

**de Gruyter Lehrbuch
Schnupp · Rechnernetze**

Peter Schnupp

Rechnernetze

Entwurf und Realisierung

2., gründlich überarbeitete Auflage



Walter de Gruyter · Berlin · New York 1982

Dr. rer. nat. *Peter Schnupp*,
Mitinhaber der Firma InterFace GmbH in München,
Lehrbeauftragter für Systemprogrammierung an der Universität Linz

Das Buch enthält 107 Abbildungen und 9 Tabellen

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Schnupp, Peter:

Rechnernetze : Entwurf u. Realisierung / Peter

Schnupp. – 2., gründl. überarb. Aufl. – Berlin ;

New York : de Gruyter, 1982.

(De-Gruyter-Lehrbuch)

ISBN 3-11-008951-3

© Copyright 1982 by Walter de Gruyter & Co., vormals G. J. Göschen'sche Verlags-
handlung, J. Guttentag, Verlagsbuchhandlung Georg Reimer, Karl J. Trübner,
Veit & Comp., Berlin 30.

Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der
Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (durch Photo-
kopie, Mikrofilm oder ein anderes Verfahren) ohne schriftliche Genehmigung des Verla-
ges reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt
oder verbreitet werden. Printed in Germany.

Satz und Druck: G. Wagner, Nördlingen.

Bindearbeiten: Lüderitz & Bauer Buchgewerbe GmbH, Berlin.

Vorwort zur ersten Auflage

Dieses Buch entstand aus einem selbst empfundenen Mangel: dem unerfüllten Wunsch nach einer Einführung in das Gebiet der Rechnernetze und der Datenverbundsysteme, welche auf die Bedürfnisse und Vorkenntnisse des Software-Entwicklers abgestimmt ist. Er ist es gewohnt, ein EDV-System „top down“, ausgehend von der Benutzerschnittstelle, zu verstehen und zu planen. Die Mehrzahl der vorliegenden Publikationen beginnt die Darstellung jedoch mit der Hardware und den sie verbindenden Leitungen – ein Ansatz, welcher dem Nachrichtentechniker weit mehr liegt als dem Informatiker.

Im Rahmen einer Reihe von Seminaren wurde deshalb die Struktur einiger typischer Rechnernetze „von oben nach unten“ analysiert. Das Ziel war, eine ähnlich geschlossene Vorstellung über den Aufbau und die wesentlichsten Entwurfsprinzipien eines Netzes zu erarbeiten, wie sie ein erfahrener Software-Planer auch für einen Compiler oder ein Betriebssystem hat.

Ausgangspunkt hierfür waren einerseits die wichtigsten Veröffentlichungen, zum anderen aber auch die Erfahrungen, welche bei der Bearbeitung einiger größerer Netzprojekte gewonnen worden waren. Aus der Aufbereitung des so erhaltenen Materials entstand das vorliegende Buch. Es ist vorwiegend für den praktisch arbeitenden Informatiker bestimmt und bemüht sich, die notwendigen theoretischen Grundlagen soweit anzudiskutieren, daß eine Vertiefung an Hand der Literatur keine allzugroßen Schwierigkeiten mehr bieten sollte.

Das Buch wendet sich also vor allem an Software-Spezialisten, erfahrene Programmierer und fortgeschrittene Studenten der Informatik. Auch als Lehrbuch für eine Einführungsvorlesung in dieses Gebiet sollte es nützlich sein. Wenn es darüber hinaus Nachrichtentechnikern und Hardware-Entwicklern hilft, die Probleme und Denkweise der Software-Planer besser zu verstehen, und damit einen Beitrag zur Verbesserung des unbedingt nötigen Dialogs zwischen diesen beiden Partnern einer Netzentwicklung leistet, würde der Verfasser sich freuen.

München, im November 1977

Peter Schnupp

Vorwort zur 2. Auflage

Wie in einem jungen, schnell wachsenden und für die Praxis wichtigen Fachgebiet nicht anders zu erwarten, wuchs das relevante Material, welches eine Einführung in das Feld der Rechnernetze bringen sollte, im Verlauf von nur zwei Jahren so stark an, daß für die Neuauflage eine gründliche Überarbeitung nötig erschien. Zwar verloren die in der ersten Auflage dargestellten Fakten nicht ihre Gültigkeit, und auch der Ansatz, sie dem Leser „top down“, von der Benutzerschnittstelle zur physikalischen Realisierung, nahezubringen, scheint gelungen zu sein und wurde für die neue Fassung beibehalten.

Andererseits stellte sich aber schon bei einem Seminar, das der Verfasser im Spätherbst 1979 abhielt, heraus, daß etwa die Hälfte des dort behandelten Stoffs in dem doch gerade erst etwas über ein Jahr alten Lehrbuch noch nicht enthalten war!

Die Neuauflage wurde deshalb zum Anlaß genommen, das Buch wesentlich zu ergänzen. Neben kleineren Korrekturen und Zusätzen in fast jedem Abschnitt wurden umfangreiche Erweiterungen vor allem dort vorgenommen, wo die Forschung und Entwicklung der letzten zwei Jahre die meisten neuen Begriffsbildungen und Einsichten brachte:

- in den Spezifikations- und Entwurfsmethoden (Kap. 12) und
- der internationale Normungs- und Standardisierungsarbeit und insbesondere dem allgemeinen Konzept des „offenen Systems“, welches dem „ISO-Architekturmodell“ zu Grunde liegt (Kap. 9).

Mit der Aufnahme dieser Ergänzungen hofft der Verfasser, mit der neuen Auflage wieder dem Leser all das Material über Rechnernetze zur Verfügung zu stellen, was er für einen Überblick über den derzeitigen Stand dieses Gebiets und als Grundlage für das Verständnis der speziellen Fachliteratur braucht.

München, im Juni 1982

Inhalt

Einführung	10
1. Grundbegriffe und Grundstrukturen	12
1.1 Kommunikation zwischen Rechnern	12
1.2 Aufgaben und Aufbau typischer Rechnernetze	17
1.3 Physische und logische Struktur von Rechnernetzen	24
1.4 Benutzeranforderungen	30
1.5 Realisierungsprobleme	34
2. Die Benutzerschnittstellen	37
2.1 Die nichtlokalisierte Maschine	37
2.2 Bedienungssprache und Terminals	39
2.3 Die verteilte Datenbasis	44
2.4 Zugriffsverfahren	48
2.5 Das Netz-Betriebssystem	55
3. Das verteilte System	59
3.1 Betriebsmittel- und Prozeßverwaltung	59
3.2 Betriebsmittladressierung im Netz	62
3.3 Adressierung in der SNA	64
3.4 Ports	66
4. Netzwerkmaschinen und Netzzustände	74
4.1 Nachrichten und Netzwerkmaschinen	74
4.2 Steuerinformationen und Zustandsübergänge	78
4.3 Untersysteme und ihre Zustände	81
4.4 Grundforderungen an Protokolle im Netz	84
5. Protokolle	87
5.1 Die Protokollhierarchie im ARPA-Netz	87
5.2 Die Protokolle des Datenübertragungs-Netzes	87
5.3 Höhere Protokollebenen	91
5.4 Grundregeln für die Protokollplanung	95
5.5 Die Realisierung eines Protokolls	96
6. Die Netzschnittstelle und die Flußsteuerung	99
6.1 Das Netz-Kommunikationsprogramm	99
6.2 Netzüberlastung und Betriebsmittelverwaltungs-Strategien	103
6.3 Grundverfahren der Flußsteuerung	106

