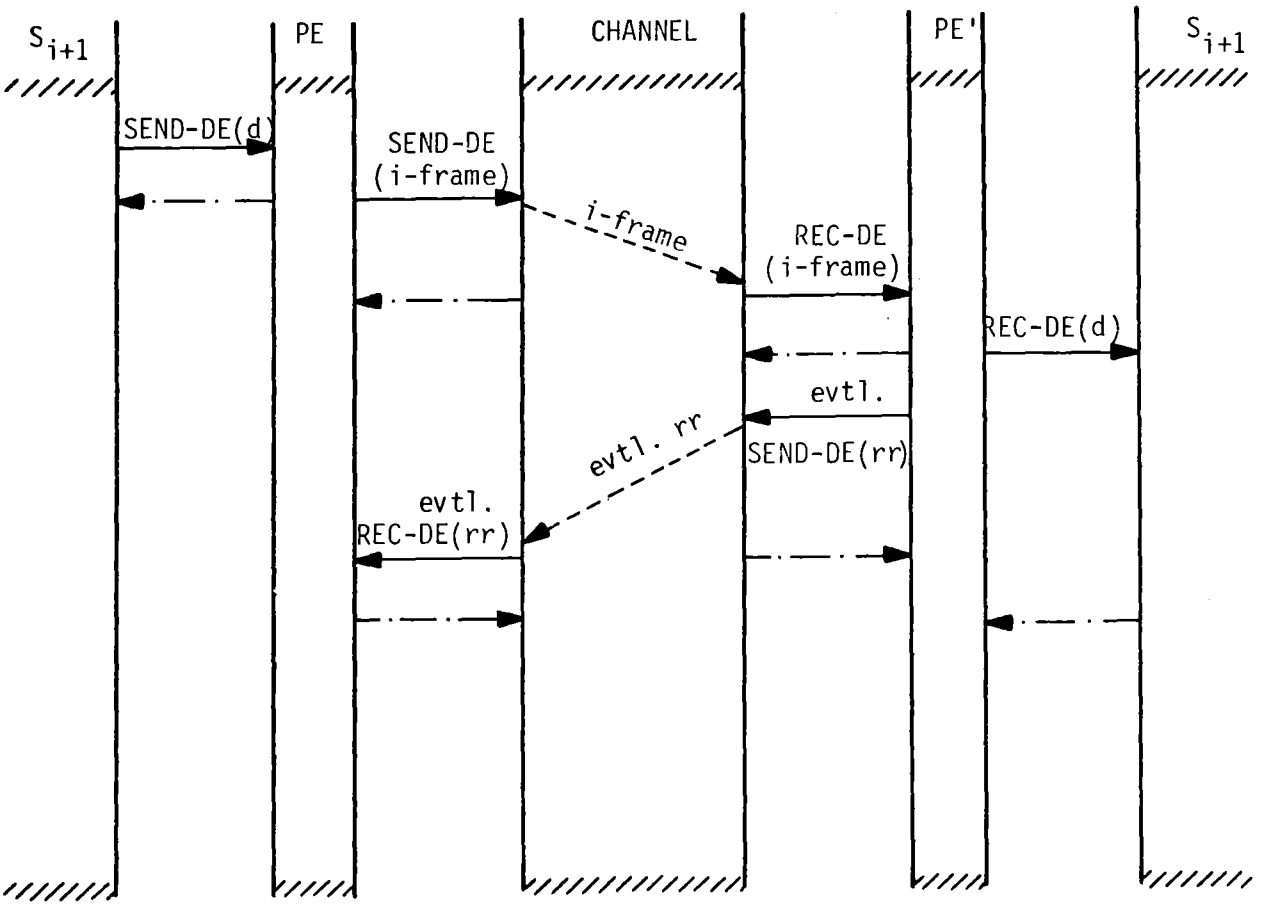


Abb. 4.11.: Abarbeitung eines SEND-DE-Schnittstellenauftrags in PE und Abarbeitung eines REC-DE in PE' (HDLC)

(7) Zeitliches Verhalten:

Das zeitliche Verhalten bei der Bearbeitung der Schnittstellenaufträge OPEN,CLOSE,SEND-DE und REC-DE kann den Abb. 4.9. - 4.11. entnommen werden.

(8) Allgemeine Bemerkungen:

HDLC ist in MOSAIC durch NUMPROTOCOL = 10 gekennzeichnet.

Da Punkt-zu-Punkt-Verbindungen vorausgesetzt werden, gilt:

Anzahl der Korrespondenten für eine HDLC-Protokolleinheit = 1.

Da durch HDLC überdies keine Benutzer einer höheren Schicht unterschieden werden, kennt eine HDLC-Protokolleinheit zu jedem Zeitpunkt genau einen Protokollzustand (s. Abb. 4.12.)

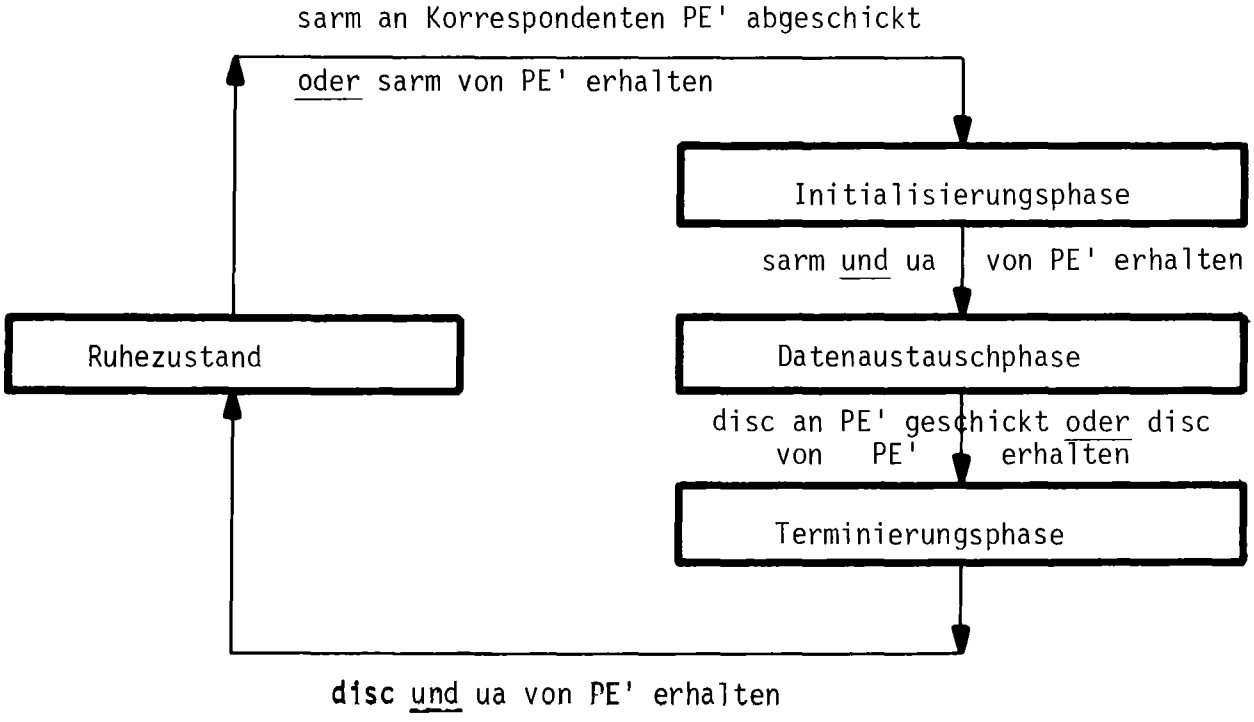


Abb. 4.12.: Die wichtigsten Protokollzustandsübergänge einer HDLC-Protokolleinheit

4.3.2.3. Das Protokoll X.25 (Paketebene)

(1) Literatur:

Für die Implementierung des X.25-Protokolls in MOSAIC wurde /CCITT77/ zugrundegelegt. X.25 steht im folgenden ausschließlich für den level 3 der CCITT-Empfehlung; X.25 (level 2) s. HDLC in Abschnitt 4.3.2.2. und X.25 (level 1) s. Modell eines Übertragungskanals.

(2) Benötigte Schnittstellenaufträge:

Neben den Schnittstellenaufträgen OPEN, CLOSE, SEND-DE, REC-DE sowie den entsprechenden Rückmeldungen, wird der Schnittstellenauftrag SEND INTERRUPT von der nächsthöheren Schicht S_{i+1} entgegengenommen.

SEND INTERRUPT [Kurzform: SEND-INT] mit den Parametern CORR, USER und IRPT dient als Aufforderung, die Interruptinformation IRPT für den Benutzer USER zu einem seiner Korrespondenten CORR weiterzuleiten.

Zu diesem Schnittstellenauftrag existiert die Rückmeldung SEND-INT RESP.

(3),(4) Schnittstelle, die der nächsthöheren Schicht S_{i+1} angeboten wird und Schnittstelle, die von der nächsttieferen Schicht S_{i-1} vorausgesetzt wird:

Die Schnittstellen zwischen einer X.25-Protokolleinheit PE und den benachbarten Schichten sind in Abb. 4.13. dargestellt.

(5) Benötigte Dateneinheiten:

Die unterschiedlichen Typen von X.25-"Packets" wurden auf folgende Typen von Dateneinheiten abgebildet (Typ-Angabe in Klammern):

(a) Übergabe von DTE an DCE:

- data & interrupt packets:
  DTE data (2001), DTE interrupt (2002), DTE interrupt confirmation (2003)
- call setup & clearing packets:
  call request (2004), call accepted (2005), clear request (2006), DTE clear confirmation (2007)

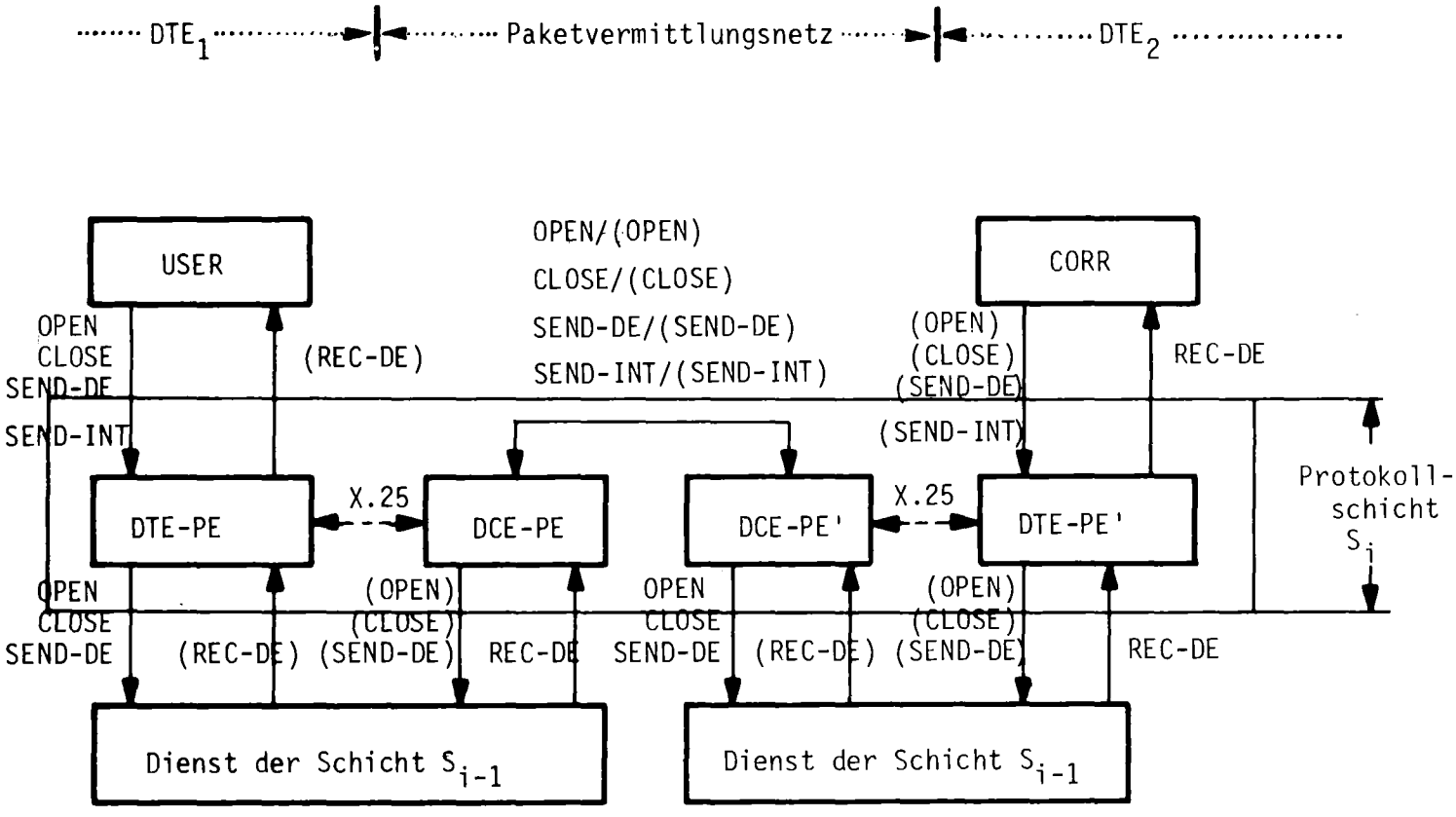


Abb. 4.13.: Schnittstellen, die zwischen X.25-Protokolleinheiten und benachbarten Protokollschichten existieren
[Schnittstellenaufträge ohne Klammern bei Initiative durch die Protokolleinheit USER; Schnittstellenaufträge in Klammern bei Initiative durch CORR; Rückmeldungen wurden nicht dargestellt]

- control & reset packets:
  DTE receive ready (2008), DTE receive not ready (2009), DTE reject (2010), reset request (2011), DTE reset confirmation (2012)
- restart packets:
  restart request (2013), DTE restart confirmation (2014)

(b) Übergabe von DCE an DTE:

- data & interrupt packets:
  DCE data (2021), DCE interrupt (2022), DCE interrupt confirmation (2023)
- call setup & clearing packets:
  incoming call (2024), call connected (2025), clear indication (2026), DCE clear confirmation (2027)

- control & reset packets:
  DCE receive ready (2028), DCE receive not ready (2029), reset indication (2031), DCE reset confirmation (2032)
- restart packets:
  restart indication (2033), DCE restart confirmation (2034)

(6) Aktionen einer X.25-Protokolleinheit (im DTE):

(a) Erhalt von OPEN durch einen Nachbarn USER (s. Abb. 4.14.):

- die X.25-Protokolleinheit baut,im Falle von "virtual call", für USER einen logischen Kanal auf, über den der Nachbar USER anschließend Dateneinheiten transferieren kann (im Falle von "virtual circuit" entfällt diese Initialisierung);
- die Existenz des logischen Kanals wird der nächsthöheren Schicht durch OPEN RESP zurückgemeldet.

(b) Erhalt von CLOSE durch einen Nachbarn USER (s. Abb. 4.15.):

- im Falle des "virtual call" wird der existierende logische Kanal gemäß dem X.25-Protokoll abgebaut;
- dem Nachbarn USER wird anschließend ein CLOSE RESP übergeben.

(c) Erhalt von SEND-DE durch einen Nachbarn USER (s. Abb. 4.16.):

- die referenzierte Dateneinheit wird in ein "DTE data packet" eingebettet und über die Schicht S_{i-1} an das Paketvermittlungsnetz übergeben;
- im Paketvermittlungsnetz findet eine Umwandlung des "DTE data packet" in ein "DCE data packet" statt, welches an das Empfangs-DTE weitergeleitet wird (Fluß- und Fehlerkontrolle werden dabei durchgeführt).

(d) Erhalt von REC-DE durch einen Nachbarn der nächsttieferen Schicht S_{i-1} (s. Abb. 4.16):

- die referenzierte Dateneinheit (z. B. data packet), die über S_{i-1} erhalten wurde, ist zu interpretieren und an die Protokolleinheit der nächsthöheren Schicht weiterzuleiten (falls es sich um ein "DCE data packet" handelt) bzw. in ein "DCE data packet" einzubetten (falls es sich um ein " DTE data packet" handelt);

- dem Nachbarn der nächsttieferen Schicht wird für jede erhaltene Dateneinheit ein REC-DE RESP übergeben.

(e) Erhalt von SEND-INT durch einen Nachbarn USER (s. Abb. 4.17.):

die Aktionen sind entsprechend (c) mit dem Unterschied, daß der Transport von Interrupts keiner Flußkontrolle unterliegt.

(7) Zeitliches Verhalten:

Das zeitliche Verhalten bei der Bearbeitung der Schnittstellenaufträge OPEN, CLOSE, SEND-DE, REC-DE und SEND-INT kann den Abb. 4.14. - 4.17. entnommen werden.

(8) Allgemeine Bemerkungen:

X.25 (Paketebene) ist in MOSAIC durch NUMPROTOCOL = 20 gekennzeichnet.

Da das X.25-Protokoll für DTE und DCE unsymmetrisches Verhalten vorschreibt, wird der Protokolleinheiten-Parameter MASTER dazu verwendet, zwischen DTE und DCE zu differenzieren; und zwar gilt:

MASTER = true für DTE, sowie

MASTER = false für DCE.

Für die Anzahl der Korrespondenten NUMCORR einer X.25-Protokolleinheit gilt NUMCORR = 1; für die maximale Anzahl der zu unterscheidenden Benutzer MAXUSERS einer höheren Schicht gilt MAXUSERS $\leq$ 4096. Damit kennt eine X.25-Protokolleinheit in MOSAIC bis zu 4096 (= NUMCORR×MAXUSERS) logische Verbindungen; ein Protokollzustand (s. Abb. 4.18.) wird geführt für jede dieser Verbindungen.

Bei den logischen Verbindungen kann es sich entweder um permanente oder um temporäre Verbindungen handeln. (Permanente Verbindungen kennen nur die Datenaustauschphase, die sie nicht verlassen).

RESTART und RESET wurden bislang nicht implementiert; hingegen sind

- Verbindungsaufbau / -abbau für logische Kanäle,
- getrennte Interrupt- und Datenkanäle,
- Flußkontrolle,
- Quittungsverkehr

im Modell einer X.25-Protokolleinheit in MOSAIC enthalten.

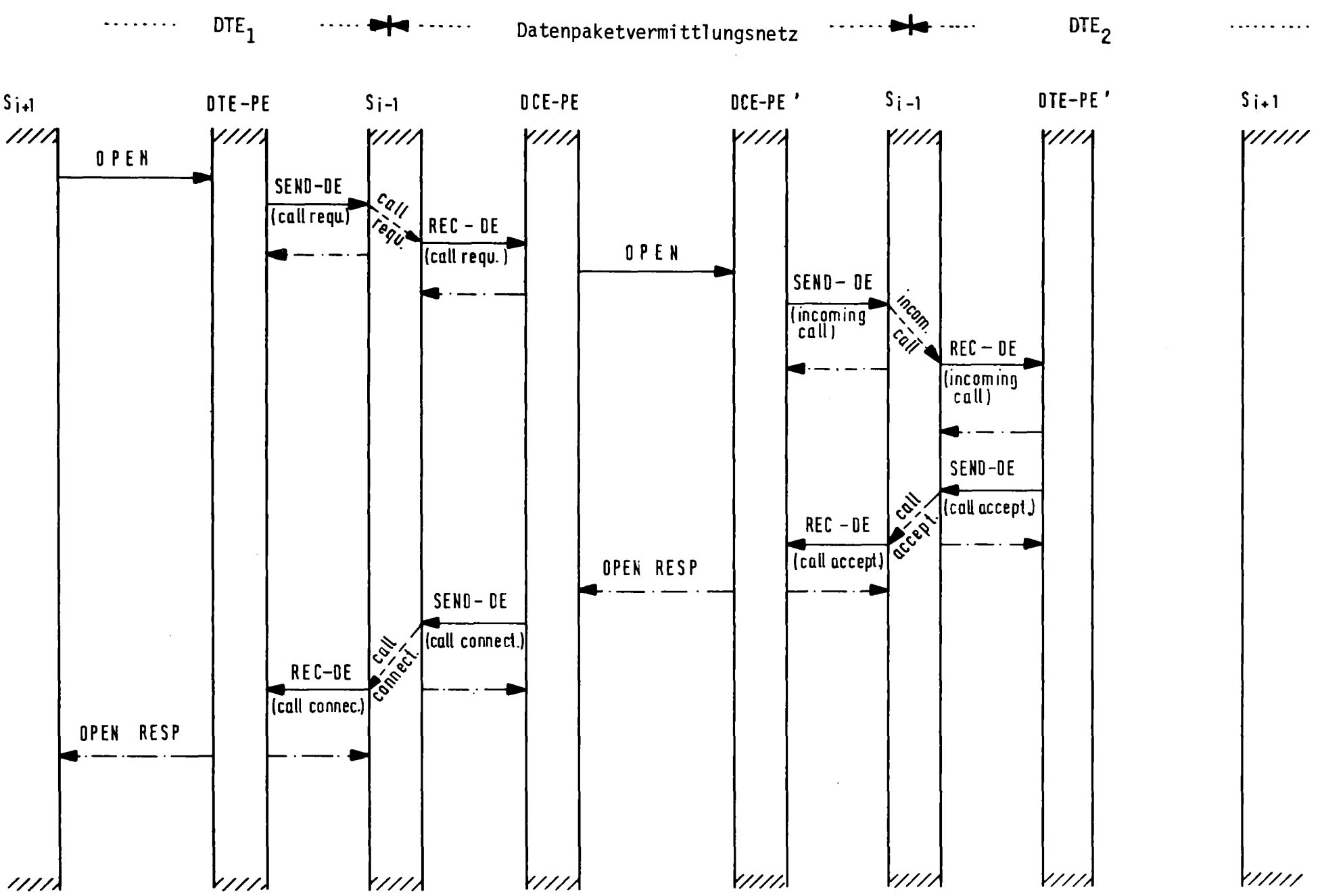


Abb. 4.14.: Abarbeitung eines OPEN-Schnittstellenauftrags (X.25)

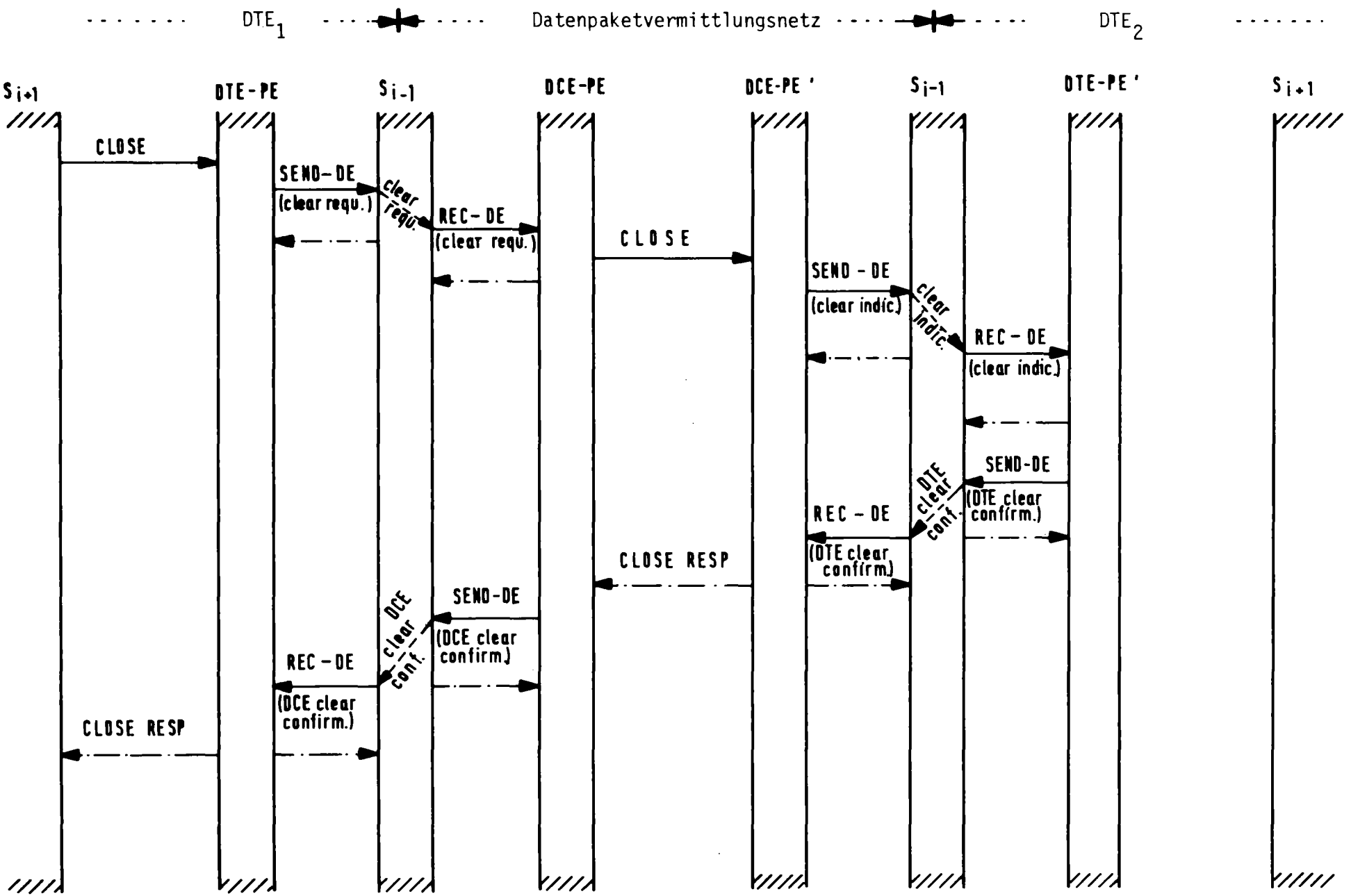


Abb. 4.15.: Abarbeitung eines CLOSE-Schnittstellenauftrags (X.25)

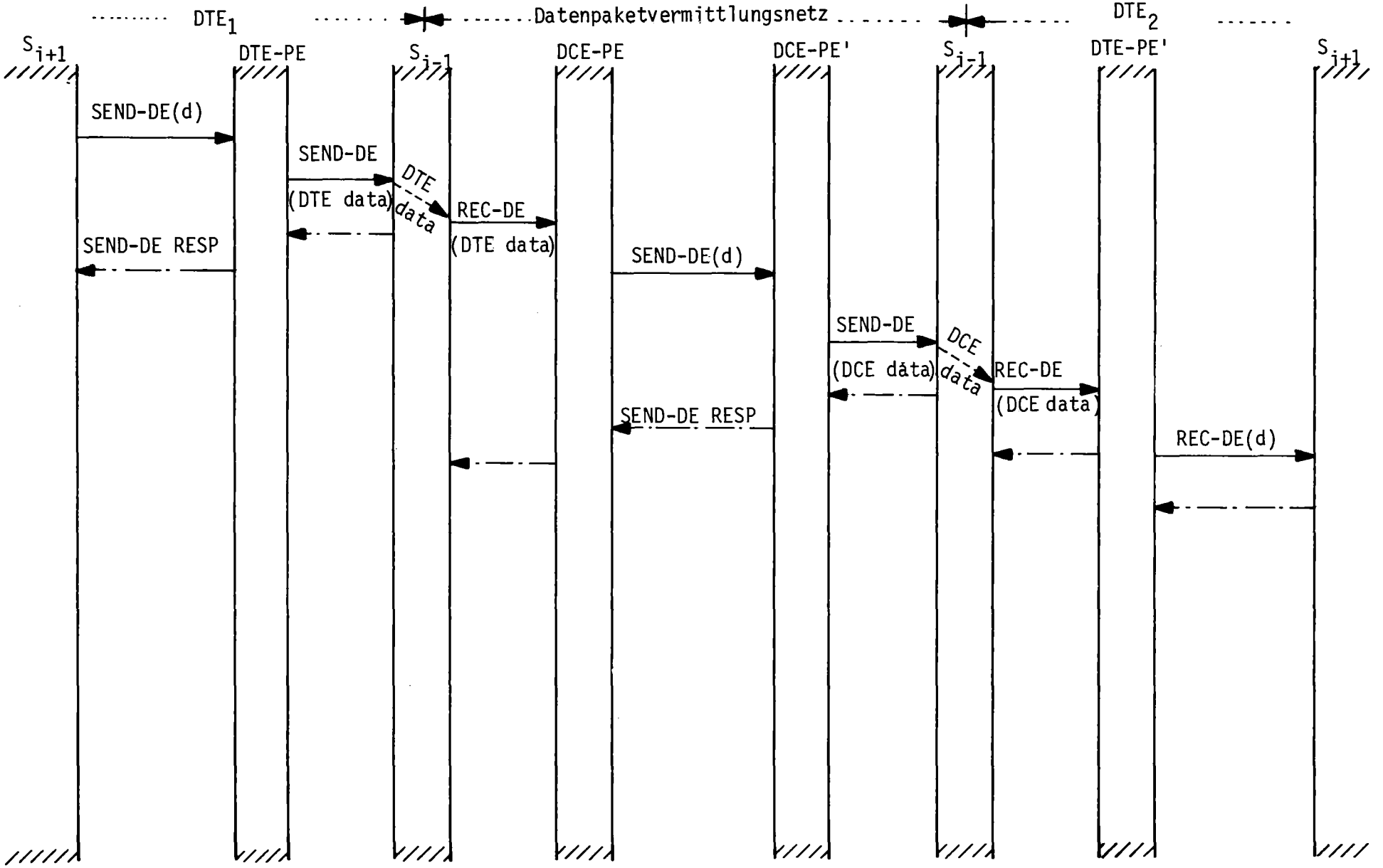


Abb. 4.16.: Abarbeitung eines SEND-DE-Schnittstellenauftrags in DTE-PE und Abarbeitung eines REC-DE in DTE-PE' (X.25)