6.4 Globale Flußsteuerungen auf Verbindungsbasis	107
6.5 Globale Flußsteuerungen auf Nachrichten- oder Paketbasis	110
7. Netzstrukturen und Laufwege	113
7.1 Netz-Topologien	113
7.2 Durchschaltungen, virtuelle Verbindungen und Datagramme	117
7.3 Pakete	119
7.4 Wegwahl	124
7.5 Der „kürzeste Weg“ als Laufwegbestimmungs-Strategie	130
7.6 Blockierungen im Netz	133
8. Leitungen und Knotenrechner	137
8.1 Struktur der IMP-Software	137
8.2 Übertragungsleitungen	141
8.3 Modulation und Leitungssteuerung	144
8.4 Ökonomie der Leitungsauslastung	146
8.5 Lokale Belastungen und globale Übertragungseigenschaften	148
8.6 Kanalbelastungs-Begrenzung	150
9. Das ISO-Architekturmodell	153
9.1 Standards und Normen	153
9.2 Schichten, Instanzen und Dienste	154
9.3 Die Transportschichten	158
9.4 Die Anwendungsschichten	160
10. Auslegung und Optimierung eines Netzes	164
10.1 Der Verwaltungsaufwand in den Protokollebenen	164
10.2 Optimierungsgesichtspunkte	169
10.3 Prioritäten für Nachrichten und Pakete	173
10.4 Fehlerbehandlung und Flußsteuerung	174
10.5 Sende- und Empfangsfenster	178
10.6 Resynchronisierung durch „Attention and Mark“	180
11. Realisierung und Betrieb eines Rechnernetzes	183
11.1 Spezifikation	183
11.2 Planung	185
11.3 Implementierung und Weiterentwicklung	188
11.4 Betriebsüberwachung und Netzsteuerung	191
11.5 Der Verbund von Netzen	194
12. Entwurfs- und Programmiermethoden in einer Netzumgebung	202
12.1 „Lokalisierte“ und „verteilte“ Anwendungsprogrammierung	202
12.2 Zum Datenmodell: Kommunikationsvariable	202
12.3 Zum Ablaufmodell: Zustandsdiagramme	204

Inhalt	9
12.4 Zum Kommunikationsmodell: Interaktionsdiagramme	210
12.5 Zum Modularisierungsmodell: Spezifikationsmittel	215
12.6 Zur Auftrags- und Betriebsmittel-Steuerung: Auftragsvariablen	219
Anhang: Datennetz-Terminologie	221
Literatur	245
Sachregister	258

Einführung

Spezifikation, Planung und Implementierung von Rechnerverbundnetzen werden noch weitgehend „Bottom Up“ betrieben, von der physikalischen Realisierung der Übertragungsleitungen und der Netzknoten über den Anschluß der Terminals und Teilnehmerrechner (Hosts) bis zur Benutzerschnittstelle. Dies ist verständlich. Schließlich ist der Grund für den Aufbau eines Netzes die räumliche Verteilung der Benutzer und der Betriebsmittel, der Rechner und Datenbestände, und damit die notwendige Datenübertragung.

Trotzdem – die Neue Softwaretechnologie bevorzugt die Top Down-Entwicklung. Auch für eine Netzplanung ist dies ein sinnvoller Ansatz, wenn auch die Berücksichtigung der übertragungstechnischen, physischen Gegebenheiten und Einschränkungen hier bei jedem Entwicklungsschritt von noch größerer Bedeutung sein wird als auf fast allen anderen Gebieten der Datenverarbeitung. Deshalb verfolgt dieses Buch die innere Logik und die sachlichen Alternativen eines Netzentwurfs von „oben“, den Benutzern, nach „unten“, den physischen Komponenten.

Wichtig für das Verständnis eines Netzentwurfs ist die Trennung der *logischen Konzepte* von ihrer *physischen Realisierung*. In den Abbildungen dieses Buches soll dem Leser hierbei eine grundsätzliche Unterscheidung helfen:

- logische Einheiten (z. B. die abstrakte „Netzwerkmaschine“) werden als runde oder ovale Symbole,
- physische Einheiten (z. B. ein Host- oder IMP-Rechner) als eckige Symbole dargestellt.

Im übrigen versuchte der Autor weitgehend auf Formeln, Algorithmen und Formalismen zu verzichten – über die Literaturzitate kann der Leser sie leicht im Direktzugriff erreichen.

Das Ziel, möglichst viel Information in deutscher Prosa zu vermitteln, drohte an einer unvorhergesehenen Schwierigkeit zu scheitern. Für eine Reihe von Begriffen (z. B. „Gateway“) kennt der Verfasser kein deutsches Wort, für andere (z. B. „Netz-Kommunikationsprogramm“) mehr als eines. Deshalb wurde in diesen Fällen willkürlich ein Ausdruck ausgewählt und vor Anglizismen nicht zurückgeschreckt – die im Anhang gegebene „Datennetz-Terminologie“ hilft vielleicht, den in diesem jungen Zweig der Datenverarbeitung leider vorhandenen Begriffswirrwarr zu entflechten.

1. Grundbegriffe und Grundstrukturen

1.1 Kommunikation zwischen Rechnern

Voneinander unabhängige Rechner können auf die verschiedensten Weisen gekoppelt werden. Abb. 1.1–1 skizziert die hierfür am häufigsten eingesetzten Methoden.

Das Hauptthema dieses Buches ist die in Abb. 1.1–1 d gezeigte Kommunikation über Datenübertragungsleitungen – nicht nur zwischen zwei, sondern auch zwischen grundsätzlich unbeschränkt vielen, weitgehend voneinander unabhängigen EDV-Anlagen, einem *Rechnerverbund* oder *Rechnernetz*.

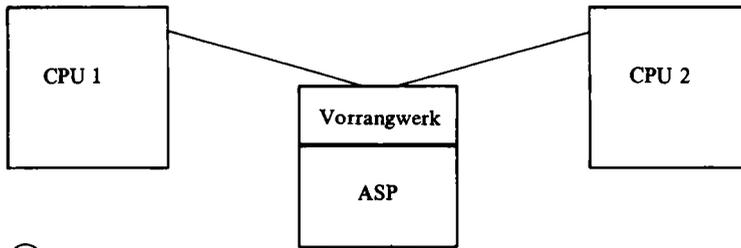
Der Grund für das Zwischenschalten einer Telex-, Datex-, Telefon- oder sonstigen Übertragungsleitung ist hierbei meist eine räumliche Entfernung von mehr als einigen 100 m zwischen den einzelnen teilnehmenden Rechnern, welche die übrigen in Abb. 1.1–1 dargestellten Kopplungsmöglichkeiten aus technischen Gründen ausschließt:

- einen gemeinschaftlich benutzten Arbeitsspeicher, in welchem ein oder mehrere Kommunikationsbereiche von den beteiligten Rechnern zur Ablage und zum Empfang von Nachrichten verwendet werden,
- eine Kopplung der E/A-Kanäle der Rechner, wobei jeder Rechner den anderen als „Externspeicher“ ansieht,
- einen gemeinsamen Externspeicher, der über einen Mehrkanalschalter rechnergesteuert mal der einen, mal der anderen Anlage zugeordnet wird.

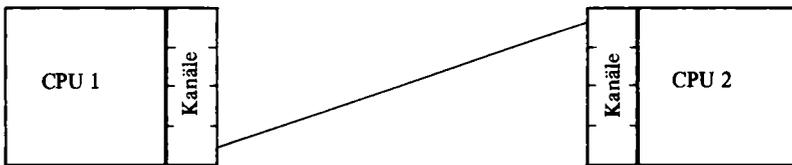
Viele der Probleme einer Rechnerkopplung sind jedoch unabhängig von ihrer technischen Realisierung. Dies gilt vor allem für die softwaretechnischen Fragen der Kommunikation zwischen den in den einzelnen Rechnern ablaufenden *Prozesse* und der *Protokolle* zur Verbindungsaufnahme, zur Sicherung des Nachrichtenverkehrs und zur Fehlererkennung und -behebung bei einem derartigen Dialog zwischen Rechnern. Welches Verfahren gewählt wird, bestimmen technische, ökonomische und einsatzorientierte Erwägungen. Hierbei ist die – unvermeidliche oder gewünschte – räumliche Trennung zwischen den beteiligten Rechnern sowie den gegebenenfalls zu ihrer Benutzung dienenden Terminals nur eines der Kriterien, freilich oft auch das ausschlaggebende.