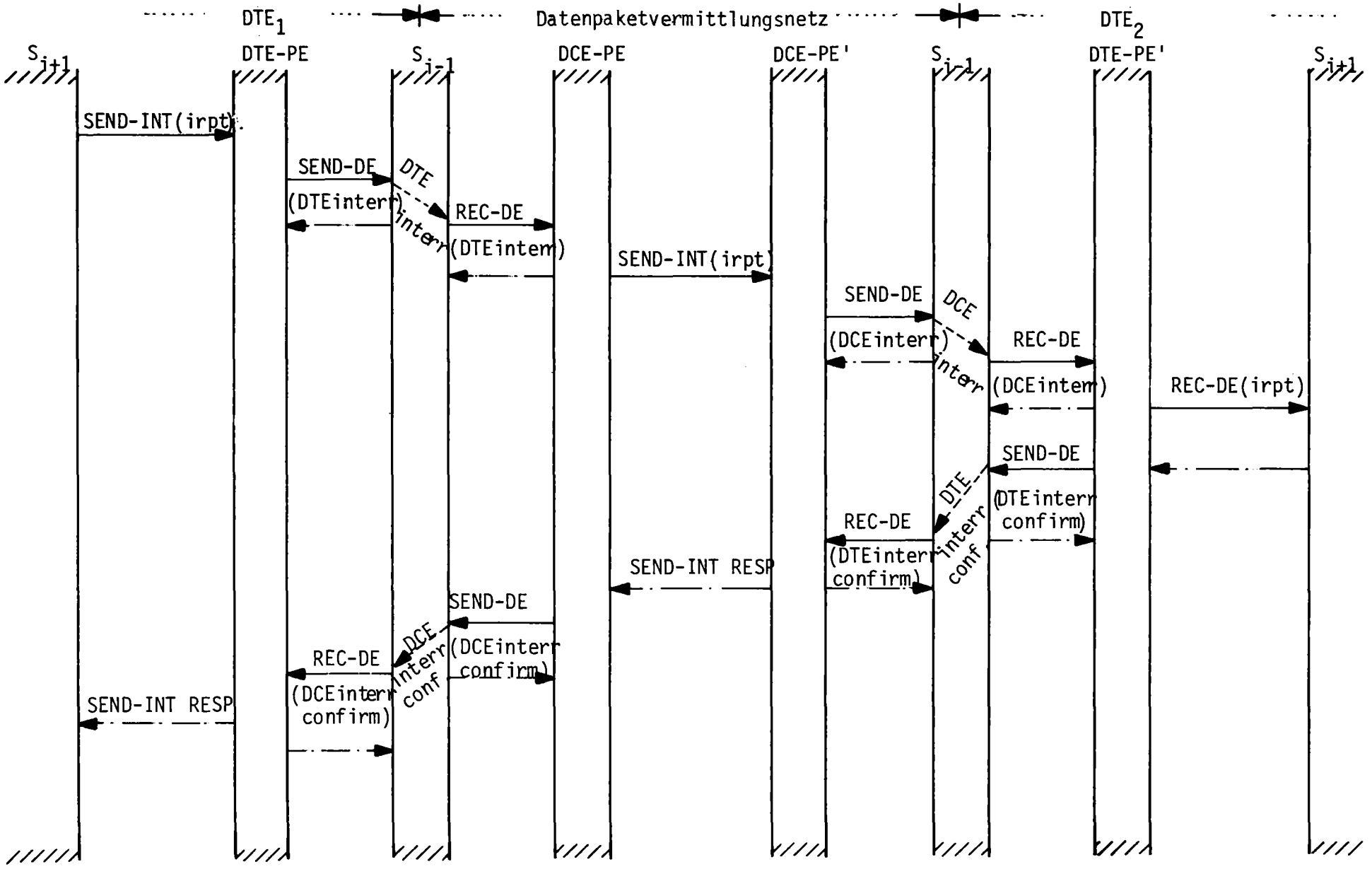


Abb. 4.17.: Abarbeitung eines SEND-INT - Schnittstellenauftrags in DTE-PE (X.25)

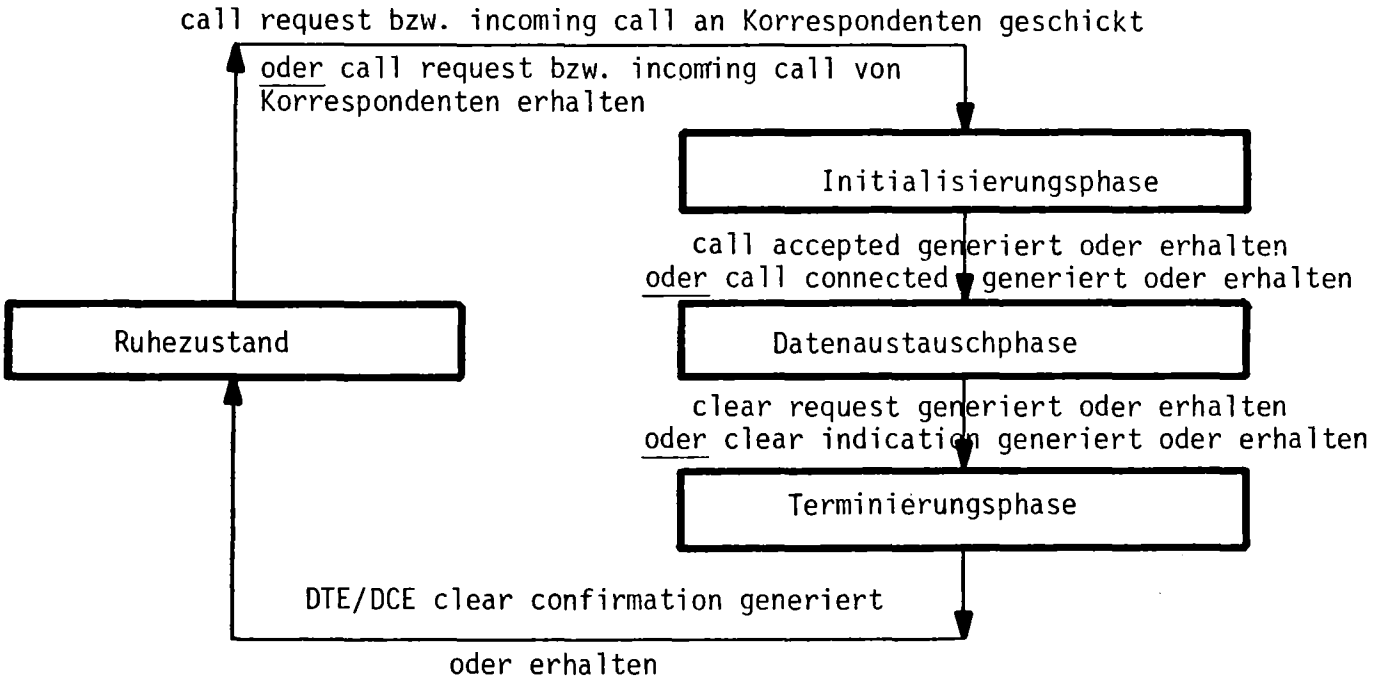


Abb. 4.18.: Die wichtigsten Protokollzustandsübergänge einer X.25-Protokolleinheit

4.3.2.4. Das Message Link Protokoll

(1) Literatur:

Es wurde ein Teil des durch den Arbeitskreis PIX definierten "Message link"-Protokolls implementiert (s. "minimum ML-protocol" in /HERT78/).

(2) Benötigte Schnittstellenaufträge:

Zur Kommunikation mit den benachbarten Schichten benötigt eine MLP-Protokolleinheit die Schnittstellenaufträge OPEN, CLOSE, SEND-DE, REC-DE und SEND-INT sowie die entsprechenden Rückmeldungen. Die Schnittstellenaufträge OPEN und CLOSE, die eine MLP-Protokolleinheit von der nächsthöheren Schicht S_{i+1} entgegennimmt, entsprechen den Kommunikationsprimitiven CONNECT und DISCONNECT in /HERT78/.

(3),(4) Schnittstelle, die der nächsthöheren Schicht S_{i+1} angeboten wird und Schnittstelle, die von der nächsttieferen Schicht S_{i-1} vorausgesetzt wird:

Die Schnittstellen zwischen einer MLP-Protokolleinheit PE und den benachbarten Schichten sind in Abb. 4.19. dargestellt.

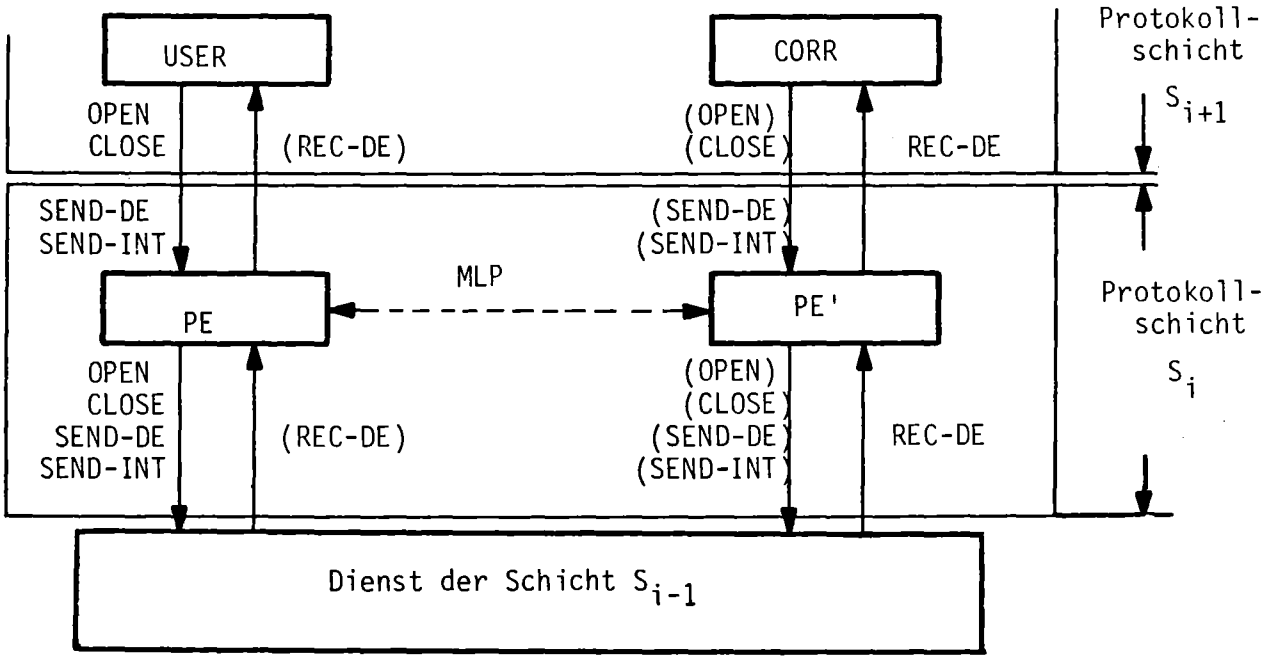


Abb. 4.19.: Schnittstellen, die zwischen MLP-Protokolleinheiten und benachbarten Protokollschichten existieren [Notation s. Abb. 4.13.]

(5) Benötigte Dateneinheiten:

Die unterschiedlichen Typen von MLP-"Messages" wurden auf folgende Typen von Dateneinheiten abgebildet (Typ-Angabe in Klammern):

(a) data messages:

data unnumbered, keine Quittung verlangt = ud (3001); data unnumbered, Quittung verlangt = ud & ackreq (3002); data numbered, keine Quittung verlangt = nd (3003); data numbered, Quittung verlangt = nd & ackreq (3004); data receive ready = darr (3005)

(b) connection control messages:

connect = conn (3006); connect response = conn resp. (3007); disconnect = disc (3008); disconnect response = disc resp. (3009); interrupt = irpt (3010); connect reject = crej (3025); reconnect = reco (3026); reconnect response = reco resp. (3027); release = rels (3028); release response = rels resp. (3029); resume = resu (3030); resume response = resu resp. (3031); reset = rset (3032)

(c) communication control messages:

mark (3011); protocol close = pcls (3012); protocol close response = pcls resp. (3013); protocol open = popn (3014); protocol open response = popn resp. (3015); protocol open reject = prej (3016); protocol reopen = reop (3017); protocol reopen response = reop resp. (3018); status = stat (3019)

Bemerkung: Für die aktuelle Implementierung sind nur die Message-Typen 3001, ... , 3011 von Bedeutung.

(6) Aktionen einer MLP-Protokolleinheit (PE bzw. PE'):

(a) Erhalt von OPEN durch einen Nachbarn USER (s. Abb. 4.20.):

- Verbindung zu dem entsprechenden Korrespondenten PE' über die nächsttiefere Schicht S_{i-1} (z.B. X.25-Protokolleinheiten) wird hergestellt;
- dem Korrespondenten PE' wird eine Initialisierungsaufforderung (connect) übermittelt;
- nach Erhalt einer Antwort (connect response) von PE' wird dem Korrespondenten ein mark gesendet;
- nach Erhalt eines mark von PE' wird ein OPEN RESP an die Protokolleinheit bzw. die Benutzertask USER übergeben, durch die der ursprüngliche OPEN-Schnittstellenauftrag generiert worden war.

(b) Erhalt von CLOSE durch den Nachbarn USER (s. Abb. 4.21.):

- dem Korrespondenten PE' wird eine Terminierungsaufforderung (disconnect) übermittelt;
- nach Erhalt einer Antwort (disconnect response) von PE' wird an USER ein CLOSE RESP übergeben und die existierende Verbindung zu PE' in der nächsttieferen Schicht S_{i-1} abgebaut.

(c) Erhalt von SEND-DE durch den Nachbarn USER (s. Abb. 4.23.):

- die referenzierte Dateneinheit (d) wird an den entsprechenden Korrespondenten PE' über die nächsttiefere Schicht S_{i-1} übertragen; im Normalfall wird hierzu eine Dateneinheit des Typs 3001 bzw. 3002 verwendet (abhängig von der Bedingung, ob eine Quittung erwartet wird), bei Fragmentierung müssen allerdings

mehrere Dateneinheiten des Typs 3003 bzw. 3004 benutzt werden;

- dem Nachbarn USER wird bei Beendigung des Sendevorgangs ein SEND-DE RESP übergeben.

(d) Erhalt von REC-DE durch einen Nachbarn der nächsttieferen Schicht S_{i-1} (s. Abb. 4.23.):

- die referenzierte Dateneinheit - die durch S_{i-1} an PE' übergeben wurde - ist zu interpretieren, an die Protokolleinheit bzw. Benutzertask CORR der Schicht S_{i+1} weiterzuleiten (im Falle von ud & ackreq, ud oder irpt), zu reassemblieren (im Falle von nd & ackreq oder nd) und mit einer Dateneinheit darr zu quittieren (im Falle von ud & ackreq oder nd & ackreq);
- dem Nachbarn der nächsttieferen Schicht wird für jede erhaltene Dateneinheit ein REC-DE RESP übergeben.

(e) Erhalt von SEND-INT durch den Nachbarn USER (s. Abb. 4.22.):

- die Interruptinformation wird mittels einer Dateneinheit des Typs interrupt unter Benutzung eines Dienstes der Schicht S_{i-1} an den entsprechenden Korrespondenten PE' übertragen;
- nach Erhalt der Rückmeldung von der Schicht S_{i-1} wird dem Nachbarn USER ein SEND-INT RESP übergeben.

(7) Zeitliches Verhalten:

Das zeitliche Verhalten bei der Bearbeitung der Schnittstellenaufträge OPEN, CLOSE,SEND-DE, REC-DE und SEND-INT kann den Abb. 4.20. - 4.23. entnommen werden.

(8) Allgemeine Bemerkungen:

Das Message Link Protokoll ist in MOSAIC durch NUMPROTOCOL = 30 gekennzeichnet.

Der Protokollzustand in einer MLP-Protokolleinheit (s. Abb. 4.24.) wird geführt für NUMCORR x MAXUSERS parallele "Message links". Setzt man MLP - wie im BERNET - auf X.25 auf, dann setzen wir voraus, daß eine 1 : 1 Abbildung zwischen "Message links" und " X.25 logical channels" besteht.

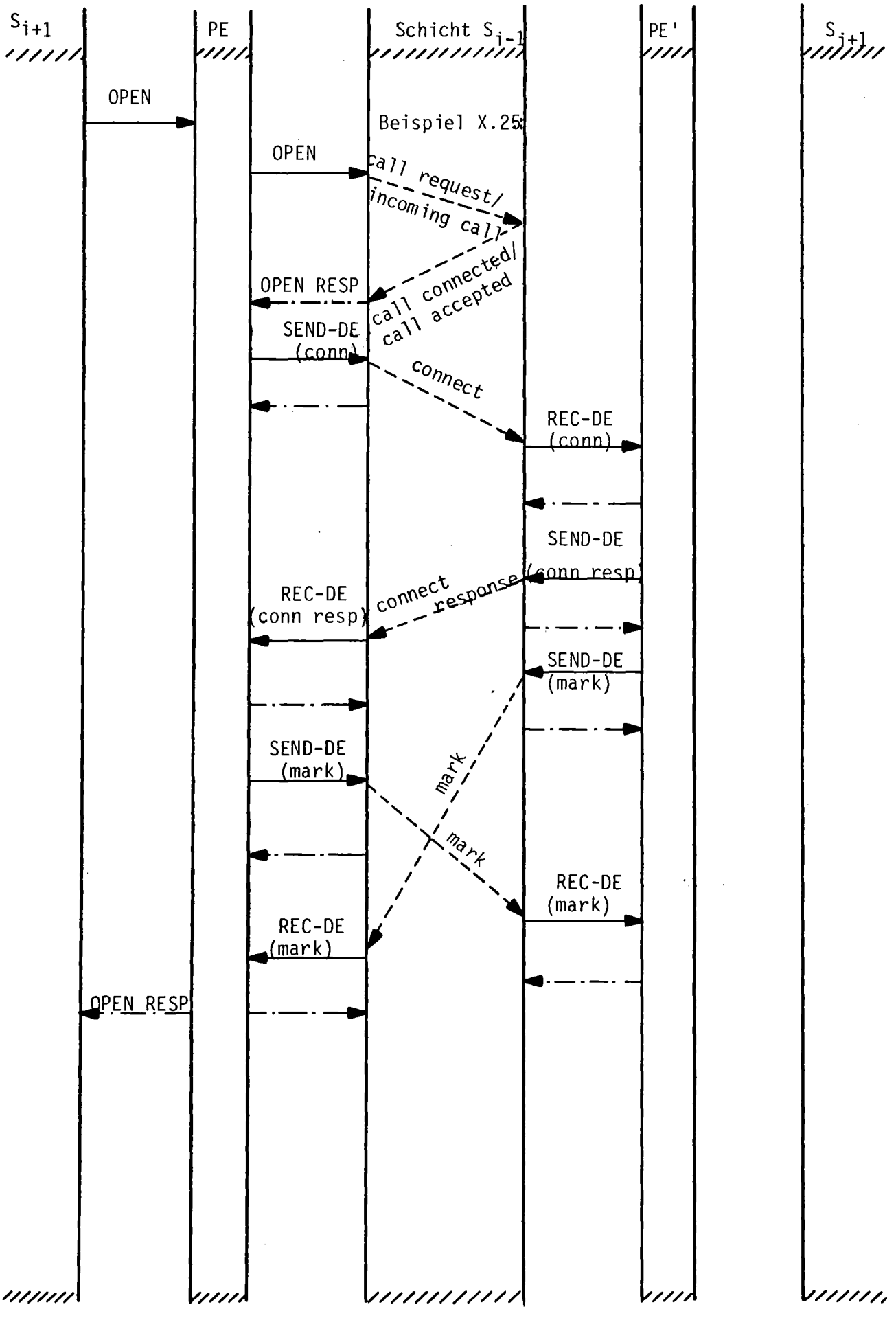


Abb. 4.20: Abarbeitung eines OPEN-Schnittstellenauftrags (MLP)

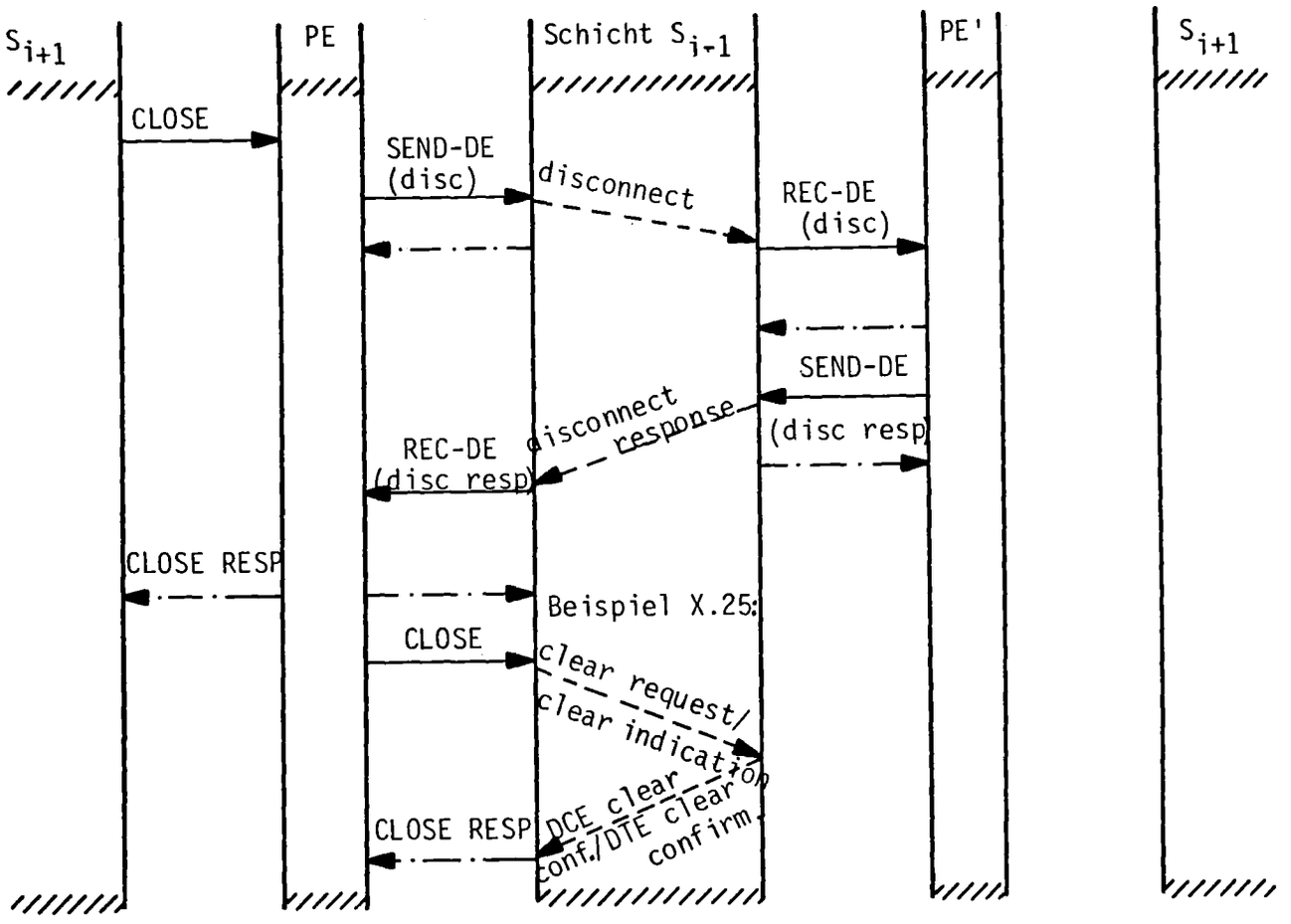


Abb. 4.21.: Abarbeitung eines CLOSE-Schnittstellenauftrags (MLP)

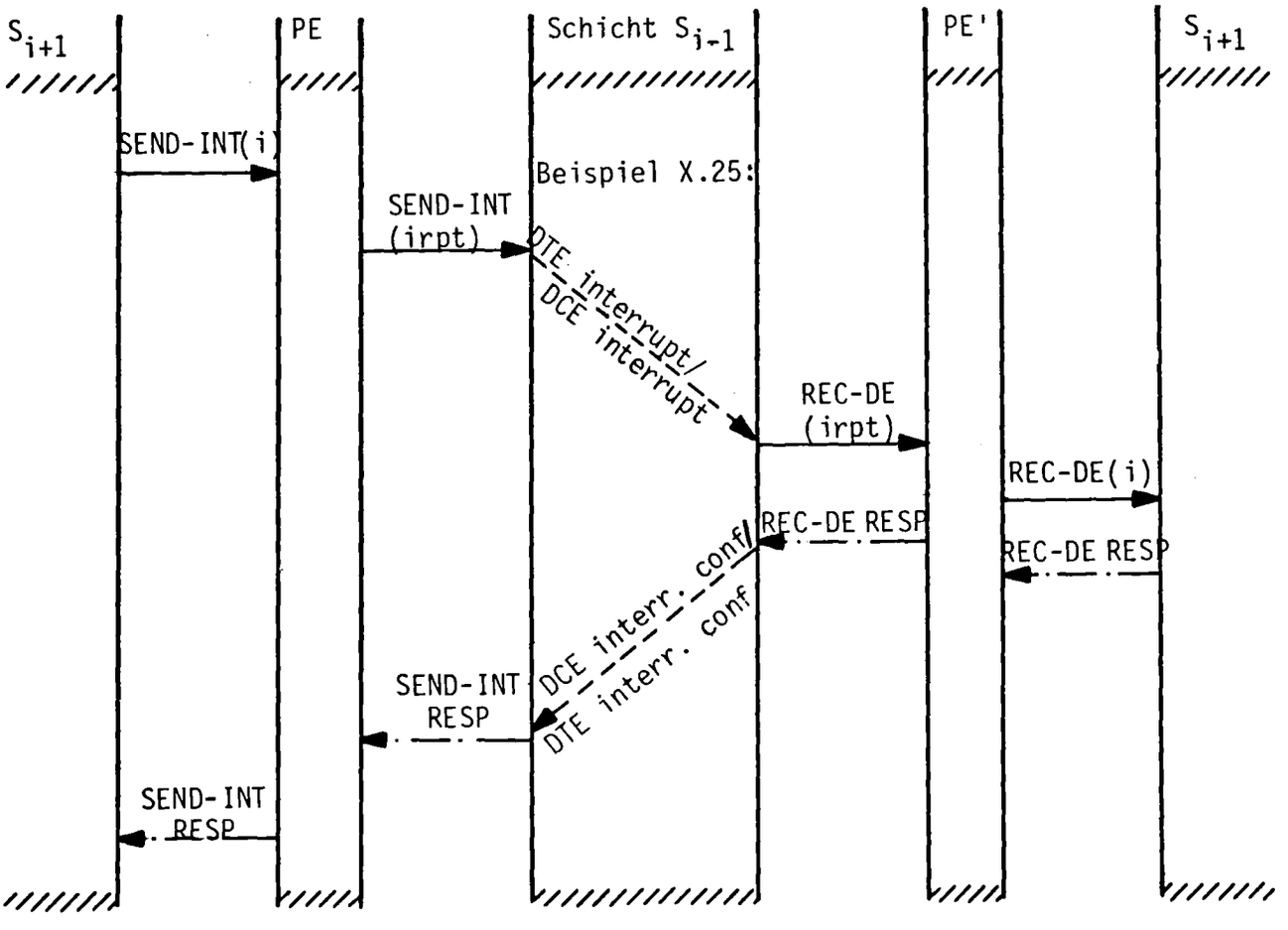
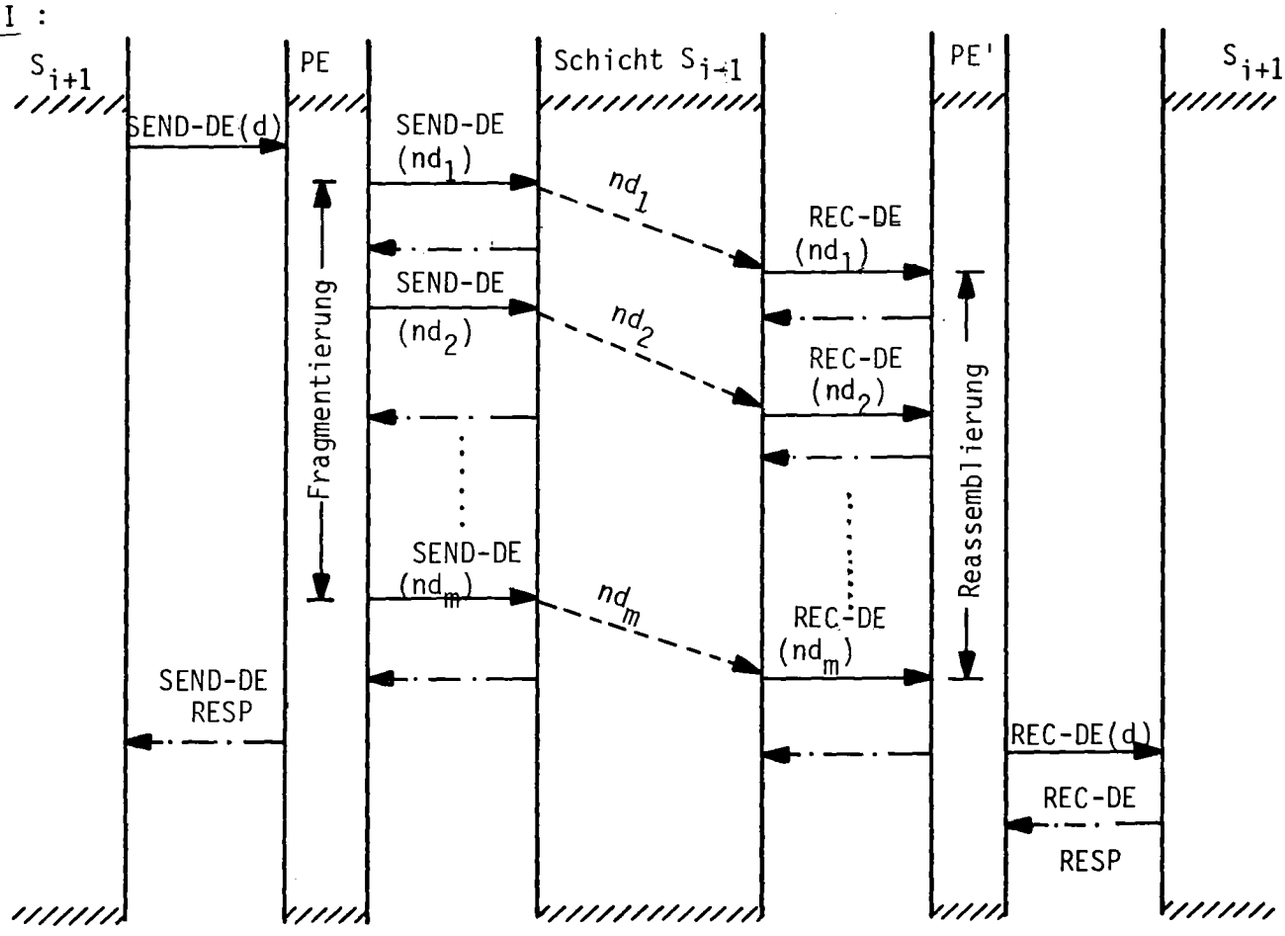


Abb. 4.22.: Abarbeitung eines SEND-INT-Schnittstellenauftrags (MLP)

Fall I :



Fall II :

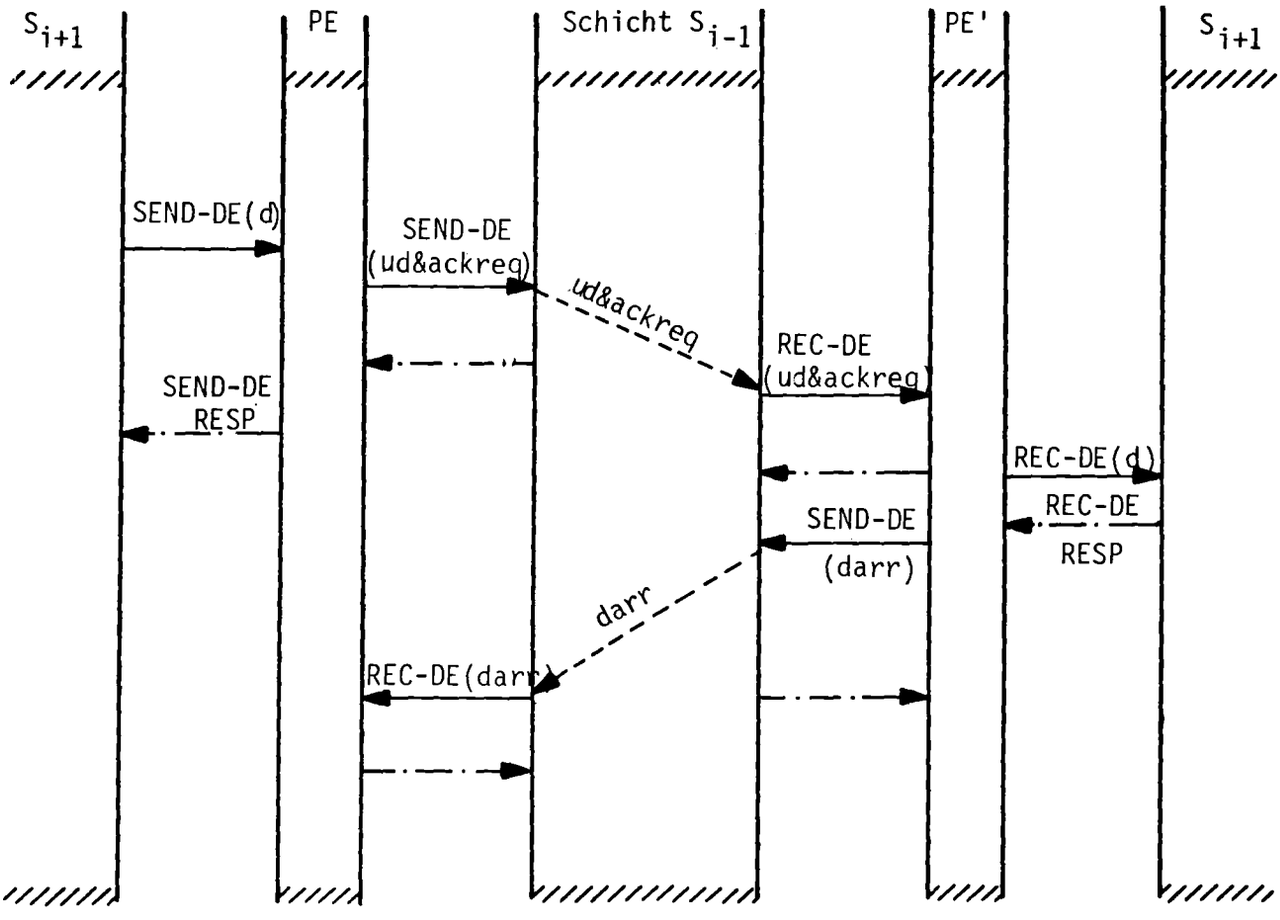


Abb. 4.23.: Abarbeitung eines SEND-DE-Schnittstellenauftrags in PE und Abarbeitung eines REC-DE in PE' (MLP; Fall I : Fragmentierung, Fall II: Quittierung)

Folgende Eigenschaften des MLP wurden modelliert:

- Fragmentierung/Reassemblierung von Dateneinheiten;
- Quittungsgenerierung für Dateneinheiten ausschließlich auf entsprechende Anforderung der zu quittierenden "message";
- getrennter Interrupt- und Datenkanal;
- expliziter Verbindungsaufbau und -abbau;
- grundsätzlich beliebig viele parallele logische Verbindungen;
- minimum MLP aus /HERT78/ ausgenommen RESUME.

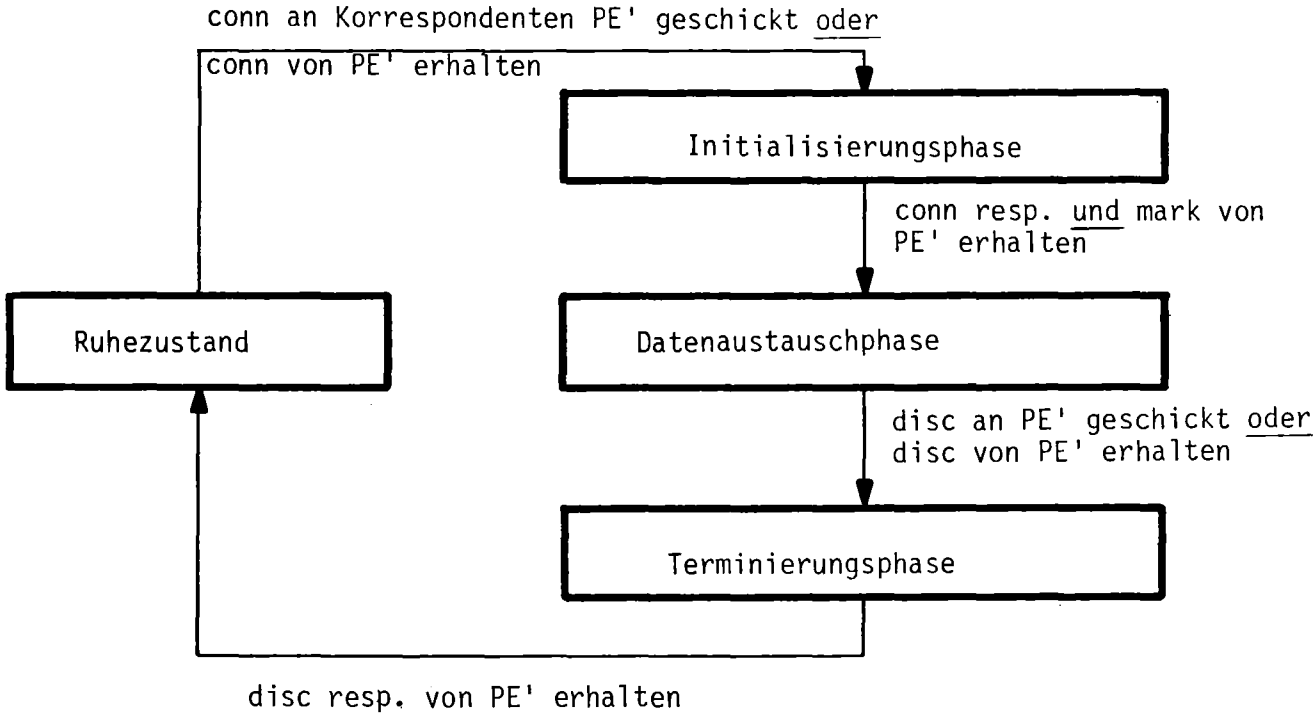


Abb. 4.24.: Die wichtigsten Protokollzustandsübergänge einer MLP-Protokolleinheit

4.3.2.5. Das Remote Job Entry Protokoll

(1) Literatur:

Es wurde ein Teil des durch den Arbeitskreis PIX definierten "Remote Job Entry"-Protokolls (RJEP) implementiert /s. HEIN78/.

(2) Benötigte Schnittstellenaufträge:

Die Schnittstellenaufträge, die zur Kommunikation zwischen einer Benutzertask und einer RJE-Protokolleinheit benötigt werden, sind

SEND-DE und REC-DE

sowie die entsprechenden Rückmeldungen.
Die Schnittstelle zwischen einer RJE-Protokolleinheit und der nächsttieferen Schicht basiert ebenfalls auf dem Austausch von SEND-DE und REC-DE.

(3),(4) Schnittstelle, die der nächsthöheren Schicht S_{i+1} angeboten wird und Schnittstelle, die von der nächsttieferen Schicht S_{i-1} vorausgesetzt wird:

Die Schnittstellen zwischen einer RJE-Protokolleinheit PE und den benachbarten Schichten sind in Abb. 4.25. dargestellt.

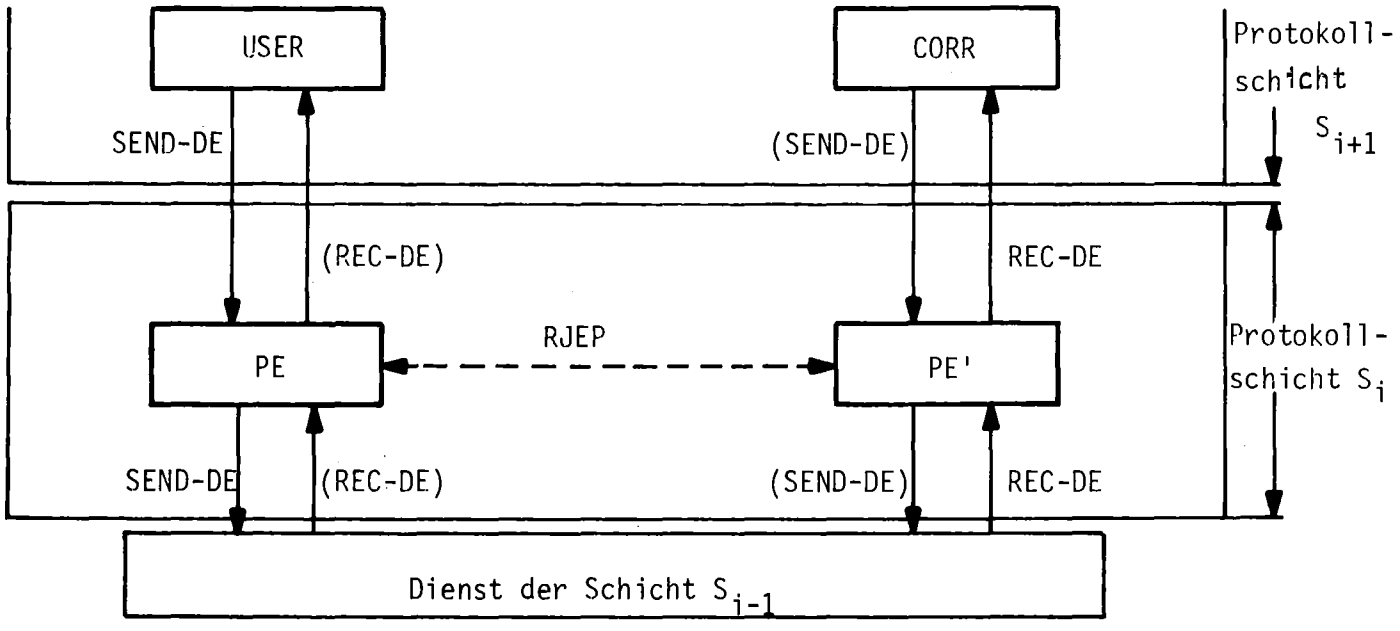


Abb. 4.25.: Schnittstellen, die zwischen RJE-Protokolleinheiten und benachbarten Protokollschichten existieren [Notation s. Abb. 4.13.]

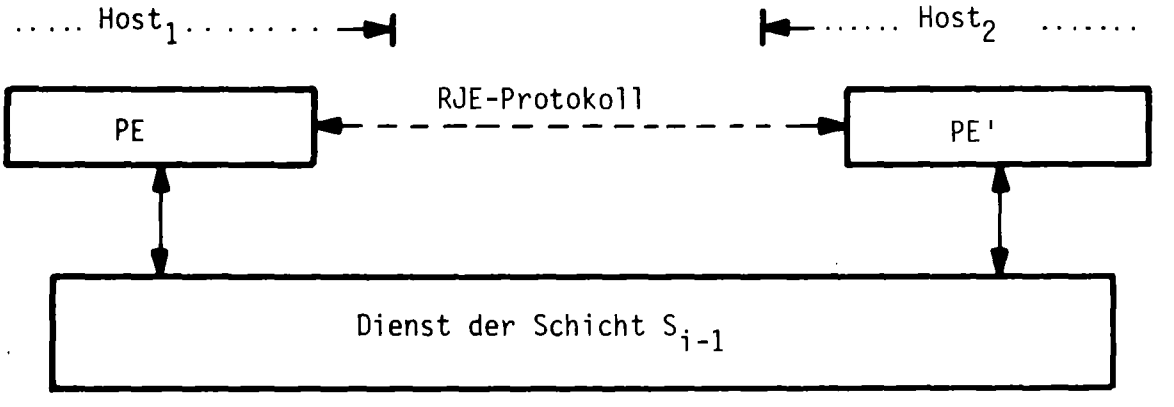
(5) Benötigte Dateneinheiten:

Folgende Typen von RJE-Dateneinheiten existieren in MOSAIC (Typ-Angabe in Klammern):

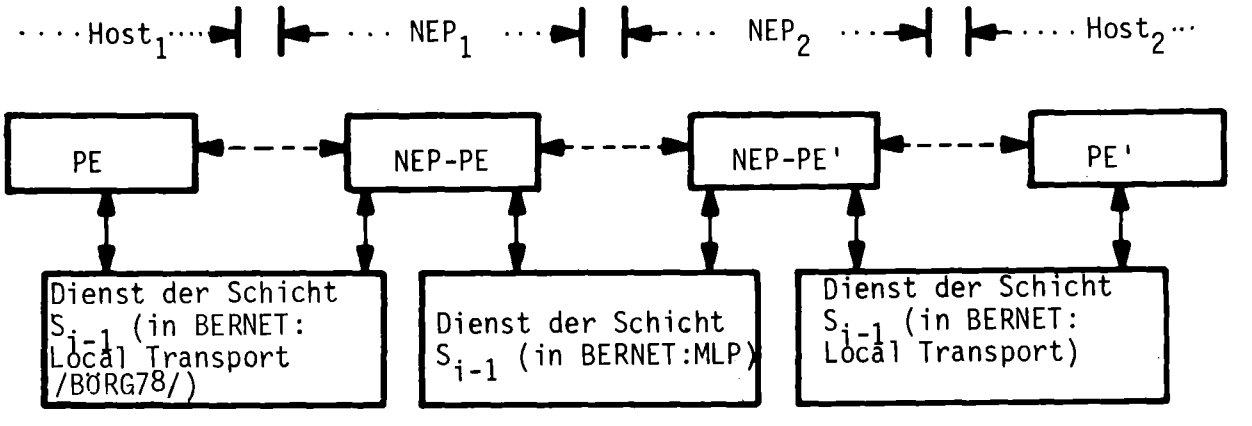
job file oder output file (6001), job/output accept request - im Host generiert (6002), job/output accept request - im NEP generiert (6012), job/output accept response positive (6003), job/output accept response negative (6004), job/output end request - im Host generiert (6005), job/output end request - im NEP generiert (6006), job/output end response (6007), abort signal (6008), repeat signal (6009), rewind signal (6010), stop signal (6011), protocol error signal (6013), resume signal (6014).

Bemerkungen:
- Für die aktuelle Implementierung sind nur die Dateneinheitentypen 6001, ... ,6007 und 6012 von Bedeutung.
- Wie der interessierte Leser bereits aus den Typen der Dateneinheiten ersieht, wurden in MOSAIC zwei Versionen eines RJE-Protokolls implementiert:

Fall I : RJE direkt zwischen Hosts:



Fall II : RJE unter Benutzung von NEP'en (s. BERNET):



(6) Aktionen einer RJE-Protokolleinheit:

Fall I : RJE direkt zwischen Hosts:

(a) Erhalt von SEND-DE durch den Nachbarn USER (s. Abb. 4.26.):

- ein job/output accept request wird an den RJE-Korrespondenten PE' übertragen, um anzufragen, ob Bereitschaft zum Empfang des vorliegenden job/output file gegeben ist;
- mittels eines job/output accept response teilt PE' seine Empfangsbereitschaft mit;
- bei Erhalt eines job/output accept response positive kann der job file/output file für den Nachbarn USER an PE' übertragen werden (wobei evtl. eine Fragmentierung zu erfolgen hat);
- nach Beendigung der Job-/Outputübertragung wird von PE an PE' ein job/output end request übermittelt und von PE' mit einem job/output end response beantwortet.

(b) Erhalt von REC-DE durch einen Nachbarn der nächsttieferen Schicht S_{i-1} (s. Abb. 4.26.):

- die referenzierte Dateneinheit - die durch S_{i-1} an PE' übergeben wurde - ist zu interpretieren,und eine Benutzertask ist zu starten (falls die erhaltene Dateneinheit einen job file darstellt) bzw. der Output ist an einem peripheren Gerät auszugeben (falls die erhaltene Dateneinheit einen output file darstellt);
- dem Nachbarn der nächsttieferen Schicht wird für jede erhaltene Dateneinheit ein REC-DE RESP übergeben.

Fall II : RJE unter Benutzung von NEP' en:
Jeder RJE-Protokolleinheit im Host ist eine RJE-Protokolleinheit im NEP zugeordnet, die z.B. Probleme wie Abbildung des lokalen auf das netzweite RJE-Protokoll löst. Man kann sich die NEP-PE als verlängerten Arm der Host-PE vorstellen. Was die Aktionen einer RJE-Protokolleinheit im Host betrifft, so gilt das bei Fall I Gesagte entsprechend (s. Abb. 4.27.).

(7) Zeitliches Verhalten:

Das zeitliche Verhalten bei der Bearbeitung der Schnittstellenaufträge SEND-DE und REC-DE kann Abb. 4.26. (RJE-Version Fall I) und Abb. 4.27. (RJE-Version Fall II) entnommen werden.

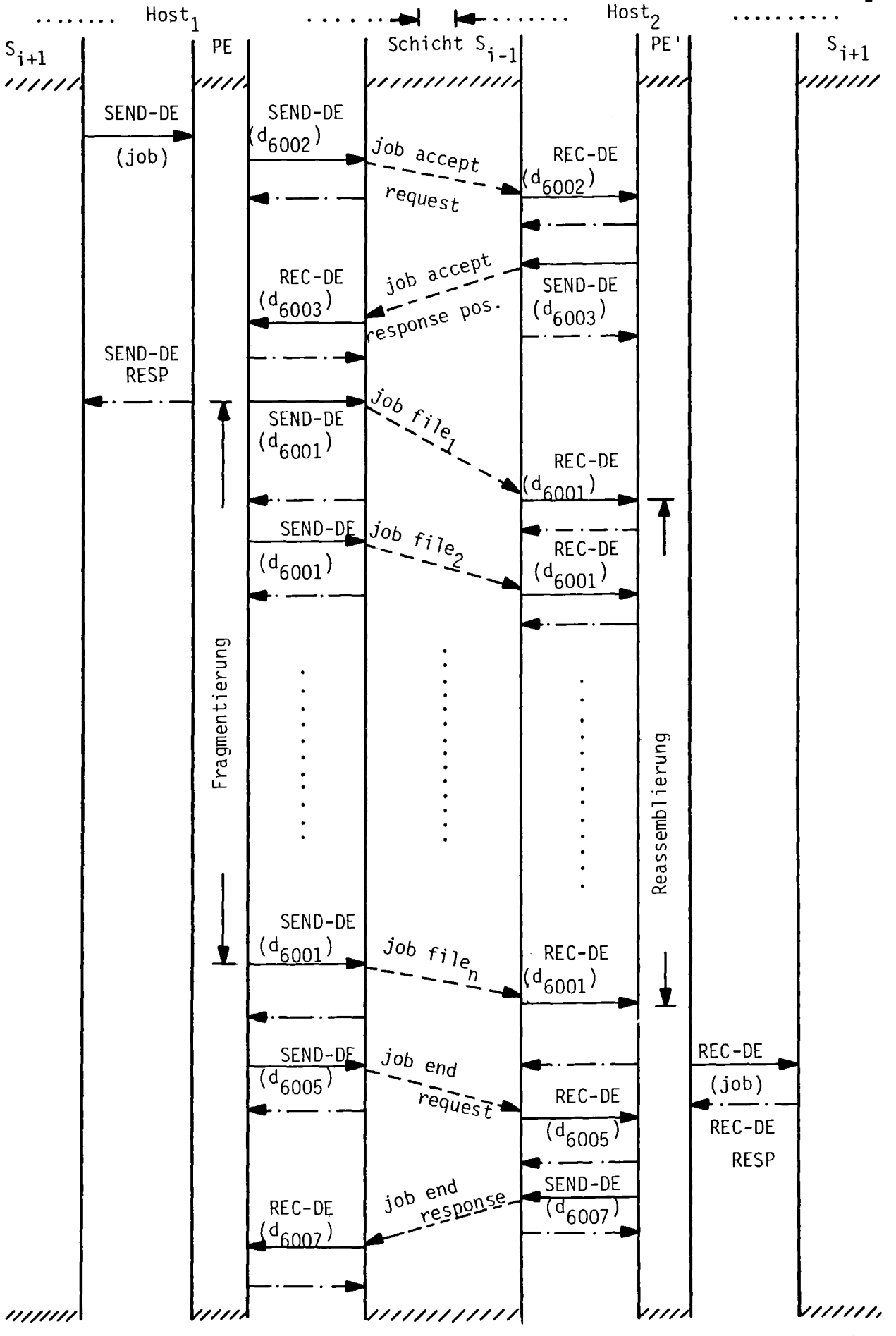


Abb. 4.26: Abarbeitung eines SEND-DE-Schnittstellenauftrags durch eine RJE-Protokolleinheit PE (RJE-Implementierung Fall I) - Übertragung eines Jobs

(Notation: d_i kennzeichnet eine Dateneinheit des Typs i)

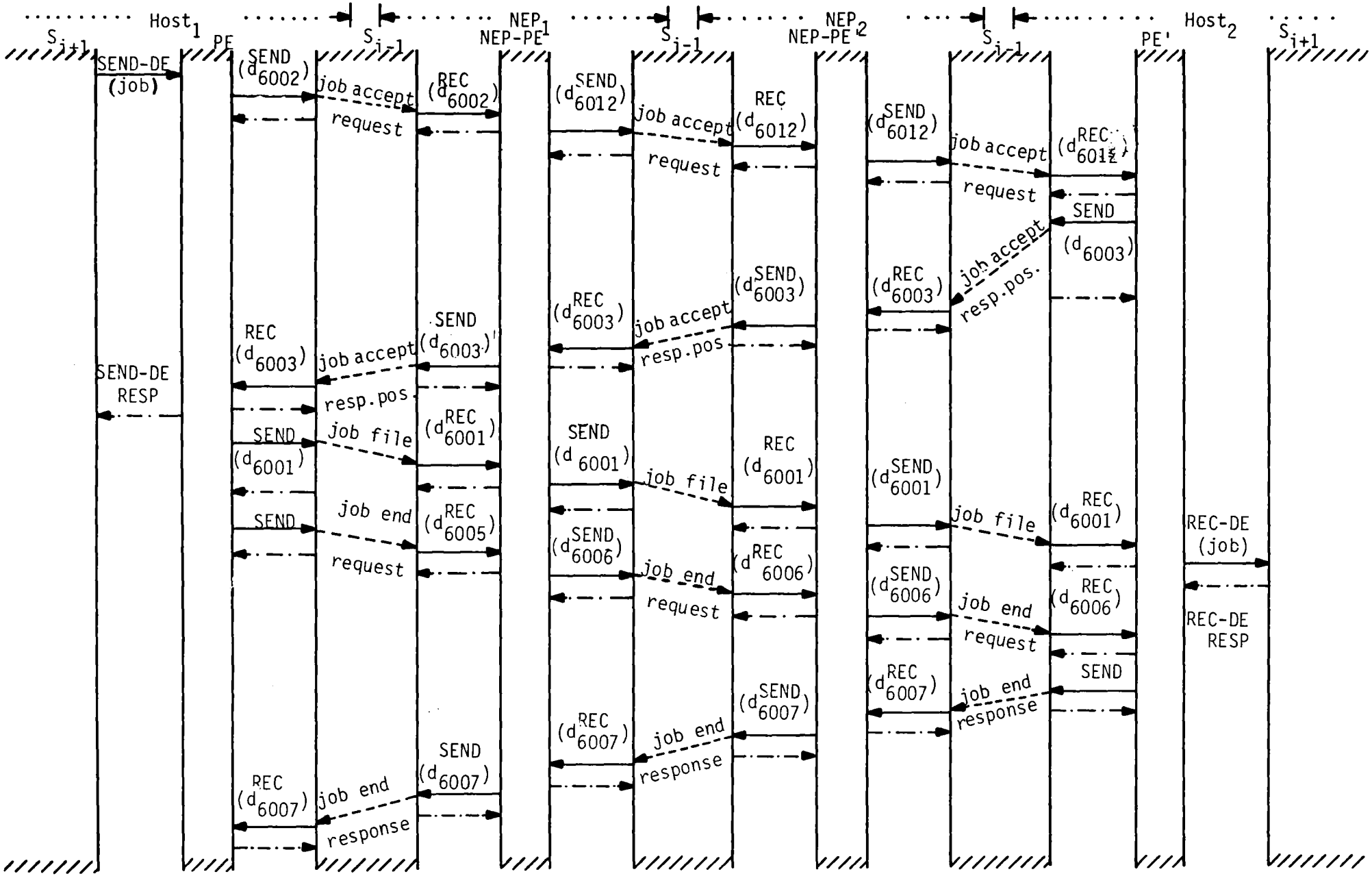


Abb. 4.27.: Abarbeitung eines SEND-DE-Schnittstellenauftrags durch eine RJE-Protokolleinheit PE (RJE-Implementierung Fall II)
- Übertragung eines Jobs (Notation: SEND = SEND-DE; REC = REC-DE)

(8) Allgemeine Bemerkungen:

Das RJE-Protokoll ist in MOSAIC durch NUMPROTOCOL = 60 gekennzeichnet. Die RJE-Dienste JOBINPUT und JOBOUTPUT können in jedem Rechner durch genau eine Protokolleinheit angeboten werden (s. Gegebenheiten in BERNET).

RJE kann verschiedene Benutzer (MAXUSERS) und verschiedene Korrespondenten (NUMCORR) unterscheiden. Auch für RJE definiert das Tupel (Benutzer, Korrespondent) eine logische Kommunikationsbeziehung und der Protokollzustand wird geführt für jede dieser Kommunikationsbeziehungen (s. Abb. 4.28.).