Der Wunsch, EDV-Anlagen miteinander kommunizieren zu lassen, hat meist einen oder mehrere der folgenden Gründe:

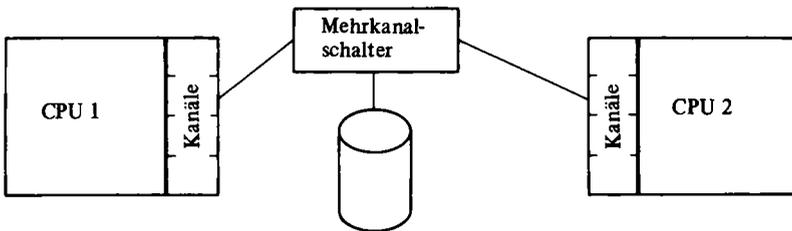
Eine *Spezialisierung für bestimmte Dienstleistungen* läßt es oft sinnvoll erscheinen, für unterschiedliche Arbeiten auch unterschiedliche Rechner einzusetzen. Erst ihr Zusammenwirken bietet dem Benutzer das volle Leistungsspektrum. So können



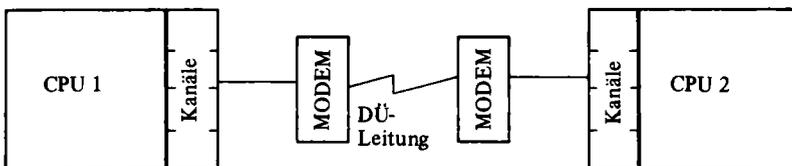
(a) Arbeitsspeicherkopplung



(b) Kanalkopplung



(c) Externspeicherkopplung



(d) Kopplung über eine Datenübertragungsleitung

Abb. 1.1-1 Kopplung zwischen unabhängigen Rechnern

für Meß- und Steuerungsaufgaben in einem Produktionsbetrieb oder einem Krankenhaus Prozeßrechner die zweckmäßige Hardware sein, während für die langfristige Haltung und Auswertung der umfangreichen Datenbestände kommerzielle Großrechner eingesetzt werden [RAIM 76].

In einem weitverzweigten Filialunternehmen oder einer öffentlichen Verwaltung werden für die Erfassung, Prüfung und Bearbeitung lokal relevanter Daten kleine Erfassungsrechner gebraucht, für die globale Auswertung der Datenbestände jedoch größere Anlagen in den Zentralen.

Eine EDV-Installation für wissenschaftlich-technische Forschungsaufgaben verwendet für die „normale“ Datenhaltung und -verarbeitung auch „normale“ Großrechenanlagen, für besondere Aufgaben wie die schnelle Durchführung von Matrixoperationen benötigt sie jedoch vielleicht mit ihnen zusammenarbeitende Spezialmaschinen wie Feld- und „Pipeline“-Rechner.

Eine der frühesten Spezialisierungen schließlich war die hardwaremäßige Trennung von Ein-/Ausgabe und Verarbeitungsvorgängen: die Architektur der CDC 6600- und 7600-Systeme beruhte auf der Aufteilung des EDV-Systems in einen Zentralrechner zur effizienten Bearbeitung numerischer Probleme und zehn mit ihm gekoppelten, kleineren *Peripheren Prozessoren* zur Bedienung der Peripherie und zur Steuerung der Gesamtanlage.

Die *Erhöhung der Zuverlässigkeit und der Verfügbarkeit* ist das zweite Motiv zum Verbund von Rechnern. Für kritische Steuerungs- und Kommunikationsaufgaben, vor allem im militärischen und industriellen Bereich, wurden bereits früh mehrere Rechner gekoppelt: entweder mit einer parallelen Bearbeitung identischer Daten und Aufgaben durch zwei bis drei grundsätzlich unabhängige Anlagen (*hot stand-by*) oder mit der Bereitstellung von Zweit- und Drittrechnern, auf die automatisch oder manuell bei Ausfall der primären Anlage umgeschaltet werden konnte (*cold stand-by*). Auch die räumlich getrennte Führung identischer Datenbestände wird aus Sicherheitsgründen zuweilen verlangt – zum Schutz wichtiger Daten gegen Zerstörung durch Krieg, Terrorismus, Unfälle oder Naturkatastrophen.

Eine *Verringerung der mittleren Wartezeiten* kann durch den Verbund mehrerer identischer Hardware-Systeme erreicht werden, die über den Arbeitsspeicher, über eine gemeinsame Plattendatei für das SPOOL-System oder auch über Datenübertragungsleitungen gekoppelt sind.

Wie stark die durchschnittlichen Wartezeiten bei hohen Auslastungen durch einen Lastausgleich zwischen mehreren Prozessoren gesenkt werden können, zeigt eine Abschätzung für die beiden in Abb. 1.1–2 dargestellten Modellsituationen.

Im *Single Server-Fall* ① treffen für m gleiche Prozessoren P_1 bis P_m in statistischer Folge Aufträge ein. Sie werden jeweils in der Warteschlange ihres Prozessors zwischengepuffert. Die mittlere Auslastung jedes Prozessors durch die Aufträge sei A , die mittlere Ausführungszeit (Service-Zeit) für einen Auftrag sei T_s .

Im *Multiserver-Fall* ② werden hingegen alle eintreffenden Aufträge in eine einzige Warteschlange zusammengeführt, aus der einem freiwerdenden Prozessor jeweils der nächste Auftrag zugeteilt wird. Im Unterschied gegenüber Fall ① ist sichergestellt, daß jeder der m Prozessoren arbeitet, solange überhaupt noch Aufträge warten.

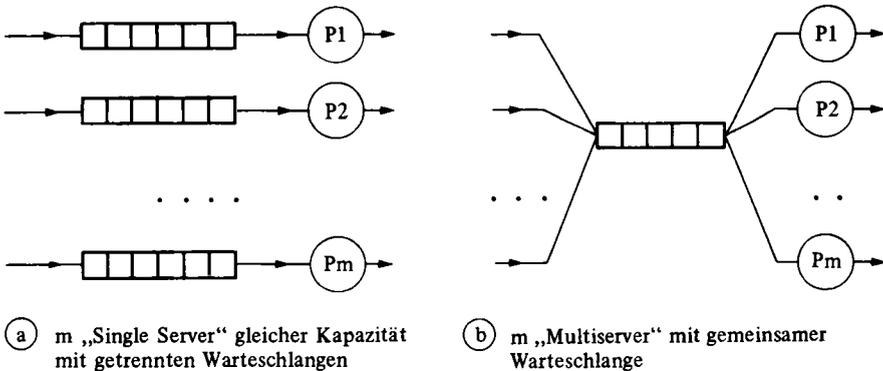


Abb. 1.1-2 Single Server- und Multiserver-System

Nach der Warteschlangentheorie ([KLEI76b], S. 280ff.) gilt bei exponentieller Verteilung der statistischen Größen für die mittlere Wartezeit T von der Anmeldung eines Auftrags bis zur Fertigmeldung

$$T = T_s \cdot \left(1 + \frac{P_m(A)}{m(1-A)} \right).$$

$P_m(A)$ ist die Wahrscheinlichkeit, daß das Gesamtsystem m oder mehr Aufträge bearbeiten muß, d. h. daß alle Prozessoren ausgelastet sind; sie ist gegeben durch

$$P_m(A) = \left(1 + \frac{(1-A)^m \cdot m!}{(mA)^m} \sum_{k=0}^{m-1} \frac{(mA)^k}{k!} \right)^{-1}$$

Für $m = 1$ ist sie gleich der Auslastung A . Für das Single Server-System folgt als mittlere Wartezeit

$$T = T_s \cdot \left(1 + \frac{A}{1-A} \right).$$

Bei einer Auslastung von 80% hat sie mit $A = 0,8$ den Wert $T = 5 \cdot T_s$ - sie ist also fünfmal länger als die reine Ausführungszeit eines Auftrags.