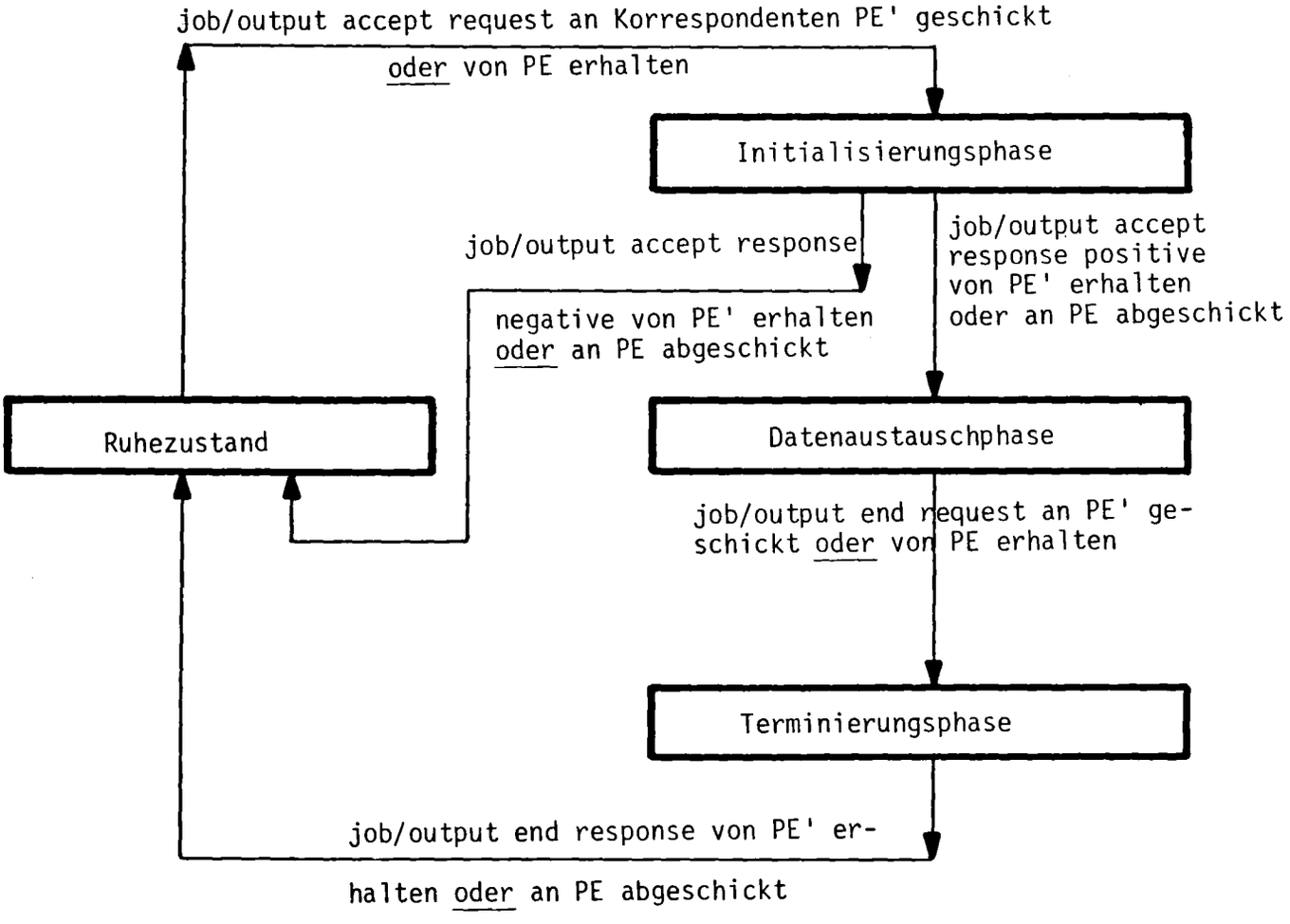


Abb. 4.28.: Die wichtigsten Protokollzustandsübergänge einer RJE-Protokolleinheit (PE bzw. PE')

4.3.2.6. Weitere Kommunikationsprotokolle

Die restlichen in BERNET verwendeten Kommunikationsprotokolle lassen sich durch geeignete Wahl der Parameter des Basisprotokolls approximieren.

4.4. Modellierung der Aufträge im BERNET-Rechnernetz

Die realitätsgetreue Nachbildung des Auftragsprofils in den einzelnen Rechnernetzknoten stellt einen kritischen Faktor bei der Modellierung von Rechnernetzen dar. Aus diesem Grund bietet MOSAIC Möglichkeiten, Aufträge in präziser Weise zu beschreiben und die Generierungszeitpunkte für Aufträge flexibel vorzugeben.

Abhängig vom Detaillierungsgrad der Simulationsstudie kennt MOSAIC zwei grundlegend verschiedene Typen von Rechnernetzaufträgen, die zu generieren sind:

(a) Benutzertasks (falls die Simulation die Benutzerschicht mit einschließt),

(b) Schnittstellenaufträge (falls ausschließlich Protokollschichten ohne die Benutzerschicht in die Betrachtung einbezogen werden).

4.4.1. Benutzertasks

Zunächst können Benutzertasks in bezug auf ihren Betriebsmittelbedarf festgelegt werden (s. Abschnitt 3.2.2.), z.B. hinsichtlich

- der Sequenz der sukzessive angeforderten Betriebsmittel (wie CPU,Speicher, periphere Geräte, Netzdienste), und
- des Umfangs des Betriebsmittelbedarfs (z.B. Länge der benötigten Rechenzeit bei CPU-Belegung).

Ein wichtiges Unterscheidungsmerkmal für Benutzertasks liegt darin , ob eine Task ausschließlich lokalen Charakter besitzt (d.h. lokal verfügbare Ressourcen beansprucht) oder ob sie auch Rechnernetzdienste anfordert. Das Verhältnis lokale Benutzertasks: kommunizierende Benutzertasks wird in MOSAIC durch Vorgabe einer (rechnerspezifischen) Wahrscheinlichkeit gesteuert. Ferner sind verschiedene Typen von kommunizierenden Benutzertasks bekannt, deren Erzeugung ebenfalls durch Vorgabe einer Wahrscheinlichkeit determiniert ist.

In MOSAIC(BERNET) sind neben den lokalen Benutzertasks (Typ = 0) aktuell folgende Typen kommunizierender Benutzertasks vorgesehen:

- Typ 1 : Benutzertask, die Schnittstellenaufträge des Typs SEND-DE generiert und Schnittstellenaufträge des Typs REC-DE entgegennimmt und zwar in beliebiger Reihenfolge;
- Typ 2 : Benutzertask, die abwechselnd einen Schnittstellenauftrag des Typs SEND-DE generiert und anschließend einen Schnittstellenauftrag des Typs REC-DE entgegennimmt;
- Typ 3 : Benutzertask, die ausschließlich SEND-DE-Schnittstellenaufträge generiert;
- Typ 4 : Benutzertask, die ausschließlich REC-DE-Schnittstellenaufträge entgegennimmt.

Bei der Erzeugung kommunizierender Benutzertask-Paare wird sichergestellt, daß eine korrekte Typenzuordnung für jeweilige Korrespondenten erfolgt, z.B. kann eine Typ 3-Benutzertask mit einer Typ 4- (nicht aber mit einer Typ 3-) Benutzertask kommunizieren.

Für eine Dialog-Anwendung erscheinen Benutzertasks des Typs 2 als Korrespondenten sinnvoll; bei einer RJE-Anwendung ist eine der beiden korrespondierenden Benutzertasks vom Typ 3, die andere vom Typ 4.

Eine Limitierung der Menge der Benutzertasks wird durch Berücksichtigung verschiedener Schwellwerte bei der Taskerzeugung erreicht, und zwar ist die Anzahl der kommunizierenden Benutzertasks limitiert, die sich zu einem Zeitpunkt

- im gesamten Rechnernetz befinden können
- in einem Rechnernetzknoten aufhalten können
- in einem Rechnernetzknoten einen Dienst D der höchsten Schicht benutzen können.

Für die Generierungszeitpunkte von Benutzertasks können sämtliche Verteilungen verwendet werden, die die Programmiersprache SIMULA anbietet (z.B. Erlang-Verteilungen). Es wäre jedoch auch problemlos möglich, sämtliche Generierungszeitpunkte direkt von einer Eingabedatei einzulesen.

Um die Einschwingphase eines Simulationsexperimentes zu verkürzen, bietet MOSAIC(BERNET) die Möglichkeit, die gesamte Grundlast, die in den Rechnernetzknoten zu erwarten ist, bereits zum Simulationszeitpunkt t=0 vorab zu erzeugen (erst die spätere Benutzertaskerzeugung wird dann durch Verteilungen gesteuert).

4.4.2. Schnittstellenaufträge

Falls Schnittstellenaufträge nicht durch Benutzertasks sondern durch den MOSAIC-Auftragsgenerator erzeugt werden, ist dafür zu sorgen, daß ausschließlich solche Schnittstellenaufträge generiert werden, die von den Protokolleinheiten der höchsten Schicht bearbeitet werden können.

In MOSAIC(BERNET) hat der Experimentator die Möglichkeit, für die Generierung von Schnittstellenaufträgen typenspezifische Wahrscheinlichkeiten vorzugeben, und zwar existieren Wahrscheinlichkeiten für die Erzeugung von OPEN, SEND-DE, SEND-INT und CLOSE.

Für die Vorgabe der Generierungszeitpunkte von Schnittstellenaufträgen gilt das bei Benutzertasks Gesagte entsprechend.

5. Schlußwort

In der vorliegenden Arbeit wurde das Modellierungssystem MOSAIC(BERNET) vorgestellt, das eine Untersuchung der Kommunikationsflüsse im BERNET-Rechnernertz durch rechnergestützte Simulation gestattet. Bei der Entwicklung von MOSAIC (BERNET) wurde besonderer Wert auf eine detaillierte Berücksichtigung der Kommunikationssoftware des BERNET gelegt. Darüber hinaus wurde versucht, wiederverwendbare Bausteine zu erarbeiten, die bei der Modellierung anderer Rechnernetze direkt benutzt werden können. Eine benutzerfreundliche Handhabbarkeit des Modellierungssystems wurde angestrebt und durch Bereitstellung eines interaktiven, graphischen Benutzerinterfaces erreicht.

MOSAIC(BERNET) gestattet Aussagen über Durchsatz, Warte- und Verweilzeiten für Dateneinheiten und Benutzertasks, Speicherbelegungen, Auslastungen der Rechnernetzkomponenten u.ä. in Abhängigkeit von Protokollparametern, unterschiedlicher Rechnernetzbelastungen, verschiedenartiger Rechnernetztopologien, von Übertragungskanalcharakteristika etc. Dank der Ermöglichung eines variablen Detaillierungsgrades reicht das Anwendungsspektrum für MOSAIC(BERNET) von der Untersuchung eines speziellen Kommunikationsprotokolls bis hin zur Untersuchung der gesamten Protokollhierarchie unter Miteinbeziehung der auf den Rechnernetzknoten ablaufenden Benutzertasks.

Das Laufzeitverhalten von MOSAIC für BERNET-Konfigurationen kann als äußerst zufriedenstellend bezeichnet werden (CPU-Zeiten für Simulationsexperimente auf einer IBM 3033 waren nahezu immer kleiner als die abgebildete Realzeit), wohingegen der Speicherbedarf - insbesondere als Konsequenz der Forderung nach allgemeiner Anwendbarkeit für MOSAIC - nicht unerheblich ist (ca. 600-800 [kByte] auf der IBM 3033 für komplette BERNET-Konfigurationen).

Die Modellvalidation wird bei Erhalt weiterer Meßdaten über das zeitliche Verhalten des realen Rechnernetzes fortgesetzt werden.

Aktuelle Erweiterungen des Modellierungssystems MOSAIC bestehen in der Anpassung an bestimmte Klassen lokaler Rechnernetze und in der Erzeugung einer MOSAIC-Version zur Simulation der Rechnernetzarchitektur DECNET des Rechnerherstellers Digital Equipment.

6. Literaturverzeichnis

/AEG/ AEG-Telefunken
Prozeßrechner AEG 80-40 / 80-60 MARTOS
- Systemdienstaufrufe -

/BÖRG78/ J. Börger
A concept to connect CDC host computers to BERNET, concerning primarily the RJE service.
GI-Tagung (Berlin, 1978)

/CCITT77/ CCITT
Recommendation X.25: Interface between data terminal equipment (DTE) and data circuit terminating equipment (DCE) for terminals operating in the packet mode on public data networks.
Orange Book, Vol. VIII.2, (Genf, 1977)

/CUNN77/ I.M. Cunningham, W.J. Older, A.K. Trivedi
DATAPAC software architecture.
Proc. COMPSAC77 (Chicago, 1977), S. 752-758

/DAHL68/ O.J. Dahl, B. Myhrhaug, K. Nygaard
SIMULA 67 - Common base language.
Norsk Regnesentral (Oslo, 1968)

/DROB74/ O. Drobnik, F. Schumacher, R. Senger
Simulation von Warteschlangensystemen - Programmsystem zur Stichprobenerhebung und -auswertung.
Kernforschungszentrum Karlsruhe KfK 1979 (Karlsruhe, 1974)

/HEIN78/ W. Heinze, B. Struif, M. Wilhelm
The PIX-RJE protocol.
PIX/RJE/TEK/78/01, GMD (Darmstadt, 1978)

/HERT78/ F.R. Hertweck, E. Raubold, F. Vogt
The ML - protocol - description.
PIX/HLP/TAG/78/01, GMD (Darmstadt, 1978)

/ISO77/ International Organization for Standardization
ISO/TC97/SC 6
HDLC - Proposed balanced class of procedures.
Dokument Nr. ISO/TC97/SC6 N1444 (1977)

/ISO79/ International Organization for Standardization
ISO/TC97/SC 16
Reference model of open systems interconnection.
Dokument Nr. ISO/TC97/SC16 N227 (1979)

/MÜLL80/ T. Müller
Interaktives graphisches Interface für ein Modellierungssystem zur Simulation von Kommunikationsflüssen in Rechnernetzen.
Diplomarbeit (Karlsruhe, 1980)

/NIED79/ J. Niedereichholz
Kostenvergleich der Simulationssprachen GPSS 1100 und SIMULA I.
Angew. Informatik 1 (1979), S. 1-8

/SIEM/ SIEMENS
Virtual memory operating system (VMOS); system management.
Reference Manual: Siemens System 4004 (1973)

/STRA78a/ H.W. Strack-Zimmermann, M. Wilhelm
BERNET-The Berlin (West) computer network.
EUROCOMP 1978 (London, 1978)

/STRA78b/ H.W. Strack-Zimmermann, F.H. Vogt, M. Wilhelm
X.25 and datagram based networks in Berlin (West).
ICCC (Kyoto, 1978)

/WOLF78/ B. Wolfinger, O. Drobnik
Simulation of protocol layers of communication in computer networks.
in: S. Schoemaker (ed.) "Computer networks and simulation", North-Holland Publishing Company (Amsterdam,New York, Oxford, 1978) S. 119-140

/WOLF79b/ B. Wolfinger
Modelle zur rechnergestützten Simulation von Kommunikationsflüssen in Rechnernetzen.
Dissertation (Karlsruhe,1979)

/WOLF79c/ B. Wolfinger
An interactive graphical interface to a computer network modeling system.
Proc. 7th SIMULA Users' Conference (Cernobbio, 1979), S. 43-51

Folgende Primärberichte enthalten unveröffentlichte Informationen von vorläufigem und betriebsinternem Charakter. Eine zur Verfügungstellung der Berichte ist nach entsprechender einzelvertraglicher Vereinbarung über die Nutzung des darin enthaltenen know how (know-how-Vertrag) möglich.
Entsprechende Anfragen sind an die Stabsabteilung Patente und Lizenzen des KfK zu richten.

/WOLF79a/ B. Wolfinger
MOSAIC - Ein Modellierungssystem zur rechnergestützten Simulation von Kommunikationsflüssen in Rechnernetzen.
Bericht-Nr.: 09.01.01P07A, KfK (Karlsruhe, 1979)

/WOLF80/ B. Wolfinger, H. Marker
Das Modellierungssystem MOSAIC und seine Anwendung zur Simulation der Kommunikationsflüsse im BERNET-Rechnernetz.
Bericht-Nr.: 09.01.01P10A, KfK (Karlsruhe, 1980)

Anhang A:

Benutzeranleitung für MOSAIC(BERNET)

MOSAIC-Eingabebeschreibung

In der nachfolgenden Beschreibung der Eingabedaten fuer das Modellierungssystem MOSAIC wird zunaechst die generelle Vorgehensweise bei der Eingabe dargelegt. Im Anschluss daran erfolgt eine Detaillierung der verschiedenen Eingabe- Formate.

```
**********************************
*                                *
* 1. Strukturierung der Eingabe *
*                                *
**********************************
```

HAUPTPROGRAMM :
==========

A. Grundlegende Information in bezug auf das geplante Simulationsexperiment erhaelt MOSAIC durch Einlesen der KARTEN 1.1. und 1.2. im aeussersten Block des Modellierungssystems. Diese erste Information dient u.a. zur Festlegung der Dimension von Vektoren in tiefer geschachtelten Bloecken. Alle weiteren Daten werden durch die Prozedur INITIALIZE (Konfigurator) entgegengenommen.

Prozedur INITIALIZE :
==========
Die Eingabe ist in der folgenden Sequenz durchzufuehren :

B. Durch die KARTE 2.1. werden zuerst Parameter zur Charakterisierung des grundsaetzlichen Aufbaus der Protokollschichten vorgegeben.

C. Anschliessend werden -nach fortlaufender Rechnernummer (s.u.)- fuer jeden Rechnernetzknoten RCOMP<i>,(i=1,...Anzahl der Rechner) saemtliche Karten der Serie 3 eingelesen :

- (a) KARTE 3.1. zur Vorgabe allgemeiner Rechnerparameter (incl. Parameter zur Charakterisierung der Belastung) fuer RCOMP<i>.
- (b) KARTE 3.2. zur Kennzeichnung der Schnittstellenauftraege zur Inanspruchnahme der durch Rechner RCOMP<i> angebotenen Rechnernetzdienste.
- (c) Innerhalb von RCOMP<i> fuer jede Protokollschicht REFLAY<j>, (j=1,...Anzahl der Schichten in RCOMP<i>) :
  - (c1) KARTE 3.3. zur Charakterisierung der Schicht REFLAY<j>
  - (c2) Innerhalb von REFLAY<j> fuer jede Protokolleinheit PE<k>,(k=1,...Anzahl der Protokolleinheiten in REFLAY<j>) :
    - (c2.1) KARTE 3.4. zur Vorgabe der allgemeinen Parame-

ter fuer PE<k>.
(c2.2) KARTE 3.5.1. und 3.5.2. zur Charakterisierung des Sende-Protokollmoduls innerhalb der Protokolleinheit PE<k>.
(c2.3) KARTE 3.6. zur Charakterisierung des Empfangsprotokollmoduls innerhalb der Protokolleinheit PE<k> .

D. Fuer jeden der 'Rechner mit Vermittlungsfunktionen' (Definition s. Bemerkung(1)) wird die KARTE 4.1. eingelesen, die zur Vorgabe der Rechnernetzvermaschung dient. Die Eingabe erfolgt (wie bei C) nach fortlaufender Nummer der Rechner.

E. Nach der Kennzeichnung saemtlicher Rechnernetzknoten wird fuer jeden der Uebertragungskanaele RCHAN<l>,(l=1,...Anzahl der Kanaele) jeweils die KARTE 5.1. zur Vorgabe der Kanalparameter eingelesen.

F. Abschliessend werden durch Einlesen der KARTE 6.1. einige Parameter zur Spezifikation der Durchfuehrung des geplanten Simulationsexperiments gesetzt.

Bemerkungen :
---

(1) Die Eingabe der Rechnerparameter in C und der Rechnernetztopologie in D setzt das Vorhandensein eines Numerierungsschemas voraus. Als Basis hierfuer dienen Zerlegungen der Menge der Rechnernetzknoten in disjunkte Teilmengen, die auf zwei unterschiedliche Arten vorgenommen werden:

Einerseits definieren wir Klassen aequivalenter Rechner (Anzahl der Klassen = NUMCTYPE), wobei wir zwei Rechner aequivalent nennen, sofern der Aufbau ihrer gesamten Protokollhierarchie identisch ist.

Andererseits ist es aus modellierungstechnischen Gruenden vorteilhaft, saemtliche Rechnernetzknoten zusaetzlich in eine der beiden Kategorien 'reine Arbeitsrechner' (Anzahl = NUMHOST) oder 'Rechner mit Vermittlungsfunktionen' (Anzahl = NUMCP) einzuordnen. Ziel dieser zweiten Zerlegung ist insbesondere eine Reduktion der Menge der zu definierenden Kanaele. Diese Reduktion ist umso groesser je umfangreicher die Menge der 'reinen Arbeitsrechner' gewaehlt werden kann. Einzige Nebenbedingung dieser (etwas willkuerlichen) Zerlegung ist die Voraussetzung, dass an jedem Uebertragungskanal zumindest ein 'Rechner mit Vermittlungsfunktionen' angeschlossen sein muss.

Als Beispiel fuer eine Zerlegung der zweiten Art wollen wir ein Rechnernetz mit sternfoermiger Topologie betrachten : Die guenstigste Aufgliederung in diesem Fall besteht darin, den Rechner im Zentrum des Rechnernetzes als einzigen 'Rechner mit Vermittlungsfunktionen' zu definieren.

Nachdem nun die beiden Arten der Zerlegung eines Rechnernetzes eingefuehrt wurden, koennen wir uns der Numerierung der Rechnernetzknoten zuwenden : Die Rechner werden durchgehend numeriert und zwar derart, dass

(a) an 'reine Arbeitsrechner' die niedrigen Nummern (1,...NUMHOST), an 'Rechner mit Vermittlungsfunktionen' die hohen Nummern (NUMHOST+1,...NUMHOST+NUMCP) zugeteilt werden, und
(b) an Rechner derselben Klasse fortlaufende Nummern zugewiesen werden, wobei die Reihenfolge der Klassen beliebig ist, sofern Bedingung(a) nicht verletzt wird.

(2) Die Eingabe der Kanalparameter in E erfolgt sukzessive fuer jeden der 'Rechner mit Vermittlungsfunktionen' (nach aufsteigender Rechnernummer) und zwar jeweils fuer saemtliche Rechnernetzknoten mit niedriger Rechnernummer.

```
**********************
*                    *
* 2. Eingabeformate *
*                    *
**********************
```

Vorbemerkungen :

(a) Zur Kennzeichnung der Variablentypen werden folgende Abkuerzungen verwendet

-I-	fuer INTEGER-Variablen
-R-	fuer REAL-Variablen
-B-	fuer BOOLEAN-Variablen (Eingabe von 1 bei Variablenwert = TRUE)
-IA-	fuer Variablen vom Typ INTEGER ARRAY
-RA-	fuer Variablen vom Typ REAL ARRAY

(b) Die verwendeten Dimensionen sind generell

<msec>	fuer Zeit-Parameter
<bit>	fuer Dateneinheiten-, Headerlaengen etc.
<bit/msec>	fuer Uebertragungsgeschwindigkeiten, Fehlerraten etc.

MOSAIC-Eingabebeschreibung

KARTE 1.1. : (Grundlegende Information, 1. Teil)
****

NUMCTYPE	-I-	Anzahl der Klassen aequivalenter Rechner (Definition s. o.) .
MAXCOMMUSER	-I-	Obere Grenze fuer die Gesamtzahl der kommunizierenden Benutzertasks, die sich zu demselben Zeitpunkt im Rechnernetz aufhalten koennen.
NUMHLPUCL	-I-	Anzahl von verschiedenen Protokolleinheiten-Klassen (PUCLASS) in der hoechsten Schicht (MAXLAY) .
INTACT	-B-	Variable zur Spezifikation des Wunsches nach interaktiver Durchfuehrung des Simulationsexperiments.
SCARD	-I-	Max. Anzahl von wartenden Dateneinheiten in der SENDQ-Warteschlange der hoechsten Protokollschicht.

Bemerkungen :
---

1. Es wird vorausgesetzt, dass 0 < NUMCTYPE < 10.

2. MAXCOMMUSER muss geradzahlig sein, da eine paarweise Kommunikation zwischen Benutzertasks vorausgesetzt wird.

3. Bei interaktiver Experimentdurchfuehrung ist INTACT=true einzugeben; ansonsten handelt es sich um ein Batch-Experiment.

MOSAIC-Eingabebeschreibung

KARTE 1.2. : (Grundlegende Information, 2. Teil)
****

NUMC(1:NUMCTYPE)	-IA-	Anzahl der Elemente in jeder der Klassen aequivalenter Rechner.
NUMLAY(1:NUMCTYPE)	-IA-	Anzahl der Protokollschichten fuer saemtliche Rechner einer gemeinsamen Rechnerklasse.
NUMCPLOW	-I-	Niedrigste Rechnernummer in der Menge der 'Rechner mit Vermittlungsfunktionen' (Definition s.o.).
PERIOD	-R-	Laenge des Zeitintervalls fuer das periodisch zu wiederholende Abspeichern von Experimentresultaten auf einem externen Datentraeger (z.B. Platte).

Bemerkungen :
---

1. NUMCOMP:= NUMC(1)+ ... +NUMC(NUMCTYPE) .

2. NUMHOST:= NUMCPLOW -1 .

3. Voraussetzung ist, dass 1 <= NUMCPLOW <= NUMCOMP .

4. Die Anzahl der 'Rechner mit Vermittlungsfunktionen' ergibt sich aus
   NUMCP:= NUMCOMP - NUMHOST .

5. Fuer die Anzahl der zu definierenden Uebertragungskanaele gilt die Gleichung
   NUMCHAN:= NUMCP * NUMHOST + NUMCP * (NUMCP-1)/2 .

MOSAIC-Eingabebeschreibung

KARTE 2.1. : (Genereller Aufbau der Protokollschichten)
****

NUMPU(I,J)	-IA-	Anzahl der Protokolleinheiten fuer Rechner der Klasse I innerhalb der Protokollschicht J .

Bemerkungen :
---

1. Die Reihenfolge des Einlesens der Werte ist derart, dass sukzessive fuer jede der Klassen (nach aufsteigender Numerierung) die Anzahl der Protokolleinheiten innerhalb saemtlicher Protokollschichten (beginnend bei der niedrigsten Schicht) eingegeben wird.

2. Fuer alle I,J gilt : NUMPU(I,J) >= 1 .

MOSAIC-Eingabebeschreibung

KARTE 3.1. : (Allgemeine Rechnerparameter)
****

NCPUS	-I-	Anzahl der Zentraleinheiten im Rechner .
TIMESLICE	-R-	Dauer einer Zeitscheibe bei der Zuordnung der CPU an eine Task .
MINPRIOR MAXPRIOR	-I- -I-	Bei der Kreierung einer Task (durch den Belastungsgenerator) wird dieser eine Prioritaet aus dem Intervall <MINPRIOR, MAXPRIOR> zugewiesen .
LAYLOW	-I-	Bei Datenaustausch zwischen Benutzertasks in demselben Rechner : Nummer der Protokollschicht in welcher Dateneinheiten von einem Sende-Protokollmodul direkt zu einem Empfangs-Protokollmodul weitergereicht werden koennen.
CPUTMIN CPUTMAX	-I- -I-	Die Dauer saemtlicher aufeinanderfolgender Rechenanweisungen wird bei Eintritt in den internen Benutzertask-Zustand S1' bzw. S1'' aus dem Intervall <CPUTMIN,CPUTMAX> gezogen.
IOTMIN IOTMAX	-I- -I-	Die Dauer der Ein-/Ausgabeanweisungen wird bei Eintritt in den internen Benutzertask-Zustand S2 aus dem Intervall <IOTMIN, IOTMAX> gezogen.
MINLENG MAXLENG	-I- -I-	Bei der Generierung einer Dateneinheit (durch eine Benutzertask bzw. durch den Belastungsgenerator) wird dieser eine Laenge aus dem Intervall <MINLENG,MAXLENG> zugewiesen .
MEAN	-R-	Mittlere Zeit zwischen jeweils zwei aufeinanderfolgenden Generierungen von Benutzertasks bzw. Schnittstellenauftraegen durch den Belastungsgenerator im Rechner .
TPROB(1)	-R-	Wahrscheinlichkeit fuer die Generierung lokaler Benutzertasks durch den Belastungsgenerator.
TPROB(2)	-R-	Wahrscheinlichkeit fuer die Generierung kommunizierender Benutzertasks durch den Belastungsgenerator.
PRACKML	-R-	Wahrscheinlichkeit fuer die Erzeugung von Quittungen (MLP-spezifisch) .
TOVHR	-R-	Overhead im Rechner bei einer Eingabe von einem Kanal.

MOSAIC-Eingabebeschreibung

TOVHS	-R-	Overhead im Rechner bei einer Ausgabe zu einem Kanal.
MEANCPUT	-R-	Mittelwert fuer die Bestimmung der CPU-Rechenzeit einer Benutzertask (unter Beruecksichtigung einer Erlang-Verteilung).
SDEVCPUT	-R-	Standardabweichung fuer die obige Erlang-Verteilung.
MAXCUSER	-I-	Maximale Anzahl von kommunizierenden Benutzertasks,die sich zu demselben Zeitpunkt in dem Rechner aufhalten koennen.

Bemerkungen :
---

1. Da die aktuelle Implementierung von MOSAIC auf Einprozessorsysteme zugeschnitten ist, ist nur NCPUS = 1 sinnvoll.

2. Die Bestimmung der Prioritaet, der Dauer von Rechen- bzw. E/A-Anweisungen fuer Benutzertasks sowie der Laenge von Dateneinheiten aus den jeweils vorgegebenen Intervallen erfolgt unter Zugrundelegung einer Gleichverteilung.

3. Der Parameter MEAN ergibt sich als reziproker Wert der mittleren Ankunftsrate von Benutzertasks bzw. Schnittstellenauftraegen.

4. Trivialerweise gilt :
   0 <= TPROB(i) <= 1 fuer i=1,2 ;
   und 0 <= TPROB(1)+TPROB(2) <= 1 .

5. Die Wahrscheinlichkeit fuer die Generierung von Schnittstellenauftraegen ergibt sich zu :
   TPROBPMREQ = 1 - (TPROB(1) + TPROB(2)) .

6. MEANCPUT und SDEVCPUT werden in Einheiten von 100 <sec> eingelesen.

MOSAIC-Eingabebeschreibung

KARTE 3.2. : (Fortsetzung allg. Rechnerparameter und
********** Kennzeichnung der Schnittstellenauftraege)

COMMBUFFER	-I-	Groesse des gesamten fuer die Kommunikation zur Verfuegung stehenden Speichers.
PREEMPTPROB	-R-	Wahrscheinlichkeit fuer die Unterbrechbarkeit von Benutzertasks.
SSAPROB(1)	-R-	Wahrscheinlichkeit zur Erzeugung von (Schnittstellen-)Auftraegen des Typs OPEN.
SSAPROB(2)	-R-	Wahrscheinlichkeit zur Erzeugung von (Schnittstellen-)Auftraegen des Typs SENDDU.
SSAPROB(3)	-R-	Wahrscheinlichkeit zur Erzeugung von (Schnittstellen-)Auftraegen des Typs SENDINTERRUPT.
MAXPUUSER(1:NUMHLPUCL)	-IA-	Max. Anzahl von Benutzertasks, die eine Protokolleinheit benutzen koennen (NUMHLPUCL s. Karte 1.1.) .
PROTPROB(1:i0)	-RA-	Wahrscheinlichkeit fuer die Auswahl einer Protokolleinheit.
(Es gilt : i0 = 1 und i0 = NUMHLPUCL-1		falls NUMHLPUCL = 1 NUMHLPUCL > 1).
WABTNUM	-I-	Anzahl der Unterbrechungsanfragen durch den Master (BSC-spezifisch).
NUMSTARTTASK	-I-	Anzahl der Benutzertasks im Rechner zum Zeitpunkt t=0 .

Bemerkungen :
---

1. Die Wahrscheinlichkeit fuer die Erzeugung von Verbindungsabbau-Auftraegen (CLOSE) ergibt sich zu :
   SSAPROB(4)= 1.0 - (SSAPROB(1)+SSAPROB(2)+SSAPROB(3)) .
2. MAXPUUSER(I) gibt die max. Anzahl von Benutzertasks an, die ueber die Protokolleinheit kommunizieren, die im Rechner die PUCLASS mit der Nummer MAXLAY*100+I repraesentiert.
3. MAXPUUSER wird nur eingelesen, wenn gilt :
   NUMPU(NUMCTYPE,MAXLAY) > 1 .

4. PROTPROB(I) gibt die Wahrscheinlichkeit dafuer an, dass eine nichtlokale Benutzertask ueber die PUCLASS mit der Nummer MAXLAY*100+I kommuniziert.
5. PROTPROB(I) wird nur eingelesen, wenn gilt :
   NUMPU(NUMCTYPE,MAXLAY) > 1 und NUMHLPUCL > 1 .

6. PREEMPTPROB = 0 bedeutet : Tasks sind nicht unterbrechbar,
   PREEMPTPROB = 1 bedeutet : Tasks sind zu jedem Zeitpunkt unterbrechbar.

7. WABTNUM wird im Fall des Protokolls BSC fuer Master und Slave eingelesen; ist jedoch nur im Slave von Bedeutung.

MOSAIC-Eingabebeschreibung

KARTE 3.3. : (Kennzeichnung der Protokollschichten)
****

MAXDULENGTH	-I-	Max. Laenge von Dateneinheiten innerhalb der Protokollschicht.
HEADERLENGTH	-I-	Laenge der fuer Protokolleinheiten der Schicht relevanten Kontroll-information.
REJPROB	-R-	Wahrscheinlichkeit fuer die Verfaelschung der Uebertragung einer Dateneinheit innerhalb der Protokollschicht.

Bemerkungen :
---

1. Wenn die Laenge einer Dateneinheit beim Eintritt in die Protokollschicht den Wert MAXDULENGTH ueberschreitet, ist eine Fragmentierung der Dateneinheit notwendig.

2. Die Fragmentierung wird nur durchgefuehrt, wenn die Variable FRAGMENTNEC (siehe Karte 3.5.2.) auf true gesetzt wird.

3. REJPROB = 0 bedeutet : Dateneinheiten werden ohne Ausnahme korrekt empfangen,
   REJPROB = 1 : saemtliche Dateneinheiten werden fehlerhaft uebertragen.

MOSAIC-Eingabebeschreibung

KARTE 3.4. : (Kennzeichnung der Protokolleinheiten)
****

MASTER	-B-	Parameter zur Kennzeichnung unsymmetrischer Kommunikationsprotokolle.
NUMPROTOCOL	-I-	Nummer zur Kennzeichnung des durch die Protokolleinheit realisierten Kommunikationsprotokolls.
PUCLASS	-I-	Parameter zur Kennzeichnung der Menge der untereinander kommunizierenden Korrespondenten (in verschiedenen Rechnernetzknoten, jedoch innerhalb derselben Protokollschicht) .
MAXBLOCKL	-I-	Max. Laenge von 'Dateneinheitenbloecken' (nur bei Blockbildung durch eine Protokolleinheit!) .
CORRESP	-I-	Nummer des Rechners, der den Korrespondenten enthaelt (nur bei Kommunikation zwischen genau zwei Protokolleinheiten).
PROTOPEN	-B-	True, falls vorausgesetzt werden soll, dass das Kommunikationsprotokoll nur eine Datenaustauschphase (und keine Initialisierungs- oder Terminierungsphase) kennt.
MULTIPLEXNEC	-B-	True, falls das Multiplexen mehrerer Nachbarn einer hoeheren Schicht durch die Protokolleinheit zu realisieren ist.

MOSAIC-Eingabebeschreibung

Bemerkungen:
---

1. (a) Bei X.25 gilt MASTER=true (false) fuer die Protokolleinheit im DTE (DCE) .
   (b) Bei RJE gilt MASTER=true im NEP-Rechner, sofern an ihn keine RJE-Station angeschlossen ist.

2. Beispiele fuer Nummern, die MOSAIC zur Kennzeichnung von Protokollen kennt, sind

NUMPROTOCOL = 0	fuer	Basisprotokoll,
NUMPROTOCOL = 10	fuer	HDLC (balanced class of procedures, asynchronous response mode),
NUMPROTOCOL = 13	fuer	CCP (channel communication protocol),
NUMPROTOCOL = 15	fuer	BSC,
NUMPROTOCOL = 20	fuer	X.25 (packet level),
NUMPROTOCOL = 23	fuer	SCP (subchannel protocol),
NUMPROTOCOL = 25	fuer	RSP (resource sharing protocol),
NUMPROTOCOL = 30	fuer	MLP (PIX-message link protocol),
NUMPROTOCOL = 60	fuer	RJE (PIX-remote job entry protocol).

3. In jedem Rechner darf pro Schicht ein Wert von PUCLASS nicht mehrmals auftreten.

4. Blockbildung dient dazu, Dateneinheiten geringer Laenge, die an denselben Korrespondenten adressiert sind, zu einem 'Dateneinheitenblock' zusammenzufuegen. Das Fragmentieren eines 'Dateneinheitenblockes' innerhalb einer tieferen Protokollschicht ist im Fall der Blockbildung zu vermeiden.

5. CORRESP = 0 bedeutet : die Protokolleinheit kann mit jeweils einem Korrespondenten in jedem Rechnernetzknoten kommunizieren,
   CORRESP > 0 : es existiert nur ein Korrespondent, der sich im Rechnernetzknoten mit der Rechner-Nr. CORRESP befindet.

MOSAIC-Eingabebeschreibung

KARTE 3.5.1. : (Kennzeichnung der Sende-Protokollmoduln,
********** 1. Teil)

NCORR	-I-	Anzahl der Korrespondenten.
MAXUSER	-I-	Max. Anzahl von Nachbarn einer hoeheren Protokollschicht.
NSN	-I-	Hilfsvariable zur Steuerung der Nachbarauswahl (s.u.) .
NSC	-I-	Hilfsvariable zur Steuerung der Korrespondentenauswahl (s.u.) .
SENEIGHB(i0:j0)	-IA-	Indexfeld fuer die Auswahl des Nachbarn auf der naechsttieferen Schicht beim Absenden einer Dateneinheit.
SECORR(i0:j0)	-IA-	Indexfeld fuer die Auswahl des Korrespondenten beim Absenden einer Dateneinheit.

(Es gilt: i0 = 1, j0 = NUMCOMP } falls fuer die eigene Protokolleinheit CORRESP = 0 gesetzt ist,

sowie i0 = CORRESP, j0 = CORRESP } sonst .)

CORRPROB(1:k0)	-RA-	Wahrscheinlichkeit zur Steuerung der Richtung des Dateneinheitenflusses bei Sendeauftraegen.

(Es gilt: k0 = NUMCOMP-1 } falls NCORR > 1 und LAYNUM = MAXLAY ,
k0 = 1 sonst.)

TTYPE(1:t0)	-RA-	Wahrscheinlichkeit fuer die Typ-Zuordnung einer Benutzertask.

(Es gilt: t0 = MAXTASK-1 falls LAYNUM = MAXLAY
t0 = 1 sonst.)

MOSAIC-Eingabebeschreibung

Bemerkungen :
---

1. Beim Absenden einer Dateneinheit wird primaer die Korrespondentenbestimmung durchgefuehrt und anschliessend fuer den ermittelten Korrespondenten der Nachbar festgestellt.

2. Die Korrespondentenauswahl wird durch die Variablen NSC und SECORR unterstuetzt :
   (a) NSC = 0 impliziert, dass sich der Korrespondent fuer die aktuelle Schicht in demselben Rechner befindet wie der Korrespondent fuer die naechsthoehere Schicht; wurde die abzusendende Dateneinheit allerdings in der aktuellen Schicht erzeugt, so liegt der Korrespondent trivialerweise im Zielrechner der Dateneinheit.
   (b) NSC > 0 und der Korrespondent fuer die naechsthoehere Schicht befinde sich im Rechner RECCOMP. Wir unterscheiden zwei Faelle :
   Fall 1 : SECORR(RECCOMP) = 0 impliziert,dass der Korrespondent fuer die aktuelle Schicht unter Benutzung der Prozedur SELECTCP (bei Verwendung der Matrix REACH) bestimmt wird.
   Fall 2 : SECORR(RECCOMP) > 0 impliziert,dass sich der Korrespondent fuer die aktuelle Schicht im Rechnernetzknoten mit der globalen Nummer SECORR(RECCOMP) befindet.

3. Die Auswahl des Nachbarn (Sende-Protokollmodul) innerhalb der naechsttieferen Schicht basiert auf den Variablen NSN und SENEIGHB :
   (a) NSN = 0 impliziert, dass saemtliche abzusendenden Dateneinheiten an die Protokolleinheit der naechsttieferen Schicht mit niedrigster Nummer uebergeben werden;
   (b) NSN > 0 wird derart interpretiert, dass saemtliche abzusendenden Dateneinheiten an die Protokolleinheit einer tieferen Schicht mit globaler Nummer NSN uebergeben werden;
   (c) NSN < 0 impliziert, dass die globale Nummer des Nachbarn korrespondentenabhaengig durch SENEIGHB(RECNODE) ermittelt wird.

4. Aus der Bemerkung 2 resultiert, dass der Vektor SECORR nicht fuer Protokolleinheiten einzulesen ist, die sich auf hoechster Schicht befinden, und ueberdies entfaellt, falls NSC = 0 gesetzt wurde.
   Analoges gilt fuer den Vektor SENEIGHB, der nur einzugeben ist, falls NSN < 0 und zwar generell nur fuer solche Protokolleinheiten, die sich nicht auf tiefster Schicht

befinden.

5. a) CORRPROB(I) gibt die Wahrscheinlichkeit dafuer an, dass der Korrespondent sich im Rechner mit der Nummer I befindet (falls NCORR > 1) .
   b) CORRPROB(I) gibt die Nummer des Rechners an, der den Korrespondenten enthaelt (falls NCORR = 1).
6. Im Fall 5a) gilt : 0 <= CORRPROB(I) <= 1 fuer alle I und ausserdem gilt :
   0 <= CORRPROB(1) + ... + CORRPROB(NUMCOMP-1) <= 1 .
7. Im Fall 5a) ergibt sich CORRPROB(NUMCOMP) implizit, da
   CORRPROB(1)+ ... + CORRPROB(NUMCOMP) = 1 .

8. TTYPE(I) gibt die Wahrscheinlichkeit dafuer an, dass eine Benutzertask vom Typ I ist.
9. TTYPE wird nur eingelesen fuer die hoechste Schicht (d.h. falls LAYNUM = MAXLAY); Bemerkungen 6 - 7 bitte beachten.

10. MAXTASK bezeichnet die Anzahl von implementierten Benutzertask-Typen (aktuell : MAXTASK = 4) .

MOSAIC-Eingabebeschreibung

KARTE 3.5.2. : (Kennzeichnung der Sende-Protokollmoduln,
********** 2. Teil)

CSVTSCI	-R-	Konstante Bedienungszeit fuer das Erstellen und Absenden von Kontrollinformation.
BSVTSCI	-R-	Laengenabhaengige Bedienungszeit fuer das Erstellen und Absenden von Kontrollinformation.
CSVTSDU	-R-	Konstante Bedienungszeit fuer das Erstellen und Absenden einer Dateneinheit.
BSVTSDU	-R-	Laengenabhaengige Bedienungszeit fuer das Erstellen und Absenden einer Dateneinheit.
CSVTSCOP	-R-	Konstante Bedienungszeit fuer das wiederholte Absenden einer Dateneinheit (Kopie).
BSVTSCOP	-R-	Laengenabhaengige Bedienungszeit fuer das wiederholte Absenden einer Dateneinheit (Kopie).
BLOCKINGNEC	-B-	True, falls prinzipiell ein Blockbilden fuer Dateneinheiten durch den Sende-Protokollmodul durchzufuehren ist.
FRAGMENTNEC	-B-	True, falls prinzipiell ein Fragmentieren von Dateneinheiten durchzufuehren ist.
TIMEOUTNEC	-B-	True, falls eine Zeitueberwachung fuer Sendevorgaenge durchzufuehren ist.
SCRED	-I-	'Sende-Fenstergroesse', d.h. Anzahl der Dateneinheiten, die bis zum Erhalt einer Quittung an einen Korrespondenten abgesandt werden koennen.
INTERFACE	-I-	Charakterisierung des Typs der Schnittstellen zu den Nachbarn.

MOSAIC-Eingabebeschreibung

Bemerkungen :
---

1. Die Bedienungszeiten sind so zu waehlen, dass sie den CPU-Zeiten entsprechen, die fuer die Durchfuehrung der entsprechenden Softwareaktivitaeten im realen Rechnernetz benoetigt werden. Die Genauigkeit dieser Zeiten ist von wesentlichem Einfluss auf die Guete des Modells. (Bei den laengenabhaengigen Bedienungszeiten ist die Bedienungszeit pro Bit anzugeben).

2. Um Deadlocks zu vermeiden, muss der Parameter SCRED so gewaehlt werden, dass er den Wert des Parameters fuer das Erstellen von Quittungen im Korrespondenten (s. RCRED auf Karte 3.6.) nicht unterschreitet.
   Darueberhinaus gilt : SCRED >= 1 .
   SCRED > 100 wird als unendlicher Sendekredit gewertet; die Kopienhaltung entfaellt hier (aus modellierungstechnischen Gruenden) .

MOSAIC-Eingabebeschreibung

KARTE 3.6. : (Kennzeichnung der Empfangs-Protokollmoduln)
****

NCORR	-I-	Anzahl der Korrespondenten.
MAXUSER	-I-	Max. Anzahl von Nachbarn einer hoeheren Protokollschicht.
CSVTRDU	-R-	Konstante Bedienungszeit fuer den gesamten Empfangsvorgang fuer eine Dateneinheit.
BSVTRDU	-R-	Laengenabhaengige Bedienungszeit fuer den gesamten Empfangsvorgang fuer eine Dateneinheit.
CSVTRCI	-R-	Konstante Bedienungszeit fuer die Entgegennahme und Analyse von Kontrollinformation.
BSVTRCI	-R-	Laengenabhaengige Bedienungszeit fuer die Entgegennahme und Analyse von Kontrollinformation.
RCRED	-I-	'Quittungs-Fenstergroesse', d.h. Anzahl der Dateneinheiten, die von einem Korrespondenten sukzessive empfangen werden, ehe eine Quittung zu generieren ist.
INTERFACE	-I-	Charakterisierung des Typs der Schnittstelle zu den Nachbarn.
NRN	-I-	Hilfsvariable zur Steuerung der Nachbarauswahl (s.u.) .
RECNEIGHB(i0:j0)	-IA-	Indexfeld fuer die Auswahl des Nachbarn beim Empfang einer Dateneinheit.
(i0, j0 siehe Karte 3.5.1.)		
POLLTABLE(1:NCORR)	-IA-	Indexfeld fuer die Auswahl der zu pollenden Korrespondenten (BSC-spezifisch).
WABTPROB	-R-	Wahrscheinlichkeit fuer die Rueckmeldung einer Unterbrechungsanfrage (BSC-spezifisch).

MOSAIC-Eingabebeschreibung

Bemerkungen :
---

1. Fuer die Bedienungszeiten gilt das bei Karte 3.5.2. Gesagte.

2. Die Wahl von RCRED hat der Tatsache Rechnung zu tragen, dass einerseits RCRED >= 1 und andererseits fuer saemtliche Korrespondenten SCRED >= RCRED gilt (s.o.) .

3. Die Auswahl des Nachbarn (Sende- oder Empfangs-Protokollmodul) basiert auf den Variablen NRN und RECNEIGHB :
   (a) NRN < 0 : Einlesen von RECNEIGHB (die Komponente gibt die Nummer des Sende-Protokollmoduls an, an den die Dateneinheit weitergegeben wird);
   (b) NRN = 0 : impliziert die direkte Uebergabe an den Sende-Protokollmodul derselben Protokolleinheit oder die Uebergabe an den Empfangs-Protokollmodul einer hoeheren Protokollschicht mit entsprechender PUCLASS;
   (c) NRN > 0 : Uebergabe an den Sende-Protokollmodul mit der eingelesenen Nummer.

4. POLLTABLE beinhaltet eine Zusammenstellung (lineare Anordnung) aller zu pollenden Korrespondenten.
5. POLLTABLE wird nur im Master-Rechner fuer das BSC-Protokoll eingelesen.

6. WABTPROB = 0 bedeutet : Unterbrechungsanfrage vom Master positiv beantwortet .
   WABTPROB = 1 bedeutet : negative Rueckmeldung (Master muss weitere Unterbrechungsanfragen schicken; maximal WABTNUM-mal moeglich) .
7. WABTPROB wird nur fuer das BSC-Protokoll eingelesen; nur Bedeutung im Slave .

MOSAIC-Eingabebeschreibung

KARTE 4.1. : (Kennzeichnung der Topologie)
****

NUM	-I-	Anzahl der direkten Verbindungen (auf tiefster Protokollschicht), die zwischen dem aktuell in das Rechnernetz einzufuegenden 'Rechner mit Vermittlungsfunktionen' und irgendwelchen anderen Rechnern bestehen.
K	-IA-	Globale Rechnernummern der Rechnernetzknoten, zu denen eine derartige direkte Verbindung besteht.

Bemerkungen :
---

1. Durch Vorgabe der Werte der Variablen NUM und K werden automatisch Routing-Tabellen erstellt, aus denen die Vermaschung des Rechnernetzmodells ersichtlich ist.

2. Der Vektor K enthaelt genau NUM Werte.

3. Offensichtlich ist KARTE 4.1. im Falle einer sternfoermigen Topologie trivial und braucht daher fuer eine derartige Konfiguration nicht eingegeben zu werden.

MOSAIC-Eingabebeschreibung

KARTE 5.1. : (Kennzeichnung der Uebertragungskanaele)
****

MODE	-I-	Uebertragungsmodus .
CAPACITY	-R-	Uebertragungsgeschwindigkeit des Uebertragungsmediums zwischen den beiden (durch den Kanal verbundenen) Rechnern.
MAXPACKLENGTH	-I-	Max. Laenge von Dateneinheiten im Uebertragungskanal .
FREEPLACE	-I-	Speicherbereich zum Zwischenpuffern von Dateneinheiten, die zur Uebertragung bereitstehen.

Bemerkungen :
---

1. Die in MOSAIC implementierten Uebertragungsmodi sind :
   1 = simplex (vom Rechner mit niedriger zum Rechner mit hoeherer Nummer);
   2 = simplex (vom Rechner mit hoeherer zum Rechner mit niedriger Nummer);
   3 = halbduplex ;
   4 = vollduplex .

2. Die Uebertragungszeit TRANSTIME fuer eine Dateneinheit DE ergibt sich zu :
   TRANSTIME:= DE.LENGTH/CAPACITY .

3. Eine fuer eine sinnvolle Eingabe notwendige Bedingung ist
   MAXPACKLENGTH <= FREEPLACE .

MOSAIC-Eingabebeschreibung

KARTE 6.1. : (Randbedingungen fuer das geplante Simulations-
********** experiment)

SIMTIME	-R-	Dauer der gesamten 'Belastungsphase' des Simulationsexperiments : waehrend dieses Zeitraums wird die vorgegebene Modellbelastung erzeugt (s. Details hinsichtlich Experimentdurchfuehrung).
PHASE	-R-	Dauer der gesamten 'Abklingphase' des Simulationsexperiments : in diesem Zeitabschnitt am Ende des Experiments wird - z.B. zu Verifikationszwecken - keine Modellbelastung mehr erzeugt.
TIOUT	-R-	Dauer eines Timers (z.B. bei der Zeitueberwachung von Sendevorgaengen in einer Protokolleinheit) .
TINT	-R-	Zeitintervall zwischen zwei direkt aufeinanderfolgenden Aufrufen der PROCESS class SNAPSHOT zur Ausgabe von Simulationsresultaten.
PT1	-R-	Zeitschranken zur Ausgabe detaillierter In-
PT2	-R-	formation (Trace) hinsichtlich des Experimentablaufs.
TRACEPROT	-I-	Nummer des Kommunikationsprotokolls, fuer das ein Trace verlangt wird.

Bemerkungen :
---

1. Die Ausgabe von Trace erfolgt im Intervall <PT1,PT2> .

2. Die Dauer des gesamten Simulationsexperiments T ergibt sich zu T = SIMTIME + PHASE .
   PHASE = 0 ist ein erlaubter Wert.

3. TRACEPROT < 0 bedeutet : Trace fuer jedes Protokoll des gesamten Rechnernetzes gewuenscht.

Anhang B:

Beispiel eines interaktiven Simulationsexperiments

mit MOSAIC(BERNET)

B. Beispiel eines interaktiven Simulationsexperiments mit MOSAIC(BERNET)

Neben der Möglichkeit, Simulationsexperimente als Batch-Läufe durchzuführen, unterstützt MOSAIC(BERNET) ein interaktives Experimentieren unter Einsatz graphischer Hilfsmittel. Um hierfür ein Beispiel zu geben, wird im folgenden ein Simulationsexperiment vorbereitet und seine Durchführung im Dialogbetrieb unter Benutzung eines TEKTRONIX T4014-Displays protokolliert.

B1. Wesentliche Randbedingungen der gewählten BERNET-Konfiguration

Als Rechnernetztopologie, die dem Experiment zugrundeliegt, wurden 5 Arbeitsrechner AR_1, ..., AR_5 gewählt, die über 5 Netzeingangsprozessoren NEP_1, ..., NEP_5 an einen X.25-Paketvermittlungsknoten angeschlossen sind (vgl. Abb. 4.1. in Kapitel 4 mit $r = 5$). Abb. B.1. veranschaulicht die gewählte Struktur der Kommunikationssoftware (idealisiert). Benutzertasks werden ausschließlich in den Arbeitsrechnern generiert und bearbeitet, die Netzeingangsprozessoren besitzen nur Vermittlungsfunktionen. Die Netzeingangsprozessoren sind als identisch hinsichtlich ihrer Hard- und Softwareeigenschaften vorausgesetzt.

Zu übertragende Daten werden von den Benutzern (kommunizierende Benutzertasks) in den Arbeitsrechnern an die Protokolleinheiten der höchsten Schicht (RJE- oder Basisprotokoll) übergeben. Unter detaillierter Berücksichtigung des Verhaltens der Kommunikationssoftware bis einschließlich Leitungsebene wird der Datenfluß innerhalb sämtlicher Schichten der Protokollhierarchie im Modell nachvollzogen.

Anmerkung:
Man beachte, daß die Schicht MLP nicht in sämtlichen Rechnernetzknoten vorhanden ist (vgl. Abb. B.1.). Entsprechend sind die Aussagen bzgl. Durchsatz, Dateneinheitenverweilzeiten, "Power" - die auf das Gesamtsystem bezogen sind - zu interpretieren.

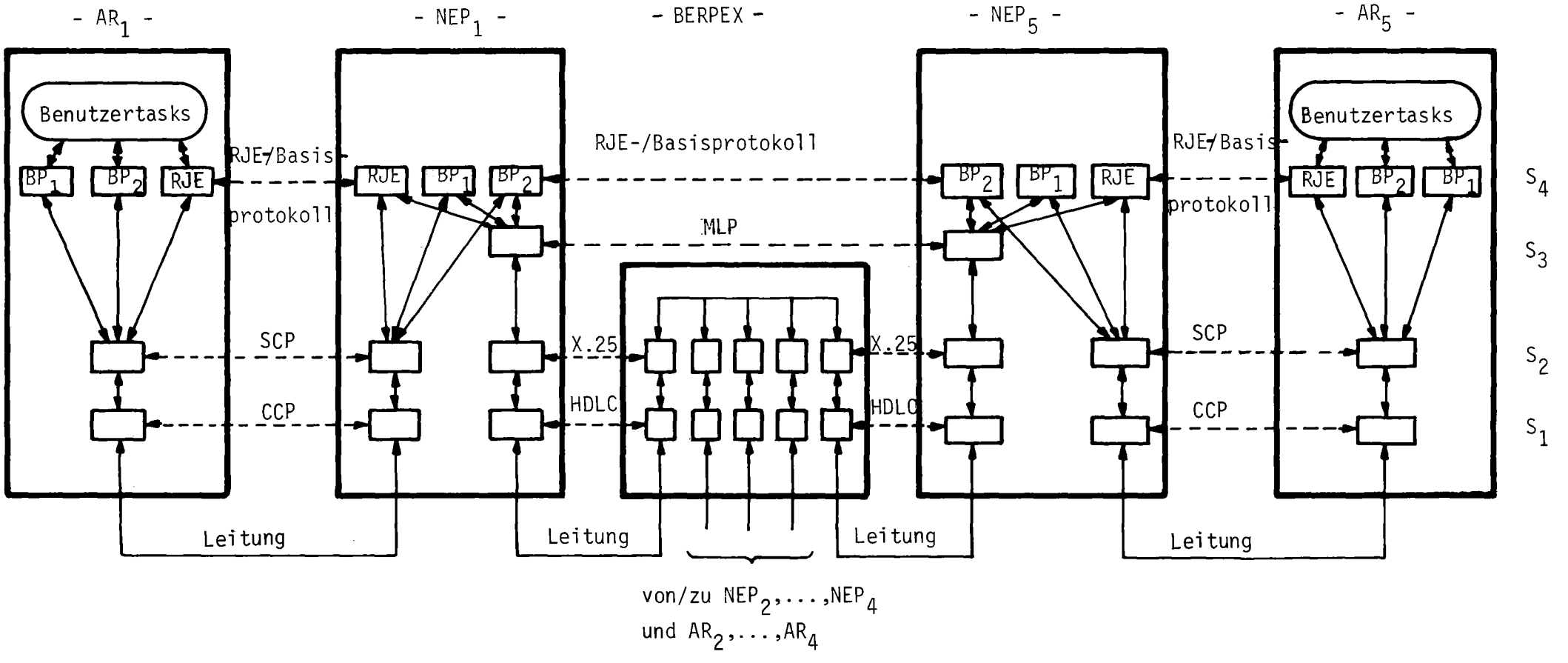


Abb. B.1.: Dem Eingabedatensatz EXP2 zugrundeliegende Rechnernetzstruktur (Notation s. Abb. 4.3.)