Arbeiten jedoch im Multiserver-Fall $m = 5$ Prozessoren auf die gleiche Warteschlange, so ergibt sich $P_5(0,8) = 0,7885$ und $T = 1,7885 \cdot T_s$. Gegenüber dem Single Server-Fall ist die mittlere Wartezeit also auf fast ein Drittel reduziert!

Noch ausgeprägter ist die Senkung der durchschnittlichen Wartezeiten, wenn die im Verbund zusammenarbeitenden Prozessoren durch die lokalen Anforderungen ohnehin nicht gleichmäßig ausgelastet wären und schwach belastete Systeme den überlasteten Arbeit abnehmen können. Dies ist auf Grund der geographischen Ausdehnung der USA bei den unter dem TENEX-System laufenden PDP10-

Installationen des ARPA-Netztes der Fall, welches mehrere Zeitzonen überdeckt.

Der Trend zur *Online-Datenverarbeitung über dialogfähige Bildschirm- oder Fernschreib-Terminals* ist ein weiteres Motiv zum Aufbau von Rechnernetzen. Abb. 1.1-3 zeigt die Zunahme der Anzahl installierter Terminals in Europa, Abb. 1.1-4 die Entwicklung der relativen Kostenanteile für Terminals, Rechner und

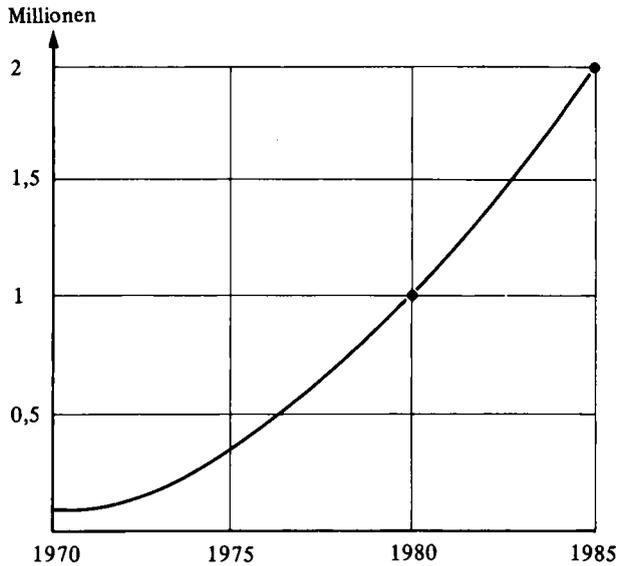


Abb. 1.1-3 Erwarteter Anstieg der Zahl der in Europa eingesetzten Terminals

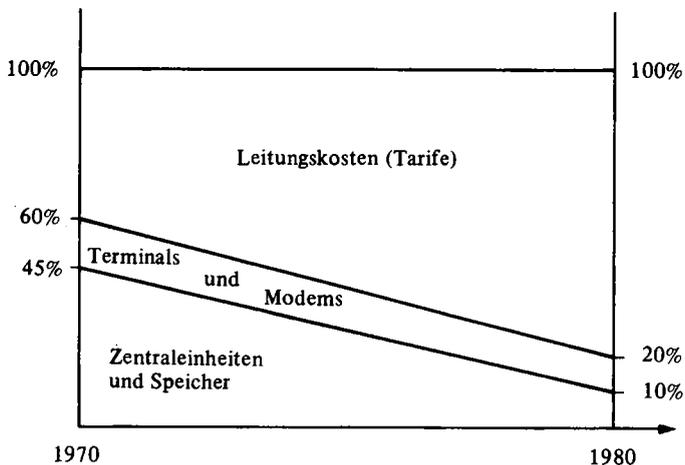


Abb. 1.1-4 Kostenverteilung einer typischen Datenübertragungs-Anwendung

Leitungsmieten bei einer typischen Datenübertragungsanwendung [INFO77]. Sobald die Leitungskosten den überwiegenden Kostenfaktor darstellen, wird es wirtschaftlich sinnvoll, durch Verteilung der Rechnerkapazität und Datenhaltung einen möglichst großen Teil der Datenverarbeitung an die Aufstellungsorte der Terminals, also etwa in die einzelnen Filialen eines Handelsunternehmens selbst zu verlagern, und über den Verbund dieser Rechner nur diejenigen Daten und Informationen zwischen ihnen auszutauschen, die wirklich überregional benötigt werden.

Die Erweiterung des Dienstleistungsangebots und der verfügbaren Betriebsmittel schließlich ist der wohl häufigste Grund zum Verbund mehrerer Rechner. Nur auf einzelnen Rechnern verfügbare Programmiersprachen, Betriebs-, Datenbank- und Informationssysteme oder besonders leistungsfähige Hardware können durch Eingliederung in ein Rechnernetz auch Benutzern anderer Anlagen zur Verfügung gestellt werden.

Für das ARPA-Netz wurden im März 1973 die Einsparungen ermittelt, welche 20 Teilnehmer-Installationen hierdurch erzielten [ROBE74]. Umgerechnet auf das volle Jahr mußten sie statt etwa 6 Mio \$ bei Eigenbeschaffung der benötigten Betriebsmittel nur ca. 2 Mio \$ an Kostenbeteiligungen und Gebühren für Rechenzeiten bezahlen. Den so ersparten 4 Mio \$ standen Kosten für den Betrieb des Netzes mit 3,5 Mio \$ gegenüber. In Anbetracht der Tatsache, daß damals das ARPA-Netz zu nur etwa 20% ausgelastet war, dürfte die Bilanz inzwischen noch wesentlich günstiger sein.

Je nachdem, ob ein Rechnerverbund auf eine Vervielfachung oder gleichmäßigere Verteilung der Rechnerleistung, auf die Verfügbarkeit von Betriebsmitteln eines EDV-Systems auch für andere Rechner oder auf eine Integration aller systemweit gehaltenen Einzel-Datenbestände abzielt, spricht man von einem *Lastverbund*, *Betriebsmittelverbund* oder *Datenverbund*. Wird schließlich eine Leistungserhöhung des Gesamtsystems dadurch erreicht, daß verschiedene Prozesse in den gekoppelten Einzelsystemen jeweils einen logisch abgetrennten Teilbeitrag zum Erreichen des Gesamtziels erbringen, so bezeichnet man dies als *Intelligenzverbund*.

1.2 Aufgaben und Aufbau typischer Rechnernetze

Die verschiedenen Motivationen zur Kopplung mehrerer EDV-Systeme zu einem Rechnernetz führen zu Systemlösungen, die sich vor allem in der Hardwarestruktur und der Topologie der physischen Leitungsführung unterscheiden. Diese Unterschiede beeinflussen die Hardwareauswahl, die Wahl der Datenübertragungsmedien, -verfahren und -techniken, die „niederer“ Ebenen der Datenübertragungssoftware, Leistungs- und Kostenoptimisierungen und – natürlich – die zu realisierenden Benutzerschnittstellen. Sie verdecken damit aber auch häufig die in

allen Netzen wiederzufindenden Gemeinsamkeiten der Grobstruktur und der Basisbausteine der Rechnernetz-Software.

Trotzdem empfiehlt es sich, vor der Idealisierung einer abstrakten Netzarchitektur als konkrete Beispiele einige realisierte Rechnernetze vorzustellen. Letztlich ist der Grund für die Planung eines Netzes immer „physisch“: eine aufgabenspezifisch vorgegebene Verteilung der Rechnerdienstleistungen, welche die einfachere und ökonomischere Lokalisierung der EDV-Installation ausschließt.