Weitere Randbedingungen des Rechnernetzmodells wurden wie folgt gewählt:

(a) Charakteristika der Kommunikationssoftware:

- keine Verfälschungen der Dateneinheiten bei ihrem Transfer durch die verschiedenen Protokollschichten
- CPU-Zeiten für die Abwicklung der Kommunikationssoftware entsprechend den bisherigen Meßergebnissen über das BERNET
- HDLC:
  - Quittung für jede Dateneinheit (frame)
  - Sendekredit = 7
- X.25:
  - permanente logische Kanäle
  - Quittung für jede Dateneinheit (packet)
  - Sendekredit = 2
- MLP:
  - Quittung für jede Dateneinheit (message)
- RJE:
  - keine Quittungen
  - Durchführung der Protokollinitialisierung
- Basisprotokoll:
  - keine Quittungen

(b) Hardware- und Betriebssystemcharakteristika der Rechnernetzknoten:

- prioritätsgesteuerte CPU-Zuteilung; innerhalb derselben Priorität: CPU-Zuteilung nach einem Zeitscheibenverfahren (Zeitscheibengröße: 50 [msec]);
- Dauer von E/A-Vorgängen gleichverteilt im Intervall [50 msec, 100 msec]

(c) Charakteristika der Übertragungskanäle:

- vernachlässigbare Fehlerraten
- Leitungen zwischen Netzeingangsprozessoren und X.25-Vermittlungsknoten:
  - vollduplex-Übertragungsmodus
  - Übertragungsgeschwindigkeit: 9.6 [KBit/sec]

- Leitungen zwischen Arbeitsrechnern und Netzeingangsprozessoren:
  - halbduplex-Übertragungsmodus
  - Übertragungsgeschwindigkeit: 1,0 [MBit/sec]

(d) Beschreibung der Rechnernetzbelastung:

- Länge der (durch Benutzertasks) erzeugten Dateneinheiten gleichverteilt im Intervall [1 KBit, 10 KBit]
- Ankunftsrate: 0.05 [Benutzertask/sec] für jeden Arbeitsrechner (hieraus resultiert eine mittlere Zwischenankunftszeit für zu generierende Benutzertasks von 20 [sec])
- zu Beginn des Simulationsexperiments sind 10 Benutzertasks pro Arbeitsrechner vorhanden
- 60% lokale und 40% kommunizierende Benutzertasks
- maximal 3 Benutzertasks können gleichzeitig einen Dienst der Schicht S_4 in einem Arbeitsrechner benutzen

B2. Eingabedatensatz

Die beschriebenen Randbedingungen werden im folgenden Eingabedatensatz für das Modellierungssystem MOSAIC(BERNET) berücksichtigt [für die Erstellung des Eingabedatensatzes kann das in /MÜLL80/ vorgestellte Eingabeinterface MOSIN Verwendung finden] :

```
00010 3 30 3 1 6
00020 5 5 1  4 4 2  6  0 NUMC NUMLAY NUMCPLOW  PERIOD
00030 1 1 0 3  2 2 1 3  5 5    NUMPU
00040 1 50 1 4 2 70 90 50 100 1024 10240 20000 .8 .4  1 1 1  10000 100 9  AR 1
00050 500000 1   0 1 0   3 3 3  .5 .3  0 2
00060 1024 72 0.0                                        LAYER = 1
00070 0 13  101  24576 6 1 0                              PE-NR. = 1
00080 1 1   0 0
00090  2 0.0   20 0.0   10 0.0    1 0 0 999 1
00100 1 1  20 0.0   5 0.0       3 1 0
00110 24576 72 0.0                                       LAYER = 2
00120 0 23 201  1024 6 1 0                                PE-NR. = 2
00130 1  1  0 0
00140  2 0.0   20 0.0   10 0.0     0 1 0 999 1
00150 1 1  20 0.0   5 0.0        3 1 0
00160 0 0 0                                              LAYER = 3
00170 10240 48 0                                         LAYER = 4
00180 0 60  401   1024 6 0 1                             PE-NR. = 3
00190 11 15  2 0  0 0.2 0.2  0.3 0.3   0 0 0 0 0   0 1 0
00200  2 0.0   20 0.0   10 0.0      0 0 0 999 1
00210 1 15  20 0.0   5 0.0          3 1 0
00220 0  0  402   1024 6 1 1                              PE-NR. = 4
00230 11 15  2 0  0 0.1 0.05 0.8 0.05  0 0 0 0 0   0 1 0
00240  2 0.0   20 0.0   10 0.0      0 0 0 999 1
00250 1 15  20 0.0   5 0.0          3 1 0
00260 0  0  403   1024 6 1 1                              PE-NR. = 5
00270 11 15  2 0  0 0.1 0.05 0.8 0.05  0 0 0 0 0   0 1 0
00280  2 0.0   20 0.0   10 0.0      0 0 0 999 1
00290 1 15  20 0.0   5 0.0          3 1 0
```

Eingabedatensatz (Fortsetzung):

```
00300 1 50 1 4 2 70 90 50 100 1024 10240 20000 .6 .4  1 1 1  10000 100 9 AR 2
00310 500000 1   0 1 0   3 3 3  .5 .3  0 2
00320 1024 72 0.0                                                LAYER = 1
00330 0 13  108  24576 7 1 0                                      PE-NR. = 1
00340 1 1  0 0
00350  2 0.0   20 0.0   10 0.0    1 0 0 999 1
00360 1 1  20 0.0   5 0.0      3 1 0
00370 24576 72 0.0                                               LAYER = 2
00380 0 23 208  1024 7 1 0                                        PE-NR. = 2
00390 1  1  0 0
00400  2 0.0   20 0.0   10 0.0     0 1 0 999 1
00410 1 1  20 0.0   5 0.0       3 1 0
00420 0 0 0.0                                                    LAYER = 3
00430 10240 48 0                                                 LAYER = 4
00440 0 60  401   1024 7 0 1                                     PE-NR. = 3
00450 11 15  2 0  0.3   0 0.3  0.2 0.2    0 0 0 0 0   0 1 0
00460  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00470 1 15  20 0.0   5 0.0       3 1 0
00480 0  0  402   1024 7 1 1                                      PE-NR. = 4
00490 11 15  2 0  0.075 0 0.15 0.7 0.075  0 0 0 0 0   0 1 0
00500  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00510 1 15  20 0.0   5 0.0       3 1 0
00520 0  0  403   1024 7 1 1                                      PE-NR. = 5
00530 11 15  2 0  0.075 0 0.15 0.7 0.075  0 0 0 0 0   0 1 0
00540  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00550 1 15  20 0.0   5 0.0       3 1 0
00560 1 50 1 4 2 70 90 50 100 1024 10240 20000 .6 .4  1 1 1  10000 100 9 AR 3
00570 500000 1   0 1 0   3 3 3  .5 .3  0 2
00580 1024 72 0.0                                                LAYER = 1
00590 0 13  107  24576 8 1 0                                      PE-NR. = 1
00600 1 1  0 0
00610  2 0.0   20 0.0   10 0.0    1 0 0 999 1
00620 1 1  20 0.0   5 0.0      3 1 0
00630 24576 72 0.0                                               LAYER = 2
00640 0 23 207  1024 8 1 0                                        PE-NR. = 2
00650 1  1  0 0
00660  2 0.0   20 0.0   10 0.0     0 1 0 999 1
00670 1 1  20 0.0   5 0.0       3 1 0
00680 0 0 0.0                                                    LAYER = 3
00690 10240 48 0                                                 LAYER = 4
00700 0 60  401   1024 8 0 1                                     PE-NR. = 3
00710 11 15  2 0  0.2   0.3 0  0.3 0.2    0 0 0 0 0   0 1 0
00720  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00730 1 15  20 0.0   5 0.0       3 1 0
00740 0  0  402   1024 8 1 1                                      PE-NR. = 4
00750 11 15  2 0  0.075 0.15 0 0.7 0.075  0 0 0 0 0   0 1 0
00760  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00770 1 15  20 0.0   5 0.0       3 1 0
00780 0  0  403   1024 8 1 1                                      PE-NR. = 5
00790 11 15  2 0  0.075 0.15 0 0.7 0.075  0 0 0 0 0   0 1 0
00800  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00810 1 15  20 0.0   5 0.0       3 1 0
00820 1 50 1 4 2 70 90 50 100 1024 10240 20000 .6 .4  1 1 1  10000 100 9 AR 4
00830 500000 1   0 1 0   3 3 3  .5 .3  0 2
00840 1024 72 0.0                                                LAYER = 1
00850 0 13  109  24576 9 1 0                                      PE-NR. = 1
00860 1 1  0 0
00870  2 0.0   20 0.0   10 0.0    1 0 0 999 1
00880 1 1  20 0.0   5 0.0      3 1 0
00890 24576 72 0.0                                               LAYER = 2
00900 0 23 209  1024 9 1 0                                        PE-NR. = 2
00910 1  1  0 0
00920  2 0.0   20 0.0   10 0.0     0 1 0 999 1
00930 1 1  20 0.0   5 0.0       3 1 0
00940 0 0 0.0                                                    LAYER = 3
00950 10240 48 0                                                 LAYER = 4
00960 0 60  401   1024 9 0 1                                     PE-NR. = 3
00970 11 15  2 0  0.5 0.5 0   0 0      0 0 0 0 0   0 1 0
00980  2 0.0   20 0.0   10 0.0     0 0 0 999 1
00990 1 15  20 0.0   5 0.0       3 1 0
01000 0  0  402   1024 9 1 1                                      PE-NR. = 4
01010 11 15  2 0  0.2 0.5 0.2 0 0.1    0 0 0 0 0   0 1 0
01020  2 0.0   20 0.0   10 0.0     0 0 0 999 1
01030 1 15  20 0.0   5 0.0       3 1 0
01040 0  0  403   1024 9 1 1                                      PE-NR. = 5
01050 11 15  2 0  0.2 0.5 0.2 0 0.1    0 0 0 0 0   0 1 0
01060  2 0.0   20 0.0   10 0.0     0 0 0 999 1
01070 1 15  20 0.0   5 0.0       3 1 0
01080 1 50 1 4 2 70 90 50 100 1024 10240 20000 .6 .4  1 1 1  10000 100 9 AR 5
01090 500000 1   0 1 0   3 3 3  .5 .3  0 2
01100 1024 72 0.0                                                LAYER = 1
01110 0 13  110  24576 10 1 0                                     PE-NR. = 1
01120 1 1  0 0
01130  2 0.0   20 0.0   10 0.0    1 0 0 999 1
01140 1 1  20 0.0   5 0.0      3 1 0
01150 24576 72 0.0             2                                 LAYER = 2
01160 0 23 210  1024 10 1 0                                       PE-NR. = 2
01170 1  1  0 0
01180  2 0.0   20 0.0   10 0.0     0 1 0 999 1
01190 1 1  20 0.0   5 0.0       3 1 0
01200 0 0 0.0                                                    LAYER = 3
01210 10240 48 0                                                 LAYER = 4
01220 0 60  401   1024 10 0 1                                    PE-NR. = 3
01230 11 15  2 0  0   0   0.5 0.5 0    0 0 0 0 0   0 1 0
```

Eingabedatensatz (Fortsetzung):

```
01240  2 0.0   20 0.0   10 0.0     0 0 0 999 1
01250 1 15  20 0.0   5 0.0          3 1 0
01260 0   0  402   1024 10 1 1                              PE-NR. = 4
01270 11 15  2 0  0.1 0.1 0.1 0.7 0    0 0 0 0 0   0 1 0
01280  2 0.0   20 0.0   10 0.0     0 0 0 999 1
01290 1 15  20 0.0   5 0.0          3 1 0
01300 0   0  403   1024 10 1 1                              PE-NR. = 5
01310 11 15  2 0  0.1 0.1 0.1 0.7 0    0 0 0 0 0   0 1 0
01320  2 0.0   20 0.0   10 0.0     0 0 0 999 1
01330 1 15  20 0.0   5 0.0          3 1 0
01340 1 50 1 4 2   0 0   0 0   0 0   999999     1 0 1  1.0 2.0  0 0  0   NEP 1
01350 900000 1    0 1 0   0 0 0    0 0    0    0
01360 1024 72 0                                            LAYER = 1
01370 0 13  101  24576 1 1 0                               PE-NR. = 1
01380 1 1   0 0
01390  2 0.0   20 0.0   10 0.0       1 0 0 999 1
01400 1 1   20 0.0   5 0.0        3 1 0
01410 0 10  102   1024 11 1 0                              PE-NR. = 2
01420 1 1   0 0
01430 0 0.035   11 0.035  0 0.035    0 0 0  7 1
01440 1 1   11 0.035  0 0     1 1 0
01450 1024 72 0                                            LAYER = 2
01460 0 23 201   1024 1 1 0                                PE-NR. = 3
01470 1 1   1 0
01480  2 0.0   20 0.0   5 0.0     0 0 0 999 1
01490 1 1   20 0.0   5 0.0     3 1     0
01500 1 20  202   1024 11 1 0                              PE-NR. = 4
01510 1 15   2 1   11 11 11 11 11 11 11 11 11 11 11 11
01520 0 0.048  24 0.048  0 0.048       0 0 0  2 1
01530 1 15  24 0.056  0 0          1 1   0
01540 1024 72 0                                            LAYER = 3
01550 1 30  301  1024 0 1 0                                PE-NR. = 5
01560 1 15  4 0
01570  2 0.0   20 0.0   5 0.0     0 0 0 13 1
01580 1 15  20 0.0   5 0.0          1 1   0
01590 1024 72 0.0                                          LAYER = 4
01600 1 60  401   1024 0 0 1                               PE-NR. = 6
01610 1 15  -1 1  3 5 5 5 5 5 5 5 5 5 5  0 7 8 9 10 6 7 8 9 10 11  0.0 0 0 0
01620 0 0.064   11 0.064  0 0.064       0 0 0  999 1
01630 1 15   11 0.056  0 0          3 1   0
01640 1  0  402   1024 0 1 1                               PE-NR. = 7
01650 1 15  -1 1  3 5 5 5 5 5 5 5 5 5  0 7 8 9 10 6 7 8 9 10 11  0.0 0 0 0
01660 0 0.064   11 0.064  0 0.064       0 1 0  999 1
01670 1 15   11 0.056  0 0          3 1   0
01680 1  0  403   1024 0 1 1                               PE-NR. = 8
01690 1 15  -1 1  3 5 5 5 5 5 5 5 5 5 5  0 7 8 9 10 6 7 8 9 10 11  0.0 0 0 0
01700 0 0.064   11 0.064  0 0.064       0 1 0  999 1
01710 1 15   11 0.056  0 0          3 1   0
01720 1 50 1 4 2   0 0   0 0   0 0   999999     1 0 1  1.0 2.0  0 0  0   NEP 2
01730 900000 1    0 1 0   0 0 0    0 0    0    0
01740 1024 72 0                                            LAYER = 1
01750 0 13  102  24576 2 1 0                               PE-NR. = 1
01760 1 1   0 0
01770  2 0.0   20 0.0   10 0.0       1 0 0 999 1
01780 1 1   20 0.0   5 0.0        3 1 0
01790 0 10  103   1024 11 1 0                              PE-NR. = 2
01800 1 1   0 0
01810 0 0.035   11 0.035  0 0.035    0 0 0  7 1
01820 1 1   11 0.035  0 0     1 1 0
01830 1024 72 0                                            LAYER = 2
01840 0 23 202   1024 2 1 0                                PE-NR. = 3
01850 1 1   1 0
01860  2 0.0   20 0.0   5 0.0     0 0 0 999 1
01870 1 1   20 0.0   5 0.0     3 1     0
01880 1 20  203   1024 11 1 0                              PE-NR. = 4
01890 1 15   2 1   11 11 11 11 11 11 11 11 11 11 11 11
01900 0 0.048  24 0.048  0 0.048       0 0 0  2 1
01910 1 15  24 0.056  0 0          1 1   0
01920 1024 72 0                                            LAYER = 3
01930 1 30  301 1024 0 1 0                                 PE-NR. = 5
01940 1 15  4 0
01950  2 0.0   20 0.0   5 0.0     0 0 0 13 1
01960 1 15  20 0.0   5 0.0          1 1   0
01970 1024 72 0.0                                          LAYER = 4
01980 1 60  401   1024 0 0 1                               PE-NR. = 6
01990 1 15  -1 1  5 3 5 5 5 5 5 5 5 5 5  6 0 8 9 10 6 7 8 9 10 11  0.0 0 0 0
02000 0 0.064   11 0.064  0 0.064       0 0 0  999 1
02010 1 15   11 0.056  0 0          3 1   0
02020 1  0  402   1024 0 1 1                               PE-NR. = 7
02030 1 15  -1 1  5 3 5 5 5 5 5 5 5 5 5  6 0 8 9 10 6 7 8 9 10 11  0.0 0 0 0
02040 0 0.064   11 0.064  0 0.064       0 1 0  999 1
02050 1 15   11 0.056  0 0          3 1   0
02060 1  0  403   1024 0 1 1                               PE-NR. = 8
02070 1 15  -1 1  5 3 5 5 5 5 5 5 5 5 5  6 0 8 9 10 6 7 8 9 10 11  0.0 0 0 0
02080 0 0.064   11 0.064  0 0.064       0 1 0  999 1
02090 1 15   11 0.056  0 0          3 1   0
```

Eingabedatensatz (Fortsetzung):

```
02100 1 50 1 4 2   0 0  0 0  0 0  999999     1 0 1  1.0 2.0  0 0  0   NEP 3
02110 900000 1    0 1 0  0 0 0   0 0  0 0
02120 1024 72 0                                          LAYER = 1
02130 0 13  107  24576 3 1 0                              PE-NR. = 1
02140 1 1  0 0
02150  2 0.0   20 0.0   10 0.0      1 0 0 999 1
02160 1 1  20 0.0   5 0.0       3 1 0
02170 0 10  106   1024 11 1 0                             PE-NR. = 2
02180 1 1  0 0
02190 0 0.035   11 0.035  0 0.035    0 0 0  7 1
02200 1 1   11 0.035  0 0    1 1 0
02210 1024 72 0                                          LAYER = 2
02220 0 23 207   1024 3 1 0                              PE-NR. = 3
02230 1 1   1 0
02240  2 0.0   20 0.0   5 0.0     0 0 0 999 1
02250 1 1  20 0.0   5 0.0    3 1    0
02260 1 20  206   1024 11 1 0                             PE-NR. = 4
02270 1 15   2 1   11 11 11 11 11 11 11 11 11 11 11 11
02280 0 0.048  24 0.048  0 0.048      0 0 0  2 1
02290 1 15  24 0.056  0 0        1 1   0
02300 1024 72 0                                          LAYER = 3
02310 1 30  301 1024 0 1 0                               PE-NR. = 5
02320 1 15  4 0
02330  2 0.0   20 0.0   5 0.0     0 0 0 13 1
02340 1 15  20 0.0   5 0.0        1 1   0
02350 1024 72 0.0                                        LAYER = 4
02360 1 60  401   1024 0 0 1                              PE-NR. = 6
02370 1 15  -1 1  5 5 3 5 5 5 5 5 5 5 5  6 7 0 9 10 6 7 8 9 10 11  0.0 0 0 0
02380 0 0.064   11 0.064  0 0.064      0 0 0  999 1
02390 1 15   11 0.056  0 0          3 1   0
02400 1  0  402   1024 0 1 1                              PE-NR. = 7
02410 1 15  -1 1  5 5 3 5 5 5 5 5 5 5 5  6 7 0 9 10 6 7 8 9 10 11  0.0 0 0 0
02420 0 0.064   11 0.064  0 0.064      0 1 0  999 1
02430 1 15   11 0.056  0 0          3 1   0
02440 1  0  403   1024 0 1 1                              PE-NR. = 8
02450 1 15  -1 1  5 5 3 5 5 5 5 5 5 5 5  6 7 0 9 10 6 7 8 9 10 11  0.0 0 0 0
02460 0 0.064   11 0.064  0 0.064      0 1 0  999 1
02470 1 15   11 0.056  0 0          3 1   0
02480 1 50 1 4 2   0 0  0 0  0 0  999999     1 0 1  1.0 8.0  0 0  0   NEP 4
02490 900000 1    0 1 0  0 0 0   0 0  0     0
02500 1024 72 0                                          LAYER = 1
02510 0 13  109  24576 4 1 0                            PE-NR. = 1
02520 1 1  0 0
02530  2 0.0   20 0.0   10 0.0      1 0 0 999 1
02540 1 1  20 0.0   5 0.0       3 1 0
02550 0 10  105   1024 11 1 0                             PE-NR. = 2
02560 1 1  0 0
02570 0 0.035   11 0.035  0 0.035    0 0 0  7 1
02580 1 1   11 0.035  0 0    1 1 0
02590 1024 72 0                                          LAYER = 2
02600 0 23 209   1024 4 1 0                              PE-NR. = 3
02610 1 1   1 0
02620  2 0.0   20 0.0   5 0.0     0 0 0 999 1
02630 1 1  20 0.0   5 0.0    3 1    0
02640 1 20  205   1024 11 1 0                             PE-NR. = 4
02650 1 15   2 1   11 11 11 11 11 11 11 11 11 11 11 11
02660 0 0.048  24 0.048  0 0.048      0 0 0  2 1
02670 1 15  24 0.056  0 0        1 1   0
02680 1024 72 0                                          LAYER = 3
02690 1 30  301 1024 0 1 0                               PE-NR. = 5
02700 1 15  4 0
02710  2 0.0   20 0.0   5 0.0     0 0 0 13 1
02720 1 15  20 0.0   5 0.0        1 1   0
02730 1024 72 0.0                                        LAYER = 4
02740 1 60  401   1024 0 0 1                              PE-NR. = 6
02750 1 15  -1 1  5 5 5 3 5 5 5 5 5 5 5  6 7 8 0 10 6 7 8 9 10 11  0.0 0 0 0
02760 0 0.064   11 0.064  0 0.064      0 0 0  999 1
02770 1 15   11 0.056  0 0          3 1   0
02780 1  0  402   1024 0 1 1                              PE-NR. = 7
02790 1 15  -1 1  5 5 5 3 5 5 5 5 5 5 5  6 7 8 0 10 6 7 8 9 10 11  0.0 0 0 0
02800 0 0.064   11 0.064  0 0.064      0 1 0  999 1
02810 1 15   11 0.056  0 0          3 1   0
02820 1  0  403   1024 0 1 1                              PE-NR. = 8
02830 1 15  -1 1  5 5 5 3 5 5 5 5 5 5 5  6 7 8 0 10 6 7 8 9 10 11  0.0 0 0 0
02840 0 0.064   11 0.064  0 0.064      0 1 0  999 1
02850 1 15   11 0.056  0 0          3 1   0
02860 1 50 1 4 2   0 0  0 0  0 0  999999     1 0 1  1.0 2.0  0 0  0   NEP 5
02870 900000 1    0 1 0  0 0 0   0 0  0     0
02880 1024 72 0                                          LAYER = 1
02890 0 13  110  24576 5 1 0                            PE-NR. = 1
02900 1 1  0 0
02910  2 0.0   20 0.0   10 0.0      1 0 0 999 1
02920 1 1  20 0.0   5 0.0       3 1 0
02930 0 10  104   1024 11 1 0                             PE-NR. = 2
02940 1 1  0 0
02950 0 0.035   11 0.035  0 0.035    0 0 0  7 1
02960 1 1   11 0.035  0 0    1 1 0
02970 1024 72 0                                          LAYER = 2
02980 0 23 210   1024 5 1 0                              PE-NR. = 3
02990 1 1   1 0
03000  2 0.0   20 0.0   5 0.0     0 0 0 999 1
03010 1 1  20 0.0   5 0.0    3 1    0
03020 1 20  204   1024 11 1 0                             PE-NR. = 4
03030 1 15   2 1   11 11 11 11 11 11 11 11 11 11 11 11
03040 0 0.048  24 0.048  0 0.048      0 0 0  2 1
03050 1 15  24 0.056  0 0        1 1   0
03060 1024 72 0                                          LAYER = 3
```

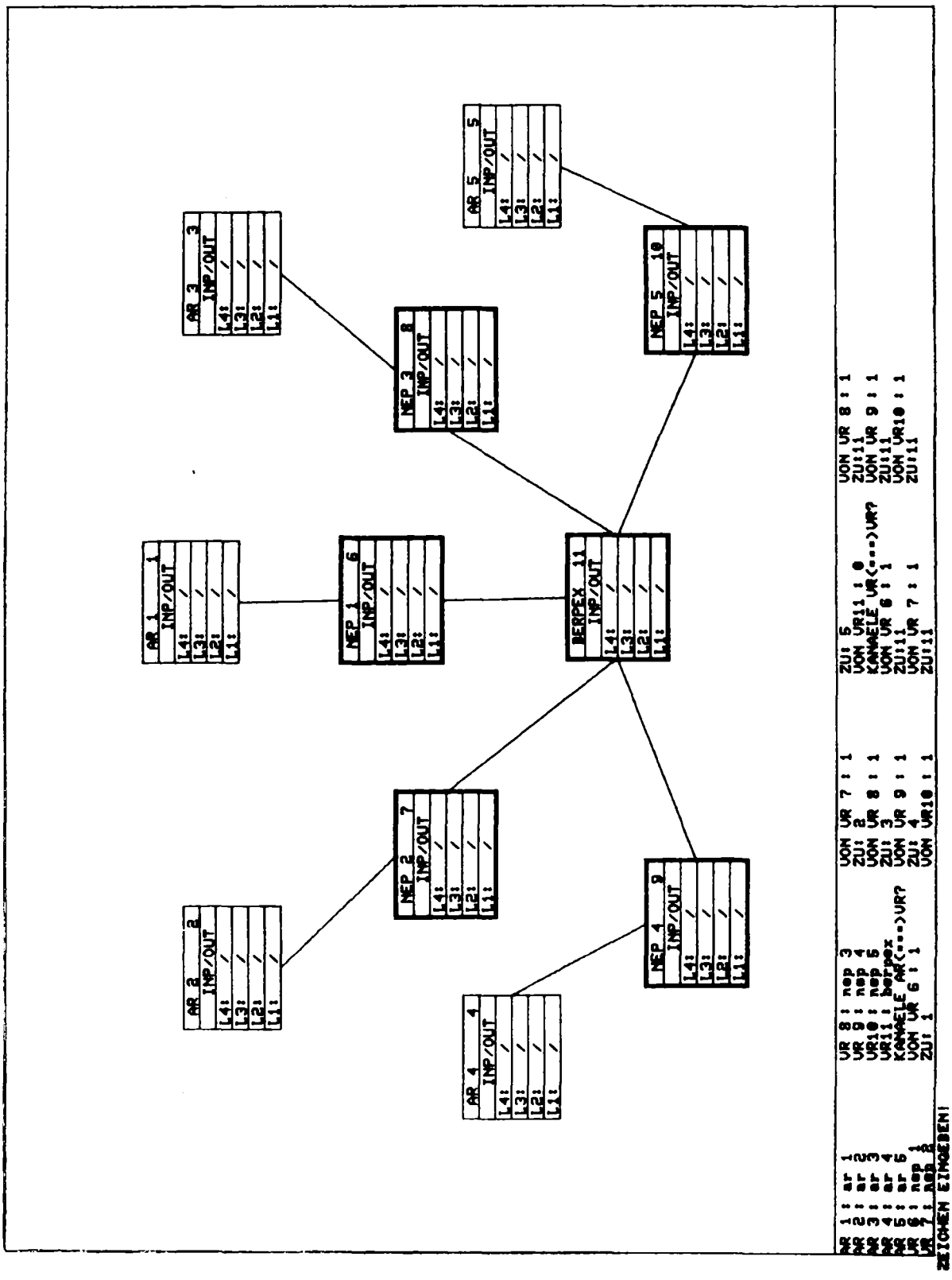
Eingabedatensatz (Fortsetzung):

```
03070 1 30  301 1024 0 1 0                                  PE-NR. = 5
03080 1 15  4 0
03090  2 0.0   20 0.0   5 0.0     0 0 0 13 1
03100 1 15  20 0.0   5 0.0        1 1   0
03110 1024 72 0.0                                           LAYER = 4
03120 1 60  401   1024 0 0 1                                  PE-NR. = 6
03130 1 15  -1 1  5 5 5 5 3 5 5 5 5 5 5  6 7 8 9 0 6 7 8 9 10 11  0.0 0 0 0
03140 0 0.064   11 0.064  0 0.064       0 0 0  999 1
03150 1 15   11 0.056  0 0          3 1   0
03160 1  0  402   1024 0 1 1                                  PE-NR. = 7
03170 1 15  -1 1  5 5 5 5 3 5 5 5 5 5 5  6 7 8 9 0 6 7 8 9 10 11  0.0 0 0 0
03180 0 0.064   11 0.064  0 0.064       0 1 0  999 1
03190 1 15   11 0.056  0 0          3 1   0
03200 1  0  403   1024 0 1 1                                  PE-NR. = 8
03210 1 15  -1 1  5 5 5 5 3 5 5 5 5 5 5  6 7 8 9 0 6 7 8 9 10 11  0.0 0 0 0
03220 0 0.064   11 0.064  0 0.064       0 1 0  999 1
03230 1 15   11 0.056  0 0          3 1   0
03240 1 20  1 4 2  0 0  0 0  0 0  999999   1 0  0   1 2  0 0  0    DERPEX
03250 2500000 1  0 1 0    0 0
03260 1024 16 0                                             LAYER = 1
03270 0 10  102  1024 6  1 0                                 PE-NR. = 1
03280  1 1  0 0
03290 0 0.035  0 0.035  0 0.035    0 0 0   7 1
03300 1 1  0 0.035  0 0     1 1 0
03310 0 10  103  1024 7  1 0                                 PE-NR. = 2
03320  1 1  0 0
03330 0 0.035  0 0.035  0 0.035    0 0 0   7 1
03340 1 1  0 0.035  0 0     1 1 0
03350 0 10  104  1024 10 1 0                                 PE-NR. = 3
03360  1 1  0 0
03370 0 0.035  0 0.035  0 0.035    0 0 0   7 1
03380 1 1  0 0.035  0 0     1 1 0
03390 0 10  105  1024 9  1 0                                 PE-NR. = 4
03400  1 1  0 0
03410 0 0.035  0 0.035  0 0.035    0 0 0   7 1
03420 1 1  0 0.035  0 0     1 1 0
03430 0 10  106  1024 8  1 0                                 PE-NR. = 5
03440  1 1  0 0
03450 0 0.035  0 0.035  0 0.035    0 0 0   7 1
03460 1 1  0 0.035  0 0     1 1 0
03470 1024 16 0                                             LAYER = 2
03480 0 20  202    1024 0 1 0                                PE-NR. = 6
03490 1 15   1 1  6 7 8 9 10  6 7 8 9 10 0
03500   0 0.035  0 0.035  0 0.035     0 0 0  2 1
03510 1 15  0 0.035  0.0 0.0     1 1  -1 6 7 10 9 8  6 7 10 9 8  0
03520 0 20  203   1024 0 1 0                                 PE-NR. = 7
03530 1 15   2 1  6 7 8 9 10  6 7 8 9 10 0
03540   0 0.035  0 0.035  0 0.035     0 0 0  2 1
03550 1 15  0 0.035  0.0 0.0     1 1  -1 6 7 10 9 8  6 7 10 9 8  0
03560 0 20  204   1024 0 1 0                                 PE-NR. = 8
03570 1 15   3 1  6 7 8 9 10  6 7 8 9 10 0
03580   0 0.035  0 0.035  0 0.035     0 0 0  2 1
03590 1 15  0 0.035  0.0 0.0     1 1  -1 6 7 10 9 8  6 7 10 9 8  0
03600 0 20  205   1024 0 1 0                                 PE-NR. = 9
03610 1 15   4 1  6 7 8 9 10  6 7 8 9 10 0
03620   0 0.035  0 0.035  0 0.035     0 0 0  2 1
03630 1 15  0 0.035  0.0 0.0     1 1  -1 6 7 10 9 8  6 7 10 9 8  0
03640 0 20  206   1024 0 1 0                                 PE-NR. = 10
03650 1 15   5 1  6 7 8 9 10  6 7 8 9 10 0
03660   0 0.035  0 0.035  0 0.035     0 0 0  2 1
03670 1 15  0 0.035  0.0 0.0     1 1  -1 6 7 10 9 8  6 7 10 9 8  0
03680 2  1 11    VERBINDUNGEN UEBER NEP 1
03690 2  2 11    VERBINDUNGEN UEBER NEP 2
03700 2  3 11    VERBINDUNGEN UEBER NEP 3
03710 2  4 11    VERBINDUNGEN UEBER NEP 4
03720 2  5 11    VERBINDUNGEN UEBER NEP 5
03730 5  6 7 8 9 10    VERBINDUNGEN UEBER DERPEX
03740 3  1024   100000  100000    KANAL NEP1 ZU AR 1
03750 3  1024   100000  100000    KANAL NEP1 ZU AR 2
03760 3  1024   100000  100000    KANAL NEP3 ZU AR 3
03770 3  1024   100000  100000    KANAL NEP4 ZU AR 4
03780 3  1024   100000  100000    KANAL NEP5 ZU AR 5
03790 4  9.6  100000  100000      KANAL DERPEX ZU NEP1
03800 4  9.6  100000  100000      KANAL DERPEX ZU NEP2
03810 4  9.6  100000  100000      KANAL DERPEX ZU NEP3
03820 4  9.6  100000  100000      KANAL DERPEX ZU NEP4
03830 4  9.6  100000  100000      KANAL DERPEX ZU NEP5
03840  10000     0  999999999 2500 0  10000   0    SIMULATIONSZEIT UND TRACE
```