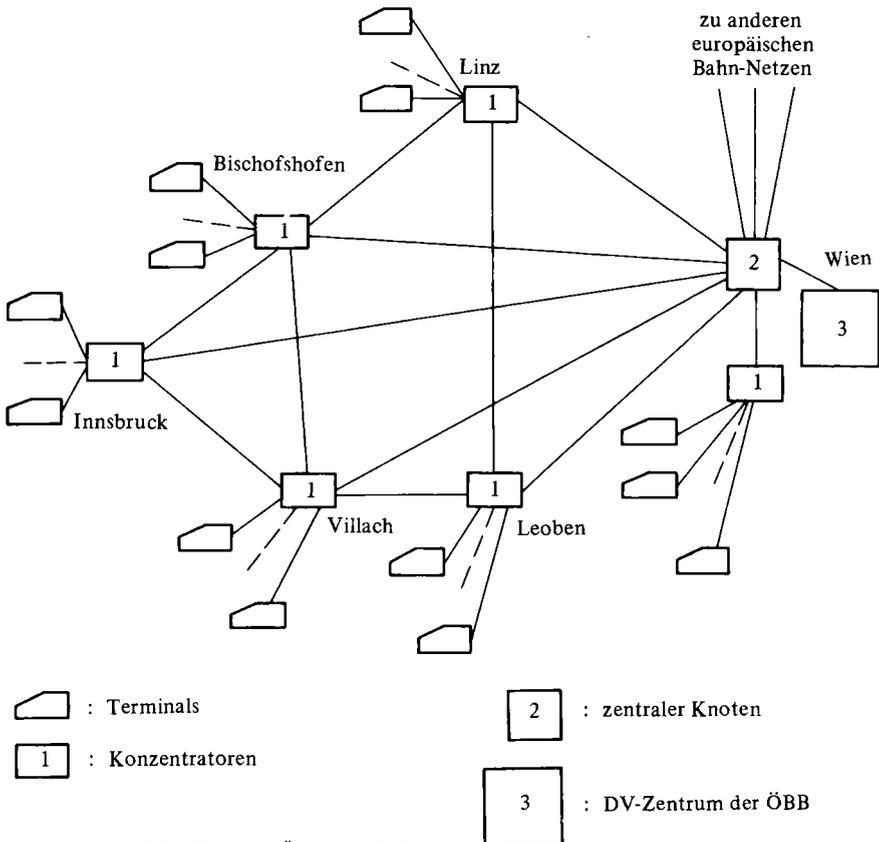


Abb. 1.2-1 (Sub-)Netz der Österreichischen Bundesbahn

Die räumliche *Verteilung der Benutzer* ist das Hauptmotiv für den Aufbau vieler Datenerfassungs- und Informationsnetze von Großfirmen, Behörden, öffentlichen Dienstleistungsunternehmen und ähnlichen Institutionen. Ein Beispiel ist das in Abb. 1.2-1 gezeigte Netz der *Österreichischen Bundesbahn*, welches zugleich Subnetz in einem Verbund der europäischen Bahnverwaltungen ist. Seine Hauptaufgaben sind

- Datenerfassung am Ort der einzelnen Bahnhöfe und Dienststellen,
- Auskunftserteilung über die 70 000 Bediensteten der ÖBB,
- Voranmeldung der Güterzüge an jedem Bahnhof und Lenkung des Güterverkehrs durch Steuerung der Zugzusammenstellung,
- Speichervermittlung von Fernschreiben,
- Platzbuchung in Zusammenarbeit mit dem automatischen Buchungssystem der Deutschen Bundesbahn.

Für jeden Bezirk gibt es ein Bezirksnetz mit einem Konzentrador und angeschlossenen Terminals unterschiedlicher Intelligenz. Die Aufgaben der Konzentradoren sind

- Plausibilitätsprüfungen der Terminaleingaben,
- Laufwegermittlung und Weiterleitung der Nachrichten an den oder von dem zentralen Knoten in Wien.

Die Konzentradoren sind mit dem zentralen Knoten über ein mehrfach vermaschtes Netz von 1200 baud-Leitungen verbunden.

Der zentrale Knoten

- übernimmt die Nachrichtenvermittlung zwischen Konzentradoren, DV-Zentrum in Wien und anderen europäischen Bahn-Netzen,
- überprüft die Terminalberechtigung bei Dienstleistungswünschen und Anfragen, wie etwa einer Personalauskunfts-Anforderung,
- überwacht das Gesamtnetz,
- führt Statistiken über den Betrieb des Netzes.

Die räumliche *Verteilung von angebotenen Dienstleistungen und Betriebsmitteln* gab Anlaß zur Entwicklung von Rechnernetzen, die im wesentlichen die Datenbestände, Programme und Betriebsmittel verschiedener lokaler Dienstleistungsrechenzentren für wissenschaftlich-technische oder kommerzielle Anwendungen miteinander verbinden und damit jedem Benutzer bei Bedarf die Leistungen aller beteiligten Rechnerinstallationen verfügbar machen sollen. Beispiele sind das amerikanische ARPA-Netz [LOGI74] oder das französische CYCLADES-Netz [POUZ73], welches in Abb. 1.2–2 skizziert ist. CYCLADES verbindet 16 Teilnehmerrechner (*Hosts*) von 6 verschiedenen Typen unter 8 verschiedenen Betriebssystemen und stellt auf ihnen den Netzbenutzern sowohl Timesharing- als auch Stapelverarbeitungs-Dienstleistungen zur Verfügung.

Das ARPA-Netz, an welches neben Rechenzentren in der gesamten USA auch noch einige europäische EDV-Installationen über Satelliten angeschlossen sind, umfaßt mehrere Zeitzonen und dient damit auch dem Ausgleich der *räumlichen Verteilung der zeitlichen Anforderungsschwerpunkte*.

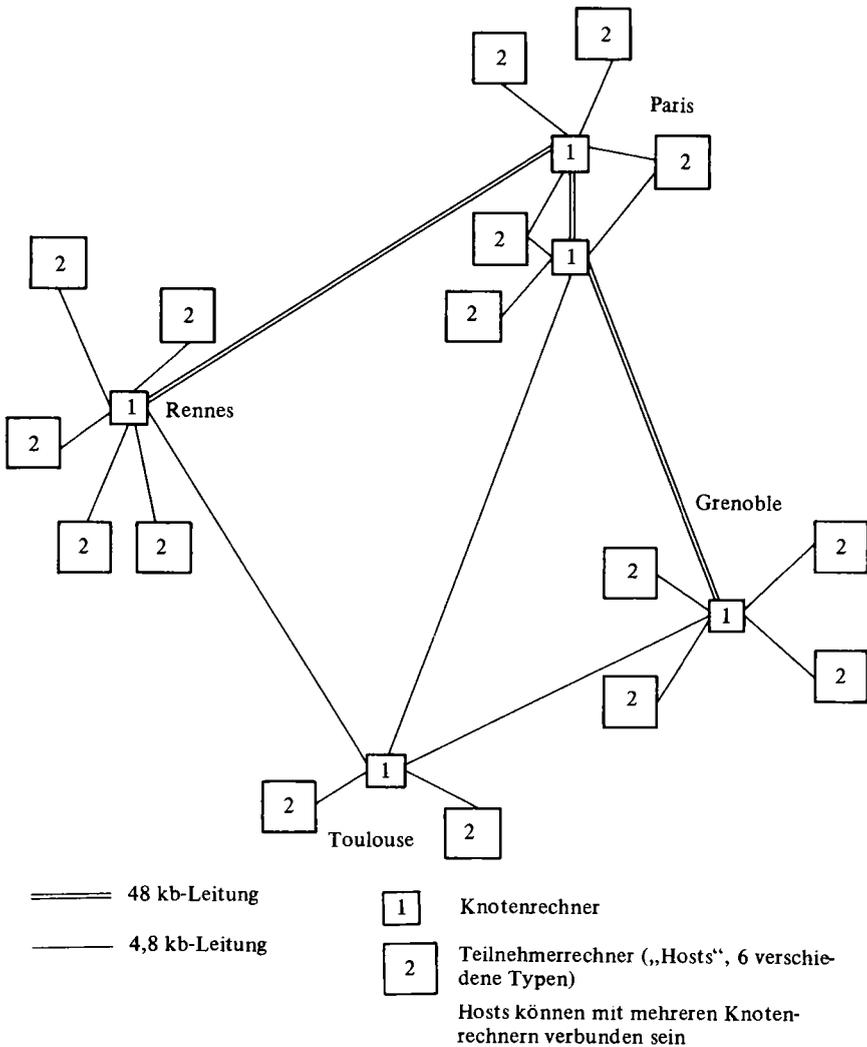


Abb. 1.2-2 Das CYCLADES-Netz

Der Wunsch nach *Nutzung aller verfügbaren Betriebsmittel* – vor allem von Peripheriegeräten – für verschiedene, vorhandene Rechner führt zur Idee des *Resource Sharing-Netzes*. Ein Vertreter dieses Typs ist das Octopus-System der Lawrence Livermore Laboratories [WATS80], welches Abb. 1.2-3 skizziert. Hier wurde für jede Betriebsmittel-Klasse ein eigenes Netz aufgebaut, an welches einerseits die entsprechenden Peripheriegeräte und andererseits sämtliche bedienten Rechenanlagen angeschlossen sind.