B3. Protokollierung des Experiments an einem Graphik-Display

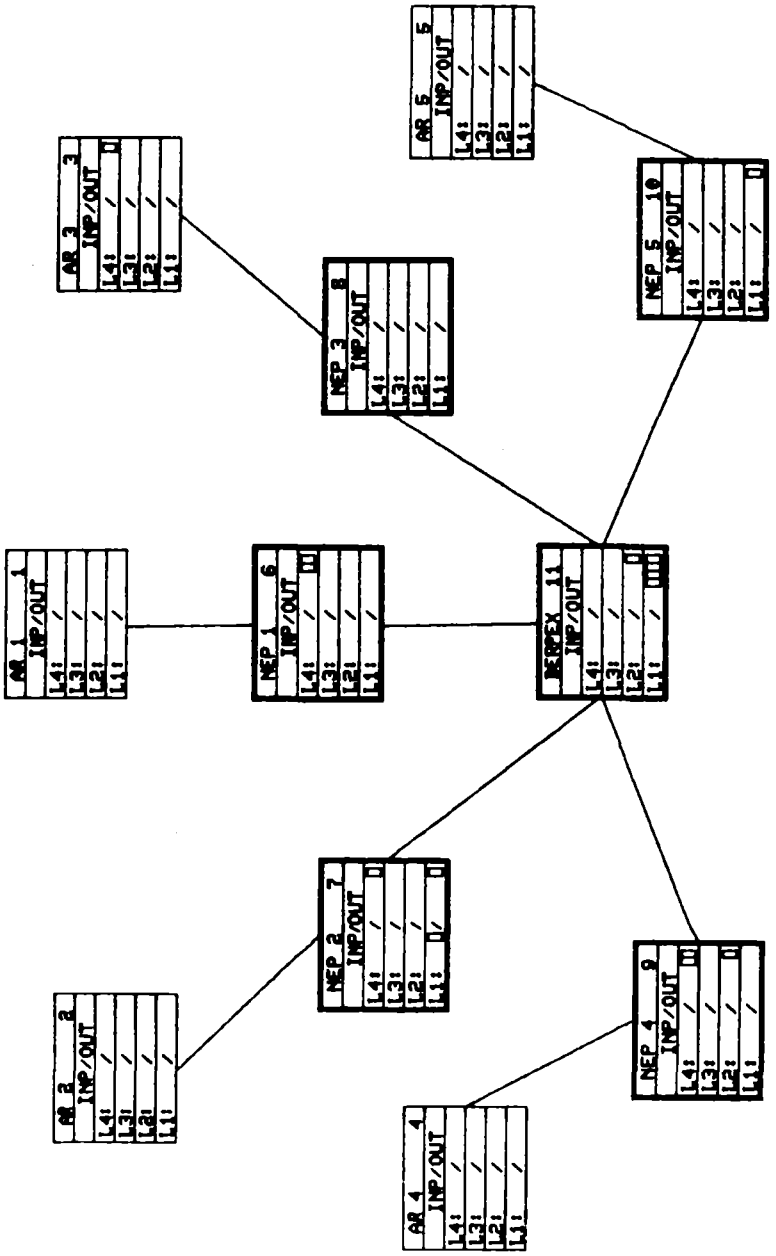
Unter Verwendung des oben gegebenen Eingabedatensatzes könnte das Simulationsexperiment an einem TEKTRONIX T4014-1 z.B. in folgender Weise durchgeführt werden:

```
**********************************************************
********************  M O S A I C  ***********************
**********************************************************
********  MODELLIERUNGSSYSTEM ZUR SIMULATION       ********
********  ALLGEMEINER  INFORMATIONSFLUESSE IN      ********
********  COMPUTERNETZEN                           ********
**********************************************************

AUSFUEHRLICHE DEFINITION DER EINGABEDATEN ERWUENSCHT (J = JA) ?

N
GRAPHISCHE DEFINITION DER RECHNERNETZ-KONFIGURATION ERWUENSCHT (J = JA) ?

J
PLOTAUSGABE DER ZWISCHENERGEBNISSE ERWUENSCHT (J = JA) ?

J
INTERVALL IV FUER DAS ABSPEICHERN DER ZWISCHENERGEBNISSE AUF
 DER DATEI DISC (10 < IV < 2000, MSEC) ?

750
FORTSETZUNG MIT GRAPHIK (RETURNTASTE DRUECKEN) ?
```

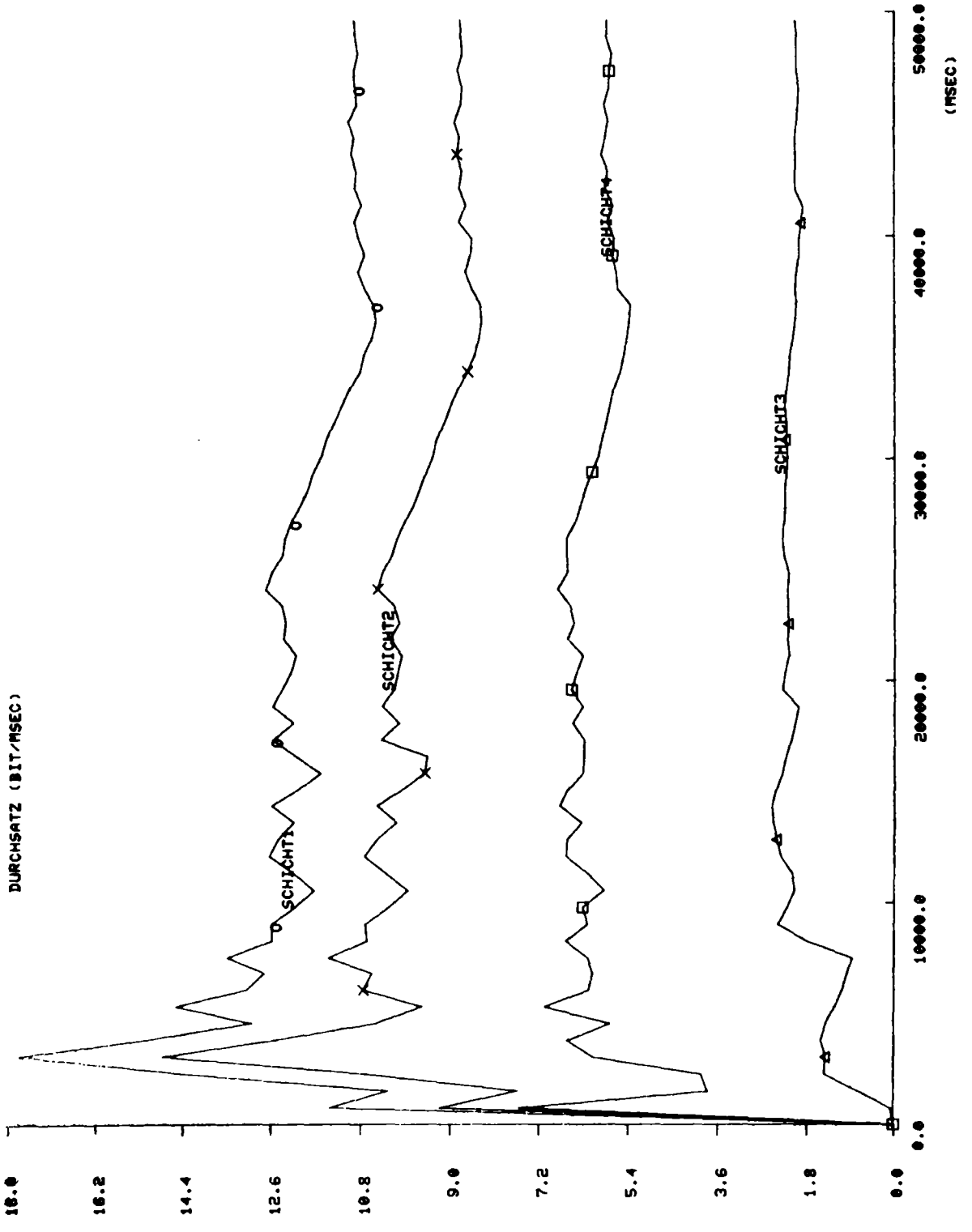

AR 1 1
INP/OUT
L4:
L3:
L2:
L1:
AR 2 2
INP/OUT
AR 3 3
INP/OUT
AR 4 4
INP/OUT
AR 5 5
INP/OUT
NEP 1 6
INP/OUT
NEP 2 7
INP/OUT
NEP 3 8
INP/OUT
NEP 4 9
INP/OUT
NEP 5 10
INP/OUT
BERPEX 11
INP/OUT
AR 1 : ar 1
AR 2 : ar 2
AR 3 : ar 3
AR 4 : ar 4
AR 5 : ar 5
NEP 1 : nep 1
NEP 2 : nep 2
NEP 3 : nep 3
NEP 4 : nep 4
NEP 5 : nep 5
BERPEX : berpex
VR 1 : ASB 1
VR 2 : ASB 2
VR 3 : ASB 3
VR 4 : ASB 4
VR 5 : ASB 5
VR 6 : ASB 6
VR 7 : ASB 7
VR 8 : nep 3
VR 9 : nep 4
VR10 : nep 5
VR11 : berpex
KANAELE AR(--->)VR?
VON VR 6 : 1
ZU: 1
VON VR 7 : 1
ZU: 2
VON VR 8 : 1
ZU: 3
VON VR 9 : 1
ZU: 4
VON VR10 : 1
ZU: 5
VON VR11 : 0
KANAELE VR(--->)VR?
VON VR 6 : 1
ZU:11
VON VR 7 : 1
ZU:11
VON VR 8 : 1
ZU:11
VON VR 9 : 1
ZU:11
VON VR10 : 1
ZU:11
ZEICHEN EINGEBEN!

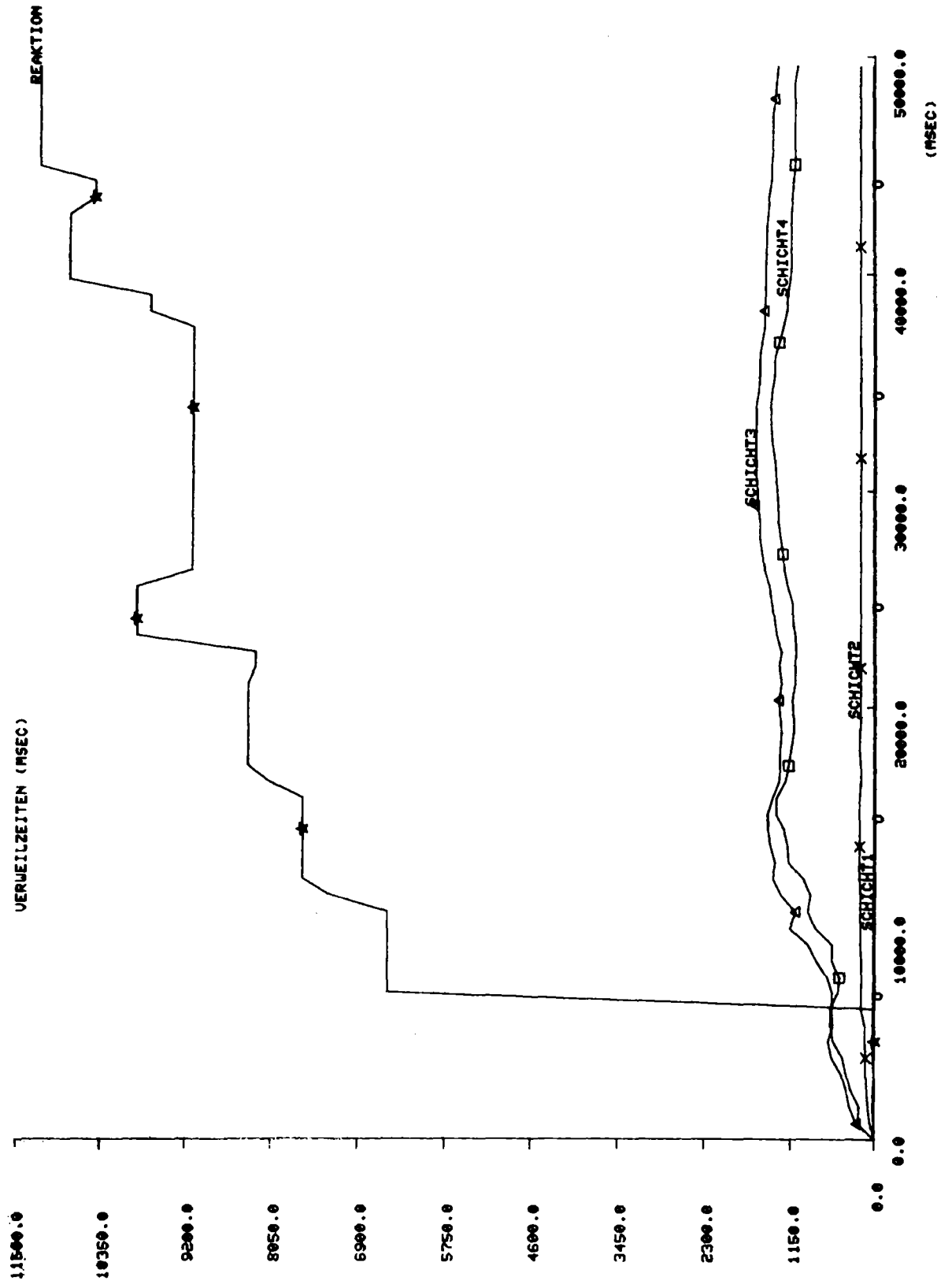
```
SIMULATIONSZEITRAUM (MSEC) ?

300000
MITTLERE ZWISCHENANKUNFTSZEIT FUER BENUTZERTASKS (IN MSEC
NACH DER NEGATIVEN EXPONENTIALVERTEILUNG) ?

20000
WAHRSCHEINLICHKEIT ZUR ERZEUGUNG VON LOKALEN TASKS ?

.6
WAHRSCHEINLICHKEIT ZUR ERZEUGUNG VON KOMMUNIZIERENDEN TASKS?

.4

EINGABE BEENDET. SOLLEN DIE EINGABEDATEN ZU KONTROLLZWECKEN
AUSGEGEBEN WERDEN (J...)JA) ?

n

********************** ENDE KONTROLLDATEN-AUSGABE ***********************************


SIMULATIONSZEIT BIS ZUR NAECHSTEN UNTERBRECHUNG (MSEC)?

50000
UMFANG DER PROTOKOLLIERUNG DES EXPERIMENTS BIS ZUR
NAECHSTEN UNTERBRECHUNG ?
 BEI VORGABE VON 0 ....> KEIN TRACE ERWUENSCHT
                 1 ....> DETAILLIERTER TRACE
                 2 ....> AUSGABE DATENEINHEITEN-VERMESSUNG
                 3 ....> AUSGABE BENUTZERTASK-VERMESSUNG
                 4 ....> DATENEINHEITEN- UND BENUTZERTASK-
                         VERMESSUNG

2


START DER SIMULATION (HOFFENTLICH).

FORTSETZUNG IN DER TRACE-AUSGABE (BILD LOESCHEN, RETURNTASTE DRUECKEN) ?


    TYP      NUMMER URSPRUNG ZIEL ENTSTEHUNG VERNICHTUNG VERWEILZEIT LAENGE
*****************************************************************************
DATENEINHEIT     1       4      2  5.0040E+01  6.4883E+03  6.4382E+03  2391
DATENEINHEIT     3       1      2  1.8913E+02  6.7432E+03  6.5541E+03  3163
DATENEINHEIT     1       5      4  1.8575E+03  1.0600E+04  8.7429E+03  8359
DATENEINHEIT     1       1      4  8.6351E+03  1.1341E+04  8.7061E+03  5613
DATENEINHEIT    14       2      4  6.6331E+03  1.6491E+04  9.8576E+03  1173
DATENEINHEIT    16       2      1  6.7932E+03  1.6542E+04  9.7492E+03  4025
DATENEINHEIT    13       1      2  1.3532E+04  2.1183E+04  7.6509E+03  7962
DATENEINHEIT     1       2      4  2.1434E+03  2.3208E+04  2.1062E+04  9128
DATENEINHEIT    12       5      4  2.2425E+04  2.5560E+04  3.1345E+03  3099
DATENEINHEIT    22       2      3  3.4382E+04  3.7689E+04  3.3069E+03  2685
DATENEINHEIT    25       1      2  1.6564E+04  3.7825E+04  2.1260E+04  4133
DATENEINHEIT     6       4      2  1.6521E+04  3.9722E+04  2.3201E+04  9032
DATENEINHEIT    30       1      5  3.7009E+04  4.3204E+04  6.1954E+03  6813
DATENEINHEIT    22       4      2  2.3256E+04  4.4338E+04  2.1082E+04  9805
DATENEINHEIT    32       2      1  3.7846E+04  4.9756E+04  1.1910E+04  4740
SOLLEN DIE ERGEBNISSE AUSGEDRUCKT WERDEN (J...)JA) ?

J

SIMULATIONSERGEBNISSE NACH   5.000E+04 MSEC
-----------------------------------------

A. ALLGEMEIN
************
---------------------- DATENEINHEITEN ----------------------------  +++++++++++++++++++ TASKS +++++++++++++++++++
  EIN  AUS SYSTEM MITTL.ANZAHL MITTL.LAENGE VERWEILZEIT  DURCHSATZ  EIN  AUS(LOKAL) AUS(KOMM) VWZ(LOKAL)  VWZ(KOMM)
   20   15   5     4.28342E+00  5.47473E+03  1.1257E+04  1.642E+00   33        9        12    6.0179E+03 7.4761E+03

SCHICHT              1           2           3           4
-------------------------------------------------------------
DURCHSATZ        1.1002E+01  8.8577E+00  2.0168E+00  5.8996E+00
-------------------------------------------------------------
VERWEILZEIT      2.6251E+01  2.1268E+02  1.3141E+03  1.0430E+03
-------------------------------------------------------------
POWER            4.1648E-02  1.5347E-03  5.6563E-03  1.4591E-04

B. RECHNER
**********
++++++++++++++++++++DATENEINHEITEN++++++++++++++++++ -----TASKS-----  WARTESCHLANGENLAENGEN
NR GENERIERT BEARBEITET CANCEL DURCHSATZ VERWEILZEIT  EIN  AUS COMM  QT0 QT1 QT2 QIO QPM QCPU AUSLASTUNG STAT BUFF
-------------------------------------------------------------------------------------------------------------------
 1     15        16        0  1.753E-01  1.0830E+04    5    3    1    0   0   1   0  10    1  3.1458E-01    1 500000
 2     21        38        0  7.297E-01  1.4364E+04   10    6    4    0   0   4   0  10    0  3.7059E-01    1 500000
 3      4         3        0  5.370E-02  3.3069E+03    4    3    1    0   0   1   0  10    0  1.2292E-01    1 498609
 4     14        23        0  5.474E-01  1.0301E+04    8    4    4    0   0   4   0  10    0  1.1751E-01    1 497144
 5      8         7        0  1.363E-01  6.1954E+03    6    5    0    0   0   0   0  10    1  1.9086E-01    1 500000
 6    155       195        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1  4.1771E-01    3 898263
 7    352       428        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1  9.0210E-01    3 892922
 8     24        24        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  16    0  4.1586E-02    1 899992
 9    287       333        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1  6.8788E-01    1 885311
10     64        75        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  16    0  1.6682E-01    1 899976
11    700       693        0  0.000E+00  0.0000E+00    0    0    0    0   0   0   0  19    1  4.1724E-01   12497664

FORTSETZUNG (RETURNTASTE DRUECKEN) ?
```

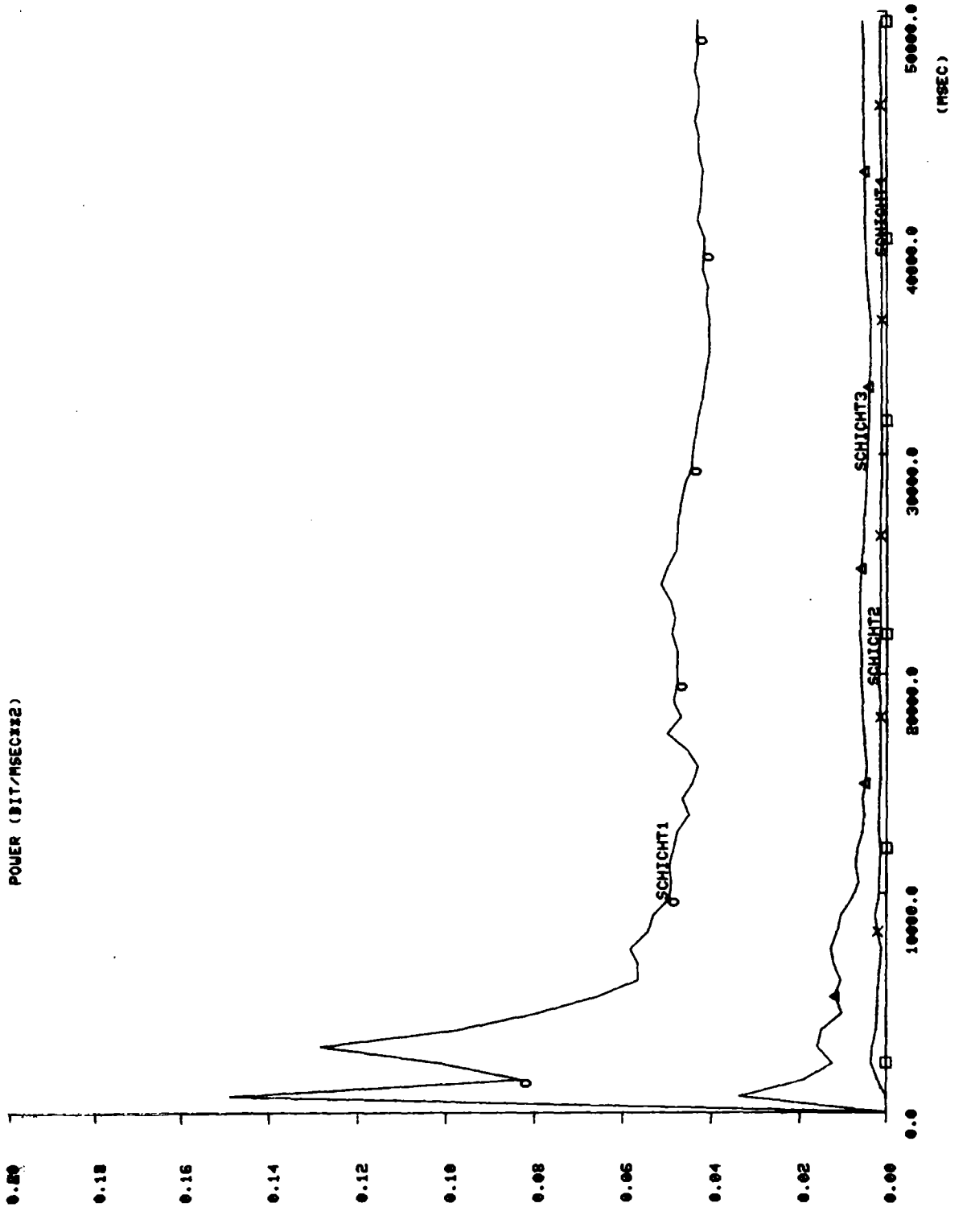
```
MATRIX DER EIN- UND AUSGELAUFENEN DATENEINHEITEN (RICHTUNGSSPEZIFISCH)
          1    2    3    4    5    6    7    8    9   10   11
-------------------------------------------------------------
 1 EIN    0    4    0    1    1    0    0    0    0    0    0
   AUS    0    3    0    1    1    0    0    0    0    0    0
-------------------------------------------------------------
 8 EIN    2    0    1    4    0    0    0    0    0    0    0
   AUS    2    0    1    2    0    0    0    0    0    0    0
-------------------------------------------------------------
 3 EIN    0    0    0    1    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
 4 EIN    0    4    0    0    0    0    0    0    0    0    0
   AUS    0    3    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
 5 EIN    0    0    0    2    0    0    0    0    0    0    0
   AUS    0    0    0    2    0    0    0    0    0    0    0
-------------------------------------------------------------
 6 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
 7 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
 8 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
 9 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
10 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
11 EIN    0    0    0    0    0    0    0    0    0    0    0
   AUS    0    0    0    0    0    0    0    0    0    0    0
-------------------------------------------------------------
SOLLEN DIE PROTOKOLLEINHEITEN AUSGEGEBEN WERDEN (J--->JA) ?

n
FORTSETZUNG (RETURNTASTE DRUECKEN) ?

D. KANAL
********
NR VON NACH  REQQ(1)  REQQ(2)  DEQ(1)  DEQ(2)  BUFF(1)  BUFF(2)  AUSLASTUNG
-------------------------------------------------------------------------
 1  1    6       0        0       0       0   100000   100000  9.4301E-04
 2  2    7       0        0       0       0   100000   100000  1.8049E-03
 3  3    8       0        0       0       0   100000   100000  7.0254E-05
 4  4    9       0        0       0       0   100000   100000  1.3589E-03
 5  5   10       0        0       0       0   100000   100000  4.3404E-04
 6  6   11       0        0       0       0   100000   100000  1.2127E-01
 7  7   11       0        1       0       1   100000    99928  2.5721E-01
 8  8   11       0        1       0       1   100000    99920  1.1377E-02
 9  9   11       0        1       0       1   100000    98896  2.1085E-01
10 10   11       0        0       0       0   100000   100000  5.3565E-02

AUSGABE DER KONFIGURATION MIT AKTUELLEN WARTESCHLANGENLAENGEN (J--->AUSGABE) ?

J
```



ZEICHEN EINGEBEN!