Die Octopus-Konfiguration hat den Vorteil, daß jedes der spezialisierten Netze auf die Eigenschaften des zugeordneten Betriebsmittels abgestimmt werden kann. Dafür müssen die beteiligten Rechner neben ihrer eigentlichen Arbeit jedoch aufwendige Durchschaltfunktionen zwischen den einzelnen Teilnetzen übernehmen: sie arbeiten als *Gateways*, auf deren Probleme in Abschn. 11.5 näher eingegangen wird. Dies kann leicht zu ihrer Überlastung führen, da ein normaler Großrechner für derartige Aufgaben technisch nicht ausgelegt ist – so wird in [WATS80] berichtet, daß zuweilen ein CDC 7600-Rechner ausschließlich mit der Übertragung einer Datei beschäftigt ist!

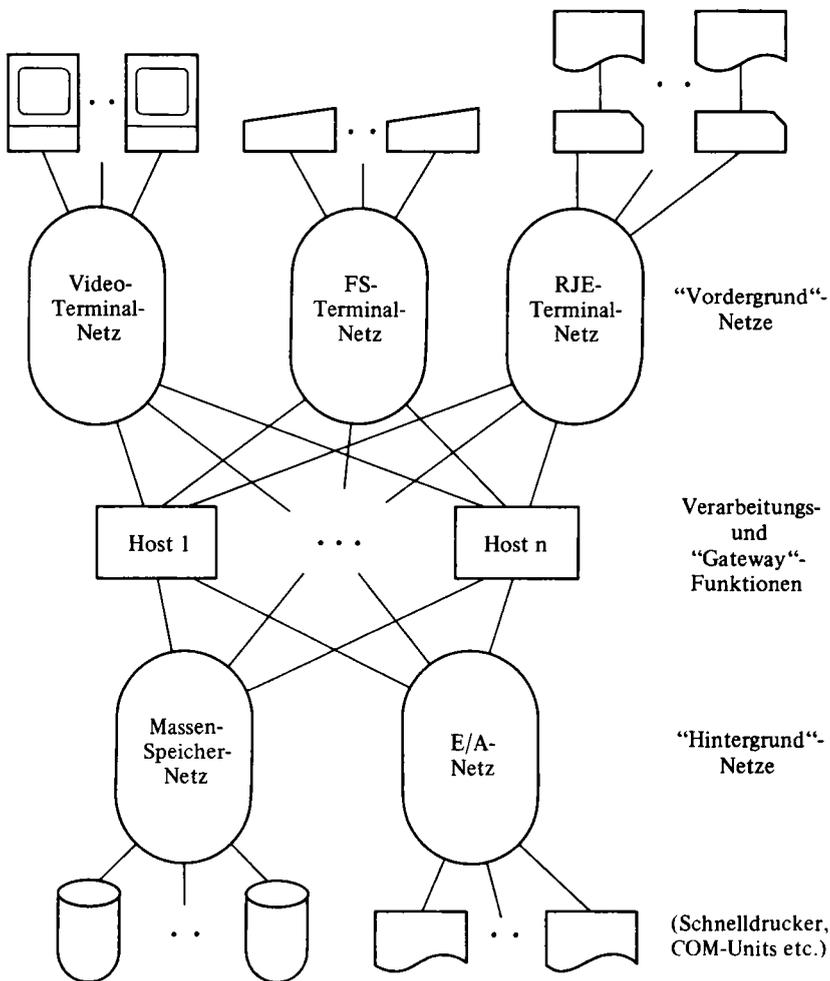


Abb. 1.2-3 „Resource Sharing“-Netz

Deshalb wurde für das Nachfolgersystem von Octopus ein anderes Konzept gewählt, bei dem sämtliche Peripheriegeräte und Verarbeitungsrechner an einem gemeinsamen „Broadcast-Trunk“-Vermittlungssystem hängen. Diese Konfiguration entspricht dann weitgehend einer „Architektur offener Systeme“, auf welche die derzeitige Standardisierungsarbeit hinzielt und auf die in Kapitel 9 ausführlich eingegangen werden wird.

Bei der Diskussion von DV-Netzen werden schließlich die im Rahmen von *öffentlichen oder privaten Datenübertragungsnetzen* zusammenarbeitenden Rechner häufig vergessen. Auch sie bilden oft sehr komplexe Rechnernetze, wobei die beteiligten EDV-Anlagen jedoch nicht die Bearbeitung von Benutzeraufgaben sondern den Nachrichtenverkehr über das Netz selbst sowie ggf. Dienstleistungen übernehmen, welche mit diesem Nachrichtenverkehr zusammenhängen (z. B. Abrechnung, Statistik, Speichervermittlung).

Neben dem Vermittlungsdienst bietet die Bundespost durch eigene Systeme auch noch netzbezogene Dienstleistungen, wie etwa die Automatische Telexauskunft, die auf Anfrage selbsttätig Auskunft über Telexnummern oder Teilnehmeranschriften erteilt.

Abb. 1.2–4 skizziert schematisch das Fernschreibnetz der Deutschen Bundespost. Die Vermittlung von Verbindungen zwischen Fernschreibteilnehmern wird über zentrale Vermittlungsrechner (*EDS = Electronic Data Switch*) durchgeführt, welche auch für die Gebührenberechnung und ähnliche Aufgaben zuständig sind. Neben dem Direktanschluß von Fernschreibern an die EDS-Zentralen ist der Anschluß von Nebenstellen-Anlagen oder Sondernetzen möglich, die über eigene, kleinere Vermittlungsrechner verfügen. Diese Nebenstellenrechner erfüllen ihrerseits weitere Dienste, wie etwa die interne Gebührenüberwachung, Speichervermittlung, externe und interne Rundschreiben, automatische Generierung von Standardtexten, Textspeicherung und -editierung.

In einigen Ländern wie den USA (TYMNET [SCHW76]), stehen derartige öffentliche Netze bereits auch ausschließlich für den Datenverkehr zwischen Rechnern zur Verfügung. Wegen der Ergänzung der Vermittlungsfunktion durch zusätzliche Dienstleistungen – vor allem der gemeinsamen Benutzung einer physischen Leitung durch mehrere Teilnehmer – und der damit verbundenen „Werterhöhung“ jeder Einzelleitung werden sie oft als *Value Added Networks (VAN)* bezeichnet.

In der Praxis wird ein konkretes Netz freilich oft mehr als einer Aufgabe dienen, und es ist nicht leicht vorauszusagen, wo später die tatsächlichen Schwerpunkte seines Einsatzes liegen werden. Laut *McQuillan* [MCQU80] war das ARPA-Netz ursprünglich zwar als Resource-Sharing-Netz konzipiert worden, wurde dann aber von seinen Benutzern primär als schnelles und bequemes Nachrichten-Medium („Elektronische Post“) eingesetzt.

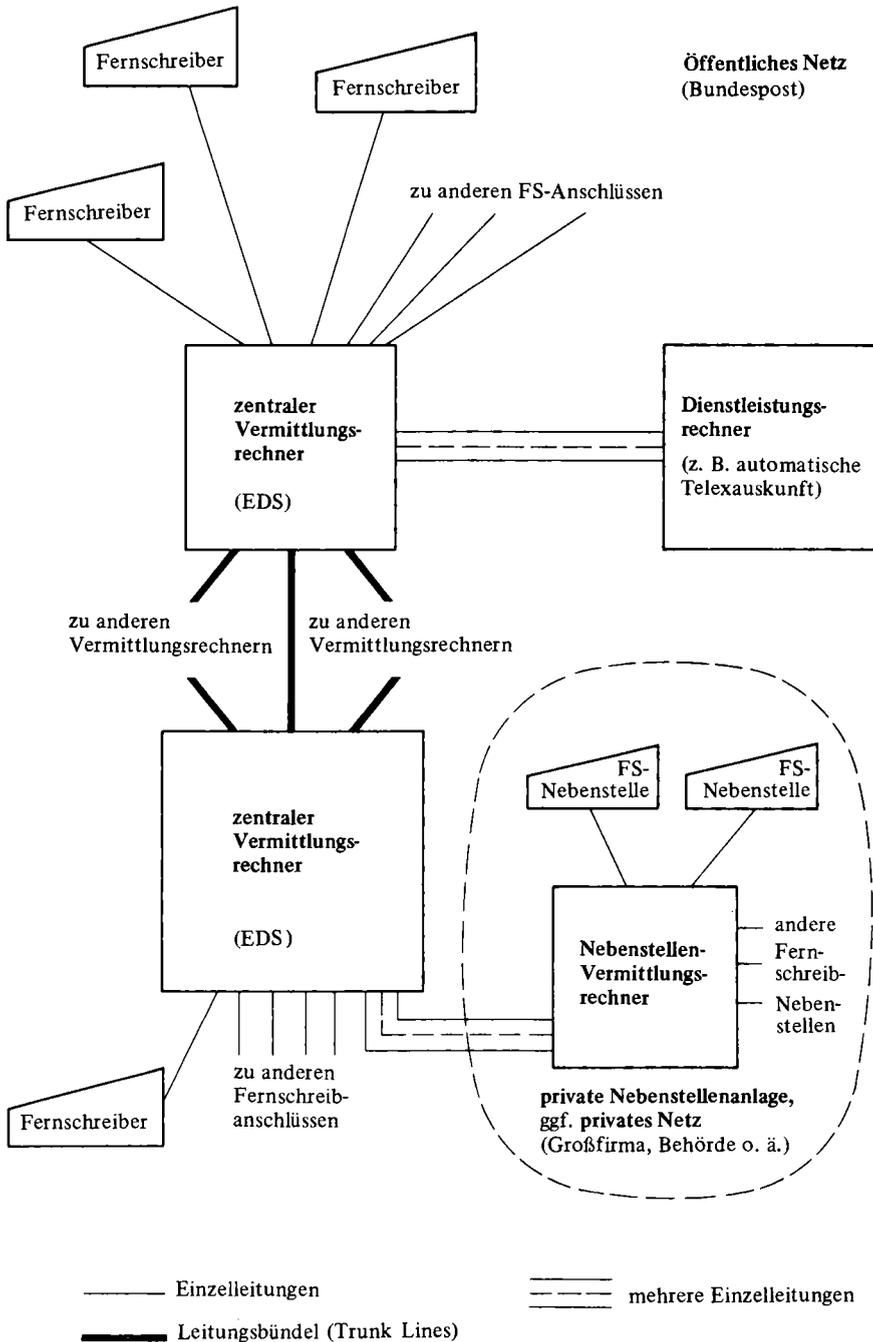


Abb. 1.2-4 Das Fernschreibnetz der DBP

1.3 Physische und logische Struktur von Rechnernetzen

Die Vielfalt der Gründe, die zum Aufbau eines Rechnernetzes führen können, bedingt eine Vielfalt der Entwurfsvorgaben, der bei der Realisierung gegeneinander abzuwägenden Zuverlässigkeits- und Effektivitätsgesichtspunkte, der sinnvollen Leitungsführung und -auslegung, der Anordnung der verschiedenen Rechner und sonstigen Betriebsmittel sowie der Aufgabenteilung zwischen ihnen. Um überhaupt eine einheitliche Entwicklungsmethode für DV-Netze erarbeiten zu können, welche jeweils einer bestimmten Aufgabe angepaßt und – wenn nötig – modifiziert werden kann, muß man von einer Netz-Grobstruktur ausgehen, die als Idealisierung der unterschiedlichen konkreten Realisierungen angesehen werden kann.

Bei jedem DV-System lassen sich zwei wesentliche Strukturen unterscheiden:

- die *physische* Struktur, d. h. die einzelnen eingesetzten Rechner, Terminals und sonstigen Betriebsmittel, die technischen Übertragungselemente wie Leitungen und Modems, sowie ihre gegenseitige (konkrete) Anordnung und Zusammenschaltung,

und

- die *logische* Struktur von Daten und Prozessen sowie deren Wechselwirkungen systemintern oder -extern in Kommunikation mit der Umgebung (Benutzern an Terminals, Meß- und Steuereinrichtungen u. ä.).

Die Aufgabe jeder *Software* ist die *Abbildung* der logischen Struktur auf die physische, der *Benutzermaschine* auf die *Hardware* und auf die als Teil von ihr aufzufassende, bereits vorhandene *Basissoftware* wie Sprachübersetzer, Hersteller-Betriebssysteme und -Dienstleistungsprogramme. Die in Abb. 1.3–1 und 1.3–2 gezeigten physischen und logischen Grundstrukturen eines Netzes charakterisieren damit zugleich die wesentlichen Teilaufgaben eines Netzentwurfs:

- die konkrete Auswahl und Anordnung der Hardware-Strukturelemente,
- die für die jeweilige Anwendung optimale Realisierung der gewünschten logischen Struktur

sowie

- die Zuordnung von logischen und physischen Strukturelementen.

Die *physische Struktur* skizziert Abb. 1.3–1: verschiedene Arten von Hardware-Elementen werden durch Datenübertragungsleitungen zu einem Netz verbunden.

Die im Netz zusammengeschlossenen Hardware-Einheiten können jeweils als an einem Ort lokalisiert angesehen und damit durch eine *Adresse* identifiziert werden.

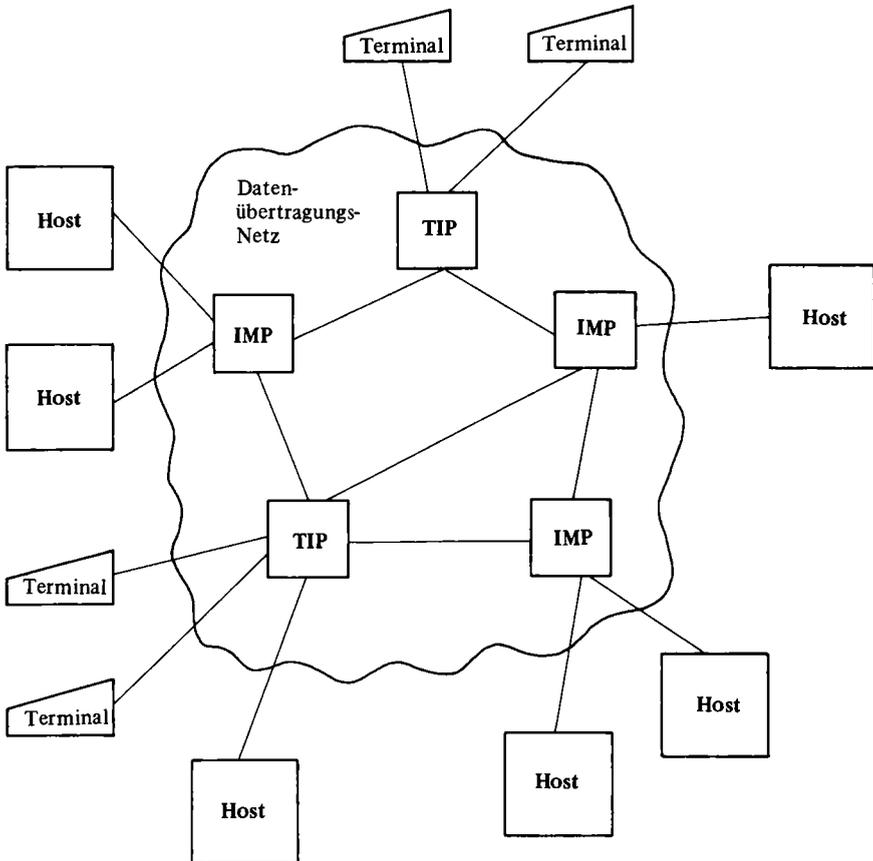


Abb. 1.3-1 Physische Struktur eines Rechnerverbundnetzes

Diese Adresse ist eine Bit- oder Zeichenfolge, welche eine mnemonische Bedeutung haben kann, aber nicht muß.