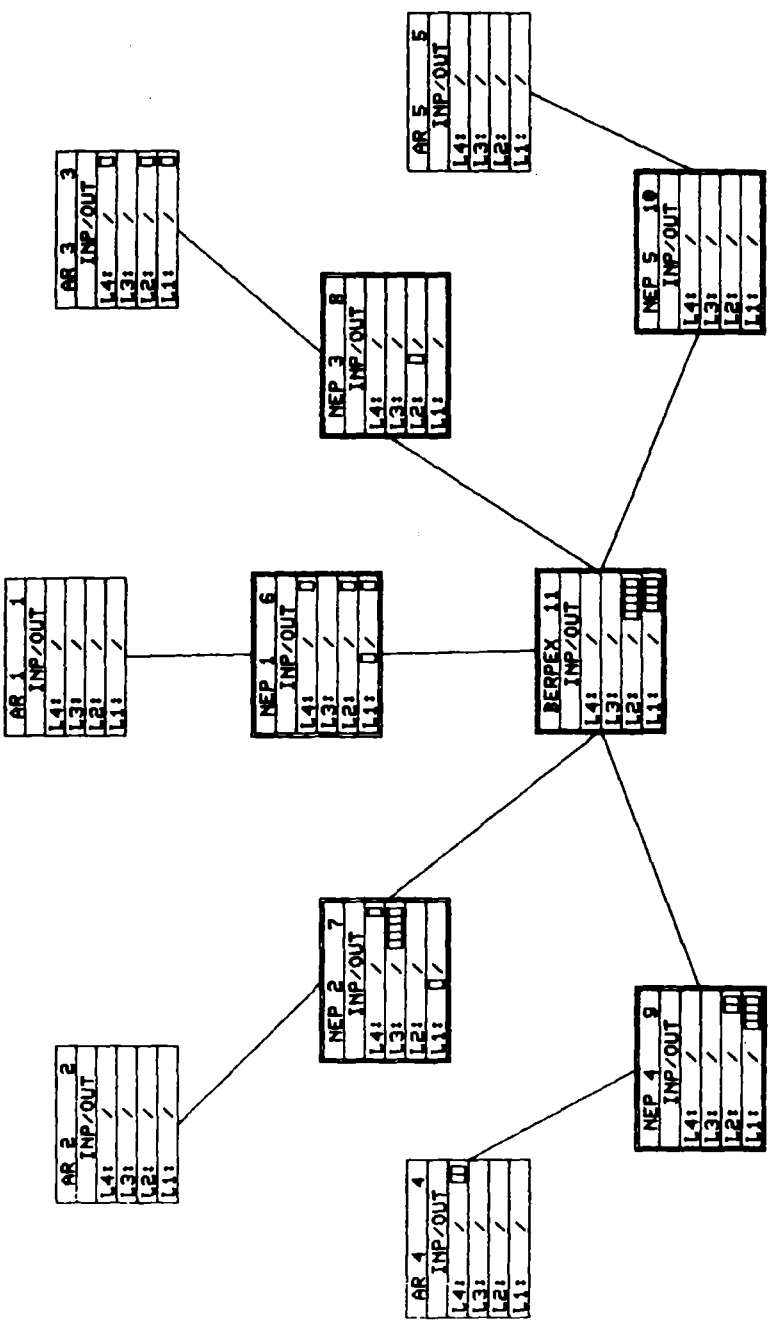

DURCHSATZ (BIT/MSEC)
18.0
16.2
14.4
12.6
10.8
9.0
7.2
5.4
3.6
1.8
0.0
0.0
10000.0
20000.0
30000.0
40000.0
50000.0
(MSEC)
SCHICHT1
SCHICHT2
SCHICHT3
SCHICHT4


REAKTION
VERWEILZEITEN (MSEC)
11500.0
10350.0
9200.0
8050.0
6900.0
5750.0
4600.0
3450.0
2300.0
1150.0
0.0
0.0
10000.0
20000.0
30000.0
40000.0
50000.0
(MSEC)
SCHICHT1
SCHICHT2
SCHICHT3
SCHICHT4

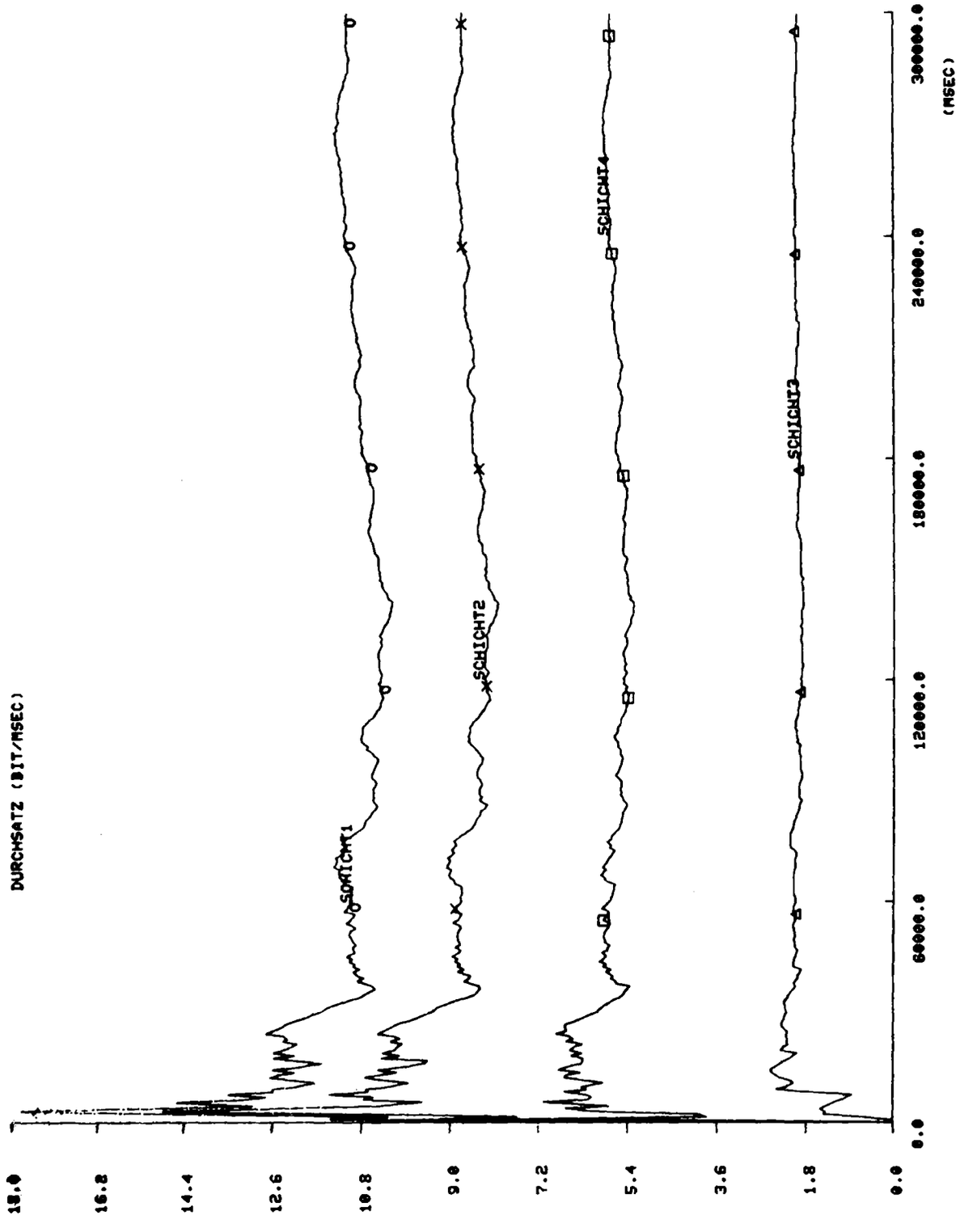

POWER (BIT/MSEC)*2)
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
10000.0
20000.0
30000.0
40000.0
50000.0
(MSEC)
SCHICHT1
SCHICHT2
SCHICHT3
SCHICHT4

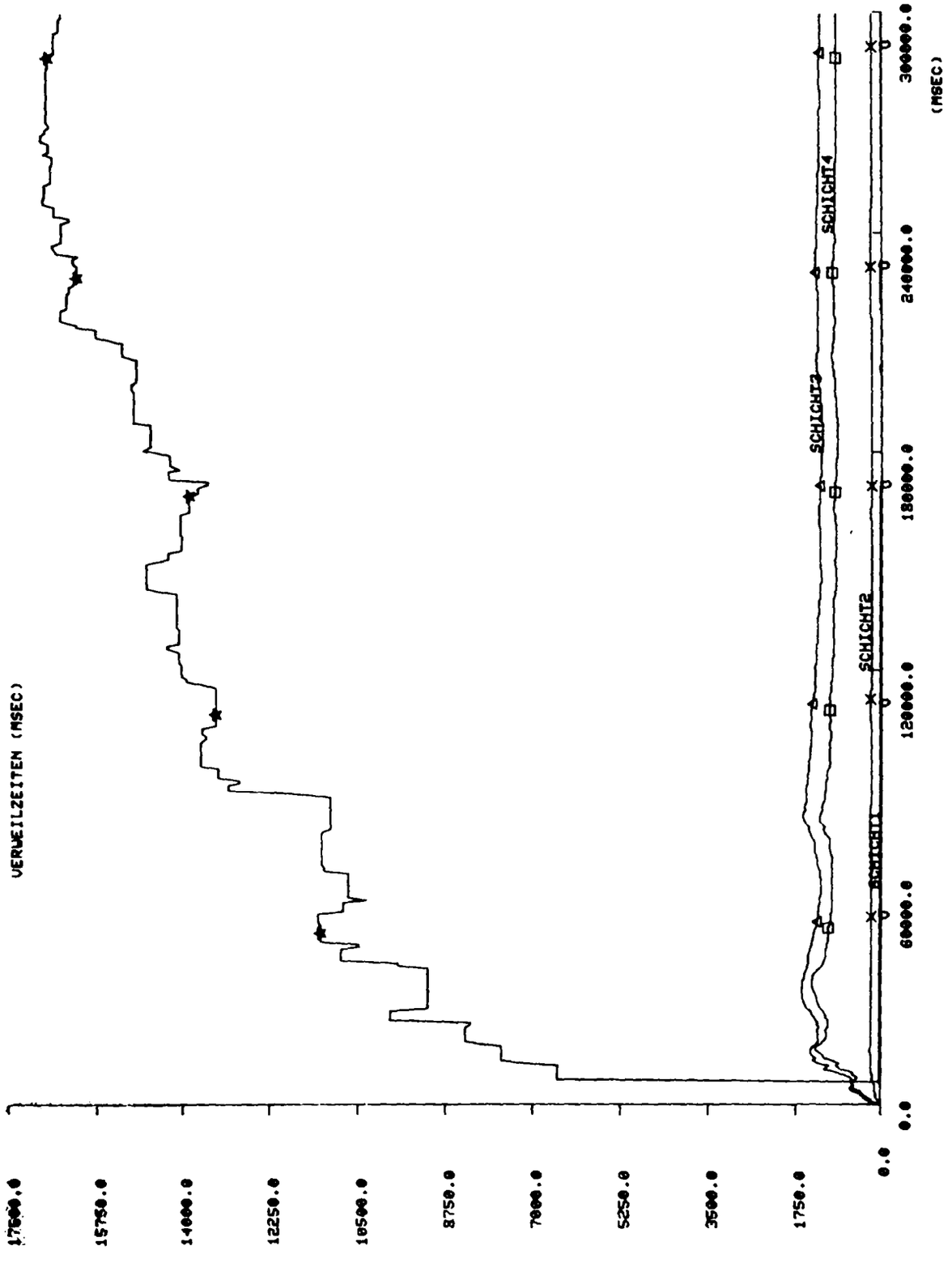
```
SIMULATIONSZEIT BIS ZUR NAECHSTEN UNTERBRECHUNG (MSEC)?

250000
UMFANG DER PROTOKOLLIERUNG DES EXPERIMENTS BIS ZUR
NAECHSTEN UNTERBRECHUNG ?
 BEI VORGABE VON 0 ----> KEIN TRACE ERWUENSCHT
                 1 ----> DETAILLIERTER TRACE
                 2 ----> AUSGABE DATENEINHEITEN-VERMESSUNG
                 3 ----> AUSGABE BENUTZERTASK-VERMESSUNG
                 4 ----> DATENEINHEITEN- UND BENUTZERTASK-
                         VERMESSUNG

3
SOLLEN MODELLPARAMETER VERAENDERT WERDEN (J=JA) ?

J
******** BITTE BEACHTEN *********
BEI EINGABE VON 0 WIRD DER URSPRUENGLICHE MODELLPARAMETER BEIBEHALTEN

ZWISCHENANKUNFTSZEIT (AKTUELLER WERT =  2.0000E+04 MSEC) ?

0
MINIMALE DATENEINHEITEN-LAENGE (AKTUELLER WERT = 1024 BIT) ?

0
MAXIMALE DATENEINHEITEN-LAENGE (AKTUELLER WERT =10240 BIT) ?

0
WAHRSCHEINLICHKEIT ZUR ERZEUGUNG VON LOKALEN TASKS (AKTUELLER WERT = 6.000E-01 ) ?

.65
WAHRSCHEINLICHKEIT ZUR ERZEUGUNG VON KOMM. TASKS (AKTUELLER WERT = 4.000E-01 ) ?

.35
FORTSETZUNG IN DER AUSGABE DER BENUTZERTASK- BZW. DATENEINHEITEN-VERMESSUNG  (RETURNTASTE DRUECKEN) ?



     TYP      NUMMER URSPRUNG ZIEL ENTSTEHUNG VERNICHTUNG VERWEILZEIT LAENGE
*****************************************************************************
     TASK          4       3      --  4.8366E+04  5.3101E+04  3.7350E+03    --
     TASK          4       4      --  4.9366E+04  5.6156E+04  6.7904E+03    --
     TASK          4       5      --  6.6193E+04  6.8658E+04  2.4645E+03    --
     TASK          4       4      --  6.6193E+04  7.5687E+04  9.4939E+03    --
     TASK          4       2      --  7.7180E+04  8.0973E+04  3.7930E+03    --
     TASK         16       1      --  2.6742E+04  8.1100E+04  5.4358E+04    --
     TASK         17       1      --  6.2966E+04  8.2431E+04  1.9465E+04    --
     TASK          4       3      --  7.7180E+04  8.7883E+04  1.0702E+04    --
     TASK         18       5      --  4.5601E+04  8.9948E+04  4.4347E+04    --
     TASK          4       5      --  9.4115E+04  9.8860E+04  4.7451E+03    --
     TASK         17       4      --  9.5366E+04  9.9657E+04  4.2907E+03    --
     TASK         19       5      --  6.7871E+04  1.0088E+05  3.3006E+04    --
     TASK          4       4      --  9.4115E+04  1.0412E+05  1.0008E+04    --
     TASK         19       3      --  8.5827E+04  1.0661E+05  2.1379E+04    --
     TASK         20       2      --  8.5926E+04  1.0963E+05  2.3709E+04    --
     TASK         18       1      --  7.8252E+04  1.1000E+05  3.1746E+04    --
     TASK          6       1      --  1.1970E+05  1.2150E+05  1.7987E+03    --
     TASK         19       1      --  9.9670E+04  1.2601E+05  2.6341E+04    --
     TASK          6       4      --  1.1970E+05  1.2632E+05  6.6180E+03    --
     TASK         21       2      --  8.8130E+04  1.3064E+05  4.2515E+04    --
     TASK         20       1      --  1.2568E+05  1.4133E+05  1.5647E+04    --
     TASK         18       3      --  7.6978E+04  1.4161E+05  6.4629E+04    --
     TASK          6       5      --  1.4251E+05  1.4393E+05  1.4234E+03    --
     TASK         18       4      --  1.2866E+05  1.4771E+05  1.9060E+04    --
     TASK          6       3      --  1.4251E+05  1.4882E+05  6.3096E+03    --
     TASK         21       1      --  1.3831E+05  1.5346E+05  1.5145E+04    --
     TASK          6       4      --  1.5105E+05  1.6625E+05  1.5193E+04    --
     TASK          6       2      --  1.5105E+05  1.7010E+05  1.9047E+04    --
     TASK          9       2      --  1.5522E+05  1.7089E+05  1.5669E+04    --
     TASK          9       4      --  1.5522E+05  1.7478E+05  1.9560E+04    --
     TASK         20       5      --  1.5030E+05  1.8816E+05  3.7863E+04    --
     TASK          6       3      --  1.8839E+05  1.8966E+05  1.2705E+03    --
     TASK         20       3      --  1.7133E+05  1.9336E+05  2.2032E+04    --
     TASK          6       5      --  1.8839E+05  1.9830E+05  9.9030E+03    --
     TASK         22       2      --  1.7207E+05  2.0106E+05  2.8985E+04    --
     TASK         22       1      --  2.0128E+05  2.0497E+05  3.6880E+03    --
     TASK          6       4      --  2.0847E+05  2.1036E+05  1.8818E+03    --
     TASK         19       4      --  1.9786E+05  2.1526E+05  1.7393E+04    --
     TASK          6       1      --  2.0847E+05  2.1853E+05  1.0054E+04    --
     TASK          9       1      --  2.1635E+05  2.1942E+05  3.0726E+03    --
     TASK          9       2      --  2.1635E+05  2.2515E+05  8.8025E+03    --
     TASK         23       1      --  2.0783E+05  2.3356E+05  2.5732E+04    --
     TASK         25       1      --  2.2878E+05  2.3545E+05  6.6846E+03    --
     TASK         24       1      --  2.1606E+05  2.3760E+05  2.1539E+04    --
     TASK         20       4      --  2.3694E+05  2.3827E+05  1.3316E+03    --
     TASK         21       3      --  2.2691E+05  2.4092E+05  1.4016E+04    --
     TASK         26       1      --  2.3506E+05  2.4482E+05  9.7603E+03    --
     TASK          6       5      --  2.4735E+05  2.4774E+05  3.8537E+02    --
     TASK          6       3      --  2.4735E+05  2.5349E+05  6.1378E+03    --
     TASK          6       1      --  2.6131E+05  2.6158E+05  2.7184E+02    --
     TASK          6       2      --  2.6131E+05  2.6631E+05  4.9971E+03    --
     TASK         27       1      --  2.5263E+05  2.7081E+05  1.7976E+04    --
     TASK         22       5      --  2.6471E+05  2.7302E+05  8.3051E+03    --
     TASK         29       1      --  2.8061E+05  2.8213E+05  1.5193E+03    --
     TASK         30       1      --  2.8562E+05  2.9047E+05  4.8545E+03    --
     TASK         28       1      --  2.7436E+05  2.9199E+05  1.7630E+04    --
     TASK         22       3      --  2.7578E+05  2.9419E+05  1.8412E+04    --
     TASK         24       3      --  2.8199E+05  2.9484E+05  1.2854E+04    --
     TASK         24       2      --  2.8994E+05  2.9859E+05  8.6430E+03    --
 SOLLEN DIE ERGEBNISSE AUSGEDRUCKT WERDEN (J--->JA) ?

 J
```

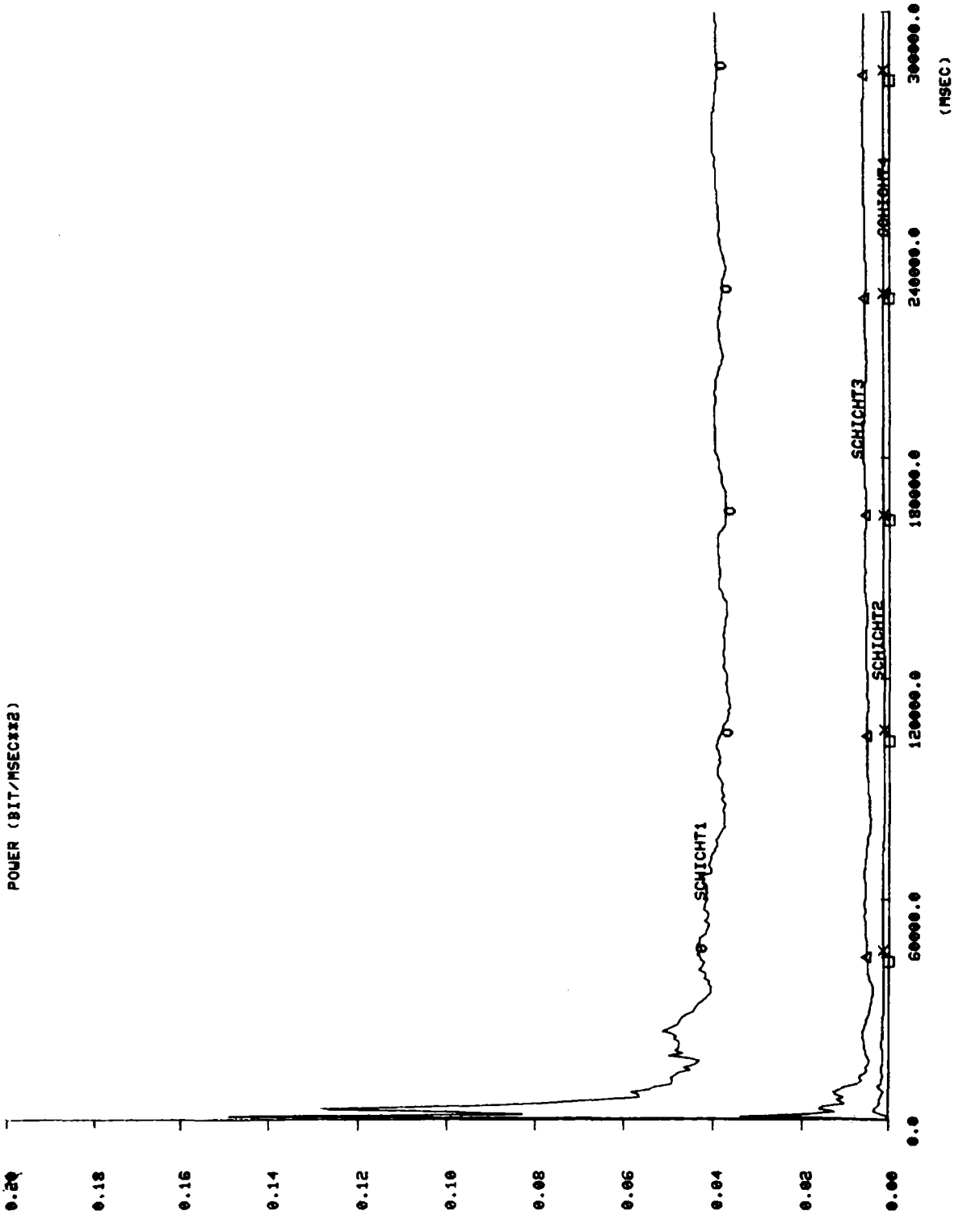
```
SIMULATIONSERGEBNISSE NACH   3.000E+05 MSEC
*****************************************

A. ALLGEMEIN
************
---------------------- DATENEINHEITEN ----------------------------  +++++++++++++++++++++ TASKS +++++++++++++++++++++
  EIN   AUS SYSTEM MITTL.ANZAHL MITTL.LAENGE VERWEILZEIT  DURCHSATZ  EIN  AUS(LOKAL) AUS(KOMM) VWZ(LOKAL)  VWZ(KOMM)
  107    95  12     6.42819E+00  5.48923E+03  1.6380E+04  1.738E+00  109       42        38    1.7834E+04 7.2062E+03

SCHICHT              1           2           3           4
-------------------------------------------------------------
DURCHSATZ        1.1162E+01  8.8212E+00  2.0034E+00  5.7721E+00
-------------------------------------------------------------
VERWEILZEIT      2.4767E+01  2.2105E+02  1.2594E+03  9.2882E+02
-------------------------------------------------------------
POWER            3.9907E-02  1.5907E-03  6.2145E-03  1.0612E-04

B. RECHNER
**********
++++++++++++++++++++++DATENEINHEITEN++++++++++++++++++ -----TASKS-----  WARTESCHLANGENLAENGEN
NR GENERIERT BEARBEITET CANCEL DURCHSATZ VERWEILZEIT  EIN  AUS COMM  QT0 QT1 QT2 QIO QPM QCPU AUSLASTUNG STAT BUFF
-------------------------------------------------------------------------------------------------------------------
 1       63        104      0 2.784E-01  1.1202E+04   25   22    3    0   0   3   0  10    0 4.6274E-01    1 498096
 2      123        249      0 7.092E-01  1.8991E+04   22   15    6    0   0   7   0  10    0 3.7928E-01    1 500000
 3       34         51      0 1.570E-01  1.2002E+04   20   14    4    0   2   4   0   9    1 3.1474E-01    3 488422
 4       96        174      0 5.199E-01  1.8260E+04   25   15    9    0   0  10   0  10    0 2.1607E-01    1 489601
 5       29         29      0 7.379E-02  7.3774E+03   17   14    2    0   0   2   0  10    1 2.9607E-01    1 500000
 6      970       1154      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1 3.9849E-01    3 892061
 7     2133       2556      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1 9.2733E-01    3 874927
 8      549        648      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  15    1 2.1740E-01    1 886362
 9     1757       2134      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  16    0 7.3314E-01    1 897212
10      307        381      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  16    0 1.3278E-01    1 900000
11     4637       4606      0 0.000E+00  0.0000E+00    0    0    0    0   0   0   0  20    0 4.3397E-01   12480855

FORTSETZUNG (RETURNTASTE DRUECKEN) ?



MATRIX DER EIN- UND AUSGELAUFENEN DATENEINHEITEN (RICHTUNGSSPEZIFISCH)
          1   2   3   4   5   6   7   8   9  10  11
-------------------------------------------------------
 1 EIN    0  16   0   2   2   0   0   0   0   0   0
   AUS    0  16   0   2   2   0   0   0   0   0   0
-------------------------------------------------------
 2 EIN   13   0   2  82   0   0   0   0   0   0   0
   AUS   12   0   2  18   0   0   0   0   0   0   0
-------------------------------------------------------
 3 EIN    0   0   0   9   3   0   0   0   0   0   0
   AUS    0   0   0   7   2   0   0   0   0   0   0
-------------------------------------------------------
 4 EIN    2  21   6   0   0   0   0   0   0   0   0
   AUS    1  20   5   0   0   0   0   0   0   0   0
-------------------------------------------------------
 5 EIN    2   0   3   4   0   0   0   0   0   0   0
   AUS    1   0   3   4   0   0   0   0   0   0   0
-------------------------------------------------------
 6 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
 7 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
 8 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
 9 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
10 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
11 EIN    0   0   0   0   0   0   0   0   0   0   0
   AUS    0   0   0   0   0   0   0   0   0   0   0
-------------------------------------------------------
SOLLEN DIE PROTOKOLLEINHEITEN AUSGEGEBEN WERDEN (J--->JA) ?

N
FORTSETZUNG (RETURNTASTE DRUECKEN) ?

D. KANAL
********
NR VON NACH  REQQ(1)  REQQ(2)  DEQ(1)  DEQ(2)  BUFF(1)  BUFF(2)  AUSLASTUNG
-------------------------------------------------------------------------
 1  1    6       0        0       0       0   100000   100000  7.5188E-04
 2  2    7       0        0       0       0   100000   100000  1.6992E-03
 3  3    8       0        0       0       0   100000   100000  4.0814E-04
 4  4    9       0        0       0       0   100000   100000  1.3694E-03
 5  5   10       0        0       0       0   100000   100000  2.9397E-04
 6  6   11       0        1       0       1   100000    99952  1.1605E-01
 7  7   11       0        1       0       1   100000    99952  2.5104E-01
 8  8   11       0        0       0       0   100000   100000  6.4639E-02
 9  9   11       0        0       0       0   100000   100000  2.0916E-01
10 10   11       0        0       0       0   100000   100000  3.9829E-02

AUSGABE DER KONFIGURATION MIT AKTUELLEN WARTESCHLANGENLAENGEN (J--->AUSGABE) ?

J
```



ZEICHEN EINGEBEN!


DURCHSATZ (BIT/MSEC)
SCHICHT1
SCHICHT2
SCHICHT4
SCHICHT3
18.0
16.2
14.4
12.6
10.8
9.0
7.2
5.4
3.6
1.8
0.0
0.0
60000.0
120000.0
180000.0
240000.0
300000.0
(MSEC)


VERWEILZEITEN (MSEC)
17500.0
15750.0
14000.0
12250.0
10500.0
8750.0
7000.0
5250.0
3500.0
1750.0
0.0
0.0
60000.0
120000.0
180000.0
240000.0
300000.0
(MSEC)
SCHICHT1
SCHICHT2
SCHICHT3
SCHICHT4


POWER (BIT/MSEC**2)
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
60000.0
120000.0
180000.0
240000.0
300000.0
(MSEC)
SCHICHT1
SCHICHT2
SCHICHT3