Nach ihrer Funktion im Netz unterscheiden wir vier Grundtypen dieser Hardware-Elemente.

Terminals (Datenendplätze, Datenstationen) sind die Ein-/Ausgabegeräte, über welche die Außenwelt, in der Regel der menschliche Benutzer, mit dem Netz kommuniziert. Es gibt *dialogfähige* und *nicht dialogfähige* Terminals. Dialogfähige Terminals sind Fernschreibgeräte, Schreibmaschinen, Datensichtgeräte und graphische Terminals. Nicht dialogfähige Terminals sind Kartenleser, Schnelldrucker, Anzeigelampen, Ausweisleser und die übliche Prozeßperipherie wie Meßfühler und Steuereinrichtungen.

Hosts (Wirtsrechner, Gastrechner, Verarbeitungsrechner, Teilnehmer-Rechner, Endrechner) sind Rechner (bzw. EDV-Installationen), die für das Netz Quellen

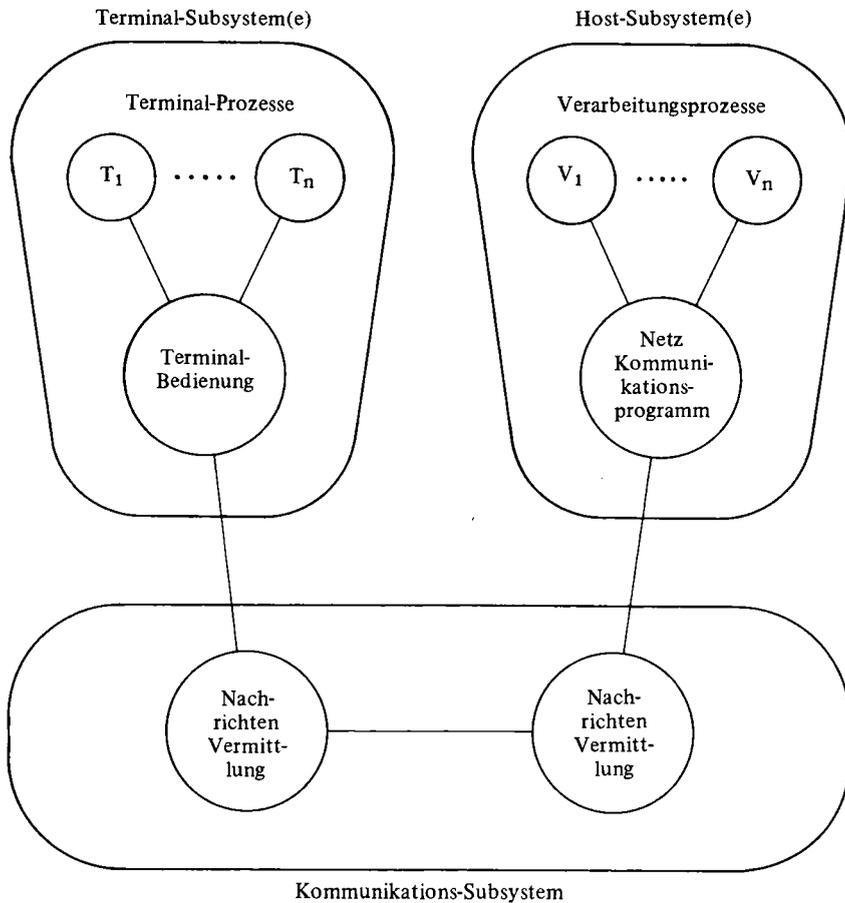


Abb. 1.3-2 Logische Struktur eines Rechnernetzes

und Senken von Informationen sind. In ihnen entstehen Nachrichten, welche dem Netz zur Übermittlung an andere Hosts oder Terminals übergeben werden, und an sie gelangen über das Netz Nachrichten von anderen Hosts oder Terminals zur Verarbeitung.

IMPs (Interface Message Processors, Knotenrechner, Kommunikationsprozessoren, Store and Forward Nodes, Vermittlungsrechner) sind Rechner, welche Nachrichten von Hosts zur Weiterleitung an eine bestimmte Zieladresse (anderer Host oder Terminal) übernehmen. Sie trennen diese Nachrichten ggf. in kleinere Einheiten (*Pakete*) auf und übermitteln sie dem Adressaten entweder unmittelbar oder unter Zwischenschaltung anderer IMPs.

TIPs (Terminal Interface Processors, Terminal-Knotenrechner) sind IMPs, welche nicht nur Hosts bedienen sondern auch den Nachrichtenverkehr mit Terminals abwickeln können.

Die IMPs und TIPs bilden das „eigentliche“ *Datenübertragungsnetz*, welches Hosts und Terminals zur Kommunikation benutzen.

Im Rahmen des groben Strukturmodells von Abb. 1.3–1 sind Hosts, IMPs und TIPs nichts weiter als lokalisierte, adressierbare „Behälter“ für bestimmte zur Verfügung stehende Betriebsmittel und ablaufende Aktivitäten (*Prozesse*). Über ihren Aufbau, Ausstattung, Größe u. ä. werden keinerlei Angaben gemacht.

Nimmt man an, daß alle Hosts einerseits und alle IMPs und TIPs andererseits vom gleichen Rechnertyp sind und sich allenfalls unwesentlich in ihrem Arbeitsspeicher-ausbau, ihrer Zentraleinheit und der angeschlossenen Peripherie unterscheiden, so spricht man von einem *homogenen Netz* – ein Sonderfall, der in der Regel die Planung und Realisierung der Netzsoftware wesentlich erleichtert. Der technische Fortschritt oder steigende Anforderungen zwingen allerdings immer irgendwann zur Eingliederung neuer Hardwaremodelle und Datenübertragungsverfahren oder zum Anschluß an andere Netze, wodurch ein bestehendes Netz zwangsläufig inhomogen wird. Wird dies nicht bereits während der ersten Planung und Realisierung bedacht, so betragen die höheren Wartungs- und Weiterentwicklungskosten leicht ein vielfaches der durch die angenommene Homogenität des Netzes erreichten Einsparungen.

In der *logischen Struktur* eines Rechnernetzes, die Abb. 1.3–2 darstellt, kann man auf oberster Ebene

- Host-Subsysteme,
- Terminal-Subsysteme, und
- Kommunikations-Subsysteme

unterscheiden. Ein Netz enthält im allgemeinen mehrere Subsysteme jeder Art, die jeweils identisch sein (z. B. gleiche Hosts mit identischer Software) oder sich voneinander unterscheiden können.

Häufig gibt es nur ein einziges Kommunikations-Subsystem – das Datenübertragungsnetz mit seiner Software. Das zunehmende Kommunikationsbedürfnis in der Datenverarbeitung führt aber immer mehr dazu, daß Hosts und/oder Terminals in mehrere Netze eingegliedert werden, wodurch ein Gesamtnetz mit mehreren Kommunikations-Subsystemen entsteht.

Ein *Host-Subsystem* enthält ein *Netz-Kommunikationsprogramm*, mit dessen Hilfe Verarbeitungsprozesse einen Nachrichtenverkehr über das Kommunikations-Subsystem abwickeln können. Diese Subsystem-Komponente wird in fast jedem Netz mit einem anderen Namen bezeichnet. Im ARPA-Netz heißt sie *Network Control*